Modeling Transitions of Focal Entities for Conversational Knowledge Base Question Answering

Yunshi Lan
Singapore Management University
yslan@smu.edu.sg

Jing Jiang
Singapore Management University
jingjiang@smu.edu.sg

Abstract

Conversational KBQA is about answering a sequence of questions related to a KB. Follow-up questions in conversational KBQA often have missing information referring to entities from the conversation history. In this paper, we propose to model these implied entities, which we refer to as the focal entities of the conversation. We propose a novel graph-based model to capture the transitions of focal entities and apply a graph neural network to derive a probability distribution of focal entities for each question, which is then combined with a standard KBQA module to perform answer ranking. Our experiments on two datasets demonstrate the effectiveness of our proposed method.

1 Introduction

Recently, conversational Knowledge Base Question Answering (KBQA) has started to attract people's attention (Saha et al., 2018; Christmann et al., 2019; Guo et al., 2018; Shen et al., 2019). Motivated by real-world conversational applications, particularly personal assistants such as Apple Siri and Amazon Alexa, the task aims to answer questions over KBs in a conversational manner.

Figure 1 shows an example of conversational KBQA. As we can see, the conversation can be roughly divided into two parts: Q1, Q2 and Q3 revolve around the book "The Great Gatsby," while Q4 and Q5 revolve around its author, "F. Scott Fitzgerald". Although these entities are not explicitly mentioned in the questions, they are implied by the conversation history, and they are critical for answering the questions. For example, Q3, when taken out of context, cannot be answered because Q3 itself does not state the title of the book being discussed. But since Q3 is a follow-up question of Q1, humans can easily infer that the book of interest here is "The Great Gatsby" and can hence answer the question correctly. We therefore can

Q1: What novel has the character named Nick Carraway?	
R1: The Great Gatsby	
Q2: Where is Jay Gatsby born?	The Great Gatsby
R2: North Dakota	
Q3: What is the name of the author?	The Great Gatsby
R3: F. Scott Fitzgerald	
Q4: What's his first novel?	F. Scott Fitzgerald
R4: This Side of Paradise	
Q5: Who was his child?	F. Scott Fitzgerald
R5: Frances Scott Fitzgerald	

Figure 1: An example conversation in conversational KBQA. The entities shown in blue are what we call the *focal entities*, which are implicit but important for answering the questions.

regard the entity "The Great Gatsby" as the *focus* of the conversation at this point. When we move on to Q4, again, if the question is taken out of context, we cannot answer it. But by following the conversation flow, humans can guess that at this point the focus of the conversation has shifted to be "F. Scott Fitzgerald" (the answer to Q3), and based on this understanding, humans would have no problem answering Q4. We refer to "The Great Gatsby" and "F. Scott Fitzgerald" as the *focal entities* of the conversation.

Based on the observation above, we hypothesize that it is important to explicitly model how a conversation transits from one focal entity to another in order to effectively address the conversational KBQA task. There are at least two scenarios where knowing the current focal entity helps answer the current question. (1) The current focal entity is the unspecified topic entity[1] of the current question. E.g., "The Great Gatsby" is the unspecified topic entity for Q3, which effectively should be "what is the name of the author of *The Great Gatsby*?" (2) The current focal entity is closely related to the

[1]In KBQA, a *topic entity* is an entity mentioned in the question and the starting point in the KB to search for answers.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 3288–3297
August 1–6, 2021.

topic entity of the current question and can help narrow down the search space in case of ambiguity. E.g., knowing the focal entity is "The Great Gatsby" for Q2, the system can identify the correct subgraph of the KB that contains both "Jay Gatsby" (the topic entity) and "The Great Gatsby" for answer prediction, which is critical if there are more than one entities in the KB named "Jay Gatsby." We can also see that simple entity coreference resolution techniques (e.g., Lee et al. (2017)) may not always help for conversational KBQA as no pronouns are used in many cases.

Although existing work on conversational KBQA has tried to address the challenges of missing information in follow-up questions by modeling conversation history, most of it simply includes everything in the conversation history without considering focal entities. For example, Saha et al. (2018) leveraged a hierarchical encoder to encode all the questions and responses in the conversation history, but there was no explicit modeling of anything similar to focal entities. Guo et al. (2018) concatenated previous questions with the current question to fill in the missing information, but again there was no special treatment of entities. A more recent work (Christmann et al., 2019) believed that the answers to sequential questions should be closely connected to each other in the KB. Thus, they proposed an algorithm to keep a context graph in memory, expanding it as the conversation evolves to increase the connections between the questions. However, their method is inefficient in capturing the most significant information related to focal entities in a conversation history.

In this paper, we explicitly model the focal entities and their transitions in a conversation in order to improve conversational KBQA. Based on several observations we have with focal entities, such as their tendencies to be topic entities or answer entities in the conversation history and their stickiness in a conversation, we propose to construct an *Entity Transition Graph* to elaborately model entities involved in the conversation as well as their interactions, and apply a graph-based neural network to derive a *focal score* for each entity in the graph, which represents the probability of this entity being the focal entity at the current stage of the conversation. The key intuition behind the graph neural network is to propagate an entity's focal score in the i-th turn of the conversation to its neighboring entities in the $(i+1)$-th turn of the conversation. This derived focal entity distribution is then incorporated into a standard single-turn KBQA system to handle the current question in the conversation.

We evaluate our proposed method on two conversational KBQA datasets, ConvQuestions (Christmann et al., 2019) and ConvCSQA (which is a subset we derived from CSQA (Saha et al., 2018)). Experiment results show that compared with either a single-turn KBQA system or a system that simply encodes the entire conversation history without handling focal entities in a special way, our method can clearly perform better on both datasets. Our method also outperforms several existing systems that represent the state of the art on these benchmark datasets. We also conduct error analysis that sheds light on where further improvement is desired.

We summarize our contributions of this paper as follows: (1) We propose to explicitly model the focal entities of a conversation in order to improve conversational KBQA. (2) We propose a graph-based neural network model to capture the transitions of focal entities and derive a focal entity distribution that can be plugged into a standard single-turn KBQA system. (3) We empirically demonstrate the effectiveness of our method on two datasets. Our method can outperform the state of the art by 9.5 percentage points on ConvQuestions and 14.3 percentage points on ConvCSQA[2].

2 Background

2.1 Problem Formulation

A KB $\mathcal{K}$ consists of a large nubmer of triplets $\langle e_s, r, e_o \rangle$, where e_s and e_o are entities and r indicates their relation.

We first define *single-turn* KBQA as follows. Given a KB $\mathcal{K}$ and a question q, the system is supposed to return one or more entities from $\mathcal{K}$ as the answer to q. In single-turn KBQA, different question-answer pairs $\mathcal{D} = \{(q_1, a_1), (q_2, a_2), \ldots\}$ are independent.

Conversational KBQA is a *multiple-turn* KBQA problem, where a sequence of question-answer pairs $c = ((q_1, a_1), (q_2, a_2), ..., (q_m, a_m))$ forms a complete conversation and a set of independent conversations $\mathcal{D} = \{c_1, c_2, \ldots\}$ forms a conversational KBQA dataset. We refer to each question-answer pair as *one turn of the conversation*. A conversational KBQA system is supposed to return

[2]Our code is available at https://github.com/lanyunshi/ConversationalKBQA.

the correct answer to the current question q_t based on not only q_t but also the preceding questions $(q_1, q_2, ..., q_{t-1})$ in the same conversation.

2.2 Pipeline for Single-turn KBQA

A standard single-turn KBQA includes two main components: a *Query Generator* and an *Answer Predictor*. The Query Generator generates a set of candidate query graphs $\mathcal{C}$ for a given q. Specifically, we first assume that some entities relevant to q are first identified. These can be entities directly mentioned in q or other entities relevant to q but implicitly mentioned, such as the focal entities we introduced earlier. Starting from these entities, the Query Generator generates a set of candidate query graphs (Yih et al., 2016) from $\mathcal{K}$, which lead to some candidate answers to the question. The second component of a single-turn KBQA system, the Answer Predictor, is a neural-network-based ranker that takes in the question as well as the generated query graphs as input and outputs a predicted answer $\hat{a}$.

For conversational KBQA, the initial question q_0 in a conversation c can be answered directly using an existing single-turn KBQA approach (Yu et al., 2017; Luo et al., 2018; Yih et al., 2016; Lan et al., 2019). When the single-turn KBQA system is used for answering follow-up questions, we make the following modifications: First, we assume that a focal entity distribution (which is the core of our method and will be presented in detail below) is derived from the conversation history. Then each focal entity is considered relevant to the current question and will be used to generate candidate query graphs by the Query Generator. Meanwhile, the probabilities of these focal entities (i.e., their focal scores) will be used by the Answer Predictor when it ranks the candidate query graphs.

3 Our Method

3.1 Overview

Our proposed method hinges on the notion of *focal entities* that we introduced in Section 1. Recall that a focal entity is the focus of the conversation at its current stage. To model focal entities, we propose to first use an *Entity Transition Graph* to model all the entities involved in the conversation so far and their interactions. These entities are candidate focal entities. The edges of the graph reflect how the conversation has shifted from one entity to another, and such transitions can help us estimate how likely an entity is the current focal entity, as we will explain in Section 3.2. This graph is incrementally constructed by a *Graph Constructor* after each turn of the conversation. To derive a *focal score* (i.e., a probability) for each entity in this graph, a *Focal Entity Predictor* employs a graph-based neural network and generates a new focal entity distribution based on the previous focal entity distribution as well as the conversation history, which is encoded by a *Conversation History Encoder* using a standard sequence model. Finally, the derived focal entity distribution is incorporated into the single-turn KBQA module presented in Section 2.2 to perform answer prediction. The overall architecture of our method is illustrated in Figure 2.

3.2 Entity Transition Graph and Graph Constructor

Our Graph Constructor builds the Entity Transition Graph as follows. The initial Entity Transition Graph $\mathcal{G}^{(0)}$ is set to be an empty graph. Let $\mathcal{G}^{(t-1)}$ denote the Entity Transition Graph before the t-th turn of the conversation, and suppose we have processed the t-th question and obtained the answer entity $\hat{a}_t$ (which is predicted) with the help of $\mathcal{G}^{(t-1)}$. We now need to construct $\mathcal{G}^t$, which will be used to help answer q_{t+1}. Recall that the Answer Predictor presented in Section 2.2 obtains the answer entity $\hat{a}_t$ by identifying a top-ranked query graph, which starts from either an entity in $\mathcal{G}^{(t-1)}$ or a topic entity mentioned in q_t. Let $\mathcal{S}_t$ denote all the entities except $\hat{a}_t$ in this top-ranked query graph. The Graph Constructor adds the following nodes and edges to $\mathcal{G}^{(t-1)}$ in order to build $\mathcal{G}^t$.

- For each entity $e \in \mathcal{S}_t$, add e to the graph as a node if it does not exist in the graph yet. Also add $\hat{a}_t$ to the graph as a node if it does not exist yet.
- For each newly added node e, add a "self-loop" edge from e to itself.
- For each entity $e \in \mathcal{S}_t$, add a "forward" edge from e to $\hat{a}_t$.
- For each entity $e \in \mathcal{S}_t$, add a "backward" edge from $\hat{a}_t$ to e.
- For each entity $e \in \mathcal{S}_1$, i.e., the entities relevant to the first question, add a "backward" edge from $\hat{a}_t$ to e.

The way we construct the Entity Transition Graph as described above is based on the following observations with focal entities: (1) A focal entity is often an answer entity to a previous question.

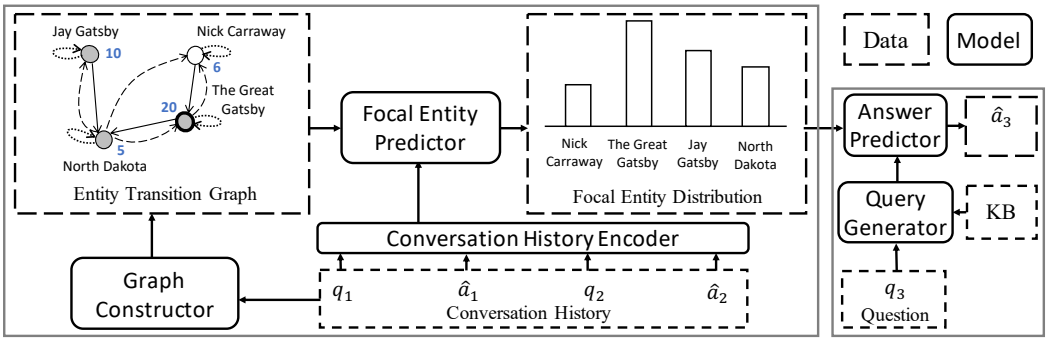

Figure 2: Architecture of our method. q_1, $\hat{a}_1$, q_2 and $\hat{a}_2$ correspond to the example conversation in Figure 1. Specifically, we show the prediction procedure for q_3, where the entities "*Nick Carraway*", "*The Great Gatsby*", "*Jay Gatsby*" and "*North Dakota*" form the Entity Transition Graph. After predicting the focal entity distribution at that stage, we leverage both the distribution and q_3 to generate $\hat{a}_3$. The single-turn KBQA system is shown inside the rectangle on the right and our proposed component is shown inside the rectangle on the left.

Therefore we include all previous answer entities in the graph. (2) A focal entity is also likely to be an entity relevant to a previous question that has led to the answer entity. We therefore also include those entities in the query graphs into the Entity Transition Graph. (3) The focal entity tends to stay unchanged and thus has a "stickiness" property in a conversation. Thus we add a self-loop edge for each node. (4) The focal entity may often go back to some entity relevant to the first question. Therefore, we always add an edge from the latest answer entity to entities relevant to the first question. (5) If an entity is frequently discussed in the conversation history, it might be more likely to be a focal entity. We thus give such entities more connectivities in the graph.

To give a concrete example of the Entity Transition Graph, let us take a look at Figure 3. When we answer Q2, "Nick Carrayway" and "The Great Gatsby" are included in the graph because the top-ranked query graph of Q1 contains the entity "Nick Carrayway" and returns the entity "The Great Gatsby". As the conversation proceeds, the Entity Transition Graph grows dynamically and we eventually obtain Figure 3 (d) when we answer Q5.

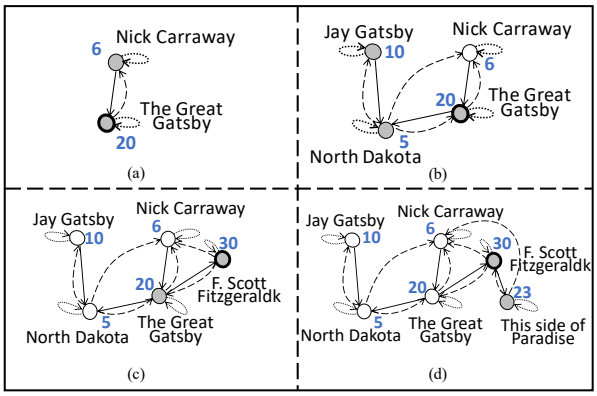

Figure 3: An example of the construction process of the Entity Transition Graph. (a) to (d) show the graph when we answer Q2 to Q5, respectively. The nodes in gray are the most recently added entities. The numbers in blue are the out-degrees of the entities in the KB, which are used in Section 3.4. The edges shown in solid, dashed or dotted lines indicate "forward", "backward" and "self-loop", respectively. The nodes highlighted with thick borders are the actual focal entities of the current questions.

3.3 Conversation History Encoder

The objective of the Conversation History Encoder is to encode the textual context of the previous questions and their predicted answers, particularly information other than the entities (which is already captured by the Entity Transition Graph). The output of the Conversation History Encoder is a single vector and it will be fed into the Focal Entity Predictor as an additional input.

Similar to previous methods (Serban et al., 2017; Saha et al., 2018), we leverage a hierarchical encoder to encode the conversation history, where a lower layer encodes individual questions and predicted answers independently and an upper layer connects the sequence of questions and answers to derive a single vector. Specifically, suppose we have completed $(t-1)$ turns of the conversation. The lower-layer encoder employs a standard sequence encoder (in our case a BiLSTM) to encode each question and each predicted answer so far. Let $\mathbf{q}_i \in \mathbb{R}^d$ $(1 \leq i \leq (t-1))$ denote the encoded vector representation of q_i, and similarly, let $\hat{\mathbf{a}}_i \in \mathbb{R}^d$ denote the encoded vector for $\hat{a}_i$. Next, the

upper-layer encoder leverages a recurrent network to encode the vector sequence $\mathbf{q}_1, \hat{\mathbf{a}}_1, \mathbf{q}_2, \hat{\mathbf{a}}_2, \ldots$ and generate a sequence of hidden vectors. The last hidden vector, which we denote as $\mathbf{h}_{t-1} \in \mathbb{R}^d$, will be used as the representation of the conversation history.

It is worth noting that although our Conversation History Encoder is similar to how previous work encodes conversation history (Serban et al., 2017), previous work uses the representation $\mathbf{h}_{t-1}$ directly as part of the representation of the current question, which introduces noise. In contrast, we use it to help predict our focal entity distribution only.

3.4 Focal Entity Predictor

The Focal Entity Predictor employs a graph convolution network (GCN) (Kipf and Welling, 2017; Schlichtkrull et al., 2018) to derive a focal score for each node in the Entity Transition Graph at each turn of the conversation. First, we assume that each entity (i.e., node) e in the graph has a vector representation, and this representation is updated at each turn. Let us use $\mathbf{e}_t$ to represent this vector at the t-th turn. For each interaction relation label (i.e., "forward", "backward" and "self-loop"), we also use a vector to represent it at each turn, which we denote as $\mathbf{r}_t$.

At the t-th turn, the vector representations of the entities and interaction relations are updated as follows:

$$\mathbf{e}_t = \sum_{(r,e') \in \mathcal{N}(e)} \alpha_r \mathbf{e'}_{t-1}, \quad (1)$$

$$\alpha_r = \text{softmax}_{(r,e') \in \mathcal{N}(e)}(\mathbf{h}_{t-1}^\intercal \mathbf{r}_{t-1}), \quad (2)$$

where $\mathcal{N}(e)$ is the set of nodes connect to e together with the connecting edges, and $\mathbf{h}_{t-1}$ is the output of the Conversation History Encoder as we have explained earlier. The formulas above show that the representation of e will be aggregated from the representations of its neighborhood entities from the last turn of the conversation, and the aggregation weights α are derived based on the conversation history $\mathbf{h}_{t-1}$ as well as the nature of the interaction relation.

For each node that is newly added to the Entity Transition Graph and each of the interaction relation labels, we initialize its vector representation to a random vector.

To derive the focal score of entity e at the current turn, we make use of both $\mathbf{e}_t$ and two additional features. Specifically, we obtain the out degree of each entity from the entire KB as one additional feature. We also assign a label to each entity to indicate whether it is from $\mathcal{S}_t$ (as defined in Section 3.2) or is $\hat{a}_t$. We denote these two features as $e^{\text{out-degree}}$ and $\mathbf{e}^{\text{temporal}}$, where $e^{\text{out-degree}}$ is a scalar and $\mathbf{e}^{\text{temporal}} \in \mathbb{R}^d$ is represented using embeddings.

We now concatenate $\mathbf{e}_t$ and $\mathbf{e}^{\text{temporal}}$ as well as $e^{\text{out-degree}}$ to derive focal scores as follows:

$$\tilde{\mathbf{e}}_t = [\mathbf{e}_t \oplus \mathbf{e}^{\text{temporal}} \oplus e^{\text{out-degree}}], \quad (3)$$

$$\text{FocalScore}_t(e) = \text{softmax}_{e \in \mathcal{G}_c}(\mathbf{w}_t^\intercal \tilde{\mathbf{e}}_t + b_t), \quad (4)$$

where $\oplus$ denotes concatenation, both $\mathbf{w}_t$ and b_t are parameters to be learned and they are specific to the t-th turn. Here $\text{FocalScore}_t(e)$ denotes the focal score, i.e., the probability that entity e would be the focal entity for the t-th question.

3.5 Training Objectives

Our training objective comes from two parts: First, we want to minimize the loss from incorrectly answering a question. For this, we use a standard cross entropy loss. Second, we want to supervise the training of the Focal Entity Predictor, but we do not have any ground truth for the focal entity distributions. We therefore produce pseudo ground truth as follows: If there is an entity that could generate at least one query graph resulting in the correct answer, we treat it as a correct focal entity for that question and assign a value of 1 to the entry for this entity in the distribution; otherwise, the value remains 0. Finally, we normalize the distribution and obtain a pseudo distribution. We then try to minimize the KL-divergence between this pseudo ground truth of focal entity distribution and our predicted focal entity distribution.

4 Experiments

In this section, we first introduce two benchmark datasets and our experiment settings in Section 4.1 and Section 4.2. Next, we discuss the main results and analysis in Section 4.3 and Section 4.4. We further show the comparison with SOTA systems in Section 4.5 and some error analysis in Section 4.6.

4.1 Data Sets

We use two datasets to evaluate our proposed method. The latest WikiData dump[3] is used as the KB for both datasets. Average accuracy and F1 score are employed to measure the performance.

[3] https://query.wikidata.org

ConvQuestions: This is a large-scale conversational KBQA dataset[4] created via Amazon Mechanical Turk (Christmann et al., 2019). The questions cover topics in five domains. Each conversation contains 5 sequential questions with annotated ground truth answers. There are many questions with missing information in the conversations, which makes the dataset very suitable for evaluating our method. The dataset contains 6K, 2K and 2K conversations for training, development and testing, each evenly distributed across domains.

ConvCSQA: This dataset comes from the the CSQA dataset[5] (Saha et al., 2018), originally created for a setting similar to conversational KBQA. However, one of the focuses of the original CSQA data was complex questions, which is not related to our work. Also, the CSQA data contains many questions in a conversation that do not have connections with preceding questions. We therefore elaborately selected conversational questions from CSQA to suit our needs, using the following strategies: 1) We collected the topic entities as well as the answer entities in the conversation history. If a follow-up question contains one of these entities, we kept the question; otherwise, we omitted it. 2) If the question type description did not explicitly mention that this question contains an "indirect" subject, we removed it. 3) We also filtered out the conversations with a length smaller than 5. As a result, we obtained a subset of CSQA that consists of 7K, 0.5K and 1K conversations for training, development and testing, respectively. The average number of questions per conversation is 5.36. We call this the **ConvCSQA** dataset.

4.2 Experiment Settings

To evaluate the effectiveness of our proposed Entity Transition Graph and Focal Entity Predictor, we mainly compare the following three methods:

SingleTurn: This is the method described in Section 2.2. Specifically, we first recognize the named entities in the questions via the AllenNLP NER tool[6] and retrieve the corresponding entities via SPARQL. To generate candidate query graphs, we consider all subgraphs that are 1 hop or 2 hops away from the topic entities (or focal entities in the case when the SingleTurn system is used in our method). Next, we employ the Answer Predictor that consists of two BiLSTMs to encode the question as well as each candidate subgraph independently. The final score is computed via the dot product of these two vectors.

ConvHistory: This method follows a standard way of encoding the conversational history using a two-level hierarchical encoder (Serban et al., 2017). It does not explicitly model any focal entity.

Our Method: This is our proposed method where we model the focal entities through the Entity Transition Graph and the Focal Entity Predictor. This method also uses the same hierarchical encoder as above to encode the conversation history.

Implementation Details: We implement our method by PyTorch on Nvidia V440.64.00-32GB GPU cards. We employ GloVe[7] as our initialized word embeddings and set the maximum number of GCN layers as 10. We apply grid search through pre-defined hyper-parameter spaces, specifically, hidden dimensionality amongst $\{200, 300, 400\}$, learning rate amongst $\{3e-3, 3e-4, 3e-5\}$ and dropout ratio amongst $\{0.2, 0.1, 0.0\}$. The best hyper-parameter configuration is based on the best F1 score on the development set. Eventually, for each neural network model, we set the hidden dimensionality to 300. A dropout layer is set before each MLP with a ratio of 0.1. We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of $3e-5$, and the batch size is 1. The training epoch number is 100.

4.3 Main Results

Table 1 shows the overall results. As we can see, our method clearly outperforms both SingleTurn and ConvHistory on both datasets. This confirms that with the additional components we added that model the focal entities, the method is able to make use of the conversation history more effectively to answer the follow-up questions compared with ConvHistory (which simply encode the entire history without specifically modeling focal entities). Surprisingly, we find that simply modeling the conversation history through a standard two-level hierarchical sequence model does not consistently

[4] https://convex.mpi-inf.mpg.de/

[5] https://amritasaha1812.github.io/CSQA/

[6] https://demo.allennlp.org/named-entity-recognition

[7] https://nlp.stanford.edu/projects/glove/

Methods	ConvQuestions		ConvCSQA	
	Dev	Test	Dev	Test
SingleTurn	29.7	27.3/30.5	61.8	56.8/65.0
ConvHistory	29.1	27.2/30.2	62.0	57.0/65.1
Our Method	31.9	**29.8/33.3**	63.2	**57.8/66.9**

Table 1: F1 results on development and Acc/F1 results on test of ConvQuestions and ConvCSQA.

Model Configuration	Acc/F1
Our Method (full model)	**29.8/33.3**
- historical conversation	28.7/32.1
- entity transition graph	28.2/31.6
- entity property	28.3/31.7

Table 2: The ablation results on ConvQuestions.

improve the performance. It suggests that including all the historical conversation information in a brute-force manner may not capture the most important conversation contexts effectively.

4.4 Further Analysis

Ablation Studies. Next, we remove the major components in Our Method one at a time and show the ablation results conducted on ConvQuestions in Table 2. Specifically, we 1) remove the effect of modeling conversation history by replacing α_r in Eqn. (1) with a uniform distribution; 2) remove graph information by replacing $\mathbf{e}_t$ with $\mathbf{h}_{t-1}$ in Eqn. (3); 3) remove entity property by omitting $e^{\text{out-degree}}$ in Eqn. (3). The results in Table 2 show that all the above information helps our method to predict focal entities accurately and achieve the best KBQA results.

Breakdown by Turns of Conversation. Our method is specifically designed for follow-up questions. Therefore, it would be interesting to see how the method fares for questions at different turns of the conversation. Is it more difficult to answer a question at a later turn of the conversation than an earlier question? We therefore show the results breakdown by turns of conversation in Table 3. We observe that as expected, for questions at later turns of a conversation, the performance drops for all three methods. We believe that for both ConHistory and Our Method, this is partially due to error propagation. On the other hand, compared with SingleTurn and ConvHistory, Our Method is still more robust when handling the follow-up questions at later turns of a conversation.

Methods	Q1	Q2	Q3	Q4	Q5
SingleTurn	49.2	33.5	22.1	19.6	12.3
ConvHistory	**50.1**	31.7	23.5	19.2	9.2
Our Method	50.0	**35.0**	**28.7**	**20.1**	**15.4**

Table 3: Accuracy results breakdown by conversation turns on ConvQuestions.

Case Studies. To verify if our predicted focal entities are meaningful, we use two concrete examples to conduct a case study. Figure 4 displays two example conversations from ConvQuestions. We show the focal entity distributions for the sequence of questions in bar charts. We can see that the predicted focal entity distribution indeed follows the flow of the conversation. For example, the entity with the largest focal score in the first conversation transits from "F.Scott Fitzgerald" to "Zelda Fitzgerald," and then to "St. Patrick's Cathedral," while in the second conversation it remains as "Tupac Shakur" throughout the conversation.

4.5 Comparison with SOTA

We compare our proposed method with existing state-of-the-art systems in Table 4. Our method outperforms other systems on most questions and achieves overall 9.5 and 14.3 percentage points of improvement on ConvQuestions and ConvCSQA, respectively. CONVEX, Star and Chain employ expansion-based or rule-based strategies to identify the answer entities for follow-up questions. HRED+KVmem combines the hierarchical encoder with a Key-Value Memory network. D2A and MaSP are two seq2seq models to translate the questions into logical forms. Our system is developed based on a standard single-turn KBQA system. We strengthen it by modeling focal entity transitions, and it shows outstanding capability in answering co-referenced, ellipsis and verification questions.

4.6 Error Analysis

To better understand where our method has failed, we randomly sampled and analysed 100 questions with wrong predictions and manually inspected them. We find that the errors are mainly due to the following reasons.

Mis-prediction of Relations (43%) The major errors come from relation mis-predictions. In our model, relation prediction is done by a simple answer predictor. We expect that employing a more advanced encoder could reduce this type of errors.

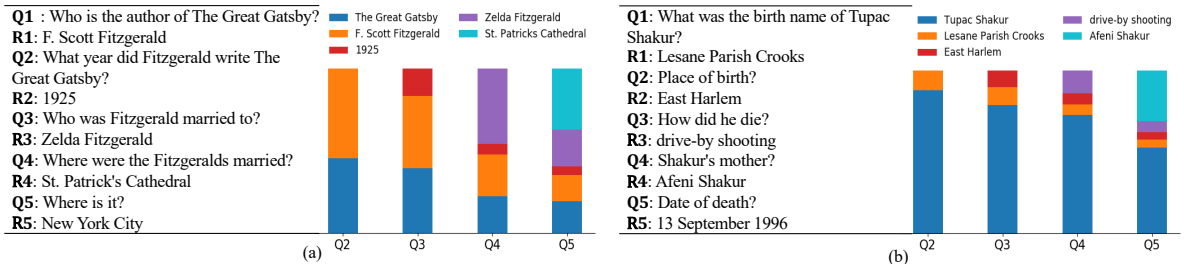

Figure 4: Two conversations in ConvQuestions and our predicted focal entity distributions. Each stacked bar shows the probabilities of the focal entity candidates for each question, where each entity is shown in its own color.

Methods	ConvQuestions					ConvCSQA			
	Movies	TV series	Music	Books	Soccer	QT_1	QT_2	QT_3	QT_4
CONVEX (Christmann et al., 2019)	25.9	17.8	19.0	19.8	18.8	38.9	14.8	4.6	47.8
Star (Christmann et al., 2019)	25.7	19.4	24.1	24.1	17.9	-	-	-	-
Chain (Christmann et al., 2019)	9.4	3.1	4.0	5.3	1.6	-	-	-	-
HRED+KVmem (Saha et al., 2018)	-	-	-	-	-	13.6	7.1	8.8	21.4
D2A (Guo et al., 2018)	9.0	6.7	7.2	12.1	10.7	61.0	43.4	4.7	45.8
MaSP (Shen et al., 2019)	-	-	-	-	-	**82.7**	45.2	3.8	46.3
Our Method	**29.0**	**30.4**	**30.1**	**30.1**	**29.6**	81.2	**64.6**	**23.1**	**58.0**

Table 4: Comparison with other systems. ConvQuestions results (Acc) are shown with different domains and ConvCSQA results (F1) are shown with different question types ("Simple", "Co-referenced", "Ellipsis" and "Verification"). Results of ConvQuestions are copied from (Christmann et al., 2019). Results of ConvCSQA are based on our re-implementation using the official source code[8].

Query Generation Failure (29%) There are many cases where the correct query graphs are difficult to be collected from the KB due to the incompleteness of the KB or the limitation of the query generator.

Mis-linking of Topic Entities (22%) The errors caused by wrong identification of the topic entities of questions also lead to incorrectness of the final answers, because if the entity linker links the question to a wrong entity, it is unlikely to answer the question correctly. This is a general challenge for KBQA.

5 Related Work

Single-turn KBQA task has been studied for decades. Traditional methods tried to retrieve the correct answers from the KB via either embedding-based methods (Bordes et al., 2014; Xu et al., 2019; Sun et al., 2018, 2019; Qiu et al., 2020; He et al., 2021) or semantic parsing-based methods (Berant et al., 2013; Yih et al., 2015; Luo et al., 2018; Zhang et al., 2019; Lan and Jiang, 2020). Conversational KBQA is a relatively new direction that builds on top of single-turn KBQA.

Conversational KBQA is related to dialogue systems and conversational QA in general, which require techniques to sequentially generate responses based on the interactions with users (Ghazvininejad et al., 2018; Rajendran et al., 2018; Das et al., 2017). A conversation history can be encoded via different techniques such as a hierarchical neural network (Serban et al., 2017; Reddy et al., 2019) or modeling the flow of the conversation along with a passage (Huang et al., 2019; Gao et al., 2019, 2020). Our work also intends to capture the flow of the conversation but we specifically model the transitions of focal entities.

Regarding conversational KBQA, Saha et al. (2018) proposed a model consisting of a hierarchical encoder, a key-value memory network and a decoder. Guo et al. (2018) and Shen et al. (2019) employed a seq2seq model to encode the conversation history then output a sequence of actions to form an executable command. Some follow-up work (Guo et al., 2019; Shen et al., 2020) focused on the meta-learning setting or the effective search strategy under weak supervision, which is beyond the focus of this paper. Christmann et al. (2019) detected frontier nodes by expanding a subgraph, which are potential answer entities to the current question.

[8]Since the original D2A and MaSP codes leverage the ground truth topic entities and relations to pre-train the entity linker and relation predictor but we do not, we skip the pre-training procedure in our re-implementation.

Their motivation is relevant to ours but we target at modeling the focal entities in the conversation.

6 Conclusion

In this paper, we present a method to model the transitions of focal entities in a conversation in order to improve conversational KBQA. Our method can outperform two baselines and achieve state-of-the-art performance on two benchmark datasets.

Acknowledgements

This research was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 615–620.

Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. 2019. Look before you hop: Conversational question answering over knowledge graphs using judicious context expansion. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 729–738.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual Dialog. In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–335.

Yifan Gao, Piji Li, Irwin King, and Michael R. Lyu. 2019. Interconnected question generation with coreference alignment and conversation flow modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4853–4862.

Yifan Gao, Chien-Sheng Wu, Shafiq Joty, Caiming Xiong, Richard Socher, Irwin King, Michael R. Lyu, and Steven C.H. Hoi. 2020. Explicit memory tracker with coarse-to-fine reasoning for conversational machine reading. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5110–5117.

Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-action: Conversational question answering over a large-scale knowledge base. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2942–2951.

Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2019. Coupling retrieval and meta-learning for context-dependent semantic parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 855–866.

Gaole He, Yunshi Lan, Jing Jiang, 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*.

Hsin-Yuan Huang, Eunsol Choi, and Wen tau Yih. 2019. FlowQA: Grasping flow in history for conversational machine comprehension. In *Proceedings of International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of International Conference on Learning Representations*.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 969–974.

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*, pages 5046–5052.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197.

Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny 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*, pages 2185–2194.

Yunqi Qiu, Yuanzhuo Wang, Xiaolong Jin, and Kun Zhang. 2020. Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 474–482.

Janarthanan Rajendran, Jatin Ganhotra, Satinder Singh, and Lazaros Polymenakos. 2018. Learning end-to-end goal-oriented dialog with multiple answers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3834–3843.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Amrita Saha, Vardaan Pahuja, Mitesh M. Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 705–713.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of European Semantic Web Conference*, pages 593–607.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, page 3295–3301.

Tao Shen, Xiubo Geng, Guodong Long, Jing Jiang, Chengqi Zhang, and Daxin Jiang. 2020. Effective search of logical forms for weakly supervised knowledge-based question answering. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2227–2233.

Tao Shen, Xiubo Geng, Tao Qin, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. 2019. Multi-task learning for conversational question answering over a large-scale knowledge base. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2442–2451.

Haitian Sun, Tania Bedrax-Weiss, and William W. 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*, pages 4231–4242.

Kun Xu, Yuxuan Lai, Yansong Feng, and Zhiguo Wang. 2019. Enhancing key-value memory neural networks for knowledge based question answering. In *Proceedings of The 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2937–2947.

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 (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 (Volume 2: Short Papers)*, pages 201–206.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero 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 (Volume 1: Long Papers)*, pages 571–581.

Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. 2019. Complex question decomposition for semantic parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4477–4486.