Incremental Knowledge-Based Question Answering

Yongqi Li, Wenjie Li, Liqiang Nie
1 The Hong Kong Polytechnic University
2 Shandong University
liyongqi0@gmail.com, wenjie.li@polyu.edu.hk, nieliqiang@gmail.com

Abstract
In the past years, Knowledge-Based Question Answering (KBQA), which aims to answer natural language questions using facts in a knowledge base, has been well developed. Existing approaches often assume a static knowledge base. However, the knowledge is evolving over time in the real world. If we directly apply a fine-tuning strategy on an evolving knowledge base, it will suffer from a serious catastrophic forgetting problem. In this paper, we propose a new incremental KBQA learning framework that can progressively expand learning capacity as humans do. Specifically, it comprises a margin-distilled loss and a collaborative exemplar selection method, to overcome the catastrophic forgetting problem by taking advantage of knowledge distillation. We reorganize the SimpleQuestion dataset to evaluate the proposed incremental learning solution to KBQA. The comprehensive experiments demonstrate its effectiveness and efficiency when working with the evolving knowledge base.

Introduction
Knowledge bases, such as DBpedia freebase and Wikidata, contain a large amount of facts about the real world. These facts are commonly represented as triples and each triple is in the form of \((s, r, o)\), where \(s\), \(r\) and \(o\) represent the subject, the relation and the object, respectively. The task of Knowledge-Based Question Answering (KBQA) aims to answer the questions presented in natural language using the relevant facts available in a knowledge base.

KBQA is an area well developed in the past years. The state-of-the-art research solution is to develop neural network-based models to learn the semantic similarity between a given question and the candidate facts in a knowledge base. A typical neural network-based KBQA model often involves two fundamental question analysis steps, i.e., entity linking and fact selection. First, the entity in the question is identified and linked to its corresponding entities in the knowledge base so that only a small subset of facts remains as candidates. Then, the answer to the question is extracted from the candidate fact that best matches the question. Essentially, the KBQA task is deemed as the matching problem with such models.

The current KBQA models all assume the static knowledge base, i.e., once the knowledge is acquired, it no longer changes. However, in the real world, the knowledge base is evolving over time while the new entities and relations are constantly added into it. There have been some work exploring evolving KBs for link prediction and knowledge reasoning (Garcia-Duran, Dumancic, and Niepert 2018; Trivedi et al. 2017). In question answering, when KB evolves, the questions related to the new knowledge should be readily answered, as illustrated in Figure 1. However, it is difficult for a typical KBQA model to answer these questions because it has no ability to detect the relations that are not available in training (Wu et al. 2019). To answer these questions, we can certainly re-train a new robust KBQA model over the entire data once the additional KB knowledge comes. Unfortunately, this is often impractical limited by the memory and computation resources, because it must rerun the model over the entire huge data even when the KB changes a little. Alternatively, it might be a better choice to fine-tune the parameters of the existing model with the new KB knowledge. However, when the model is fine-tuned using the new coming entities and relations alone, it will suffer from the serious problem of catastrophic forgetting (McCloskey and Cohen 1989). Since fine-tuning focuses more on the new KB knowledge, when the model learns to apply the new knowledge, it will get to forget about what it has learned before and its ability to apply the old KB knowledge more or less diminishes. Therefore, how to develop a more intelligent QA model that can gradually expand its capacity and continually learn new knowledge while still preserving existing knowledge in maximum is a big challenge. To the best of my knowledge, this problem has not yet been explored in the KBQA area.

We cast the aforementioned problem caused by knowledge base evolution into an incremental KBQA task. Incremental learning is critical to the problems where the data comes in a stream form (He et al. 2011) and only a hand-
Questions

Who is the author for The Old Man and the Sea?

What country produced the film Titanic?

What position in baseball does Michael Jordan play?

Knowledge Base

Ernest Miller Hemingway
USA

The Old Man and the Sea

Titanic

Michael Jordan
Shooting Guard

Related Work

KBQA

The previous KBQA models can be roughly divided into two categories, i.e., semantic parsing based (Yih, He, and Meek 2014) [green] and information retrieval based (Bordes et al. 2015) [green] etc. In our work, we consider inter-relationships among KB relations when selecting exemplars rather than random (Rebuffi et al. 2017), herding (Castro et al. 2018), generation (Shin et al. 2017), and end-to-end training (Liu et al. 2020), etc. In our work, we consider inter-relationships among KB relations when selecting exemplars rather than

Incremental Learning in Classification

Incremental learning has a long history in machine learning (Cauwenberghs and Poggio 2001). Recent studies were generally conducted in the incremental-class setting, i.e., classes come in a sequence. Grounded on the knowledge distillation technique (Hinton, Vinyals, and Dean 2015) [green], which was firstly applied in incremental learning by (Rebuffi et al. 2017), these studies tried to overcome the catastrophic forgetting problem via the distillation loss and exemplars. Several methods were proposed for exemplar selection, such as herding (Castro et al. 2018) [green] and end-to-end training (Liu et al. 2020), etc. In our work, we consider inter-relationships among KB relations when selecting exemplars rather than
only paying attention to the individual relations themselves. Moreover, different from the previous incremental learning approaches, which were specially proposed for classification, we design an incremental learning framework for matching-based KBQA.

Furthermore, there are some researches applying incremental learning to NLP tasks. For example, Shan et al. (2020) proposed a teacher-student model for text classification, Wang et al. (2019) studied incremental learning for task-oriented dialogue systems, where developers did not have to define user needs in advance. As for question answering, to the best of our knowledge, this is the first work to focus on the continual learning abilities of models in KBQA.

**Incremental KBQA**

**Problem Formulation**

We define the task of incremental KBQA as follows. Fact triples and the relevant questions are assumed to be incrementally available in sequence as the datasets $D_0, D_1, D_2, \ldots$, where $D_i = \{Q_i, F_i\}$. $Q_i$ is the set of questions and $F_i$ is the knowledge base represented by the fact triples $(s, r, o)$ at time $i$, where $s$, $r$, and $o$ represent the subject, the relation, and the object respectively. The relations involved in the facts given in $D_i$ are completely new compared with the previously available ones. At time $i$, only $D_i$ and a handful of samples from previous time steps (called exemplars) can be used to train the KBQA model and the learned model is evaluated on all the test data from $D_1$ to $D_i$. The challenge for a successful incremental KBQA model is not only to learn the ability to effectively answer questions using the accumulated knowledge but also to learn from the dynamic streaming data efficiently.

**Overview**

The main workflow of the proposed incremental KBQA approach is illustrated in Figure 2. Assume there are $N$ phases, including one initial phase and $N - 1$ incremental phases. The initial phase actually trains a KBQA model in a conventional way that feeds the positive and negative pairs of questions and fact triples in data $D_0$ to the model to learn the model parameter $\Theta_0$ (Section 3.3). In each subsequent incremental phase $i$, inspired by the work of incremental learning for classification (Rebuffi et al. 2017), we feed not only the question and fact triple pairs in $D_i$ but also a number of exemplars, i.e., $Exemplar_{0:i-1}$, selected from previous phases by applying the proposed collaborative exemplar selection strategy (Section 3.4). The incremental model in phase $i$ is then trained on data $D_i$ and $Exemplar_{0:i-1}$ collectively using a novel margin-distilled loss function (Section 3.5).

**Incremental KBQA Learning**

Given data $D_i = \{Q_i, F_i\}$ and $Exemplar_{0:i-1}$, the model $\Theta_i$ is expected to learn answering questions with the new knowledge and at the meanwhile avoiding to lose the ability to handle the questions relevant to the old knowledge. In all incremental phases, entity linking is implemented to generate positive and negative pairs for training, and a deep neural network model is followed to encode the text inputs into embedding vectors. The gradient approach is used to update the weights of the network with the margin-distilled loss function.

**Entity Linking.** Similar to the previous work (Dai, Li, and Xu 2016, Wang et al. 2018), we train a BiGRU-CRF model to detect the entity mentions in questions. Each question is then converted into a (mention, pattern) tuple, where the pattern is obtained by simply replacing the mention in the question with $<$ e $>$. For each question, candidate fact triples are selected as follows. (1) We firstly apply the entity linking as previous methods (He and Golub 2016, Hao et al. 2018). In this work, we employ the Active Entity Linker, to obtain the top-20 entities for each question. Active Entity Linker is proposed in (Yin et al. 2016) and the top-20 entities are publicly released. It has been shown to be able to achieve better coverage of ground truth. The triples whose subjects are within the top-20 entities are selected as candidate fact triples. (2) For each of the top-20 entities, we randomly select some other relations (Hsiao, Huang, and Chen 2017) to generate new triples as candidate fact triples. (3) Some triples randomly selected from the knowledge bases are also included in the set of candidate fact triples. We build the candidate fact triples set from various aspects to be consistent with previous KBQA approaches for effective training and fair comparison. What is different from the previous approaches is that we enlarge the number of candidate fact triples in order to more reliably evaluate the model's
ability to find the correct KB triple from a larger collection of candidates.

Now, for each question, which has been converted into (mention, pattern), we combine it with its truly linked fact (subject, relation) as the positive pair, and the question (mention, pattern) with the candidate fact triples excluding the true one as negative pairs. Besides, we denote the positive pair and negative pairs in $D_i$ as $pair_i^+$ and $pair_i^-$, and those pairs in $Exemplar_{0:i-1}$ as $pair_i^+$ and $pair_i^-$, respectively.

**Text Encoding.** We feed the texts of mention, pattern, subject, and relation into a deep neural network model to obtain their corresponding embedding vectors. This model plays the same role as the embedding layer in most conventional KBQA models. We choose the model presented in [Yin et al., 2016]. Specifically, the mentions and subjects are encoded by a char-based CNN network. Considering they are short, the generated representations are more robust even in the presence of typos, spaces and other character violations via character-level rather than commonly-used word-level encoding. As for patterns and relations, a word-level CNN with attentive maxpooling is applied. These obtained embeddings are used to train the model using the proposed margin-distilled loss, which will be introduced later.

**Collaborative Exemplar Selection**

In the $i$-th phase, the model can be retrained using all the relations and their corresponding questions accumulated up to this phase, i.e., $D_0, D_1, \ldots, D_i$, the model $\Theta_i$ can learn to answer both the new and old types of questions well. However, when the amount of the data increases, the retraining will consume more and more computing resources. Due to memory limitation, retraining with the entire available data is often impossible. In practice, only a handful of samples are selected as $Exemplar_i$ from each phase $i$ to retrain the model $\Theta_j$ ($j > i$). Typically, the number of exemplars is set to be much smaller than that of the original data. As a result, how to select the most representative and effective exemplars always plays a very important role in incremental learning.

The previous incremental learning approaches focus on the classification task generally select the samples that are near to the average vector (called herding) for per class. Considering the particularity of KBQA where the relations have their own text inputs, we propose a novel collaborative exemplar selection strategy based on the semantic information carried in texts. Different from the previous approaches that only consider the interior of one class, our approach considers semantic relationships among relations to collaboratively select the "best" learning samples from a global view. Inspired by the importance of support vectors in Support Vector Machine (SVM) [Suykens and Vandewalle, 1999], we assume that the samples near the boundary between relations contain more significant information for the model to detect the relation. For a relation, we consider the other relations semantically similar to it. As such, we are able to select the most effective samples for all the related relations collaboratively rather than only replying on the individual relations alone.

**Algorithm 1 Collaborative Exemplar Selection.**

**Require:**

- The input data in phase $i$, $D_i$;
- The model parameters, $\Theta_i$;
- The previous exemplars, $Exemplar_{0:i-1}$
- The number of considered relations, $m$;
- The number of samples selected for each considered relation, $n$;

**Ensure:**

- The selected exemplars from $D_i$, $Exemplar_i$;

1: $Exemplar_i = []$
2: for relation $r_1$ in $D_i$:
3: \hspace{10pt} $V_{r1} = \text{Model}(\Theta_i, r_1)$
4: \hspace{10pt} $s1 = []$
5: \hspace{10pt} for relation $r_2$ in $D_i \cup Exemplar_{0:i-1}$:
6: \hspace{15pt} $V_{r2} = \text{Model}(\Theta_i, r_2)$
7: \hspace{15pt} s1.append($\cos(V_{r1}, V_{r2})$)
8: \hspace{10pt} $s1 = \text{argmax}(s1)[:m]$  
9: \hspace{10pt} for relation $r_3$ in s1:
10: \hspace{15pt} $s2 = []$
11: \hspace{15pt} for pattern corresponding to relation $r_1$:
12: \hspace{20pt} $V_{pattern} = \text{Model}(\Theta_i, pattern)$
13: \hspace{20pt} s2.append($\cos(V_{r3}, V_{pattern})$)
14: \hspace{15pt} $s2 = \text{argmax}(s2)[:n]$  
15: \hspace{10pt} $Exemplar_i$.add(s2)
16: RETURN $Exemplar_i$;

The exemplar selection procedure is specified in Algorithm 1. For each relation $r_1$ in $D_i$, we firstly feed its text to the encoder to obtain its embedding vector $V_{r1}$. Similarly, we obtain the embedding vector for each relation $r_2$ in $D_i$ and $Exemplar_{0:i-1}$. Then, we calculate the cosine similarities between the relation $r_1$ and $r_2$. We select the most $m$ similar relations. Next, for each relation $r_3$ from the $m$ most similar relations, we select $n$ questions corresponding to the relation $r_3$ as exemplars. Specifically, we input all the patterns corresponding to the relation $r_3$ to the model and obtain embedding vectors. We then calculate the cosine similarities between relation $r_3$ and all the patterns. Furthermore, we select the $n$ most similar ones because we think these samples allow the model to distinguish similar relations. At the end $n$ samples for $m$ relations are selected.

**Margin-Distilled Loss**

Knowledge distillation is a technique initially proposed to transfer information between different neural networks or network structures. In this work, we apply it to a single model to distill knowledge between different time steps. It is expected that the knowledge in the model is retained as much as possible over time to overcome the catastrophic forgetting problem as mentioned before.

We design a margin-distilled loss, a joint loss function combining a margin loss and a mean squared error (MSE) loss, especially for the KBQA matching problem. We feed the samples in the new data to the margin loss, which is same as those used in most previous KBQA approaches learning to rank the candidate fact triples. The margin loss term guaran-
tees the model to learn the new KB knowledge. Meanwhile, we feed the samples in the exemplars to the MSE loss. This term is a distillation term that guides the model to retain the old KB knowledge across time steps. In short, the margin loss is applied to the samples from $D_i$ while the MSE loss is used on the samples from $Exemplar_{0:i-1}$.

Specifically, for each pair, let’s denote the embedding vectors of the mention, the pattern, the subject and the relation obtained via text encoding as $V_{mention}$, $V_{pattern}$, $V_{subject}$, and $V_{relation}$, respectively. The matching similarity score is calculated as follows,

$$s^+_D = \cos(V_{mention}, V_{subject}) + \cos(V_{pattern}, V_{relation}),$$

(1)

where $\cos()$ is a cosine similarity function and the mention, the pattern, the subject and the relation are in $pair^+_D$. The similarity scores for $pair^-_D$, $pair^+_E$, and $pair^-_E$ computed by Eq. (1) are denoted as $s^-_D$, $s^+_E$, and $s^-_E$, respectively. Formally, the margin-distilled loss function is defined as follows:

$$L = L_M + L_S,$$

(2)

where $L_M$ is the margin loss and $L_S$ is the MSE loss.

The margin $L_M$ is computed by

$$L_M = \max(0, \lambda + s^-_D - s^+_D),$$

(3)

where $\lambda$ is a hyper parameter, $s^+_D$ and $s^-_D$ are the similarity scores of the positive pair and negative pair in $D_i$, respectively. Based on the margin loss, the model learns to increase similarity scores of positive pairs and decrease similarity scores of negative pairs. The MSE loss $L_S$ is formulated by,

$$L_S = (s^+_E - l^+_E)^2 + (s^-_E - l^-_E)^2,$$

(4)

where $s^+_E$ and $s^-_E$ are the similarity scores of the positive pair and negative pair in $Exemplar_{0:i-1}$, respectively. $l^+_E$ and $l^-_E$ are the labels calculated when samples are selected as exemplars in the preceding phases. For example, assume some samples have been selected as exemplars and the model $\Theta_j$ has been trained in phase $j$ ($j < i$). Then, for an exemplar, we obtain the positive and negative pairs via the entity linking step as mentioned before. Next, we load $\Theta_j$ and feed the positive and negative pairs into the model to obtain the embedding vectors. The two similarity scores of positive and negative pairs, i.e., $l^+_E$ and $l^-_E$ calculated by Eq. (1) are deemed as target labels in the following phases. We can see that the MSE loss allows the model to keep the previous scores from the old samples as much as possible. As a result, the model can still keep a fair performance on the old samples when it learns the new knowledge. In general, this is how we alleviate the catastrophic forgetting problem.

### Experiments and Evaluations

#### Dataset

As claimed before, there is no appropriate QA dataset that contains the evolving knowledge and QA pairs for this new task. Following the previous work (Rebuffi et al. [2017]; Liu et al. [2020]) in incremental learning, we re-organize the

| Dataset | $D_0$ | $D_1$ | $D_2$ | $D_3$ | $D_4$ |
|---------|-------|-------|-------|-------|-------|
| # q in train | 18,001 | 22,055 | 10,768 | 11,672 | 15,961 |
| # q in validation | 2,250 | 2,757 | 1,346 | 1,459 | 1,995 |
| # q in test | 2,251 | 2,757 | 1,346 | 1,460 | 1,996 |
| # relations in $F_i$ | 1696 | 3,392 | 5,088 | 6,784 | 8,480 |
| # relations of q | 330 | 305 | 339 | 329 | 336 |

SimpleQuestion (SQ) Dataset (Bordes et al. [2015]) to evaluate and compare the performance of KBQA models on the incremental setting. The SQ dataset consists of 108,442 single-relation questions and their corresponding fact triples $(s, r, o)$.

We build the Incremental SimpleQuestion (Incremental-SQ) dataset from the SQ dataset as follows. (1) We firstly randomly shuffle and split the relation types into 5 sets $R_i$, where $i = 0 \to 4$. (2) Then we create 5 fact sets $F_i = \{(s, r, o)\}$, where $r \in \sum_{j=1}^i R_j$. (3) Afterwards, we construct 5 question sets $Q_i$, where these questions in $Q_i$ match with the relations in $R_i$. (4) Finally, we obtain 5 sub-datasets $D_i = (Q_i, F_i)$, where $i = 0 \to 4$. It is guaranteed that for every question in $Q_i$, its corresponding fact triple is in $F_i$, and $F_{i-1}$ is the subset of $F_i$, which reflects the evolving knowledge base where the new relations types and entities are added successively. The statistics of the datasets is summarized in Table 1. The reported experimental results in this paper are based on these datasets. Specifically, we use 80% of questions in each $Q_i$ to form the training set and the rest 10% as the validation and test sets, respectively. It is worth mentioning that the number of the relations relevant to the questions in $Q_i$ is different from that in $F_i$, since not all relations in a knowledge base have the corresponding questions.

#### Experimental Setups

**Implementation Details.** In our approach, the dimensions of character embedding and word embedding are both set to 128. The CNN is equipped with 128*2 filters in two distinct sizes [2, 3]. The margin $\lambda$ in Eq. [3] is 0.5. The number of relations $m$ used for selecting exemplars for each relation is 5 and $n$ is 2. We optimize the parameters using Adam with the initial learning rate 0.001, and the batch size is 256.

**Evaluation Metric.** The evaluation metric is the accuracy. For each phase $i$, we calculate the accuracy $Accuracy_i$ as the number of the correctly answered questions divided by the total number of questions, where the questions are those in $Test_0$ to $Test_i$, i.e., the test sets in $D_0$ to $D_i$. If a single number is preferable, we report the average of these $Accuracy_i$ as $Accuracy_{all}$ to reveal an average performance on all $N$ phases as follows:

$$Accuracy_{all} = \frac{\sum_{i=0}^{N-1} Accuracy_i}{N}.$$

(5)

*See Appendix A. for the details of the corresponding validation performance for each reported test result, the number of hyperparameter search trials and the expected validation performance.
Table 2: Performance comparison between our proposed model and the previous KBQA methods. We calculate \textit{Accuracy}_i in each phase \(i\) and \textit{Average}_a for all phases.

| Approach                     | \textit{Accuracy}_0 | \textit{Accuracy}_1 | \textit{Accuracy}_2 | \textit{Accuracy}_3 | \textit{Accuracy}_4 | \textit{Average}_a |
|------------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Yin et al.                   | 0.8956               | 0.8225               | 0.7711               | 0.7144               | 0.6635               | 0.7734               |
| Golub et al.                 | 0.8573               | 0.71                 | 0.5211               | 0.4792               | 0.4595               | 0.6047               |
| Hao et al.                   | 0.8764               | 0.8303               | 0.7982               | 0.7555               | 0.7743               | 0.8109               |
| Ours                         | 0.8956               | 0.8648               | 0.8396               | 0.8317               | 0.8312               | 0.8525               |
| Ours w/o distillation loss   | 0.8956               | 0.8097               | 0.7950               | 0.7871               | 0.7639               | 0.8103               |
| Ours with random exemplars   | 0.8956               | 0.8389               | 0.8107               | 0.8094               | 0.8077               | 0.8324               |
| Ours with herding exemplars  | 0.8956               | 0.8453               | 0.8198               | 0.8179               | 0.8153               | 0.8388               |
| Upper bound                  | 0.8956               | 0.8670               | 0.8478               | 0.8494               | 0.8486               | 0.8617               |

Figure 3: (a) Performance comparison between our proposed model and the previous KBQA methods. (b) Performance comparison between our proposed method and other exemplar selection methods; (c) Parameter analysis of the number of exemplars; and (d)-(e) the time and memory consuming between our method and the upper bound, respectively.

**Overall Performance**

To demonstrate the effectiveness of our proposed framework, we compare it with the following methods. \cite{Yin2016} utilized an attentive CNN network to encode questions and fact triples at both word and character levels. \cite{He2016} leveraged a character-level attention-based encoder-decoder LSTM model. \cite{Hao2018} added the pattern-revising procedure after entity linking to mitigate the error propagation problem. The performance of the above approaches is state-of-the-art on the Simple-Question dataset. As claimed by \cite{Mohammed2018}, more sophisticated models are unnecessary. Since these models are developed for conventional KBQA, we fine-tune them with the new KB knowledge at each time step to suit for the incremental KBQA setup. Specifically, for phase \(i\), we load the model parameters in the phase \(i - 1\) and continue training on the current \(D_i\). We also report the upper bound performance, which retrains a new robust KBQA model over the entire data in each phase. The results of all models are summarized in Table 2 and the curves of the \textit{Accuracy}_i for inspection are plotted in Figure 3(a) from which we obtain several observations as follows.

1. As the time step increases, the performances of all models decrease and are certainly below the upper bound. This clearly demonstrates the existence of the catastrophic forgetting problem mentioned before, i.e., the problem that the model forgets the old knowledge when it learns the new knowledge. The performance of the upper bound also declines in the first two incremental phases. That is because the results of entity linking in these two phases are worse.

2. In Table 2 it is obvious that our model obtains the best average accuracy \textit{Accuracy}_a (excluding the upper bound), which verifies that the proposed framework is able to effectively alleviate the catastrophic forgetting problem and retains the evolving knowledge better than other models. In the initial phase, our model obtains the same accuracy as \cite{Yin2016}, because we apply the same encoding method as it. It is noticed that the model of \cite{He2016}
It thus provides more effective information for the model to select the samples that near the boundary among relations. To verify the robustness of our method, we test the statistical significance of the average similarity scores for each considered relation in Algorithm 1. The results are shown in Figure 3(d) and Figure 3(e) respectively. We can see that as phases increases, the time and memory consumed by the upper bound increase linearly, which is because that the data is accumulated and expands linearly. As for our method, the time and memory are consistent with the number of data in each phase. There is an abrupt increase in phase 1, because the data in phase 1 is comparatively big as shown in Table 1.

### Component-wise Evaluation

Next, we remove the distillation loss, i.e., $L_s$ in Eq. 2, to further investigate the effectiveness of our proposed margin-distilled loss. Specifically, we feed exemplars to the text encoder and then calculate the cosine similarities. Rather than putting the similarity scores into the distilled loss term as before, we put them into the margin loss $L_m$. This result is included in Figure 3(b) and Table 2 being “Ours w/o distillation loss”. By doing such a comparison, we gain the following insights.

1. Without the distillation loss, our model still performs better than Yin et al. although both use the same text encoder. It demonstrates that the exemplars themselves are useful to retain the old knowledge. It is expected that if the number of exemplars gets larger, the performance will be much better.
2. We can see that “Ours w/o distillation loss” is worse than “Ours”, meaning that the distillation loss takes full advantages of exemplars to retain the old knowledge.

Our framework works well because we rely on not only a handful of previous samples but also the margin-distilled loss, which actually plays an important role.

### Verification of Collaborative Exemplar Selection

We further conduct the experiments to compare our proposed collaborative exemplar selection methods with some other exemplar selection alternatives. The simplest choice is to randomly select samples for each relation. The herding-based exemplar selection applied in Rebuffi et al. (2017) and Castro et al. (2018) selects the samples near the average embedding vector of a relation. We replace the exemplar selection method in our model with the above two, and the results are illustrated in Figure 3(c). It is found that herding-based selection surpasses random selection. This may be due to the fact that the samples around the centers are more representative. It can be also seen that our collaborative exemplar selection method exceeds the other two methods.

Our method selects the samples that near the boundary among relations. It thus provides more effective information for the model to distinguish similar relations. To verify the robustness of our propose method, we test the statistical significance of the $Average_n$ difference between Ours and Ours with herding exemplars (the best among the compared methods). The p-value is smaller than 0.01.

To understand the influence of the number of exemplars, we conduct experiments by changing the number of samples $n$ selected for each considered relation in Algorithm 1. The experimental results are illustrated in Figure 3(c) where we can see that the performance improves and approaches to the upper bound when $n$ increases.

### Catastrophic Forgetting Handling

As analyzed before, one of the challenges in the incremental KBQA task is the catastrophic forgetting problem, i.e., the model may forget the old knowledge when learning the new KB knowledge. To reveal this problem, in each phase $i$, we report the accuracy on each available test set, i.e., Test$_0$ to Test$_i$, respectively. So, we can see the variation tendency on each test set as time steps increases.

We compare the method (Yin et al. 2016) with ours for fair comparison, because we both use the same text encoder. Table 3 summarizes the detailed results of Yin et al. on each available test set in every phase. It can be seen that the performance on each test set declines as time step increases. For example, the accuracy on Test$_0$ declines over phase 0, 1, 2 3, 4. Initially, the accuracy is 0.8956 and in the last phase it already decreases to 0.5852. This is because the knowledge learned in the first phase is being forgetting as time step increases. It again verifies the existence of the catastrophic forgetting problem. However, compared with Yin et al., our method performs much better, see Table 4. For example, the accuracy on Test$_0$ declines over phase 0, 1, 2 3, 4. Initially, the accuracy is 0.8956 and it only drops about 0.02 at the end. Although the catastrophic forgetting problem still somewhat exists, our model can alleviate it well.

### Conclusion

In this paper, we propose and study the incremental KBQA task by focusing on learning from the evolving knowledge
base. Considering the complexity of incremental KBQA, we design a new framework with emphasis on a novel margin-distilled loss and a semantic-based collaborative exemplar selection method. We re-organize the SimpleQuestion dataset to form an Incremental SimpleQuestion dataset to reveal the problem and evaluate the models.

References

Bordes, A.; Usunier, N.; Chopra, S.; and Weston, J. 2015. Large-scale simple question answering with memory networks. arXiv preprint arXiv:1506.02075.

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-end incremental learning. In Proceedings of the European Conference on Computer Vision (ECCV), 233–248.

Cauwenberghs, G.; and Poggio, T. 2001. Incremental and decremental support vector machine learning. In Advances in neural information processing systems, 409–415.

Dai, Z.; Li, L.; and Xu, W. 2016. CFO: Conditional Focused Neural Question Answering with Large-scale Knowledge Bases. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 800–810.

Dong, L.; Wei, F.; Zhou, M.; and Xu, K. 2015. Question answering over freebase with multi-column convolutional neural networks. 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), 260–269.

Garcia-Duran, A.; Dumančić, S.; and Niepert, M. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 4816–4821.

Hao, Y.; Liu, H.; He, S.; Liu, K.; and Zhao, J. 2018. Pattern-revising enhanced simple question answering over knowledge bases. In Proceedings of the 27th International Conference on Computational Linguistics, 3272–3282.

He, H.; Chen, S.; Li, K.; and Xu, X. 2011. Incremental learning from stream data. IEEE Transactions on Neural Networks 22(12): 1901–1914.

He, X.; and Golub, D. 2016. Character-Level Question Answering with Attention. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 1598–1607.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Hsiao, W.-C.; Huang, H.-H.; and Chen, H.-H. 2017. Integrating subject, type, and property identification for simple question answering over knowledge base. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 976–985.

Lin, X. V.; Socher, R.; and Xiong, C. 2018. Multi-Hop Knowledge Graph Reasoning with Reward Shaping. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 3243–3253.

Liu, Y.; Su, Y.; Liu, A.-A.; Schiele, B.; and Sun, Q. 2020. Mnemonics Training: Multi-Class Incremental Learning without Forgetting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12245–12254.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In Psychology of learning and motivation, volume 24, 109–165.

Mohammed, S.; Shi, P.; and Lin, J. 2018. Strong Baselines for Simple Question Answering over Knowledge Graphs with and without Neural Networks. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 291–296.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2001–2010.

Reddy, S.; Lapata, M.; and Steedman, M. 2014. Large-scale semantic parsing without question-answer pairs. Transactions of the Association for Computational Linguistics 2: 377–392.

Shan, G.; Xu, S.; Yang, L.; Jia, S.; and Xiang, Y. 2020. Learn#: A Novel Incremental Learning Method for Text Classification. Expert Systems with Applications 113198.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. In Advances in Neural Information Processing Systems, 2990–2999.

Suykens, J. A.; and Vandewalle, J. 1999. Least squares support vector machine classifiers. Neural processing letters 9(3): 293–300.

Trivedi, R.; Dai, H.; Wang, Y.; and Song, L. 2017. Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, 3462–3471.

Wang, W.; Zhang, J.; Li, Q.; Hwang, M.-Y.; Zong, C.; and Li, Z. 2019. Incremental Learning from Scratch for Task-Oriented Dialogue Systems. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 3710–3720.

Wang, Y.; Zhang, R.; Xu, C.; and Mao, Y. 2018. The APVA-TURBO approach to question answering in knowledge base. In Proceedings of the 27th International Conference on Computational Linguistics, 1998–2009.

Wu, P.; Huang, S.; Weng, R.; Zheng, Z.; Zhang, J.; Yan, X.; and Chen, J. 2019. Learning Representation Mapping for Relation Detection in Knowledge Base Question Answering. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 6130–6139.
Xu, K.; Wu, L.; Wang, Z.; Yu, M.; Chen, L.; and Sheinin, V. 2018. Exploiting Rich Syntactic Information for Semantic Parsing with Graph-to-Sequence Model. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 918–924.

Yih, W.-t.; He, X.; and Meek, C. 2014. Semantic parsing for single-relation question answering. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 643–648.

Yin, W.; Yu, M.; Xiang, B.; Zhou, B.; and Schütze, H. 2016. Simple Question Answering by Attentive Convolutional Neural Network. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 1746–1756.

Zhang, Y.; Dai, H.; Kozareva, Z.; Smola, A. J.; and Song, L. 2018. Variational reasoning for question answering with knowledge graph. In Thirty-Second AAAI Conference on Artificial Intelligence.

Zhang, Y.; Liu, K.; He, S.; Ji, G.; Liu, Z.; Wu, H.; and Zhao, J. 2016. Question answering over knowledge base with neural attention combining global knowledge information. arXiv preprint arXiv:1606.00979.

Zhu, L.; Ikeda, K.; Pang, S.; Ban, T.; and Sarrafzadeh, A. 2018. Merging weighted SVMs for parallel incremental learning. Neural Networks 100: 25–38.