Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 1202315, 14 pages
https://doi.org/10.1155/2023/1202315

Research Article

Multigranularity Pruning Model for Subject Recognition Task under Knowledge Base Question Answering When General Models Fail

Ziming Wang, Xirong Xu, Xiaoying Song, Haochen Li, Xiaopeng Wei, and Degen Huang

School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China

Correspondence should be addressed to Xirong Xu; xirongxu@dlut.edu.cn

Received 3 March 2023; Revised 5 October 2023; Accepted 19 October 2023; Published 30 October 2023

Academic Editor: Mohammad R. Khosravi

In general knowledge base question answering (KBQA) models, subject recognition (SR) is usually a precondition of finding an answer, and it is a common way to employ a general named entity recognition (NER) model such as BERT-CRF to recognize the subject. However, in previous researches, the difference between a NER task and a SR task is usually ignored, and a wrong entity recognized by the NER model will certainly lead to a wrong answer in the KBQA task, which is one bottleneck for KBQA performance. In this paper, a multigranularity pruning model (MGPM) is proposed to answer a question when general models fail to recognize a subject. In MGPM, the set of all possible subjects in the Knowledge Base (KB) is pruned by 4 multigranularity pruning submodels successively based on the constraint of relation (domain and tuple), string similarity, and semantic similarity. Experimental results show that our model is compatible with various KBQA models for both single-relation and complex questions answering. The integrated MGPM model (with the BERT-CRF model) achieves a SR accuracy of 94.4% on the SimpleQuestions dataset, 68.6% on the WebQuestionsSP dataset, and 63.7% on the WebQuestions dataset, which outperforms the original model by a margin of 3.6%, 8.6%, and 5.3%, respectively.

1. Introduction

Knowledge base question answering (KBQA) is an important natural language processing (NLP) task which is aimed to answer natural language questions automatically with facts in a knowledge base (KB). In general, there are 4 subtasks under KBQA: subject recognition (SR), entity linking, relation prediction, and answer retrieval. In SR, the subject entity in an input question is recognized, which is an entity (or a set of entities with a same name) in a KB. In entity linking, a unique entity is selected from the entity set. In relation prediction, a relation in the KB is selected as the best one to describe the question. In answer retrieval, one or more entities can be retrieved from the KB based on the subject and relation, and a unique entity is selected from them as the answer to the question.

An example is shown in Figure 1. A question "What poetry did Shakespeare write in 1604?" is fed to a KBQA system. First, in the SR task, the entity "Shakespeare" in the KB is recognized as the subject entity to the question. As there are several entities (e.g. a person named Shakespeare, a book titled Shakespeare) which have the same name "Shakespeare" in the KB, in the Entity Linking task, the unique entity "Shakespeare" with the attribute "person" is selected from all entities named "Shakespeare." Then, all relation candidates (e.g., write, birthplace, country, etc.) with the subject "Shakespeare" can be retrieved from the KB and the best-matched one "write" is selected in the relation prediction task. Finally, entities with the subject "Shakespeare (person)" and relation "write" can be retrieved from the KB, which represents all creations of Shakespeare in this example, and the best-matched one "The Sonnets" is selected in the answer retrieval task. In general, the input of each task is the input question and a set of candidates retrieved from the KB based on the output of the previous task (except the

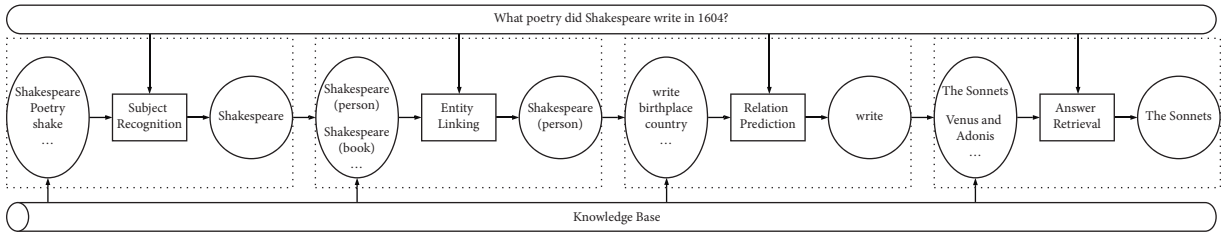

Figure 1: An example of a KBQA system.

first task SR), and the output of each task is an element of the candidate set.

In practical KBQA systems, there could be some differences in these tasks. For example, to a complex question such as "What poetry did the author of Venus and Adonis write in 1604?," several relations are required in the relation prediction task. To most questions in the SimpleQuestions dataset, a unique answer could be retrieved based on a subject name and a relation, so the entity linking task and answer retrieval task could be integrated with other tasks.

In general KBQA models, a named entity recognition (NER) model (e.g., BERT-CRF) is usually employed to recognize the entity which contains one or several successive words in a question as the subject entity in the SR task. In these models, a correct subject is the precondition of a correct relation and answer. Unfortunately, there are no models which could achieve an accuracy of 100% so there are always some questions where no or wrong entities and subjects are recognized and matched. In general, they fail for mainly 3 reasons:

(i) Question with abnormal subject (QWAS): the golden subject in the KB cannot be strictly matched to any n-grams generated from the question [1]. For example, to question "what is the name of a location in brasília standard time," the golden subject "brasília time zone" in the KB cannot be strictly matched to "brasília standard time" in the question.

(ii) No subject matched (NSM): the recognized entity (to a normal question) by the NER model cannot be strictly matched to any subject in the KB, because it contains no, wrong, not enough, or redundant words. For example, to question "what type of music does David ruffin play" with the golden subject "David ruffin," the recognized entity "play," "ruffin," or "David ruffin play" would lead to no subject matched.

(iii) Wrong subject matched (WSM): the recognized entity (to a normal question) by the NER model is strictly matched to a wrong subject in the KB. Maybe it is a correct entity in a general NER task, but it will lead to a wrong subject in a KBQA task. In the aforementioned example, "music" is a subject in the KB, but it is not the golden subject to this question.

In aforementioned examples, a correct entity in a NER task is not necessarily a correct subject in a SR task. In general, there are mainly 3 differences between a NER task and a SR task under KBQA:

(i) Aim: the aim of a NER task is to recognize several successive words in an input sentence as the entity, whereas the aim of a SR task is to select one entity in the KB as the subject of an input question.

(ii) Number: there could be no, one, or several entities for an input sentence in a NER task, whereas there should be one and only one subject for an input question in a SR task.

(iii) Constraint: a NER task is usually an independent task and there are no extra constraints for the recognized entity, whereas a SR task is a subtask of KBQA and there are mutual constrains between the subject and relation of an input question.

As a result, besides employing an error correction model to reduce QWAS and an improved NER model to reduce NSM and WSM, a compatible solution is strongly required to focus the difference between a NER task and a SR task and answer questions when general models fail. In this paper, we propose the multigranularity pruning model (MGPM) to recognize subjects and answer questions in these cases. Compared with general NER models, the search space in our model is not successive words in questions, but subjects in the KB, so our model could find correct subjects even to some QWAS. The original massive search space in the KB is narrowed down by 4 multigranularity submodels gradually based on relation dependence, string similarity, and semantic similarity. In this way, the subject with the highest score (calculated by our model) would be considered as the recognized subject and then a general KBQA model could be employed to answer the question based on the recognized subject.

Our main contributions are as follows:

(i) We focus on the difference between a NER task and a SR task and propose a method which is still effective in the case that general models fail to recognize subjects.

(ii) Our method is dataset-agnostic and effective on datasets of both simple questions and complex questions.

(iii) Our method is model-agnostic and compatible with various KBQA models. Experimental results show that the integrated MGPM model with the BERT-CRF model (or Efficient GlobalPointer, EGP)

outperforms the original model by a margin of 3.6% (or 4.7%) on the SimpleQuestions dataset, 8.6% (or 8.1%) on the WebQuestionsSP dataset, and 5.3% (or 4.8%) on the WebQuestions dataset.

2. Related Work

The research of KBQA has evolved from earlier domain-specific question answering [2] to open-domain QA based on large-scale KBs such as Freebase [3]. The model of KBQA has also evolved from semantic parsing-based models [4], which parse questions into structured queries, to neural network-based models [5, 6], which learn semantic representations of both the question and the knowledge from observed data. Some researchers [7–9] also attempt to combine multiple models to utilize information in natural language questions and KBs.

After pretrained models such as BERT [10], ALBERT [11], XLNet [12], and ELECTRA [13] are proposed, they have been widely employed in various NLP tasks [14–17]. Many researches employed NER and RE models based on pretrained models and achieved good results. For example, Gangwar et al. [18] employed pretrained models in the span extraction, classification, and relation extraction task focused on finding quantities, attributes of these quantities, and additional information. Luo et al. [19] proposed a BERT-based approach for single-relation question answering (SR-QA), which consists of two models, entity linking, and relation detection. Zhu [20] designed a comprehensive search space for BERT-based relation classification models and employ neural architecture search method to automatically discover the design choices. However, in different situations, the best-performance model is also different. For example, ELECTRA achieves better performance in some tasks in GLUE [21], ALBERT requires less training cost, and RoFormer is more effective in Chinese NLP tasks. As a result, it would be satisfactory if a proposed method could be a model-agnostic solution which is not relied on a specific KBQA model or a specific dataset. In this paper, our proposed model works well as a plug-in approach to different KBQA models to improve their results on different datasets.

For the SR task under KBQA, in traditional methods, as the performance of general NER models is not satisfactory, researchers usually employ constraint (e.g., relation constraint, similarity constraint, type constraint, etc.), which does not exist in general NER tasks, to achieve a better performance. For example, in the CFO model [1], the subject candidates in a KB are pruned based on the constraint of relation candidates generated by the model. After pretrained models are proposed, as they show satisfactory performance in general NER tasks, which is even better than traditional SR models in KBQA tasks, it is a common way to regard the SR task under KBQA as a general NER task and simply employ a NER model [22–26]. However, we cannot find a NER model which can achieve an accuracy of 100%, so if constraint could also be employed in a SR task, it would probably achieve a better performance. Unfortunately, constraint is not well-compatible with pretrained models and it is difficult to employ constraint in a pretrained NER model directly.

Besides BERT-CRF, which is a common model for both KBQA and general NER tasks, researchers have proposed many models for various NER tasks in recent years. For example, some researchers [27] propose a unified generative framework for various NER subtasks to recognize flat, nested, and discontinuous entities, some researchers [28] focus on utilizing both segment-level information and word-level dependencies in NER tasks, and some researchers [29] employ a maximal clique discovery method in a discontinuous NER task. However, a model which is effective in a general NER task may show worse performance in a subject recognition task because of the difference between them, so it is difficult to employ them directly in subject recognition. In addition, there is not a model which could achieve an accuracy of 100%, so there are always some questions where a general NER model fails.

In practical applications, a NLP model is often supposed to answer noisy and abnormal questions caused by various reasons (e.g., noise in the processes of transmission, transformation, or translation). Sometimes the input to a NLP system is even transformed from a piece of voice, video, or image. If the raw voice, video, or image is available, we could feed them directly into special models such as VL-BERT [30], LXMERT [31], VideoBERT [32], ClipBERT [33], wav2vec [34], or SpeechBERT [35], to avoid errors caused by transformation. In addition, the structure of the original model could also be improved so that such noise could be handled automatically by the model. For example, Yang et al. [36] proposed a robust and structurally aware table-text encoding architecture TableFormer, where tabular structural biases are incorporated completely through learnable attention biases. Su et al. [37] proposed a pretrained Chinese Bert that is robust to various forms of adversarial attacks like word perturbation, synonyms, and typos. Liu et al. [38] proposed a robustly optimized bidirectional machine reading comprehension method by incorporating four improvements. Besides, there are also some researches who focus on finding and eliminating noisy labels in datasets so that the model could be trained without noise. For example, Zhu and Michael [39] showed that for text classification tasks with modern NLP models like BERT, over a variety of noise types, existing noisehandling methods do not always improve its performance. Ye et al. [40] proposed a general framework named label noise-robust dialogue state tracking to train DST models robustly from noisy labels, instead of improving the annotation quality further. Nguyen and Khatwani [41] studied the impact of instance-dependent noise to performance of product title classification by comparing our data denoising algorithm and different noise-resistance training algorithms which were designed to prevent a classifier model from overfitting to noise. However, compared to a RE model [42], a NER model is much more sensitive to noise and an entity with a wrong character would be matched to a wrong subject. As a result, it is difficult to employ these methods directly in subject recognition to answer these QWAS in KBQA.

As the golden subject is even not included in a QWAS, it is impossible for a general NER model to recognize it correctly. To answer these QWAS, it is a feasible strategy to correct possible errors in input by a spelling error correction model [43–45]. Another strategy is feeding the raw data (e.g. voice) to a multimodal model to avoid errors in transformation [35]. Some researchers [46] also study NER under a noisy labeled setting with calibrated confidence estimation. However, it is usually impractical to ensure that there are no errors in all input questions. In addition, even if a question contains no errors, it could still be a QWAS because the "correct" entity in it may be unmatched to all subjects in the KB. As a result, it is necessary to propose an effective model to recognize subjects when general models fail to recognize matched subjects.

3. Approach

3.1. Overview. A KB, such as Freebase [3], contains three components: a set of entities E, a set of relations R, and a set of facts $F = \{ < s, r, o > | s, o \in E, r \in R\} \subseteq E \times R \times E$, where $\langle s, r, o \rangle$ are subject-relation-object tuples. To answer a single-relation question, a best-matched subject s' and a best-matched relation r' would be found by a model so that the predicted object o' could be retrieved from F as the answer. To answer a complex question, several candidates (path, subgraph, SPARQL statement, etc.), which generally consist of a subject and several transition relations and entities, would be generated and scored to find a best-matched one to retrieve the answer.

Since there can be millions of entities and thousands of relations in a KB, it is usually difficult and ineffective for a model to find s' and r' directly. In general, a NER model is usually employed to select several successive tokens in the input question as s'. Then, RE or other models would be employed to generate relation candidates based on s' and find the best-matched one.

However, in the NER model, s' is selected from tokens in the input question instead of E, so it is possible that $s' \notin E$ or $s' = \emptyset$. In this case, no candidates would be generated and no answer would be found. In this paper, MGPM is proposed to answer such questions.

The overall structure of our model is shown in Figure 2. R and E are generated based on all possible relations and subjects in the KB. Then, R is pruned by Pruning Model I to generate R_1, and R_1 is further pruned by Pruning Model II to generate R_2. Then, E is pruned based on the constraint of the relation constraint (domain and tuple) of R_2 to generate S_1. Especially, R_2 is a subset of R_1, so the pruned subject set by Pruning Model II (S_1) is also a subset of the pruned subject set by Pruning Model I. As a result, the pruning process by Pruning Model I is shown as a dashed line and the pruned subject set generated by it is not shown in the figure. S_1 is further pruned by Pruning Model III (based on the string similarity constraint) to generate S_2. S_2 is also further pruned by Pruning Model IV (based on the semantics similarity constraint) to get the set of the best-matched subject $S' = \{s'\}$. Thus, a general KBQA model can be employed to find the answer based on s'. In addition, we can simply exclude some of these submodels by setting corresponding parameters to 0. However, experimental results in Section 4.4 show that the whole model with all 4 submodels achieve the best performance.

Figure 3 shows the pruning process of MGPM with an example question "what kind of release is the best of cinema music?." Subject candidates would be pruned in the following steps:

(i) Sets of relations, facts, and subjects are generated based on the KB.

(ii) Pruning Model I is employed to score each relation domain, and domains with scores below a threshold β (determined in our experiments) will be eliminated (gray background area). In this example, the domain "book" is eliminated.

(iii) Pruning Model II is employed to score each relation tuple, and tuples with scores below a threshold γ (determined in our experiments) will be eliminated (red background area). In this example, the relation tuple "music/album/artist" is eliminated.

(iv) The set of subjects is pruned based on the constraint of the relation constraint in the set of facts. Subjects which cannot satisfy the constraint will be eliminated (green background area).In this example, the subject "the" is eliminated. Especially, subjects which satisfy the constraint of relation tuple can certainly satisfy the constraint of relation domain (the dashed line in the figure).

(v) Pruning Model III is employed to score each subject, and subjects with scores below a threshold α (determined in our experiments) will be eliminated (blue background area). In this example, the subject "beginnings" is eliminated.

(vi) Pruning Model IV is employed to score each subject again, and all subjects except the subject with the top score will be eliminated (yellow background area). In this example, the subject "best" is eliminated.

(vii) Only one subject remains in the set, which will be output as the best-matched subject in the model. In this example, the best-matched subject "the best of cinema music" is output, which is also the golden subject of the input question.

In general, there are two main differences between our method and subject recognition with a general NER model:

(i) Search space: the search space of a general NER model contains successive words in the question, whereas the search space of our method contains subjects in the KB.

(ii) Matching strategy: in a general NER model, the recognized entity is matched to each subject in the KB and the identical one is considered as the matched subject. In our method, the subject with the highest score (calculated by our model) is considered as the matched subject.

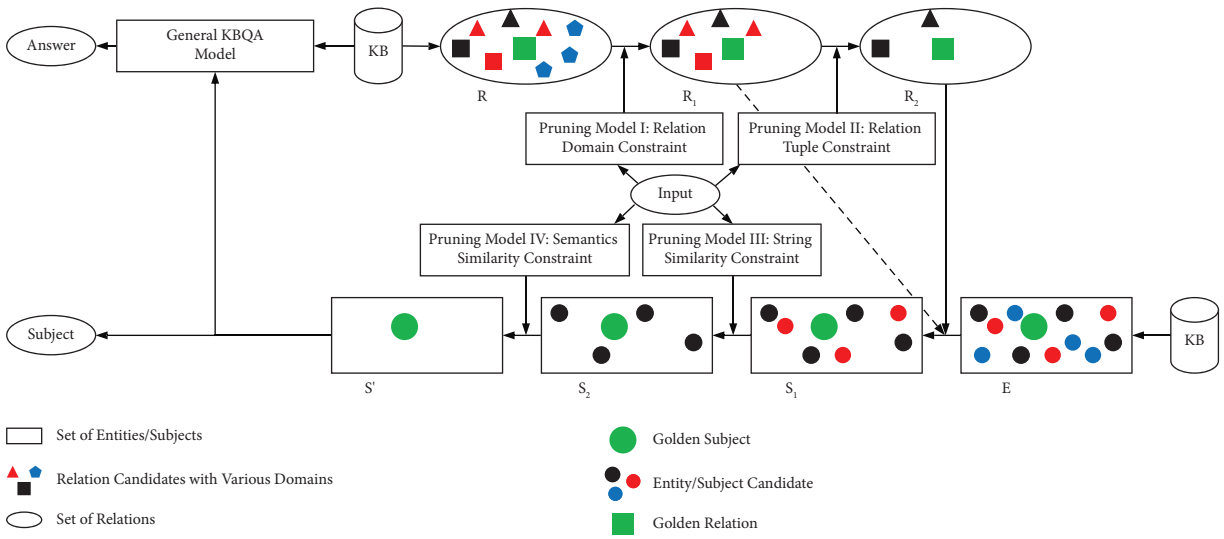

Figure 2: The overall structure of MGPM.

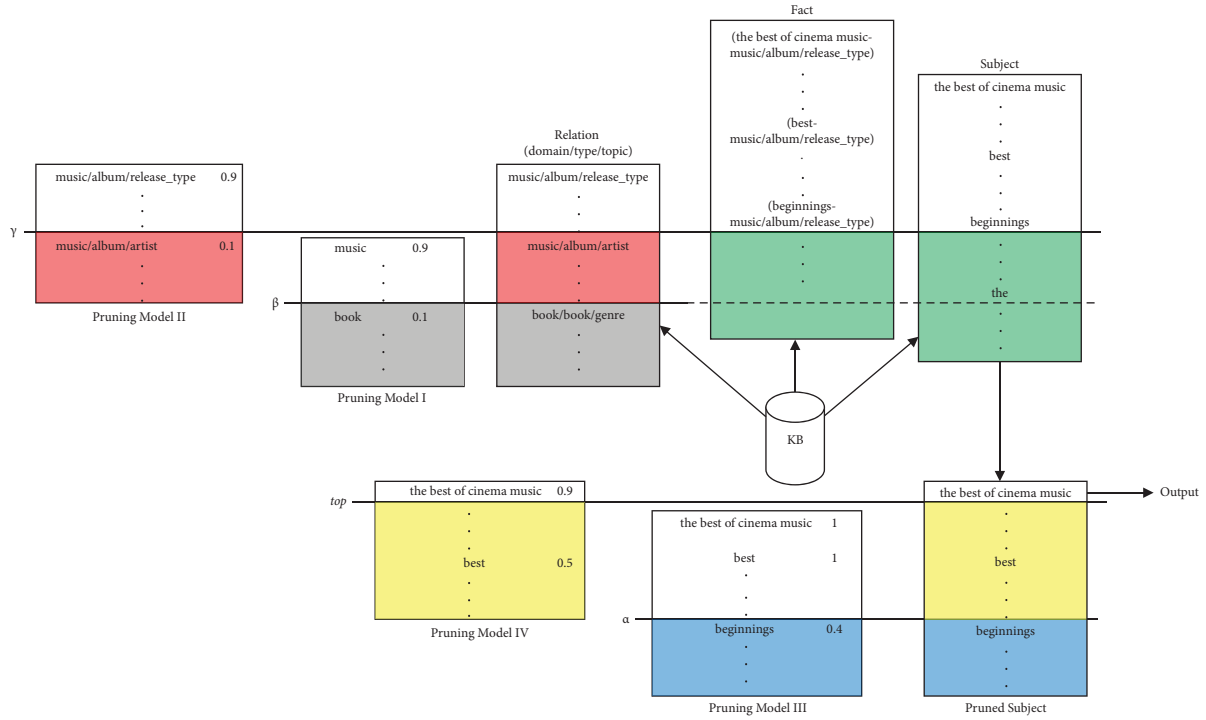

Figure 3: The pruning process of MGPM with an example question.

In addition, the order of these submodels in MGPM has no influence to the final result because s' is the subject with the highest score in Pruning Model IV, which should not be eliminated by any of other submodels. However, different orders lead to different time cost and the order in the figure leads to lowest time cost because of least calculations in total. For example, to the question "what is the name of a location in brasília standard time," if Pruning Model IV is employed first, there are 3972k candidates to be scored by the model. After Pruning Model III, which is not a neural network model and its time cost could be ignored, is employed, 287k candidates would remain. After Pruning Model I, which has 90 domain candidates, is employed, 2.5k candidates would remain then. After Pruning Model II, which has 25 relation candidates matched to the well-matched domain "time," is employed, only 75 candidates would remain at last.

3.2. Pruning Model I: Relation Domain Constraint. As there are millions of entities in the KB, it is quite difficult to find the best-matched subject s' in E directly. If the golden relation r' to a single-relation question (or the golden first-hop

relation to complex question) is known, the set of entities E can be pruned based on the relation constraint and the pruned set $E' = \{s \mid s \in E, < s, r' > \in F\}$ would be generated. Then, it is much easier to find s' in E'.

In addition, in some KBs such as Freebase, relations are organized as hierarchical structure "domain/type/topic." Compared to thousands of relations, there are much less domains and each of them only matches hundreds of relations. Therefore, a set of domain $D = \{d_1, d_2, \ldots, d_k\}$ is generated, and then the best-matched domain and best-matched relation would be found. However, such domains and relations are often not golden ones because of too many candidates and the error transfer. As a result, Pruning Model I is employed and a set of well-matched domains D' is generated to prune R to a set of relations with well-matched domains R_1. Then, Pruning Model II is employed to prune R_1 to a set of well-matched relations R_2. To complex questions, although there could be multiple golden relations, the golden first-hop relation is still better-matched to the question than wrong relations so it is most probably contained in R_2. As a result, this method is also effective to complex questions.

In Pruning Model I, to a question q (Token I), we generate k question-domain pairs (q, d) as Token II. Then, we feed them into a BERT-based classification model and get a prediction set Pre_{d_i}:

$$\mathrm{Pre}_{d_i} = \left\{\left(p_{1,0}, p_{1,1}\right), \ldots, \left(p_{i,0}, p_{i,1}\right), \ldots, \left(p_{k,0}, p_{k,1}\right)\right\}. \quad (1)$$

In the equation, $p_{i,0}$, $(i = 1, 2, \ldots, k)$ is the probability that pair (q, d_i) belongs to Class 0 (unmatched) and $p_{i,1}$ is the probability that this pair belongs to Class 1 (matched). Then, a set of well-matched domains D' is generated as follows:

$$D' = \left\{d_i' \mid d_i' \in D, p_{i,1} \in \mathrm{Pre}_{d_i}, p_{i,1} > \beta\right\}. \quad (2)$$

In the equation, $\beta = 0.5$ is a threshold value to decide whether a domain is well-matched, which is set by our experiments. In the case that $D' = \text{ø}$, we set

$$D' = \left\{d_i' \middle| d_i' \in D, i = \underset{p_{i,1} \in \mathrm{Pre}_{d_i}}{\arg\max}\, p_{i,1}\right\}. \quad (3)$$

In this case, the set of well-matched domains D' only contains a unique best-matched domain.

3.3. Pruning Model II: Relation Tuple Constraint. After D' is generated, R can be pruned to a set of relations with such domains $R_1 = \{r_1, r_2, \ldots, r_n\}$. In Pruning Model II, R_1 will be further pruned. To a question q (Token I), we generate $n$question-relation pairs (q, r) as Token II and get a prediction set Pre_{r_j}:

$$\mathrm{Pre}_{r_j} = \left\{\left(p_{1,0}, p_{1,1}, p_{1,2}\right), \ldots, \left(p_{j,0}, p_{j,1}, p_{j,2}\right), \ldots, \left(p_{n,0}, p_{n,1}, p_{n,2}\right)\right\}. \quad (4)$$

In the equation, $p_{j,0}$, $(j = 1, 2, \ldots, n)$ is the probability that pair (q, r_j) belongs to Class 0 (unmatched), $p_{j,1}$ is the probability that this pair belongs to Class 1 (matched), and $p_{j,2}$ is the probability that this pair belongs to Class 2 (related). A pair belonged to Class 2 means that the relation is unmatched to the question but matched to the subject and experimental results show that it is effective to add Class 2. Experimental results in Section 4.4 show the effectiveness of this strategy.

Then, a set of well-matched relations $R_2 = \{r_1, r_2, \ldots, r_m\}$ $(m \le n)$ is generated:

$$R_2 = \left\{r_j \middle| r_j \in R_1, p_{j,1} \in \mathrm{Pre}_{r_j}, p_{j,1} > \gamma\right\}. \quad (5)$$

In the equation, $\gamma = 0.5$ is a threshold value to decide whether a relation is well-matched, which is set by our experiments. In the case that $R_2 = \text{ø}$, we set

$$R_2 = \left\{r_j \middle| r_j \in R_1, j = \underset{p_{j,1} \in \mathrm{Pre}_{r_j}}{\arg\max}\, p_{j,1}\right\}. \quad (6)$$

In this case, the set of well-matched relations R_2 only contains a unique best-matched relation. Especially, in the situation that $\gamma = 1$, the best-matched relation would be found for each question.

3.4. Pruning Model III: String Similarity Constraint. Based on the relation constraint of R_2, E in the KB is pruned to a set of subject candidates $S_1 = \{s \mid s \in E, < s, r > \in F, r \in R_2\}$. However, it is still difficult and ineffective to recognize the best-matched subject from them. As a result, Pruning Model III is employed to prune S_1 further based on the string similarity.

To SimpleQuestions and WebQuestionsSP datasets, the golden subject is mentioned in a question, so proposition "Entity X is the golden subject to a question" is a necessary but not sufficient condition for proposition "Entity X is identical to some successive words in a question." To a question, there may be multiple entities in the KB which are identical to some successive words in a question, but only one of them is the golden subject. To QWAS, the golden subject is probably string similar to the abnormal subject as the limited impact by errors because it is impractical and meaningless to change most characters in a subject.

Levenshtein algorithm [47] is a common way to calculate the similarity between two strings. In the algorithm, Levenshtein ratio is calculated by the following equation:

$$L(\mathrm{str1}, \mathrm{str2}) = \frac{(\mathrm{sum} - \mathrm{ldist})}{\mathrm{sum}}. \quad (7)$$

In the equation, sum is the total length of the two strings and ldist is the edit distance between them. L is positively related to the similarity between two strings and L for two same strings is 1.

However, L for a subject and a question is nonsensical, because the subject is generally a part of a question. Therefore, the original Levenshtein algorithm is modified:

$$L^{'}(q,s) = \frac{\sum_{t_s \in T_s} \max_{t_q \in T_q} L\left(t_q, t_s\right)}{g}. \tag{8}$$

In the equation, $T_s = \{t_{s,1}, t_{s,2}, \ldots, t_{s,g}\}$ is the set of tokens of a subject (g tokens in total) and $T_q = \{t_{q,1}, \ldots, t_{q,h}\}$ is the set of tokens of a question (h tokens in total). To mitigate interferences of ineffective information, we eliminate tokens with symbols, numbers, and repetitive words. In this way, for each token in a subject s, the token with the maximum similarity (maximum Levenshtein ratio) in question q will be found. Then, the average Levenshtein ratio is calculated to evaluate the similarity between a subject s and the most similar words in a question q. Obviously, L' for a normal question and its golden subject is 1.

Then, a set of similar subjects S_2 is generated:

$$S_2 = \left\{ s \mid s \in S_1, L^{'}(q,s) > \alpha \right\}. \tag{9}$$

In the equation, $\alpha = 0.6$ is a threshold value to decide whether a subject is similar, which is set by our experiments.

3.5. Pruning Model IV: Semantics Similarity Constraint. In previous submodels, a set of similar subjects $S_2 = \{s_1, s_2, \ldots, s_x\}$ is generated. Then, Pruning Model IV, which is also based on a BERT-based classification model, is employed to find out the best-matched subject based on the semantic similarity. In this model, to a question q (Token I), we generate x question-subject pairs as Token II and get a prediction set Pre_{s_u}:

$$\mathrm{Pre}_{s_u} = \left\{ \left(p_{1,0}, p_{1,1}\right), \ldots, \left(p_{u,0}, p_{u,1}\right), \ldots, \left(p_{x,0}, p_{x,1}\right) \right\}. \tag{10}$$

In the equation, $p_{u,0}$, $(u = 1, 2, \ldots, x)$ is the probability that pair (q, s_u) belongs to Class 0 (unmatched) and $p_{u,1}$ is the probability that this pair belongs to Class 1 (matched). Then, we get the best-matched subject (set) by the following equation:

$$S^{'} = \left\{ s_u \,\middle|\, s_u \in S_2, u = \underset{p \in \mathrm{Pre}_{s_u}}{\operatorname{argmax}} p_{u,1} \right\}. \tag{11}$$

Then, the answer could be found by a general KBQA model based on s'.

3.6. Combination of Submodels. In MGPM, the aforementioned submodels are combined as pipeline processes and the core algorithm of the whole model is shown in Algorithm 1.

In our model, question q is first fed to Pruning Model I and D (generated from the KB) is pruned based on the score (calculated by Pruning Model I) of each domain to generate D'. Then, R (generated from the KB) would be pruned to R_1 by selecting all relations which contain a domain in D'. Pruning Model II is then employed to calculate the score of each relation in R_1, and R_1 would be pruned to R_2 based on the score. Based on R_2, E would be pruned to S_1 by selecting all entities which belong to a fact (in F) which contains a relation in R_2. Pruning Model III is then employed to calculate the score of each subject in S_1, and S_1 would be pruned to S_2 based on the score. Finally, Pruning Model IV is employed to calculate the score of each subject in S_2, and the subject with the highest score would be output as the recognized subject in our model.

4. Experiments

4.1. Dataset. The SimpleQuestions dataset [48] is a KBQA dataset of single-relation questions. It provides 108,442 single-relation questions with their answer facts, which are paired with subject-relation-object tuples from Freebase. The dataset is split into a training set, a validation set, and a test set, with 75,910, 10,845, and 21,687 question-fact pairs, respectively. Among all pairs in the test dataset, there are 1,385 QWAS which are unmatched to FB5M (a subset of Freebase) or all n-grams which could be generated from the question [1]. To evaluate the adaptability of our model to normal questions and QWAS, we divide the test set into Dataset I which contains 20,302 pairs of normal questions and Dataset II which contains 1,385 pairs of QWAS.

The WebQuestionsSP dataset [49] is a KBQA dataset of complex questions (also based on Freebase), which contains 3,098 samples in the training set and 1,639 samples in the test set. We also divide the test set into Dataset III which contains 1,233 samples of normal questions and Dataset IV which contains 406 samples of QWAS.

The WebQuestions dataset [50] is a KBQA dataset of the mixture of single-relation and complex questions (also based on Freebase), which contains 3,778 samples in the training set and 2,032 samples in the test set. It is selected to evaluate the performance of our model for mixed types of questions.

4.2. Experiment Setting. Our model is based on the BERT-base model where the number of transformer blocks is 12, the hidden size is 768, and the number of self-attention heads is 12. For each BERT-based classification model in this paper, parameters are trained by an Adam optimizer [51] with a learning rate of $5e-5$, a loss function of sparse categorical crossentropy, an activation function of tanh, and a batch size of 64. In addition, a dropout layer with 0.1 dropout rate and a SoftMax layer with 2 units (3 units in Pruning Model II) are appended to prevent overfitting and output classification results, respectively.

Each of submodels (except Pruning Model III) in our MGPM is trained independently by the training set in SimpleQuestions and the whole MGPM is evaluated by the test set of all datasets. During the training of Pruning Models I and IV, for each question, we generate 1 positive sample (golden domain or subject) with label 1 and at most 5 negative samples (randomly selected from all candidates) with label 0. In Pruning Model II, for each question, we generate 1 sample (golden relation) with label 1, at most 3 samples (randomly selected from relations unmatched to the golden subject) with label 0 and at most 3 samples (randomly

Input: Question q, Entity set E, Relation set R, Domain set D, Fact set F
Output: Recognized subject s

Initialize $D^{'} = R_1 = R_2 = S_1 = S_2 = X = []$
for d **in** D **do**
  Calculate $\text{Score}_I(q, d)$ by Pruning Model I
  if $\text{Score}_I(q, d) > \beta$ **then** $D^{'}$.append(d)
end for
for $r =$ < domain/type/topic > **in** R **do**
  ifdomain **in** $D^{'}$ **then** R_1.append(r)
end for
for r **in** R_1 **do**
  Calculate $\text{Score}_{II}(q, r)$ by Pruning Model II
  if $\text{Score}_{II}(q, r) > \gamma$ **then** R_2.append(r)
end for
for $f =$ < subject/relation/object > **in** F **do**
  if relation**in**R_2 **and** subject**not in** S_1 **then** S_1.append(subject)
end for
for s **in** S_1 **do**
  Calculate $\text{Score}_{III}(q, s)$ by Pruning Model III
  if$\text{Score}_{III}(q, s) > \alpha$ **then** S_2.append(s)
end for
for s **in** S_2 **do**
  Calculate $\text{Score}_{IV}(q, s)$ by Pruning Model IV
  X.append(Score_{IV})
end for
$s = \text{argmax}(X)$
return s

Algorithm 1: The core algorithm of MGPM.

selected from relations matched to the golden subject) with label 2. All models are trained in 3 epochs (approximately 40 minutes per epoch) on a computer with an AMD R9-5950X CPU and a GeForce RTX 3090 GPU.

In general, we choose BERT-CRF (one of the most popular fine-trained models in NLP field), which is achieved by *bert4keras* (https://github.com/bojone/bert4keras), EGP [52] (proposed in 2022, one latest fine-trained model in NER task), and several models without pretrained models (e.g., CFO, BiLSTM-CRF, etc.) as the comparison of our proposed model. For dataset, we choose SimpleQuestions (a widely used dataset for single-relation questions), WebQuestionsSP (a widely used dataset for complex questions), several subsets of them (Datasets I–V), and WebQuestions (dataset of mixed single-relation questions and complex questions).

It seems that there are other countless NER methods and datasets could be the comparison of our model; however, many of them are incompatible with our model, for the following reasons:

(i) As the difference between a NER task and a SR task under KBQA (introduced in our introduction), many NER models would output all possible entities but not the only subject entity to an input question, which is effective for a NER task but inapplicable for a SR task.

(ii) Although there are some NER models (latter than BERT) which outperform BERT in NER tasks and could be also employed in a SR task, e.g., ELECTRA, T5, etc., they fail to outperform BERT in our experiments (not shown in our paper). Even EGP could not outperform BERT on all datasets. In fact, BERT-CRF has still been the default model for the SR task under KBQA in many researches up to now, and it is significant for our model to outperform BERT, EGP, and many other models in our experiments.

(iii) SimpleQuestions, WebQuestionsSP, and WebQuestions are widely used open-domain KBQA datasets, which have been the evaluator of a general KBQA model up to now. The comparison of different models on these datasets has been a common way to evaluate different KBQA models. Just like many researches of general KBQA, we also select these datasets to evaluate our model (in fact, even the latest research [53] also chose SimpleQuestions as an evaluator).

(iv) In addition, catastrophic forgetting [54] is a feature of neural network models, especially pretrained models. As a result, the best versatility and the best performance are usually incompatible in many tasks. For example, the latest research [53] proposed a model with a high versatility of multilingual QA tasks, whereas it fails to outperform even a traditional model [55] if we only focus on the performance on the SimpleQuestions dataset. As a result, we only select models which focus on the best

performance on some specific type datasets as the comparison of our model and experimental results show that our model achieves a better performance, which shows the effect of model. In addition, our model also shows the versatility in the complex KBQA task (WebQuestionsSP dataset) where compared models are inapplicable, which further shows the effect of our model.

4.3. Experiment Results. Accuracy, recall, precision, and F1 score are all optional indicators in deep learning. In the KBQA task, as the search space for each input is usually different and some other reasons, accuracy is usually selected as the indicator in many researches. In this paper, we also select accuracy as the indicator to evaluate our model. Experimental results for subject recognition are shown in Table 1. In Table 1, MGPM means our model is employed as a standalone model, and it achieves the accuracy of 89.5% on SimpleQuestions (SQ), 52.8% on WebQuestionsSP (WSP), and 52.1% on WebQuestions (WQ), which shows that the SR task for complex questions is more difficult than that for single-relation questions. For normal questions, MGPM achieves the accuracy of 92.4% on Dataset I and 61.6% on Dataset III, which both outperform the accuracy on original datasets. It shows that if we could ensure that all input questions are normal questions in practical, a KBQA model would achieve a better performance. For QWAS, MGPM achieves the accuracy of 46.0% on Dataset II and 26.1% on Dataset IV, which are both much lower than the accuracy on normal questions. It shows that the SR task for normal questions is much easier than that for QWAS.

BERT-CRF is one of the most popular models in NER tasks, which also could be employed in SR tasks after fine-tuning. It achieves the accuracy of 90.8% on SimpleQuestions, 60.0% on WebQuestionsSP, and 58.4% on WebQuestions, which outperform MGPM (by the margin of 1.3%, 7.2%, and 6.3%). For normal questions, it achieves the accuracy of 97.0% on Dataset I and 79.8% on Dataset III, which further outperform MGPM (by the margin of 4.6% and 18.2%). However, it is inapplicable for QWAS and cannot answer any questions on Datasets II and IV.

Fortunately, MGPM not only could work as a standalone model but also could work as a plug-in approach to another KBQA model. "+ MGPM" in Table 1 means that a question is answered by a traditional KBQA model (e.g., BERT-CRF) first, and in the case that no answers could be found (no-matched or no subjects are recognized), MGPM will be employed to answer the question as the alternative model. As an integrated model, BERT-CRF + MGPM achieves the accuracy of 94.4% on SimpleQuestions, 68.6% on WebQuestionsSP, and 63.7% on WebQuestions, which both outperform BERT-CRF (by the margin of 3.6%, 8.6%, and 5.3%). For normal questions, it achieves the accuracy of 98.4% on Dataset I and 87.3% on Dataset III, which also outperform BERT-CRF (by the margin of 1.4% and 7.5%). For QWAS, it achieves the accuracy of 36.0% on Dataset II and 11.8% on Dataset IV, which both outperform BERT-CRF but fail to outperform standalone MGPM, because BERT-CRF outputs wrong answers to some QWAS and MGPM is not employed to these questions.

EGP is another NER model which is proposed recently and could also be employed in SR tasks. Experimental results show that it achieves the better performance to complex questions than BERT-CRF but shows worse performance to single-relation questions. The integrated model EGP + MGPM outperforms EGP on all datasets and also outperforms BERT-CRF on WebQuestionsSP and WebQuestions.

In general, as a standalone model, MGPM achieves highest accuracies of 46.0% and 26.1% on datasets of QWAS (II and IV). However, on datasets of normal questions (I and III), MGPM fails to outperform baseline models. As a result, to original datasets, it is a better strategy to integrate MGPM and a baseline model. In this strategy, a question is answered by a baseline model and in the case that no answers could be found (no-matched or no subjects are recognized), MGPM will be employed to answer the question as the alternative model. Experimental results show that BERT-CRF + MGPM outperforms the baseline BERT-CRF by margins of 3.6%, 8.6%, and 5.3% on the whole SimpleQuestions, WebQuestionsSP, and WebQuestions, and EGP + MGPM outperforms the baseline EGP by margins of 4.7%, 8.1%, and 4.8% on these datasets.

In practice, it is usually unknown whether the subject in an input question is abnormal and sometimes a wrong-matched subject would be found by the general model. As a result, the performance of the integrated MGPM model is worse than standalone MGPM on Datasets II and IV. If the quality of input questions can be evaluated, it is better to employ the standalone MGPM for questions in poor quality (e.g., translated from other languages).

In addition, Table 1 also shows that different models achieve different performance on different datasets: Efficient GlobalPointer outperforms BERT-CRF on SimpleQuestions while BERT-CRF outperforms Efficient GlobalPointer on WebQuestionsSP. However, no matter which baseline model is chosen, the integrated MGPM model would always outperform the corresponding baseline model on a whole dataset. In other words, our MGPM is compatible with various KBQA models.

Then, a general BERT-based RE model could be employed for relation prediction to single-relation questions in a KBQA task. Experimental results for overall accuracies (%) of our models and traditional models on SimpleQuestions are shown in Table 2. Among these methods, KEQA [56] and M3M [53] are recent methods which show satisfactory performance in multilingual KBQA or other KBs, whereas they fail to outperform the traditional method BiLSTM-CRF + BiLSTM [55] on the SimpleQuestions dataset. We choose BERT-CRF by *bert4keras* as the baseline model, which is widely employed in various KBQA tasks and achieves good performances, and the integrated MGPM model further outperforms it by a margin of 3.4%, showing the effectiveness of our models.

For complex questions, relation prediction and subject recognition are usually considered as two individual tasks, and rather than the overall accuracy, the accuracy for

Table 1: Experimental results for subject recognition.

Method	SQ	Dataset I	Dataset II	WSP	Dataset III	Dataset IV	WQ
MGPM	89.5	92.4	**46.0**	52.8	61.6	**26.1**	52.1
BERT-CRF by *bert4keras*	90.8	97.0	0	60.0	79.8	0	58.4
+MGPM	**94.4** (3.6↑)	**98.4** (1.4↑)	36.0	68.6 (8.6↑)	87.3 (7.5↑)	11.8	63.7 (5.3↑)
EGP	89.4	95.4	0	62.2	82.7	0	60.4
+MGPM	94.1 (4.7↑)	98.0 (2.6↑)	37.0	**70.3** (8.1↑)	**89.5** (6.8↑)	11.8	**65.2** (4.8↑)

Bold values represent the best-performance.

Table 2: Experimental results for overall accuracies (%).

Method	Accuracy (%)
MemNN-ensemble [48]	63.9
CFO [1]	75.7
BiLSTM-CRF + BiLSTM [55]	78.1
Structure attention + MLTA [5]	82.3
KEQA [56]	75.4
M3M [53]	76.9
BERT-CRF (baseline)	82.6
EGP [52]	81.4
SSMFRP [57]	83.0
BERT-CRF + MGPM	**86.0**

Bold values represent the best-performance.

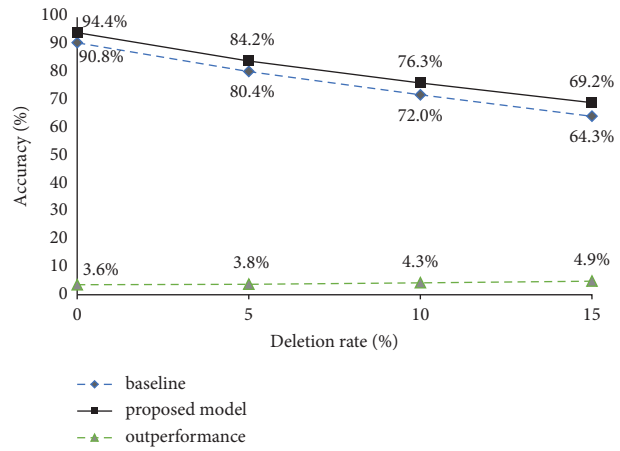

Figure 4: Experimental results for accuracies (%) of subject recognition on noisy datasets.

relation prediction is more often chosen to evaluate a model to answer complex questions. As a result, it is unnecessary to conduct additional experiments to evaluate the overall accuracy on WebQuestionsSP.

To further evaluate the robustness of our model, we delete 5%, 10%, and 15% words, respectively, in questions of the SimpleQuestions dataset and evaluate our proposed model (BERT-CRF + MGPM) and the baseline model (BERT-CRF) on these noisy datasets. Experimental results for accuracies of subject recognition are shown in Figure 4. The baseline model achieves accuracies of 90.8%, 80.4%, 72.0%, and 64.3% respectively, on datasets of deletion rate of 0% (original dataset), 5%, 10%, and 15%, respectively. Our proposed model achieves accuracies of 94.4%, 84.2%, 76.3%, and 69.2%, respectively, on these datasets, which outperform the baseline model by a margin of 3.6%, 3.8%, 4.3%, and 4.9%, respectively. In general, our proposed model outperforms the baseline model on all noisy datasets, and the outperformance shows a positive correlation with the deletion rate, which shows the robustness of our proposed model.

4.4. Parameter Determination and Ablation Experiments. In our models, $\alpha, \beta, \gamma \in [0, 1]$ are hyperparameters which should be optimized in the experiment. A higher parameter value means a stronger pruning, less candidates in Pruning Model IV, and less prediction time, so the values should be as high as possible, and in the case that several values lead to a similar accuracy, the highest value will be selected. As the output of Pruning Model III influences the direct input to Pruning Model IV, we first set $\beta = \gamma = 1$ and gradually decreasing α. As it would take too much time to evaluate all combinations on the whole dataset, we only evaluate them on a dataset of the most significant questions. This dataset (Dataset V) contains all 1,334 questions where the general model find no answers (no-matched or no subjects are recognized) on SimpleQuestions. In the integrated MGPM model, MGPM is employed to answer these questions so a well-performed model on these questions leads to a well-performed integrated MGPM model on the whole dataset. We evaluate our model with these hyperparameter values on Dataset V, and experimental results in Table 3 show that $\alpha = 0.6$ and $\alpha = 0.5$ lead to the similar accuracy, so the optimized value of α is 0.6. Compared with α, $\beta$and γ show less influence to the result, and we evaluate several value combination and results show that $\beta = \gamma = 0.5$ shows the best accuracy (lower values lead to the similar accuracy and are not shown in the table.) As a result, we set $\alpha = 0.6, \beta = \gamma = 0.5$ as the optimized values of the hyperparameters in our model.

In Table 3, "Candidate" means the average number of candidates to a question in Pruning Model IV, which has a positive correlation to the prediction time. Among the 4 submodels in our model, Pruning Model IV is indispensable because it directly outputs the recognized subject. However, if we only employ Pruning Model IV ($\alpha = \beta = \gamma = 0$), there would be a huge number of candidates (3,972k), which leads to unacceptable time and space cost. After Pruning Model III is included ($\alpha = 0.6, \beta = \gamma = 0$), the number of candidates is much smaller (109k), but still leads to unacceptable time and space cost. In these cases, our computer cannot calculate the accuracy. After Pruning Model I is included ($\alpha = 0.6, \beta = 0.5, \gamma = 0$), the number of candidates is further smaller (7,9k), and the accuracy is able to be calculated (55.6%), which is lower than the accuracy of the whole

Table 3: Experimental results of MGPM with different parameter values.

α	β	γ	Accuracy (%)	Candidate
0.8	1	1	39.2	<0.1k
0.7	1	1	46.6	0.5k
0.6	1	1	49.5	2.5k
0.5	1	1	49.5	8.6k
0.6	1	0.8	55.9	2.5k
0.6	1	0.5	56.8	2.5k
0.6	0.8	0.8	57.4	2.8k
0.6	0.8	0.5	58.3	2.9k
0.6	0.5	0.8	58.3	3.2k
0.6	0.5	0.5	**59.4**	3.4k
0.6	0.5	0	55.6	7.9k
0.6	0	0	—	109k
0	0	0	—	3,972k

Bold values represent the best-performance.

model ($\alpha = 0.6, \beta = \gamma = 0.5$). As a result, the combination of all these 4 submodels has the best performance, which is the whole MGPM.

In Pruning Model II, question-relation pairs are classified into three categories (Strategy I) instead of two categories (Strategy II). To single-relation questions, after the subject is found by MGPM, we prefer to find the relation by a general RE model (Strategy III) rather than Pruning Model II in MGPM (Strategy IV). As shown in Table 4, experimental results show that Strategy II outperforms Strategy I by a margin of 1.4% and Strategy III outperforms Strategy IV by a margin of 10.9%. As a result, we choose Strategy II and Strategy III in our MGPM.

5. Discussion

In our experiments, we choose KBQA models based on BERT-base as baseline models and experimental results show the effectiveness of our MGPM and integrated models. However, our model is not confined to BERT-base and it could also be integrated with other pretrained models (e.g., BERT-Large, ELECTRA) or other methods without pretrained models. As long as there are some questions where a KBQA model fails to find answers, our model could be employed to answer them efficiently. As a result, our model has strong adaptability to the integration with various KBQA models and integrated MGPM models would probably outperform original models.

In practice, an input question to a KBQA system could contain abnormal expressions from various users. Besides, it could be fed to the system after multiple processes of transmission, transformation, or translation. As a result, it could be common for a practical KBQA system to answer QWAS. Our experiments have shown that MGPM achieves a SR accuracy of 89.5% while the baseline BERT-CRF achieves a SR accuracy of 90.8% on the SimpleQuestions dataset which contains 6.1% QWAS. Therefore, we could infer that MGPM would outperform the baseline model on the dataset (if similar to SimpleQuestions) which contains more than 9.1% QWAS. In these cases, our MGPM has great practical value.

Table 4: Experimental results of different strategies in submodels.

Strategy	RE accuracy (%)
Strategy I	86.6
Strategy II	**88.0**
Strategy III	**90.7**
Strategy IV	79.8

Bold values represent the best-performance.

However, in some particular cases such as medical QA, it is risky to output a uncertain answer. Users even prefer no answers than an unreliable answer. In these cases, "No Answer" is the safe and acceptable output to a QWAS or an imprecise question, so MGPM is inapplicable. Besides, in the case that the time and space cost is strictly restricted, the integrated MGPM (standalone MGPM, general KBQA models, or even deep learning models) is also not appropriate for deep learning itself requires higher time and space cost than traditional methods. Instead, traditional methods such as semantic parsing-based models or query models would be employed.

In fact, integrated MGPM, standalone MGPM, and traditional KBQA models have their own advantages and disadvantages, respectively: integrated MGPM is model-agnostic and compatible with various KBQA models; standalone MGPM shows better performance in answering QWAS; sometimes we could also find out a traditional KBQA model which meets all requirements of a specific task. In summary, there are some strategies about which models to choose in different situations:

(i) In most situations, especially in situations where a KBQA model (known or unknown) has been employed, integrated MGPM is a better choice, as it is compatible with various KBQA models.

(ii) In the situations where QWAS are frequently inputted or the quality of the input is difficult to guarantee, standalone MGPM should be chosen, as it shows better performance in answering QWAS.

(iii) In some situations where no answers are more acceptable than an unreliable answer, or the time and space cost is strictly restricted, MGPM is not so appropriate. Instead, traditional KBQA models should be chosen.

6. Conclusion

Among all questions in the original dataset of SimpleQuestions and WebQuestionsSP, there are mainly normal questions (Datasets I and III) and some QWAS (Datasets II and IV). A traditional KBQA model (e.g., BERT-CRF) is effective to most normal questions (an accuracy of 97.0% for Dataset I and 79.8% for Dataset III), but it is inapplicable to QWAS (an accuracy of 0% for Datasets II and IV). In most cases, it is difficult to recognize QWAS among all input questions, and even if QWAS could be recognized, a traditional KBQA model still cannot answer it. As a result, in practical applications, a KBQA model is simply employed and it can find answers (right or wrong) to some of the input

questions and fails to answer the others of them (Dataset V in our experiment).

To improve the performance of a KBQA model, in this paper, we propose a method for the SR task under KBQA when general models fail. In our model, relations in the KB are pruned to a set by two pruning submodels (I and II). Then, the set of all subjects in the KB is pruned by the constraint of these well-matched relations and two other pruning submodels (III and IV). After this multigranularity pruning process, best-matched subject could be recognized. Then, the question could be answered by a general KBQA model based on the recognized subject. In general, there are mainly the following advantages of our model:

(i) In the case of normal questions and questions which could be answered by traditional models, our model is also effective and even outperforms traditional models.

(ii) In the case of QWAS and questions where traditional models fail, our model could still answer some of these questions correctly, whereas traditional models achieve an accuracy of 0%.

(iii) After finishing the process of training, our model is effective to different types of questions (single-relation questions and complex questions) and different datasets (SimpleQuestions, WebQuestionsSP, WebQuestions, and Dataset I–V) without more training, which shows the versatility of our model.

Inevitably, there are also some weaknesses of our model:

(i) In some particular cases where it is risky to output an uncertain answer (e.g., Medical QA), our model is inapplicable because it always tries to give answer to all questions.

(ii) In the case that the time and space cost is strictly restricted (e.g., industrial systems), our model is also inappropriate because deep learning requires higher time and space cost than traditional methods. Instead, traditional methods such as semantic parsing-based models or query models would be employed.

As future work, studies will be conducted to reduce the influence of aforementioned weaknesses of our model and extend our model to multilingual KBQA tasks.

Data Availability

Data used in this study are available on request from the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Natural Science Foundation of China under Grant Nos. U21A20491, U1936109, and U1908214.

References

[1] Z. Dai, L. Li, and W. Xu, "Cfo: conditional focused neural question answering with large-scale knowledge bases," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 800–810, Berlin, Germany, August 2016.

[2] P. Liang, M. I. Jordan, and D. Klein, "Learning dependency-based compositional semantics," *Computational Linguistics*, vol. 39, no. 2, pp. 389–446, 2013.

[3] K. Bollacker, C. Evans, P. Paritosh, Tim Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 1247–1250, Vancouver, Canada, June 2008.

[4] X. Yao and B. Van Durme, "Information extraction over structured data: question answering with freebase," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 956–966, Baltimore, Maryland, June 2014.

[5] R. Wang, Z. Ling, and Y. Hu, "Knowledge base question answering with attentive pooling for question representation," *IEEE Access*, vol. 7, pp. 46773–46784, 2019.

[6] Y. Qu, J. Liu, L. Kang, Q. Shi, and D. Ye, "Question answering over freebase via attentive rnn with similarity matrix based cnn," 2018, https://arxiv.org/abs/1804.03317.

[7] W. Zhao, T. Chung, A. Goyal, and A. Metallinou, "Simple question answering with subgraph ranking and joint-scoring," 2019, https://arxiv.org/abs/1904.04049.

[8] H. Jin, Y. Luo, C. Gao, X. Tang, and P. Yuan, "Comqa: question answering over knowledge base via semantic matching," *IEEE Access*, vol. 7, pp. 75235–75246, 2019.

[9] M. Wei and Y. Zhang, "Natural answer generation with attention over instances," *IEEE Access*, vol. 7, pp. 61008–61017, 2019.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, Minneapolis, Minnesota, June 2019.

[11] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and S. Radu, "Albert: a lite bert for self-supervised learning of language representations," 2019, https://arxiv.org/abs/1909.11942.

[12] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and V. Quoc, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Proceedings of the 2019 Annual Conference on Neural Information Processing Systems*, pp. 5754–5764, ACM, Red Hook, NY, USA, 2019.

[13] K. Clark, M.-T. Luong, V. L. Quoc, and C. D. Manning, "Electra: pre-training text encoders as discriminators rather than generators," in *Proceedings of the International*

Conference on Learning Representations, Addis Ababa, Ethiopia, June 2020.

[14] M. Martinc, B. Škrlj, and S. Pollak, "Tnt-kid: transformer-based neural tagger for keyword identification," *Natural Language Engineering*, vol. 28, no. 4, pp. 409–448, 2022.

[15] M. Blšták and V. Rozinajová, "Automatic question generation based on sentence structure analysis using machine learning approach," *Natural Language Engineering*, vol. 28, no. 4, pp. 487–517, 2022.

[16] O. Wysocki, Z. Zhou, P. O'Regan et al., "Transformers and the representation of biomedical background knowledge," *Computational Linguistics*, vol. 49, no. 1, pp. 73–115, 2023.

[17] M. T. R. Laskar, E. Hoque, and J. X. Huang, "Domain adaptation with pre-trained transformers for query-focused abstractive text summarization," *Computational Linguistics*, vol. 48, no. 2, pp. 279–320, 2022.

[18] A. Gangwar, S. Jain, S. Sourav, and A. Modi, "Counts@iitk at semeval-2021 task 8: scibert based entity and semantic relation extraction for scientific data," in *Proceedings of the 15th International Workshop on Semantic Evaluation*, pp. 1232–1238, ACL, Toronto, Canada, 2021.

[19] D. Luo, J. Su, and S. Yu, "A bert-based approach with relation-aware attention for knowledge base question answering," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–8, Glasgow, UK, July 2020.

[20] W. Zhu, "Autorc: improving bert based relation classification models via architecture search," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pp. 33–43, Toronto, Canada, June 2021.

[21] A. Wang, A. Singh, M. Julian et al., "A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 7th International Conference on Learning Representations*, Brussels, Belgium, November 2019.

[22] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016.

[23] T. Shibuya and E. Hovy, "Nested named entity recognition via second-best sequence learning and decoding," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 605–620, 2020.

[24] A. Ghaddar, P. Langlais, A. Rashid, and M. Rezagholizadeh, "Context-aware adversarial training for name regularity bias in named entity recognition," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 586–604, 2021.

[25] T. Effland and M. Collins, "Partially supervised named entity recognition via the expected entity ratio loss," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1320–1335, 2021.

[26] Z. Wang, X. Xu, X. Li, H. Li, X. Wei, and D. Huang, "An improved nested named-entity recognition model for subject recognition task under knowledge base question answering," *Applied Sciences*, vol. 13, no. 20, Article ID 11249, 2023.

[27] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang, and X. Qiu, "A unified generative framework for various ner subtasks," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 5808–5822, Bangkok, Thailand, August 2021.

[28] F. Li, Z. Wang, S. C. Hui et al., "Modularized interaction network for named entity recognition," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 200–209, Toronto, Canada, August 2021.

[29] Y. Wang, B. Yu, H. Zhu, T. Liu, Y. Nan, and L. Sun, "Discontinuous named entity recognition as maximal clique discovery," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 764–774, Toronto, Canada, August 2021.

[30] W. Su, X. Zhu, Y. Cao et al., "Vl-bert: pre-training of generic visual-linguistic representations," in *Proceedings of the International Conference on Learning Representations*, Addis Ababa, Ethiopia, January 2020.

[31] H. Tan and M. Bansal, "Lxmert: learning cross-modality encoder representations from transformers," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 5100–5111, Hong Kong, China, November 2019.

[32] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: a joint model for video and language representation learning," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, pp. 7463–7472, Seoul, Korea (South), October 2019.

[33] J. Lei, L. Li, L. Zhou, Z. Gan, and L. Tamara, "Less is more: clipbert for video-and-language learning via sparse sampling," in *Proceedings of the 2021 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7331–7341, Nashville, TN, USA, 2021.

[34] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: unsupervised pre-training for speech recognition," in *Proceedings of the 20th Annual Conference of the International Speech Communication Association*, pp. 3465–3469, Hong Kong, China, September 2019.

[35] Y.-S. Chuang, C.-L. Liu, and H. Y. Lee, "Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering," in *Proceedings of the 21st Annual Conference of the International Speech Communication Association*, pp. 4168–4172, Shanghai, China, October 2020.

[36] J. Yang, A. Gupta, S. Upadhyay, L. He, R. Goel, and S. Paul, "Tableformer: robust transformer modeling for table-text encoding," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 528–537, Dublin, Ireland, May 2022.

[37] H. Su, W. Shi, X. Shen et al., "Rocbert: robust Chinese bert with multimodal contrastive pretraining," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 921–931, Dublin, Ireland, May 2022.

[38] S. Liu, K. Li, and Z. Li, "A robustly optimized bmrc for aspect sentiment triplet extraction," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 272–278, Seattle, WA, USA, July 2022.

[39] D. Zhu and A. Michael, "Is bert robust to label noise? a study on learning with noisy labels in text classification," in *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pp. 62–67, Dublin, Ireland, May 2022.

[40] F. Ye, F. Yue, and E. Yilmaz, "Assist: towards label noise-robust dialogue state tracking," in *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2719–2731, London, UK, May 2022.

[41] H. Nguyen and D. Khatwani, "Robust product classification with instance-dependent noise," in *Proceedings of the Fifth Workshop on E-Commerce and NLP*, pp. 171–180, London, UK, May 2022.

[42] Z. Wang, X. Xu, X. Li, H. Li, L. Zhu, and X. Wei, "A more robust model to answer noisy questions in kbqa," *IEEE Access*, vol. 11, pp. 22756–22766, 2023.

[43] J. Náplava, M. Straka, J. Straková, and A. Rosen, "Czech grammar error correction with a large and diverse corpus," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 452–467, 2022.

[44] J. Lichtarge, C. Alberti, and S. Kumar, "Data weighted training strategies for grammatical error correction," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 634–646, 2020.

[45] C. Napoles, M. Nădejde, and J. Tetreault, "Enabling robust grammatical error correction in new domains: data sets, metrics, and analyses," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 551–566, 2019.

[46] K. Liu, Y. Fu, C. Tan et al., "Noisy-labeled ner with confidence estimation," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3437–3445, Hong Kong, China, May 2021.

[47] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Doklady Akademii Nauk SSSR*, vol. 163, no. 4, pp. 845–848, 1966.

[48] B. Antoine, N. Usunier, S. Chopra, and J. Weston, "Large-scale simple question answering with memory networks," 2015, https://arxiv.org/abs/1506.02075.

[49] Y. Wen, M. Richardson, C. Meek, and M. W. Chang, "The value of semantic parse labeling for knowledge base question answering," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 201–206, Berlin, Germany, August 2016.

[50] J. Berant, A. Chou, F. Roy, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, Seattle, WA, USA, October 2013.

[51] D. Kingma and B. Jimmy, "Adam: a method for stochastic optimization," 2015, https://arxiv.org/abs/1412.6980.

[52] J. Su, M. Ahmed, S. Pan et al., "Global pointer: novel efficient span-based approach for named entity recognition," 2022, https://arxiv.org/abs/2208.03054.

[53] A. Razzhigaev, M. Salnikov, V. Malykh, P. Braslavski, and A. Panchenko, "A system for answering simple questions in multiple languages," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 524–537, Toronto, Canada, July 2023.

[54] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2018.

[55] M. Petrochuk and L. Zettlemoyer, "Simplequestions nearly solved: a new upperbound and baseline approach," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 554–558, Brussels, Belgium, October 2018.

[56] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based questiosn answering," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 105–113, Melbourne VIC, Australia, October 2019.

[57] Z. Wang, X. Xu, X. Li, X. Song, X. Wei, and D. Huang, "Ssmfrp: semantic similarity model for relation prediction in kbqa based on pre-trained models," in *Proceedings of the 2022 International Conference on Artificial Neural Networks*, pp. 294–306, Bristol, UK, September 2022.