big data and cognitive computing

MDPI

Article

Candidate Set Expansion for Entity and Relation Linking Based on Mutual Entity–Relation Interaction

Botao Zhang [1,2,3,†], Yong Feng [1,2,3,†], Lin Fu [1,2,3], Jinguang Gu [1,2,3] and Fangfang Xu[1,2,3,*]

[1] College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China
[2] Institute of Big Data Science and Engineering, Wuhan University of Science and Technology, Wuhan 430065, China
[3] Key Laboratory of Rich-Media Knowledge Organization and Service of Digital Publishing Content, National Press and Publication Administration, Beijing 100038, China
* Correspondence: xuff@wust.edu.cn; Tel.: +86-13995541659
† These authors contributed equally to this work.

Abstract: Entity and relation linking are the core tasks in knowledge base question answering (KBQA). They connect natural language questions with triples in the knowledge base. In most studies, researchers perform these two tasks independently, which ignores the interplay between the entity and relation linking. To address the above problems, some researchers have proposed a framework for joint entity and relation linking based on feature joint and multi-attention. In this paper, based on their method, we offer a candidate set generation expansion model to improve the coverage of correct candidate words and to ensure that the correct disambiguation objects exist in the candidate list as much as possible. Our framework first uses the initial relation candidate set to obtain the entity nodes in the knowledge graph related to this relation. Second, the filtering rule filters out the less-relevant entity candidates to obtain the expanded entity candidate set. Third, the relation nodes directly connected to the nodes in the expanded entity candidate set are added to the initial relation candidate set. Finally, a ranking algorithm filters out the less-relevant relation candidates to obtain the expanded relation candidate set. An empirical study shows that this model improves the recall and correctness of the entity and relation linking for KBQA. The candidate set expansion method based on entity–relation interaction proposed in this paper is highly portable and scalable. The method in this paper considers the connections between question subgraphs in knowledge graphs and provides new ideas for the candidate set expansion.

Keywords: joint entity and relation linking; relation linking; KBQA; candidate set expansion

Citation: Zhang, B.; Feng, Y.; Fu, L.; Gu, J.; Xu, F. Candidate Set Expansion for Entity and Relation Linking Based on Mutual Entity–Relation Interaction. *Big Data Cogn. Comput.* **2023**, *7*, 56. https://doi.org/10.3390/bdcc7010056

Academic Editor: Min Chen

Received: 1 March 2023
Revised: 17 March 2023
Accepted: 20 March 2023
Published: 22 March 2023

1. Introduction

Knowledge base question answering (KBQA) [1] has recently been widely used in various fields, such as recommendation systems [2–4] and chat bots [5,6]. KBQA enables users to obtain direct answers to questions from a structured knowledge base. Typically, this is accomplished by semantic-parsing-based methods (SP-based methods) [7]. The process can be summarized as three steps [8]: entity linking, relation linking, and query generation. The accuracy of the final query results is obviously directly impacted by the accuracy of the entity and relation linking. Hence, it is crucial to optimize the performance of the entity and relation linking tasks [9–11].

At the moment, while the KBQA system based on semantic parsing can directly return answers to users, its accuracy is low, which has a lot to do with the accuracy of the results of the entity and relation linking. Therefore, to improve the accuracy of the entity and relation linking, researchers need to continuously explore and experiment with the generation, representation, and ranking of candidate sets in the linking process. The entity/relation

candidate set is obtained by performing entity/relation prediction on entity/relation keywords. The correct entity/relation candidate set improves the entity/relation link accuracy. This is because the correct links are already included in the candidate set. Most of the current research has focused on improving and refining the entity or relation disambiguation algorithms, and more attention needs to be paid to acquiring candidate sets. Existing keyword tools are a popular way to obtain keywords. We can obtain the final link results following the prediction and ranking of the accepted keywords. When the number of words mentioned by entities in a natural language problem is too large or the related information in a sentence is unclear, it is difficult for the keyword-extraction tool to extract the correct information. Thus, we need to obtain the correct entity and relation candidate sets. Therefore, this paper's idea is to expand the keywords after obtaining the keywords through the tool and then continue to predict and sort.

This paper is a candidate set expansion study based on the existing relation linking framework, intending to improve the coverage of entity and relation candidate sets. Currently, there are three types of problems with candidate set construction, as shown in Figure 1:

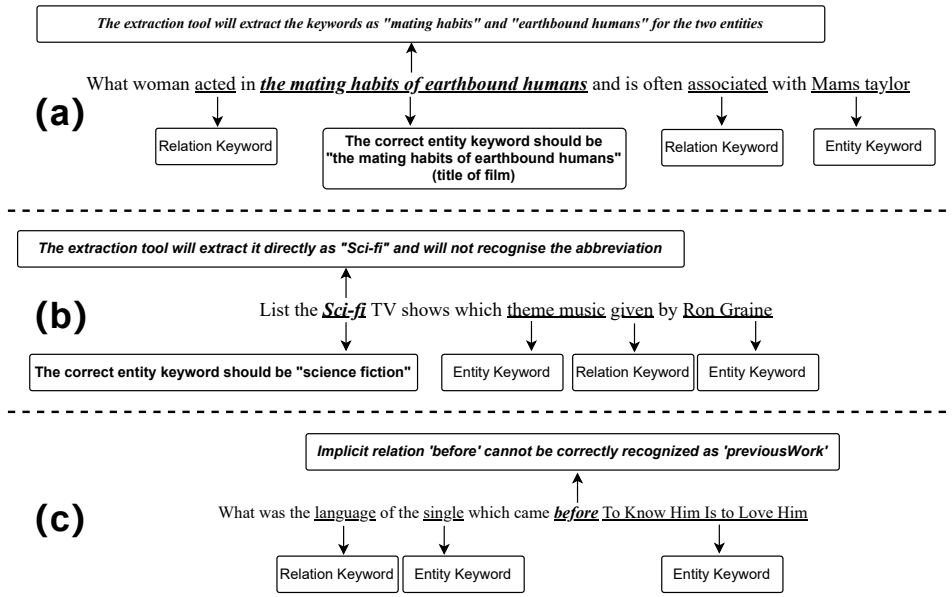

Figure 1. The issues addressed in this paper. (**a**) The number of words in the question for the entity keyword is high and existing keyword-extraction tools may split the complete entity keyword into multiple; (**b**) Existing keyword-extraction tools cannot accurately hit the correct entity keyword when it appears in the question in its abbreviated form; (**c**) When the issue relation is complex, the keyword-extraction tool may not be able to hit the correct relation candidates accurately. These three types of problems result in the inability to select the correct entity/relation candidate, thus affecting the correctness of the entity/relation linking.

In order to effectively solve the above problem, this paper proposes a candidate set generation strategy using the candidate set expansion model based on the joint entity and relation linking [12]. The aim is to increase the coverage of correct candidate words by allowing more potential candidate words to appear in the candidate set. The candidate set expansion model first obtains keywords through existing keyword-extraction tools, then obtains the initial candidate entity set and candidate relation set through entity/relation prediction and candidate element generation. The initial candidate relation set is then used to obtain the head and tail nodes connected to it in the knowledge graph. Then the cosine similarity [13] of the vector is used to filter out the less-relevant candidates to obtain the expanded entity candidate set. The relations directly connected to each node in the expanded entity candidate set are added to the initial relation candidate set. Then the

less-relevant relation terms are filtered out using semantic types, and finally, the expanded relation candidate set is obtained.

The main contributions of this paper are in two aspects:

(**1**) For joint entity and relation linking, this paper proposes a candidate set expansion module for entity and relation linking. This module improves recall by expanding the entity and relation candidate sets.

(**2**) For the candidate set expansion module of the joint entity and relation linking, this paper performs candidate set expansion through the interaction between entities and relations. We use different filtering algorithms for the expanded entity/relation candidate sets. The main application of this paper is for further expansion of the entity/relation candidate set before the two steps of entity and relation linking. The query generation module will use entity and relation linking generated after we have expanded the entity/relation candidate set. Our approach can be applied to many semantic-parsing-based KBQA queries. We provide a new way of thinking about the candidate set expansion.

The remainder of this paper is organized as follows. In Section 2, the related work of this paper is described, introducing related research and outlining the idea of the approach of this paper. Section 3 describes the candidate set generation expansion performed on the entity and relation candidate set. Section 4 experiments with and evaluates the method. Finally, Section 5 focuses on the summary and future optimization ideas.

2. Related Work

Early KBQA efforts focused on simple problems. In recent years, researchers have gradually increased their attention to complex questions, and complex questions contain more than one entity word and relation word. When multiple entity words and relation words exist in a problem, it becomes difficult to link the entity words and relation words in the question to the corresponding entities and relations in the knowledge base. At this point, many people start to increase their research efforts on entity and relation linking. In recent years, deep learning has been widely used in entity linking. Several studies have introduced neural networks into entity-linking models and used deep-learning-based models to solve entity-linking problems. For example, BERT [14] and RoBERTa [15] have been widely used in entity-linking tasks with good results. The application of entity linking in multilingual situations has also received attention. Some studies have explored cross-lingual entity-linking [16] approaches using knowledge bases in the same language for linking entities in multiple languages. Other studies have focused on using multilingual knowledge bases or multilingual training data to solve multilingual entity-linking [17] problems. Some studies have explored the use of knowledge base embedding [18] and transfer learning [19] to address entity and relation linking. Knowledge base embedding can embed entities and relations into a vector space, thus supporting vector-based entity-linking methods. Transfer learning, on the other hand, can use existing knowledge bases and training data to improve the performance of new entity-linking models. Some studies combine entity and relation linking for joint learning to improve the performance of both tasks. For example, some studies have proposed methods to perform entity and relation linking as a joint task [9,20]. Overall, recent research in entity and relation linking has focused on improving the performance of models and generalizing them to multilingual and multi-domain applications. Before that, a part of the research focused on the improvement and refinement of entity disambiguation [21–24] or relation disambiguation algorithms [25,26]. However, the candidate set generation module, which also has an important role in the overall linking system, has yet to receive much attention. When the coverage of the candidate set generation module is low, the sorted candidate set may not contain the correct answers, thus reducing the performance of the linking model and hence the accuracy of the quiz.

For entity and relation linking, the EARL [9] system is our baseline. The entity-linking tools we use are FOX [27], Babelfy [28], DBpedia Spotlight [29], Tagme [30], EARL [9], and Falcon [20], and the relation-linking tools are SIBKB [31], ReMatch [32], EARL, and Falcon.

Many researchers have proposed some famous linking tools or methods. The earliest entity recognition did not use ensemble learning, which led Speck R et al. to submit the open-source NER framework FOX [27]. They combined four ways using 15 algorithms for ensemble learning and evaluated their performance on five datasets. This framework effectively reduces the entity recognition systems' error rate. Entity linking (EL) and word sense disambiguation (WSD) are designed to address lexical ambiguity in language. However, although the two tasks are very similar, they differ in one fundamental aspect. In EL, the textual mention can be linked to a named entity that may or may not contain the exact mention, whereas in WSD, there is an exact match between the form of the word (preferably its lexical meaning) and the appropriate lexical meaning. This led Moro A to propose Babelfy [28], a unified graph-based approach to entity linking and lexical disambiguation. The researcher proposed DBpedia Spotlight [29], a system for automatically annotating text documents with DBpedia URIs, in order to link text documents with the associated open data so that the Web of Data can be used as background knowledge in document-oriented applications. This system has good results in entity disambiguation. Ferragina P et al. have annotated short texts, such as snippets of search engine results, news, etc. They proposed the Tagme [30] system, which effectively adds hyperlinks to relevant Wikipedia pages in plain text. Traditionally, entity and relation linking are executed as dependent sequential tasks or independent parallel tasks. This led Dubey M et al. to propose a framework called EARL [9], which executes entity and relation linking as a joint task. This system is also the baseline taken in this paper. The most recent study of the joint entity and relation linking related to the candidate set is the Falcon model [20] proposed by Sakor A et al. Falcon overcomes the challenges of short texts using a lightweight linguistic approach that relies on a background knowledge graph. It uses several fundamental principles of English lexicography (e.g., compounding, central word recognition) for joint entity and relation linking of short texts. It uses an expanded knowledge graph created by merging entities and relations from different knowledge sources. Its performance in terms of recall is excellent, but the model ignores the impact of the deep semantic information of the question itself on the whole linking process. Singh K et al. found that the limited availability of semantic knowledge sources and the lack of a systematic approach to maximizing the benefit of the collected knowledge affect the performance of relational linking methods. They proposed a semantic-based index SIBKB [31], which captures the knowledge encoded in the background knowledge base and significantly improves the accuracy of relational linking. Some researchers studied identifying which attribute in the knowledge graph matches with a predicate in a natural language (NL) relation. At that time, common query generation methods mainly solved this problem by retrieving named entities and their predicate lists from the knowledge graph and filtering one from all predicates of that entity. This led Mulang to try a method to directly match NL predicates with knowledge graph (KG) attributes, which can be used in a QA pipeline. He also proposed a relation-linking tool, ReMatch [32].

Under the influence of the research that found EARL and Falcon, we found that more work needs to be carried out in the candidate set expansion. This paper investigates the candidate set generation expansion in the direction of relation linking. The candidate set expansion module includes two sub-modules, entity candidate set expansion and relation candidate set expansion. Through research and analysis, the existing keyword-extraction tools cannot identify the complete entity keywords well when the number of words in the problem is large or the entity keyword appears in its abbreviated form in the problem. There is usually more than one entity in such questions. It is easy to identify the relation keywords. We can find the triads related to this relation in the corresponding knowledge graph by the relation keywords and add the entity nodes in the triads to the initial entity candidate set to improve the coverage of the correct entity candidates. When the relations in natural language problems are complex, the keyword-extraction tool may not be able to extract the correct relation keywords, or the relation candidate set may be incomplete, and the relation candidate set generated on this basis cannot cover the correct relation

disambiguation objects, which will reduce the accuracy of relation linking at this time. For such problems, we can use the existing entity words to find the triad corresponding to this entity word in the knowledge graph. The relation edges in the triad can be added as new candidate relations to the initial set of relation candidates so that the initial set of relation candidates can include the correct candidate relations as much as possible.

This research paper uses the relation words in the initial relation candidate set, selects two entity nodes in the triad paths connected by this relation in the knowledge graph to form a new entity candidate set, encodes a representation of the problem and each entity word in the new entity candidate set, then calculates the cosine similarity between these two vectors, sets a threshold for the similarity, and selects the candidate entity words outside the threshold to add to the initial entity candidate set to obtain the expanded entity candidate set. Similarly, this paper iterates through each entity in the expanded entity candidate set, adds the relation words connected with this entity word in the knowledge graph to the initial relation candidate set to obtain the new relation candidate set, filters out the relation words with low relevance with the existing algorithm, and selects the candidate relation words beyond the threshold to form the expanded relation candidate set. In this paper, by expanding these two candidate sets, we improve the coverage rate, increase the probability that the correct candidate words appear in the candidate set, and finally improve the accuracy and recall of the linking. The experiments in this paper analyze the candidate recall of entity linking and the candidate recall of relation linking separately to verify the effectiveness of our method.

3. Candidate Set Generation Expansion

This section deals with the expansion of the entity candidate set and the expansion of the relation candidate set. In order to expand the original entity candidate set, this paper proposes a relation-based entity candidate set expansion module. The entity candidate set obtained through the existing tool is called the initial entity candidate set, and the relation candidate set is called the initial relation candidate set. The main method is retrieving the entity words in the relational triad associated with the sentence by traversing the relation words in the initial relation candidate set and obtaining new candidate entity words. Finally, the number of entity candidates is reduced using filtering rules to obtain an expanded entity candidate set. In order to expand the original relation candidate set, this paper proposes an entity-based relation candidate set expansion module. The main method is to select the relations in the knowledge graph connected to the entity word as new relation candidates by traversing each entity word in the expanded entity candidate set. The number of relation candidate is then reduced using filtering rules to obtain the expanded relation candidate set. After expanding the entity and relation candidate set, we can obtain all possible candidate entity and candidate relation words. Then the correct disambiguation objects are filtered from these candidates and returned as the final linking result. Finally, the candidate words are input as the final candidate set results. We obtain the final linking results after performing joint entity and relation linking. Based on the description above, Figure 2 shows the difference between the linking method proposed in this paper and commonly used linking methods. To fully describe the improved linking model framework in this paper, we describe the data processing procedure for question (1) "What is the budget of the film directed by Paul Anderson and named Resident Evil: Retribution?" in our method, as shown in Figure 3. We generate the initial entity and relation candidate sets through the candidate set generation module, followed by relation-based entity candidate set expansion and entity-based relation candidate set expansion. Finally, we obtain the expanded entity and relation candidate sets. After applying our method to question (1), the most appropriate relation candidate set is obtained. The correct relation candidate set can generate the correct query statement during the query construction process in KBQA.

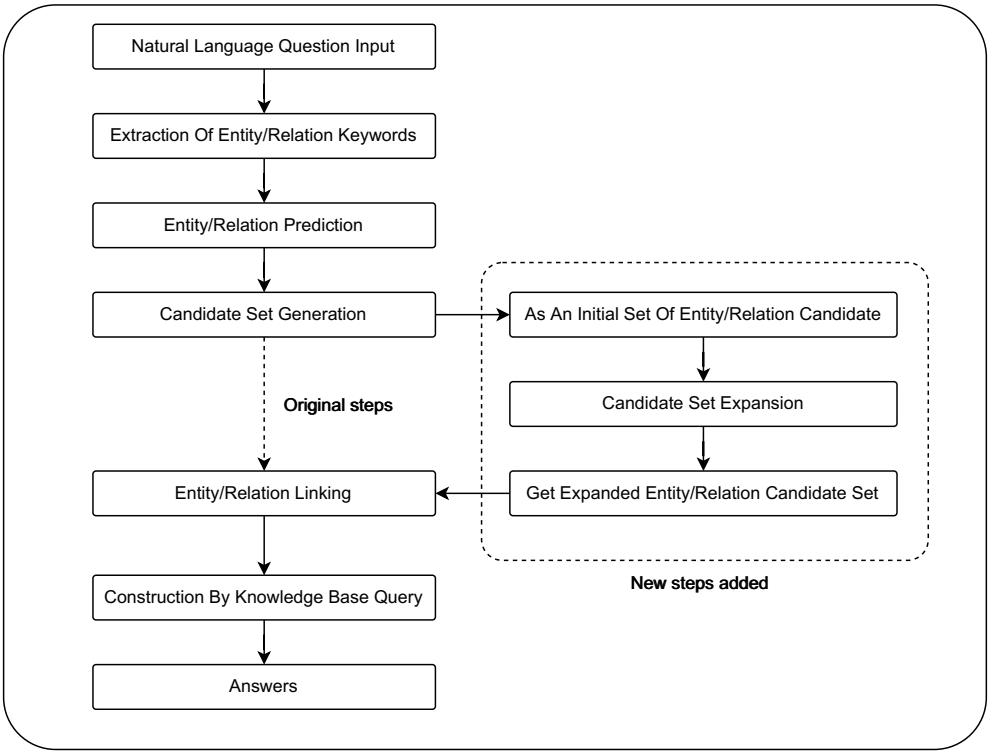

Figure 2. Relation-linking method for adding candidate set expansion.

3.1. Relation-Based Expansion of Entity Candidate Set

The entity candidate set expansion module first obtains new candidate entities based on the relations in the question and the knowledge graph subgraphs related to the natural language problem. It then filters out the less-relevant candidates using a simple similarity calculation to ensure that the correct disambiguation objects exist in the candidate entity list and to improve the efficiency of entity linking while reducing the number of candidate entities. The framework diagram of the entity candidate set expansion algorithm is shown in Figure 4. The specific steps are as follows:

(1) Keyword extraction: extraction of entity keywords and relational keywords from natural language;

(2) Entity–relation prediction: identifying which of the keywords are entity types and which are relation types;

(3) Candidate element generation: a candidate list is generated for each identified entity word and relation word, called the initial entity candidate set and the initial relation candidate set, respectively;

(4) Entity set expansion: makes full use of the relation terms in the question and adds more candidates that may contain the correct disambiguation object to the initial set of entity candidates;

(5) Entity candidate set filtering: filtering of new candidate entities using similarity matching algorithms; it selects the new candidate entities at the top of the relevance ranking to add to the initial set of entity candidates, and the resulting expanded entity candidate set will be used as the input of the entity-linkage model.

Steps (1), (2), and (3) are relatively simple and will not be repeated here. This paper focuses on (4) and (5).

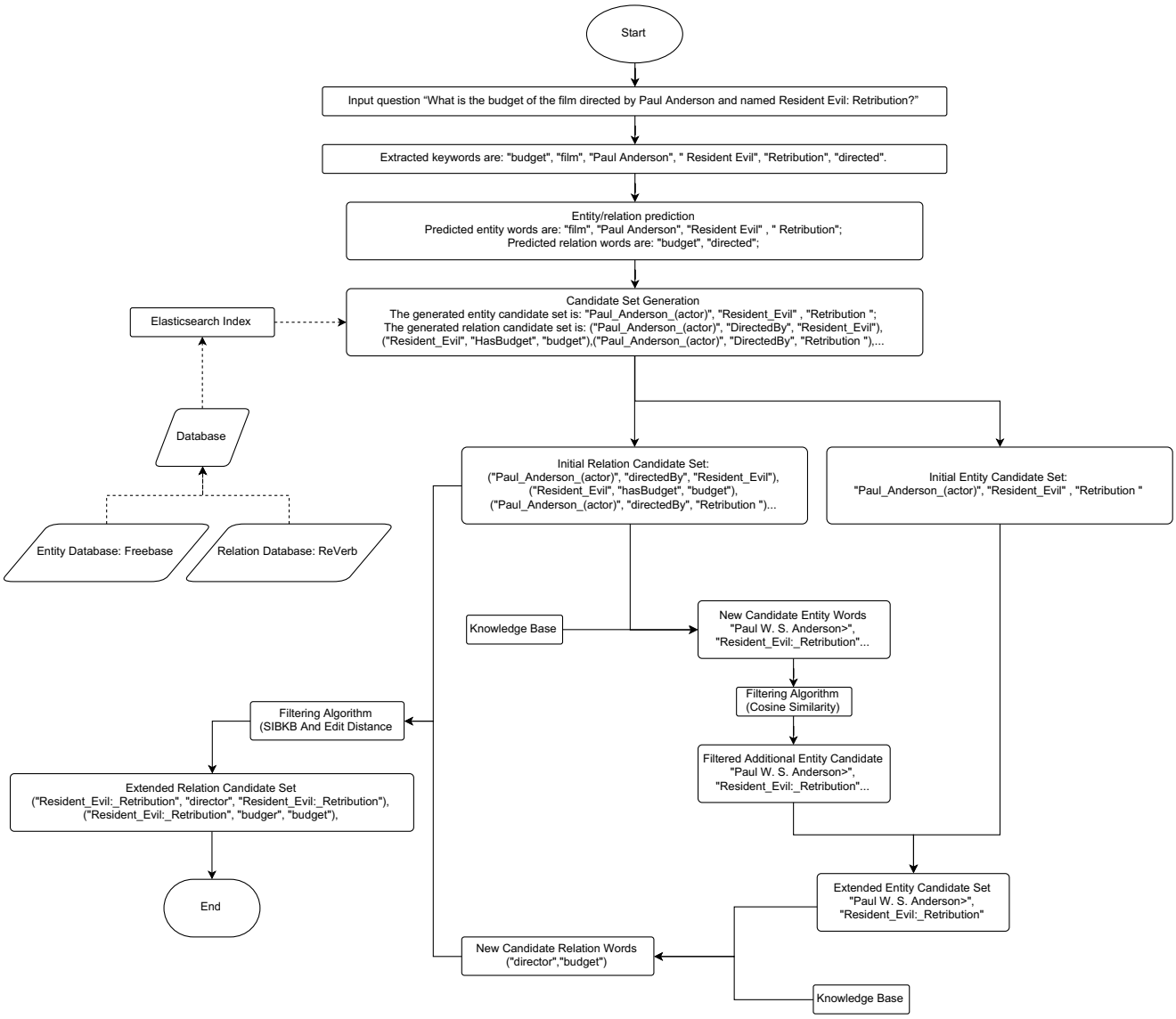

Figure 3. The data-processing procedure of question (1) in the methodology of this paper.

3.1.1. Entity Set Expansion

Taking the question "What is the budget of the film directed by Paul Anderson and named Resident Evil: Retribution?" as example (1), its correct partial RDF diagram is shown in Figure 5. This is the question (1) "What is the budget of the film directed by Paul Anderson and named Resident Evil: Retribution?" in the knowledge. This is a partial subgraph of the knowledge graph. Figure 5 mainly depicts the entities and relations related to question (1). The two correct entities for this paper are marked in green. Due to the current keyword tool itself, "Resident Evil" and "Retribution" are used as the keywords for the entity "Resident Evil: Retribution" in question (1). The entity keyword "Paul_W._S_Anderson" for "Paul Anderson" is not added to the set of entity candidates.

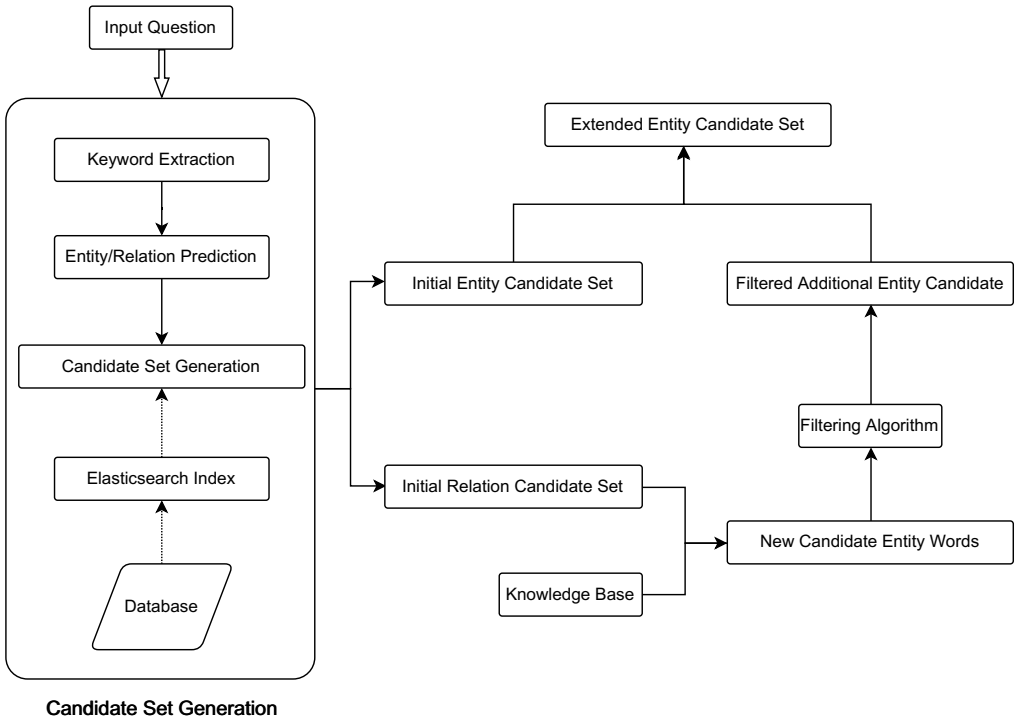

Figure 4. Framework of entity candidate set expansion algorithm.

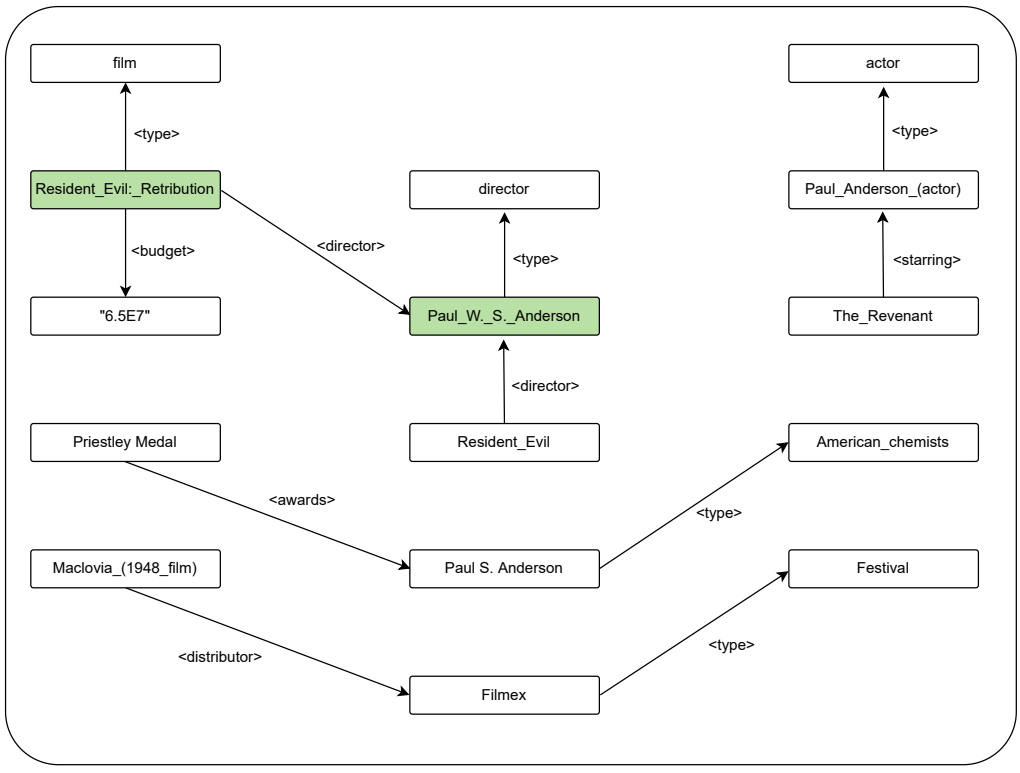

Figure 5. The correct partial RDF diagram of example (1).

Due to a flaw in the keyword-extraction tool, the entity keyword "Resident Evil: Retribution" in the question was extracted separately as "Resident Evil" and "Retribution", resulting in a list of candidate elements that did not contain the correct disambiguation object when the similarity calculation was performed. The partial candidate list of example (1) obtained with steps (1), (2), and (3) is shown in Figure 6. If the entity candidate

set expansion method proposed in this paper is not used, the correct entity candidate for "Resident Evil: Retribution" is not added to the entity candidate set during entity disambiguation for question (1). This leads to incorrect links to the entity words "Resident Evil" and "Retribution" when disambiguating entities. This will lead to an error in the final result. By looking at Figure 6, we can see that the relation keyword "direct by" in the question corresponds to the relation candidate "<director>", and the head node connected in the knowledge graph is the correct candidate entity word in this problem. At this point, both the head and tail nodes connected to this relation in the knowledge graph can be added as new candidate entity words to the initial entity candidate list, forming an expanded candidate entity set. As can be seen from Figure 5, the new candidate entity words include "<Resident_Evil:_Retribution>", "<6.5E7>", and "<Resident_Evil>".

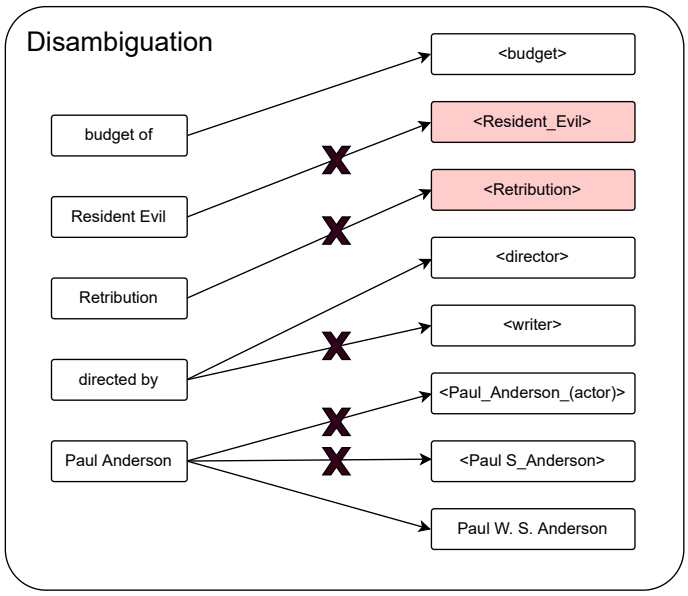

Figure 6. The partial candidate list of example (1).

3.1.2. Entity Candidate Set Filtering

Usually, there is more noise in the expanded set of entity candidates obtained from Section 3.1.1. This paper filters the added candidate entities to improve the recall while ensuring the accuracy of entity linking. First, for the added entity candidate set $NewE = ne_1, ne_2, ne_3, \ldots ne_n$, they were each transformed into vector representations using the word-embedding matrix [33]. Then the interrogatives were transformed into vector representations using the same method. We then iterated over each vector representation in the new entity candidate set and calculated the similarity with the vector representation of the question. The cosine similarity was introduced by Salton et al. as a measure of text similarity to calculate the similarity between documents and became one of the classical methods in the field of information retrieval. In the following decades, the cosine similarity has been widely studied and applied. It has become one of the most important methods for calculating text similarity. This paper uses the cosine distance to calculate the similarity (Formula (1)).

$$\frac{\sum_{i=1}^{n}(Vne_i \times Q)}{\sqrt{\sum_{i=1}^{n}(Vne_i)^2} \times \sqrt{Q^2}} \tag{1}$$

Vne_i represents the word vector representation of the candidate entity word, and Q represents the word vector representation of the interrogative sentence. We set a threshold N for similarity and filter out new candidate entities with low relevance, and all new

candidate entities greater than N are added to the initial entity candidate set to form an expanded entity candidate set.

For the new candidate entity word "<Resident_Evil:_Retribution>" in the above example, the vector representation obtained from the word-embedding matrix is used to calculate the similarity with the vector representation of the question. The candidate entity term "<Resident_Evil:_Retribution>" has a high weight factor due to such an entity keyword in the question itself. The new candidate entity word "<6.5E7>" as the answer to the question, which is not present in the question, will receive lower attention when performing the similarity calculation. Therefore the candidate entity word "<Resident_Evil:_Retribution>" is added to the initial set of entity candidates for expansion.

3.2. Entity-Based Expansion of Relation Candidate Set

The relation candidate set expansion module first obtains new candidate relations based on the entities in the question and the subgraphs of the knowledge graph associated with the natural language problem. Then it filters the less-relevant candidate relations through a ranking algorithm to ensure that the correct disambiguation objects are present in the list of candidate relations and to improve the efficiency of the relation-linking algorithm while reducing the number of candidate relations. The specific steps are as follows.

The first three steps are the same as steps (1), (2), and (3) in Section 3.1.

(4) Relation set expansion: makes full use of the entity words in the question and adds more candidate relations that may contain the correct disambiguation object to the initial set of relation candidates;

(5) Relation candidate set filtering: reuses the candidate set sorting step in the SIBKB [31] model to filter the initial set of relation candidates and the new candidate relation words. It selects the relations at the top of the correlation ranking as input to the relation-linking model.

The framework diagram of the relation candidate set expansion algorithm is shown in Figure 7. We go through steps 1, 2, and 3 for the candidate set generation operation. We perform an entity-based relation candidate set expansion operation on the initial relation candidate set obtained and the expanded entity candidate set to output in Section 3.1.2. With the entity candidate set expansion and the knowledge base, we obtain the relation terms for each entity in the expanded entity candidate set connected in the knowledge graph. We filter these relation words together with the relation words in the initial candidate set for the relation words. The final expanded relation candidate set is obtained.

3.2.1. Relation Set Expansion

We take the question "Which comic characters are painted by Bill Finger?" as example (2). If the entity keyword "Bill Finger" can be extracted with just steps (1), (2), and (3), one of the candidate entity words "dbr: Bill_Finger" is obtained. At the same time, "painted by" is extracted as another keyword, and "dbo: painter" is added to the initial set of relation candidates as a candidate according to the ElasticSearch indexing dictionary. The result of the final relation link is then entered into the query generation module along with "dbr: Bill_Finger". The result of querying on DBPedia for the constructed SPARQL [34] query statement is an empty set. In this paper, the relation candidate set is expanded based on the expanded entity candidate set. The specific method is to traverse each entity word in the expanded entity candidate set and retrieve the relation directly connected with the entity in the subgraph related to the sentence as a new candidate relation word. This expansion method can add "dbo:creator" to the relation candidate set for the above problem.

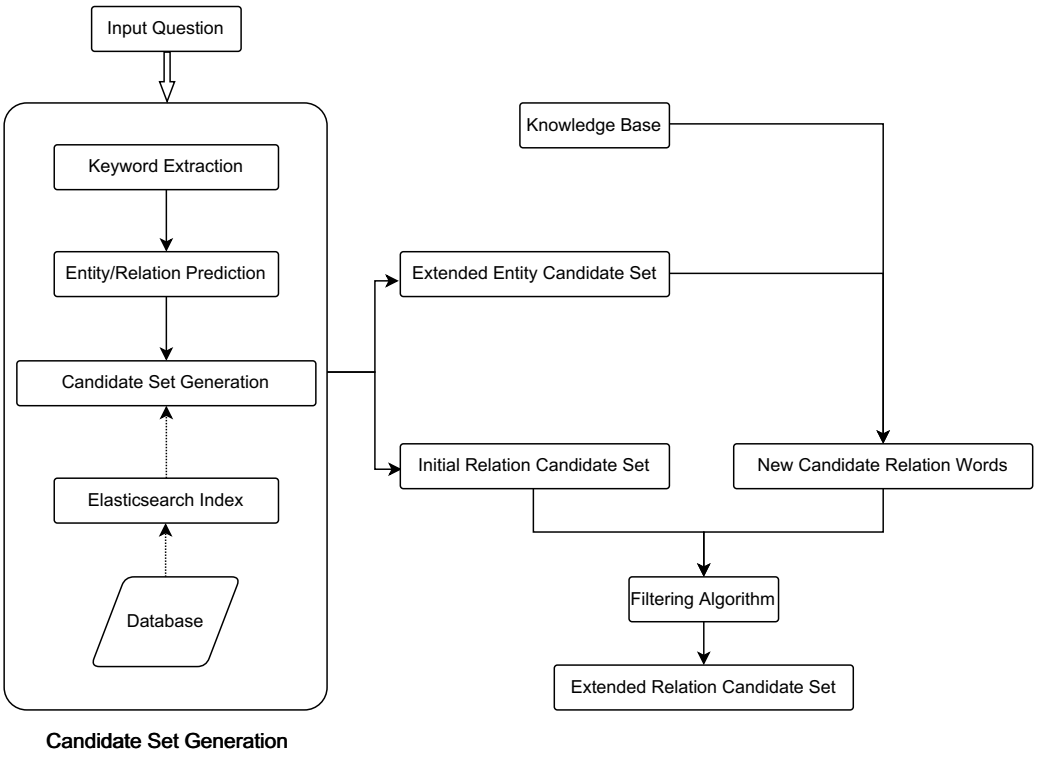

Figure 7. Framework diagram of the relation candidate set expansion algorithm.

3.2.2. Relation Candidate Set Filtering

Some of the possible candidate terms obtained in the relation set expansion phase are not highly relevant to the problem. If they are used as the input to the relation-linking model directly, it will increase the noise of the system and thus reduce the efficiency of the whole linkage system. Therefore, before performing relation linking, this paper filters the newly added candidate relation words and the initial candidate relation set.

The filtering model in this paper reuses the methods in the SIBKB model. A KEY-VALUE list is obtained by performing the candidate set sorting step in the SIBKB model. KEY represents the relation candidate word, and VALUE represents the similarity score between the candidate word and the relation keyword corresponding to the candidate word. The filtering algorithm used in this paper is based on the following steps:

Step 1: Assign weighting factors

(1) When the newly added candidate relation words or the relation word nr in the initial relation candidate set are in the relation list KEY, a higher weight factor w_1 is added to the VALUE corresponding to nr. The calculation is shown in Formula (2):

$$V_{nr} = v + w_1 \tag{2}$$

v denotes the similarity score in the KEY-VALUE list.

(2) When nr is not in the relation list KEY, nr is inserted into the KEY-VALUE list as a new element, and nr is given an initial weight coefficient w_2, as shown in Formula (3).

$$V_{nr} = w_2 \tag{3}$$

Step 2: Reorder the candidate set using the edit distance [35].

The edit distance was first introduced by the Russian mathematician Vladimir Levenshtein in 1965 and is therefore also known as the Levenshtein distance. This is achieved by comparing a query string with a candidate item and calculating the edit distance between them. The smaller the edit distance, the more similar the two strings are. We can then sort the candidates by their edit distance from the query string, with the candidate with

the smallest distance coming first. A new relation candidate set NewRelationSet can be obtained by expanding the relation candidate set described above and increasing the weight coefficients. The edit distance is calculated between each relation keyword and the relation candidate R from the NewRelationSet, the relation candidate R whose edit distance ED is between [0,1] is taken, and a new weight factor w_3 is added to R, where the larger the ED value is, the smaller the w_3 is. Formula (4) for V_nr is shown below.

$$V_{nr} = V_{nr} + w_3 \tag{4}$$

We set a threshold value of T for V_nr, and the relation candidate words with a low correlation are filtered out. All relation candidate words greater than T form an expanded relation candidate set.

4. Experiments and Evaluation

This section focuses on the datasets and baseline, the description of the experimental settings and evaluation metrics, and the evaluation of the experiment.

4.1. Datasets And Baseline

This paper exploited the Large-Scale Complex Question Answering Dataset (LC-QuAD [36]) and the 7th edition of the Question Answering over Linked Data Challenge (QALD-7 [37]) dataset. The LC-QuAD dataset comprises 5000 complex questions from DBpedia with an average length of 12.29 words. In total, 80% percent of the questions have more than one entity and relation. QALD-7 comprises 215 questions, and it is the most popular QA benchmark dataset on DBpedia. In QALD-7, the average question length is 7.41 words, and more than 50% of the questions include one entity and relation.

For entity and relation linking, the EARL [9] system is our baseline. The entity-linking tools we use are FOX [27], Babelfy [28], DBpedia Spotlight [29], Tagme [30], EARL, and Falcon [20], and the relation-linking tools are SIBKB [31], ReMatch [32], EARL, and Falcon.

FOX: It is an approach to named entity recognition based on ensemble learning. It uses multiple algorithms composed of different methods for experimentation on different datasets.

Babelfy: It is a graph-based unified approach for entity linking and word-sense disambiguation. It is based on the loose identification of candidate meanings. It is the densest subgraph heuristic algorithm selected for highly consistent semantic interpretation.

DBpedia Spotlight: It is an annotation tool for finding mentions of DBpedia resources in free text. DBpedia Spotlight allows the configuring of annotations to specific use cases through quality metrics such as the topic relevance and disambiguation confidence.

Tagme: It constructs an anchor dataset based on the linking relations of words in Wikipedia, constructs an anchor candidate set with anchor point parsing of the input text, and selects the set of candidate link entities with the largest overall relevance as the final entity-linking result.

EARL: It is a joint entity and relation disambiguation system. The disambiguation we are talking about here is link disambiguation. EARL is a single-task system that treats entity and relation linking. Its goal is also simple, i.e., to reduce the errors arising from interdependencies at each step.

Falcon: It is a rule-based tool. It can accurately map entities and relations in short texts to resources in the knowledge graph. Falcon resorts to fundamental principles of English morphology (e.g., headword identification and compounding) and performs joint entity and relation linking against a short text.

SIBKB: It is a semantic-based index. SIBKB provides a search mechanism that accurately links relational patterns to semantic types. It represents a background knowledge base as a bipartite and dynamic index over the relational patterns included in the knowledge base.

ReMatch: It attempts to directly match natural language predicate to knowledge graph properties. It models the knowledge base relations with their underlying parts of

speech, then enhances its approach with additional attributes obtained from Wordnet and dependency-parsing characteristics.

4.2. Experimental Settings and Evaluation Metrics

The max-length for entity candidates, relation candidates, and questions is 30. The dimension w of word embedding is 300. The output dimension d of the word vector is 200. The window size of max-pooling is 400. A dropout rate of 0.5 is used to avoid overfitting, and the epoch is 10. For the initialization of the matrix E_W, this paper uses the pre-trained model GloVe [33]. During the training process, we also use dynamic learning rate adjustment to improve the model's performance. The metrics accuracy (Acc) and recall (Rec) are commonly used when evaluating the performance of the joint entity- and relation-linking system.

Accuracy refers to the ratio of the number of entities and relations correctly identified by the system to the total number of entities and relations output by the system, i.e., Formula (5).

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where *TP* denotes the true cases, i.e., the entities and relations correctly identified by the system; and *TN* denotes the true negative cases, i.e., the entities and relations not correctly identified by the system. *FP* denotes the false positive cases, i.e., the non-entities or non-relations incorrectly identified by the system as entities or relations. *FN* represents the false negative cases, i.e., the entities or relations not correctly identified by the system.

Recall refers to the ratio of the number of entities and relations correctly identified by the system to the number of true entities and relations, i.e., Formula (6).

$$Rec = \frac{TP}{TP + FN} \tag{6}$$

TP and *FN* are described in Formula (5). Accuracy and recall affect each other; typically, increasing the accuracy will decrease the recall and vice versa. Therefore, when evaluating a joint entity- and relation-linking system, it is necessary to balance the accuracy and recall and to choose the appropriate threshold to achieve the optimal performance.

4.3. Experimental Evaluation

In this section, the expanded entity set and expanded relation set are added to the entity- and relation-linking model based on feature union and multi-attention [12] for experimentation, which is used to verify the method's effectiveness in this pater in the linking process. The entity-linking comparison and relation-linking comparison are performed separately. The results are analyzed with the entity-linking recall and relation-linking recall. This paper expands the existing entity candidate set and relation candidate set. We reuse the EARL method to generate entity and relation candidate sets. This paper expands them as the initial entity candidate set and initial relation candidate set, respectively. At present, the accuracy of EARL in relation linking could be better. The ultimate purpose of this paper is to verify the improvement of the method's accuracy in this paper. Therefore, EARL is used as the baseline. The Falcon model is the best research achievement in joint entity and relation linking. Although the Falcon model is not used in this paper, the final experimental results are still lower than those of Falcon. We need to compare with the better methods of the current relevant research and analyze the reasons for this. This will help us improve the candidate set expansion model based on the entity–relation interaction.

4.3.1. Entity-Linking Comparison and Result Analysis

Comparative experiments are conducted with other models regarding candidate recall in this paper. As shown in Table 1, the Falcon model performs well regarding recall due to its introduction of an expanded knowledge graph. The model proposed in this paper performs inferiorly compared to the Falcon model. Compared to the EARL model, the method in

this paper has a larger improvement in recall. The improvement is 29.8% on the LC-QuAD dataset and 17.8% on the QALD-7 dataset. This is because the method in the EARL model is reused for entity candidate set generation in Section 3 of this paper. The number of entity candidate sets containing the correct candidate words is significantly increased after using entity candidate set expansion in this paper, and consequently, the recall is improved.

Table 1. Performance of our method compared to various entity-linking tools.

Method	Dataset	Recall (Rec)%
FOX [27]	LC-QuAD [36]	51.3
Babelfy [28]	LC-QuAD	49.8
DBpedia Spotlight [29]	LC-QuAD	65.2
Tagme [30]	LC-QuAD	77.1
EARL [9]	LC-QuAD	55.3
Falcon [20]	LC-QuAD	**86.4**
Our method	LC-QuAD	85.1
FOX	QALD-7 [37]	57.1
Babelfy	QALD-7	55.3
DBpedia Spotlight	QALD-7	72.4
Tagme	QALD-7	76.2
EARL	QALD-7	60.3
Falcon	QALD-7	**79.2**
Our method	QALD-7	78.1

To further demonstrate the effectiveness of the entity candidate set expansion module proposed in this paper, this section verifies the entity-linking results from candidate recall (recall, Rec) and link accuracy (accuracy, Acc). The use of the expanded entity candidate set in the federated entity relation linkage model is represented as With Entity Set Expansion (With ESE), and the absence of the expanded entity candidate set is represented as Without Entity Set Expansion (Without ESE).

The experimental results are shown in Table 2. After expanding the entity candidate set, the correct entity-linking rate of the joint linking model in this paper improved from 83.3% to 85.4% in the LC-QuAD dataset and from 74.1% to 76.0% in the LC-QuAD dataset. This is because before we used the extended entity candidate set, the set of candidate entities did not contain the correct disambiguation object due to the long entity words in some of the questions. After using the extended entity candidate set, we improved the coverage of the correct candidate words, thus increasing the link correctness.

The method in this paper also obtains a relatively good performance regarding candidate recall. The recall was 58.3% in the LC-QuAD dataset and 60.4% in the QALD-7 dataset before using the expanded entity candidate set. The recall improved by 26.8% and 17.7% after using the expanded entity candidate set. This is because the number of complex problems in the LC-QuAD dataset is high, and the entity words in the questions are relatively complex. After using the entity candidate set expansion scheme proposed in this paper, the number of entity candidate sets containing the correct candidate words can be increased, so the enhancement effect is relatively more obvious in the LC-QuAD dataset.

Table 2. Experimental comparison results of With ESE and Without ESE models.

Method	Dataset	Recall (Rec)%	Accuracy (Acc)%
With ESE	LC-QuAD	85.1	85.4
Without ESE	LC-QuAD	58.3	83.3
With ESE	QALD-7	78.1	76.0
Without ESE	QALD-7	60.4	74.1

4.3.2. Relation-Linking Comparison and Result Analysis

Comparative experiments (Table 3) are conducted with other models regarding candidate recall in this paper. The model proposed in this paper performs inferiorly compared to the Falcon model. Compared to the EARL model, the method in this paper greatly improves the recall. The improvement is 23.1% on the LC-QuAD dataset and 31.4% on the QALD-7 dataset. This is because the entity candidate set generation in Section 3 of this paper reuses the method in the EARL model. The number of relation candidate sets containing the correct candidate words is significantly increased after using relation candidate set expansion in this paper, and consequently, the recall is improved.

Table 3. Performance of our method compared to various relation-linking tools.

Method	Dataset	Recall (Rec)%
SIBKB [31]	LC-QuAD	15.4
ReMatch [32]	LC-QuAD	17.3
EARL	LC-QuAD	21.2
Falcon	LC-QuAD	**44.6**
Our method	LC-QuAD	44.3
SIBKB	QALD-7	31.2
ReMatch	QALD-7	34.3
EARL	QALD-7	28.1
Falcon	QALD-7	**61.4**
Our method	QALD-7	**59.5**

To further demonstrate the effectiveness of the relation candidate set expansion module proposed in this paper, for relation-linking results, this section validates both candidate recall (recall, Rec) and link accuracy (accuracy, Acc). The use of expanded relation candidates in the joint entity- and relation-linking model is represented as With Relation Set Expansion (With RSE), and the absence of expanded relation candidates is represented as Without Relation Set Expansion (Without RSE).

The experimental results are shown in Table 4. This paper's joint entity- and relation-linking model improves the relation-linking accuracy rate from 46.4% to 48.0% on the LC-QuAD dataset and from 42.3% to 43.4% on the QALD-7 dataset after expanding the relation candidate set. This is because before using the extended relation candidate set, complex relations, including implicit relations, were present in some of the questions due to their presence. We could not extract the correct relation terms using the keyword-extraction tool. However, we improved the coverage of complex relations using the expanded relation candidate set, thus improving the correct linking rate.

The method in this paper also obtains a better performance in terms of candidate recall. Before using the expanded relation candidate set, the recall was 23.2% in the LC-QuAD dataset and 43.3% in the QALD-7 dataset. The recall improved by 21.1% and 16.2%, respectively, after using the expanded entity candidate set. This is because there are more complex relations in the LC-QuAD dataset. The number of relation candidate sets containing the correct candidate words is significantly increased after performing relation candidate set expansion. Consequently, the recall is also improved.

Table 4. Experimental comparison results of With RSE and Without RSE models.

Method	Dataset	Recall (Rec)%	Accuracy (Acc)%
With ESE	LC-QuAD	44.3	48.0
Without ESE	LC-QuAD	23.2	46.4
With ESE	QALD-7	59.5	43.4
Without ESE	QALD-7	43.3	42.3

5. Conclusions and Future Work

For entity and relation linking, this paper investigates the aspect of the candidate set information expansion. Two methods are proposed to expand the candidate set, making full use of the identified entity and relation keywords to improve the coverage of the candidate set. The above two approaches are used to improve the entity- and relation-linking accuracy. Experiments are conducted using two standard datasets and the DBpedia knowledge base to confirm the effectiveness of the methods in this paper. After analyzing the experimental error data, this paper's relation candidate set module can include as many relation words as possible. Still, they are all based on known entities in the problem. However, in some problems, there may be no entity words, which will lead to the model not detecting the entities, and then it will not be able to expand the relation candidate set. In some problems, the entity words and relation words do not exist in the DBpedia database or have wrong information, which leads to incorrect linking results. In this case, multiple data sources need to be added to supplement the existing knowledge base.

Although the candidate set expansion method proposed in this paper improves the effect, there is still room for improvement. This paper considers the following aspects for improvement: (**a**) The typical features of the questions are usually closely connected with the keywords in the interrogative sentences, so the type-inspired features of the questions are considered to be added to the link model in the future as a way to more comprehensively represents the entity and relation words in the questions. (**b**) Candidate set expansion is currently based on simple rules for information extraction. In the future, remote supervision is considered to enable deeper inference of information to obtain more comprehensive information about the candidate set. (**c**) This paper does not consider the parallel or alternate way to maximize the expansion of the candidate set. In the future, the order of expansion and alternate expansion should be viewed to maximize the final candidate set.

Author Contributions: Formal analysis, J.G.; Resources, L.F.; Writing—original draft, Y.F.; Writing—review & editing, B.Z.; Supervision, F.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Hubei Province Educational Committee of funder grant number B2019009.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lan, Y.; He, G.; Jiang, J.; Jiang, J.; Zhao, W.X.; Wen, J.R. A survey on complex knowledge base question answering: Methods, challenges and solutions. *arXiv* **2021**, arXiv:2105.11644.
2. Shao, B.; Li, X.; Bian, G. A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. *Expert Syst. Appl.* **2021**, *165*, 113764. [CrossRef]
3. Liu, W.; Yin, L.; Wang, C.; Liu, F.; Ni, Z. Multitask healthcare management recommendation system leveraging knowledge graph. *J. Healthc. Eng.* **2021**, *2021*. [CrossRef] [PubMed]
4. Hsu, P.Y.; Chen, C.T.; Chou, C.; Huang, S.H. Explainable mutual fund recommendation system developed based on knowledge graph embeddings. *Appl. Intell.* **2022**, *52*, 10779–10804. [CrossRef]
5. Caldarini, G.; Jaf, S.; McGarry, K. A literature survey of recent advances in chatbots. *Information* **2022**, *13*, 41. [CrossRef]
6. Hao, T.; Li, X.; He, Y.; Wang, F.L.; Qu, Y. Recent progress in leveraging deep learning methods for question answering. *Neural Comput. Appl.* **2022**, *34*, 2765–2783. [CrossRef]
7. Lan, Y.; He, G.; Jiang, J.; Jiang, J.; Zhao, W.X.; Wen, J.R. Complex knowledge base question answering: A survey. In *IEEE Transactions on Knowledge and Data Engineering*; IEEE: Piscataway, NJ, USA, 2022.
8. Singh, K. Towards Dynamic Composition of Question Answering Pipelines. Ph.D. Thesis, Universitäts-und Landesbibliothek Bonn, Bonn, Germany, 2019.
9. Dubey, M.; Banerjee, D.; Chaudhuri, D.; Lehmann, J. EARL: Joint entity and relation linking for question answering over knowledge graphs. In Proceedings of the Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, 8–12 October 2018; Proceedings, Part I 17; Springer: Cham, Switzerland, 2018; pp. 108–126.

10. De Cao, N.; Wu, L.; Popat, K.; Artetxe, M.; Goyal, N.; Plekhanov, M.; Zettlemoyer, L.; Cancedda, N.; Riedel, S.; Petroni, F. Multilingual autoregressive entity linking. *Trans. Assoc. Comput. Linguist.* **2022**, *10*, 274–290. [CrossRef]
11. Naseem, T.; Ravishankar, S.; Mihindukulasooriya, N.; Abdelaziz, I.; Lee, Y.S.; Kapanipathi, P.; Roukos, S.; Gliozzo, A.; Gray, A. A semantics-aware transformer model of relation linking for knowledge base question answering. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Virtual Event, 1–6 August 2021; pp. 256–262.
12. Fu, L.; Liu, Z.; Qiu, C.; Gao, F. Joint entity and relation linking based on jointly feature and multi-attention. *Comput. Linguist.* **2020**, *48*, 53–93.
13. Savenkov, D.; Agichtein, E. When a knowledge base is not enough: Question answering over knowledge bases with external text data. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, Pisa, Italy, 17–21 July 2016; pp. 235–244.
14. Li, X.; Li, Z.; Zhang, Z.; Liu, N.; Yuan, H.; Zhang, W.; Liu, Z.; Wang, J. Effective Few-Shot Named Entity Linking by Meta-Learning. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 178–191.
15. Zhang, Y.; Liu, J.; Huang, B.; Chen, B. Entity Linking Method for Chinese Short Text Based on Siamese-Like Network. *Information* **2022**, *13*, 397. [CrossRef]
16. Schumacher, E.; Mayfield, J.; Dredze, M. Zero-shot Cross-Language Transfer of Monolingual Entity Linking Models. In Proceedings of the 2nd Workshop on Multi-lingual Representation Learning (MRL), Abu Dhabi, United Arab Emirates, December 2022; pp. 38–51.
17. Pratapa, A.; Gupta, R.; Mitamura, T. Multilingual event linking to wikidata. *arXiv* **2022**, arXiv:2204.06535.
18. Wang, Y.C.; Ge, X.; Wang, B.; Kuo, C.C.J. KGBoost: A classification-based knowledge base completion method with negative sampling. *Pattern Recognit. Lett.* **2022**, *157*, 104–111. [CrossRef]
19. Duan, K.; Du, S.; Zhang, Y.; Lin, Y.; Wu, H.; Zhang, Q. Enhancement of Question Answering System Accuracy via Transfer Learning and BERT. *Appl. Sci.* **2022**, *12*, 11522. [CrossRef]
20. Sakor, A.; Mulang, I.O.; Singh, K.; Shekarpour, S.; Vidal, M.E.; Lehmann, J.; Auer, S. Old is gold: Linguistic driven approach for entity and relation linking of short text. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MI, USA, 2–7 June 2019; pp. 2336–2346.
21. Makris, C.; Simos, M.A. OTNEL: A Distributed Online Deep Learning Semantic Annotation Methodology. *Big Data Cogn. Comput.* **2020**, *4*, 31. [CrossRef]
22. Procopio, L.; Conia, S.; Barba, E.; Navigli, R. Entity Disambiguation with Entity Definitions. *arXiv* **2022**, arXiv:2210.05648.
23. Barba, E.; Procopio, L.; Navigli, R. ExtEnD: Extractive entity disambiguation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 2478–2488.
24. Simos, M.A.; Makris, C. Computationally Efficient Context-Free Named Entity Disambiguation with Wikipedia. *Information* **2022**, *13*, 367. [CrossRef]
25. Xiong, B.; Bao, P.; Wu, Y. Learning semantic and relationship joint embedding for author name disambiguation. *Neural Comput. Appl.* **2021**, *33*, 1987–1998. [CrossRef]
26. Li, G.; Li, H.; Pan, Y.; Li, X.; Liu, Y.; Zheng, Q.; Diao, X. Name Disambiguation Based on Entity Relationship Graph in Big Data. In Proceedings of the Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, 21–24 November 2022; Proceedings, Part II; Springer: Cham, Switzerland, 2023, pp. 319–329.
27. Speck, R.; Ngonga Ngomo, A.C. Ensemble learning for named entity recognition. In Proceedings of the Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, 19–23 October 2014; Proceedings, Part I 13; Springer: Cham, Switzerland, 2014; pp. 519–534.
28. Moro, A.; Raganato, A.; Navigli, R. Entity linking meets word sense disambiguation: A unified approach. *Trans. Assoc. Comput. Linguist.* **2014**, *2*, 231–244. [CrossRef]
29. Mendes, P.N.; Jakob, M.; García-Silva, A.; Bizer, C. DBpedia spotlight: Shedding light on the web of documents. In Proceedings of the 7th International Conference on Semantic Systems, Graz, Austria, 7–9 September 2011; pp. 1–8.
30. Ferragina, P.; Scaiella, U. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Galway, Ireland, 19–23 October 2010; pp. 1625–1628.
31. Singh, K.; Mulang', I.O.; Lytra, I.; Jaradeh, M.Y.; Sakor, A.; Vidal, M.E.; Lange, C.; Auer, S. Capturing knowledge in semantically-typed relational patterns to enhance relation linking. In Proceedings of the Knowledge Capture Conference, Austin, TX, USA, 4–6 December 2017; pp. 1–8.
32. Mulang, I.O.; Singh, K.; Orlandi, F. Matching natural language relations to knowledge graph properties for question answering. In Proceedings of the 13th International Conference on Semantic Systems, Amsterdam, The Netherlands, 11–14 September 2017; pp. 89–96.
33. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.

34. Seaborne, A.; Manjunath, G.; Bizer, C.; Breslin, J.; Das, S.; Davis, I.; Harris, S.; Idehen, K.; Corby, O.; Kjernsmo, K.; et al. SPARQL/Update: A language for updating RDF graphs. *W3c Memb. Submiss.* **2008**, *15*. Available online: http://shiftleft.com/mirrors/www.hpl.hp.com/india/documents/papers/sparql.pdf (accessed on 28 February 2023).
35. Qu, Y.; Liu, J.; Kang, L.; Shi, Q.; Ye, D. Question answering over freebase via attentive RNN with similarity matrix based CNN. *arXiv* **2018**, arXiv:1804.03317.
36. Trivedi, P.; Maheshwari, G.; Dubey, M.; Lehmann, J. Lc-quad: A corpus for complex question answering over knowledge graphs. In Proceedings of the The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, 21–25 October 2017; Proceedings, Part II 16; Springer: Cham, Switzerland, 2017; pp. 210–218.
37. Usbeck, R.; Ngomo, A.C.N.; Haarmann, B.; Krithara, A.; Röder, M.; Napolitano, G. 7th open challenge on question answering over linked data (QALD-7). In Proceedings of the Semantic Web Challenges: 4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, 28 May–1 June 2017; Revised Selected Papers; Springer: Cham, Switzerland, 2017; pp. 59–69.