



# Relation Extraction in Knowledge Base Question Answering: From General-Domain to the Catering Industry

Hung-Chen Chen<sup>1</sup>, Zi-Yuan Chen<sup>1(✉)</sup>, Sin-Yi Huang<sup>1</sup>, Lun-Wei Ku<sup>1</sup>,  
Yu-Shian Chiu<sup>2</sup>, and Wei-Jen Yang<sup>2</sup>

<sup>1</sup> Institute of Information Science, Academia Sinica, Taipei, Taiwan  
{hankchen,hsy1106,lwku}@iis.sinica.edu.tw, b02902017@ntu.edu.tw

<sup>2</sup> Institute of Digital Transformation Institute, Institute for Information Industry,  
Taipei, Taiwan  
{samuelchiu,wjyang}@iii.org.tw

**Abstract.** Knowledge base question answering (KBQA) can be decomposed into entity linking and relation extraction. In KBQA relation extraction, the goal is to find the appropriate relation given the question and its linked entity. Previous work used neural network models to process entities in a pairwise manner, which is well-suited to large relation sets in KBQA. However, such models must execute the same relation detection procedure multiple times for each question to complete an exhaustive search of the relation combinations. In this paper, we propose treating relation extraction in KBQA as a classification problem. Moreover, we introduce a masking layer which filters out less probable relations in advance. Experiments show that the masking mechanism benefits the proposed model by improving the accuracy from 72% to 77%. In addition, a catering knowledge base is constructed automatically in this paper, on which the proposed model yields an accuracy of 89%, demonstrating its effectiveness.

**Keywords:** Question-answering · Relation detection  
Deep learning application · Knowledge-base question answering  
Knowledge-base relation extraction

## 1 Introduction

It has been shown that knowledge bases (KBs) such as DBpedia [1], Freebase [2], and WordNet [3] are effective sources of knowledge for applications such as hypernym detection [4], machine translation [5], and question answering [6–8]. Among these applications, knowledge-based question answering (KBQA), the process of requesting knowledge over a KB using natural language questions, is the most direct approach to access KB knowledge. As the common representation of knowledge graph beliefs is a discrete relational triple showing one relation between two entities, such as LocatedIn (NewOrleans, Louisiana), where the

two entities are two nodes and their relation is the edge between them in the knowledge graph, the procedure of answering a question can be transformed into a traversal that starts from the question entity and searches for the appropriate path to reach the answer entity. Therefore, KBQA can be divided into two major tasks: entity linking and relation extraction. The former discovers the topic entity in the question and then locates this entity in the KB; the latter attempts to discover all the relations in the path which connects the question topic entity and the answer entity. These two tasks are illustrated in Fig. 1. Given the question “What is the name of Justin Bieber brother?”, the entity linking process identifies “Justin Bieber” as the topic entity and begins to search for candidate relations from it; the relation extraction process identifies “sibling\_s” and “sibling” as the relations in the form of a relation path between the topic entity and the answer entity. In this paper, we focus on relation extraction.

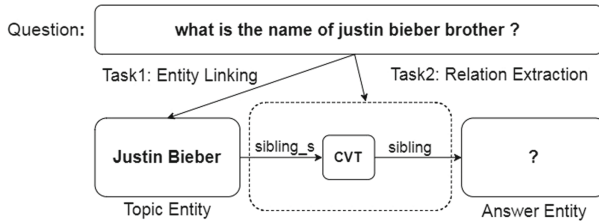


Fig. 1. Two KBQA tasks

Most previous work [6, 9] treats the relation extraction task of KBQA as a ranking problem. In contrast, we attempt to solve this task in a more straightforward fashion by regarding it as a multi-class classification task. The challenge then comes from the model optimization with a relatively large search space. Fortunately, some candidate relations can be filtered out in advance by traversing the whole KB to remove unlikely relations and reduce the search space. In this paper, we propose a masking mechanism to prevent our model from selecting these impossible relations. In this paper, we develop and discuss three models: (1) a convolutional neural network-based model (CNN), (2) a convolutional neural network-based model with a masking layer (CNN + masking) and (3) a hybrid convolutional neural network/recurrent neural network model (CNN + RNN). We evaluate the performance of these models on the WebQuestions Semantic Parses Dataset (WebQSP) [10].

In addition to providing effective models, we hope to offer a strong baseline for companies and government agencies who seek to build their own knowledge-based question-answering systems efficiently. Therefore, we describe the construction of the KB for the catering industry for which we apply the proposed models to demonstrate adaptation from the general domain to a specific domain.

The major contributions of this paper are: (1) We propose an effective relation extraction model to extract relations in both general KBs and domain-specific KBs. (2) We demonstrate strong performance on the catering KB and describe details when adapting this general model to a specific domain.

## 2 Related Work

Determining the relation between two entities is critical for natural language understanding. With the text evidence and the given two mentioned entities, the aim of text-based relation extraction is to predict the relation indicated by the text evidence. Previous methods include labor-intensive feature engineering with SVM [11] and clustering words before relation classification [12] for more sophisticated management of semantics. With the advance of deep learning techniques, relation extraction models have evolved from machine learning models based on word embeddings [13,14] to deep neural networks such as CNNs or LSTMs [15–17] and even more complicated models [18,19]. One of the assumptions in the text-based relation extraction task is that a fixed set of candidate relations is given, with a relatively small size. However, in the KBQA relation extraction task, thousands of relations are included in the KB and all are to be considered for each question. As a result, approaches for text-based relation extraction cannot be directly applied to KBQA relation extraction. Instead, KBQA relation extraction usually takes the question and a candidate relation as the input for its decision process in a pairwise fashion. That is, each time, a question and a candidate relation are given to compute a score, which eliminates the need to consider all relations at the same time.

Traditional relation extraction in KBQA started with the naive Bayes method considering rich linguistic features [7] and learning-to-rank mechanism with hierarchical relations [20]. These were followed by neural network models [6,21,22] with the attention mechanism [18,23] and the residual network [9] in recent years. Though these models are able to handle all relations in a KB sequentially, they must repeat the same process for each candidate relation many times for the final decision. Moreover, when processing one relation, other relations are neglected. This is clearly a drawback for a selection process. To address this problem, we propose a model that combines the advantages of both text-based and KBQA relation extraction, and that considers the large KB relation set using only one forward pass for each question.

## 3 Method

We first implemented a CNN-based multi-class classifier, the output layer of which is the probability of all the relations in the training data. After we analyzing the errors in this first model, we observed that the major errors were domain errors and semantic meaning errors.

Each relation in Freebase is composed of the following three fragments (tokens): Domain.Type.Property. A domain error indicates the wrong domain field in the predicted relation. For example, the predicted relation is “film.performance.actor” for the question “Who plays ken barlow in coronation street?” when the correct relation is “tv.regular\_tv\_appearance.actor”, where the domain field “tv” is mis-identified as “film”.

From our observations, domain errors arise from considering all of the relations in KB, a relative large relation set, regardless of the topic entity. In the

above example, “coronation street” is obviously a TV show and will not connect to a relation whose domain is “film”. Therefore, we attempt to reduce domain errors by ignoring those highly unlikely relations in advance: we propose a masking mechanism to filter out relations which are not connected to the topic entity within two hops, i.e., relations in the relation path whose length is not greater than two.

Semantic errors, in turn, indicate semantic mismatches between the question and the relation. For example, for the question “Where did they find Jenni Rivera’s body?” our model predicts “people.place\_lived.location” while the correct answer is “people.deceased\_person.place\_of\_death”. In this example, the model did not learn that “find the body” is related more to “death” than to “live”. This may be due to the design in which the proposed model considers each relation independently as one class but ignores the semantic meaning of its name. In order to solve this problem, we propose the third model, the hybrid CNN/RNN model (CNN + RNN).

For this hybrid model, we collect the relation paths connected to the topic entity whose lengths are equal or less than two (within two hops) and view these relation paths as the candidate answers. Then we perform a binary classification on all the candidate paths.

### 3.1 CNN

The first proposed model is a CNN-based multi-class classifier. To fully utilize the deep neural network and to achieve better language understanding, we use several different features as the input of this model. In addition, we use two channels to offer features from the raw question and its dependency-parsed question. The architecture of the proposed CNN model is illustrated in Fig. 2.

With the first channel we consider the following types of features:

- **Lexical:**

We use public pretrained GloVe word embeddings to turn words into fixed-length vectors. Given the  $D \times d$  word embedding matrix  $W$ , the  $i$ -th row indicates the embedding of the  $i$ -th word, yielding a  $d$ -dimensional vector  $x_{glove}$ .

- **WordNet:**

To gain more information on the semantic type, we use WordNet to generate another set of word embeddings as features. Each word, together with its hypernyms, forms a sequence in their hierarchical order: this is termed the hypernym path. Ten hypernym paths are generated by a random walk for each word in WordNet. All the generated paths are then treated as sentences for GloVe to train the embedding for each word in WordNet, yielding a  $d$ -dimensional vector  $x_{wordnet}$ .

- **POS:**

We randomly initialize an embedding matrix for the POS tag vocabulary. The weights in the matrix are updated during the training process. For each POS the embedding matrix yields a  $d$ -dimensional vector  $x_{pos}$ .

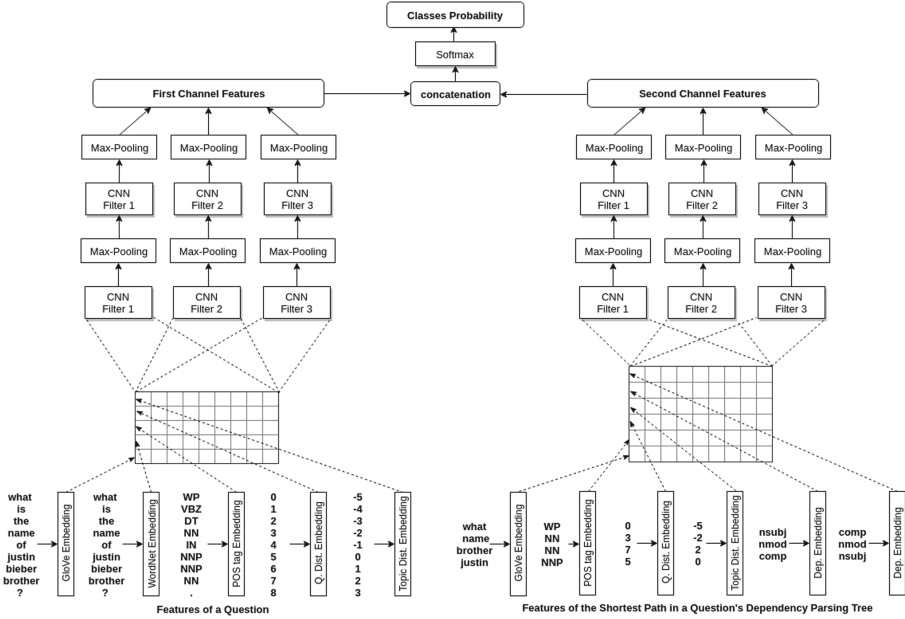


Fig. 2. Architecture of proposed CNN model

– **Distance:**

For each word, we compute two distances to the current question: its distance to the question word (e.g., Who, What, How) and its distance to the topic entity. For example, in the question “What is the name of Justin Bieber’s brother?”, the question entity distance, i.e., the distance from the word “name” to “What”, is 3 and the topic entity distance, i.e., that from “name” to “Justin Bieber”, is 2. After the two distances are computed, the randomly initialized embedding matrix turns them into the  $d$ -dimensional vectors  $x_{ques}$  and  $x_{topic}$ , respectively. The weights of the embedding matrix are also updated in the training process.

Once these features are extracted, we concatenate and feed them into the first channel of the proposed CNN model. These concatenations are shown as follows:

$$\begin{aligned}
 X &= X_{glove} \oplus X_{wordnet} \oplus X_{pos} \oplus X_{ques} \oplus X_{topic} \\
 X_{glove} &= x_{glove}^0 \oplus x_{glove}^1 \oplus x_{glove}^2 \oplus \dots \oplus x_{glove}^{n-1} \oplus x_{glove}^n \\
 X_{wordnet} &= x_{wordnet}^0 \oplus x_{wordnet}^1 \oplus \dots \oplus x_{wordnet}^{n-1} \oplus x_{wordnet}^n \\
 X_{pos} &= x_{pos}^0 \oplus x_{pos}^1 \oplus x_{pos}^2 \oplus \dots \oplus x_{pos}^{n-1} \oplus x_{pos}^n \\
 X_{ques} &= x_{ques}^0 \oplus x_{ques}^1 \oplus x_{ques}^2 \oplus \dots \oplus x_{ques}^{n-1} \oplus x_{ques}^n \\
 X_{topic} &= x_{topic}^0 \oplus x_{topic}^1 \oplus x_{topic}^2 \oplus \dots \oplus x_{topic}^{n-1} \oplus x_{topic}^n
 \end{aligned}$$

where  $\oplus$  is the concatenation operator,  $X$  is the input of channel one,  $x_k^i$  denotes the  $i$ -th vector of feature  $k$ , and  $n$  is the number of words in the question.

For the second channel, we use the Stanford CoreNLP [24] dependency parser to generate the question’s dependency parse tree. From the parse tree, we extract the shortest path from the topic entity to the question word, and then use words on the shortest path to generate the following types of features:

– **Lexical:**

The pretrained word embedding matrix for channel one is used here as well. For each word in the shortest path, from the matrix we extract a  $d$ -dimensional vector  $s_{glove}$ .

– **POS:**

For POS tags, we use the embedding matrix from channel one to transform the POS vocabulary. Each POS tag in the shortest path yields a  $d$ -dimensional vector  $s_{pos}$ .

– **Distance:**

As in channel one, we extract the question entity distance and the topic entity distance. The distance between the question entity and the topic entity in the original question is computed for each separate word in the shortest path. Again, the distance embedding matrix from channel one is used to turn the distance into  $d$ -dimensional vectors  $s_{ques}$  and  $s_{topic}$ .

– **Dependency tag:**

Words in the dependency parsed tree are connected by dependency tags, which indicate their mutual relationship. We randomly initialize a dependency tag embedding matrix for later training, and turn each dependency tag appearing in the shortest path into a  $d$ -dimensional vector  $s_{dep}$ .

– **Reversed dependency tag:**

In the end, we reverse the dependency tag feature above, indicating a traversal of the dependency parse tree from the topic entity to the question entity. The same dependency embedding matrix is used to generate  $d$ -dimensional vectors  $s_{rdep}$ .

Given these features, we concatenate them and feed them into channel two of our models.

$$\begin{aligned}
 S &= S_{glove} \oplus S_{pos} \oplus S_{ques} \oplus S_{topic} \oplus S_{dep} \oplus S_{rdep} \\
 S_{glove} &= s_{glove}^0 \oplus s_{glove}^1 \oplus s_{glove}^2 \oplus \cdots \oplus s_{glove}^{m-1} \oplus s_{glove}^m \\
 S_{pos} &= s_{pos}^0 \oplus s_{pos}^1 \oplus s_{pos}^2 \oplus \cdots \oplus s_{pos}^{m-1} \oplus s_{pos}^m \\
 S_{ques} &= s_{ques}^0 \oplus s_{ques}^1 \oplus s_{ques}^2 \oplus \cdots \oplus s_{ques}^{m-1} \oplus s_{ques}^m \\
 S_{topic} &= s_{topic}^0 \oplus s_{topic}^1 \oplus s_{topic}^2 \oplus \cdots \oplus s_{topic}^{m-1} \oplus s_{topic}^m \\
 S_{dep} &= s_{dep}^0 \oplus s_{dep}^1 \oplus s_{dep}^2 \oplus \cdots \oplus s_{dep}^{m-1} \\
 S_{rdep} &= s_{rdep}^0 \oplus s_{rdep}^1 \oplus s_{rdep}^2 \oplus \cdots \oplus s_{rdep}^{m-1}
 \end{aligned}$$

where  $S$  is the input of channel two of our model,  $s_k^i$  denotes the  $i$ -th vector of feature  $k$ , and  $m$  is the number of words in the dependency shortest path.

After providing our model with the feature vectors, for each channel, we use three filters of size 1, 2, and 3 to capture features using different window sizes. Assume a sequence of feature vectors fed into channel  $k$  is represented as

$$V^k = v_1^k \oplus v_2^k \oplus \dots \oplus v_{n-1}^k \oplus v_n^k,$$

and  $v_{i:i+j}^k$  refers to the concatenation of  $v_i^k, v_{i+1}^k, \dots, v_{i+j}^k$ . The filter  $w \in R^{hd}$  in our model generates new  $d$ -dimensional features from each window of size  $h$ , where  $h$  is either 1, 2, or 3. For instance, feature  $c_i$  is generated from a window of words  $v_{i:i+h-1}$ :

$$c_i = f(\mathbf{w} \cdot v_{i:i+h-1}), \quad (1)$$

where  $f$  is a linear activation function. This filter is applied to each possible window in the sentence  $\{v_{1:h}, v_{2:h+1}, \dots, v_{n-h+1:n}\}$  to produce a feature map  $\mathbf{c}$ :

$$\mathbf{c} = [c_1, c_2, \dots, c_{n-h+1}]. \quad (2)$$

Then two max pooling layers are applied on these feature maps to obtain the max value of each feature map. The first max pooling layer is of length 3, and the second is designed to produce the single-length feature map. After this convolution process, 6 feature maps are generated from 2 channels and 3 filters. We then concatenate these feature maps and pass them through a dense layer and then a softmax layer to calculate the class probabilities. The model is optimized by minimizing the categorical cross-entropy loss.

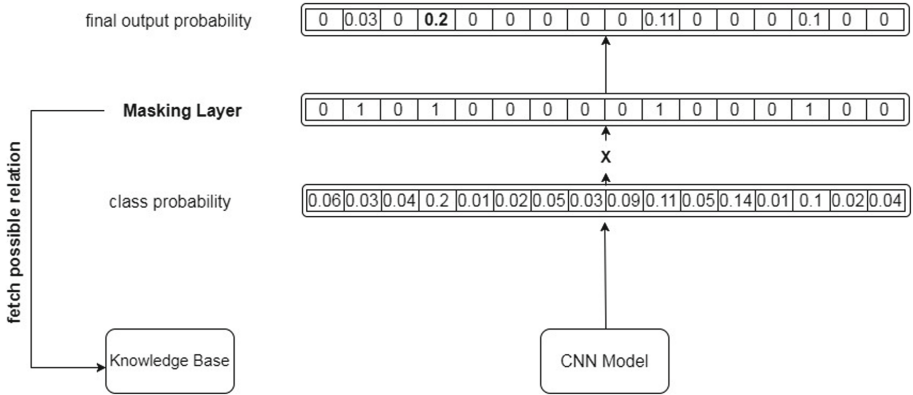
### 3.2 CNN + Masking

Further observations indicate that some candidate relations can be filtered out in advance: in the WebQSP dataset, there are 5,210 different relations in the training data, while on average there are only 141.6 relations connected to a topic entity. This substantial difference indicates that searches performed on unconnected relations of the topic entity by the proposed CNN model are inefficient.

To take into account only connected relations, we propose a masking mechanism. The masking layer is added before the output layer in the proposed CNN model to drop disconnected relations from the final prediction. Given a question and its topic entity, we retrieve these possible relations from the KB. In this work, we enumerate the possible relations  $R'$  by traversing the KB from the topic entity and recording relations within two hops. With the total relation set  $R$ , we define the masking layer  $M$  as

$$M = \{1 \text{ if } r \in R' \text{ else } 0, \forall r \in R\}. \quad (3)$$

As illustrated in Fig. 3, the output vector of the probabilities of relation classes is then multiplied element-wise with the masking layer to yield only the probabilities of the connected relations.



**Fig. 3.** Proposed CNN + masking model

### 3.3 CNN + RNN

Since the relations are organized hierarchically from domain, type, to property, we propose a third model, which not only encodes information from a question, but also considers information from candidate relations using a recurrent neural network (RNN). In this model, we treat the relation extraction task as a ranking problem. As illustrated in Fig. 4, we use the model to encode the question. For candidate relations, we first segment them into tokens according to their hierarchy, and then apply a gate recurrent unit (GRU) [25] to encode these tokens sequentially.

Consider  $t_i$  as the  $i$ -th relation token: we start by passing it through a randomly initialized embedding matrix to generate its embedding representation  $y_i$ . Then the whole  $y$  sequence is processed using the GRU layer, which is formulated as

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t \quad (4)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (5)$$

$$\tilde{h}_t = \tanh(W x_t + U(r_t \odot h_{t-1})) \quad (6)$$

$$r_t = \sigma(W_r y_t + U_r h_{t-1}), \quad (7)$$

where  $W$  and  $U$  are trainable parameters,  $\sigma$  is an arbitrary non-linear activation function, and  $x_t$  denotes the  $t$ -th relation token.

The hidden vector of the question and the candidate relation are then concatenated and passed onto a *relu* [26] activated feed-forward neural network (FFNN), which outputs a single-digit score indicating the fitness of the question and candidate relation. As with previous work that treats relation extraction



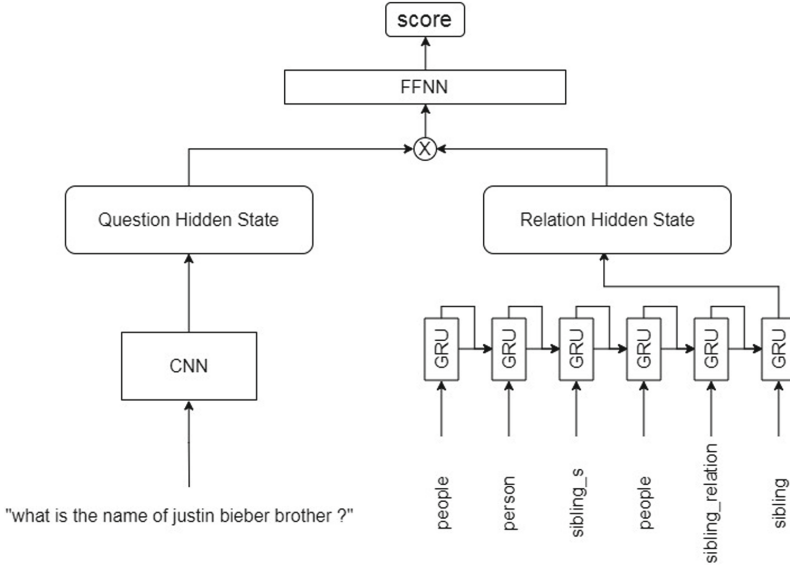


Fig. 4. Proposed CNN + RNN model

as a ranking test, hinge loss is applied to optimize the model, which can be formulated as

$$loss = \max(0, -(score^+ - score^-) + margin),$$

where  $margin$  is an arbitrary value and  $score^+$  and  $score^-$  stand for the output scores of the correct and incorrect relations, respectively.

As this model considers relations semantically, it has the potential to relate the relation to the question more correctly. Moreover, with hinge loss, we can train the model not only with positive samples but also with negative ones.

## 4 Experiment

### 4.1 The WebQSP Dataset

For experiments, we use WebQSP [10], a public QA dataset which is an annotated version of WebQuestion [27], another public QA dataset which contains question entity and answer entity pairs from the Freebase KB. Questions in WebQuestion were generated based on suggestions from the Google Search Suggestions API and were labeled with semantic parses by experts who are familiar with Freebase. In the WebQSP dataset, in addition to the question and its linked entity with the corresponding MID in Freebase, there is the annotated inferential chain which we refer as the relations (relation path) in this paper. WebQSP specifies 3,098 questions for training and 1,639 questions for testing. From these 3,098 training questions we further set aside 305 for validation.

## 4.2 Results and Discussion

We evaluated the three proposed models in terms of accuracy. As shown in Table 1, compared to previous works, the proposed models show comparable performance with a smaller search space. The state-of-the-art HR-BiLSTM [9] achieves 83% accuracy by incorporating both relation tokens and relation words.

**Table 1.** Accuracy of proposed models

Method	Accuracy
BiCNN [6]	78%
BiLSTM + words [9]	80%
BiLSTM + relation_names [9]	79%
HR-BiLSTM [9]	<b>83%</b>
CNN	72%
CNN + masking	77%
CNN + RNN	71%

Comparing the three proposed models, it is surprising that the CNN + masking model outperforms the others. The architecture of the CNN + RNN model is similar to the other related work. Moreover, the CNN + RNN model, the BiCNN model, and the BiLSTM + relation\_names model all use the same features. Therefore, we expected it to yield comparable performance. Instead, this model yielded the worst performance in our experiment; the BiCNN model and the BiLSTM + relation\_names model both far outperformed this model. This suggests that the recurrent model cannot extract information on relations effectively in the CNN + RNN model.

However, we found that masking benefits the proposed model. To the best of our knowledge, this is the first attempt to treat the relation extraction task as a classification problem. The main challenge in making it a classification problem is the large search space for selecting possible relations. Results show that simply adding a masking layer can solve this problem efficiently.

## 4.3 Error Analysis

We further analyzed the errors found in the results from the CNN + masking model. The first major type of error observed was ambiguous relations. For example, consider the question “What are the major languages spoken in Greece?”: whereas the correct relation for this question is “location.country.official\_language”, our model predicts the “location.country.languages\_spoken” relation. About 16% of the errors are of this type.

The second major error type is due to the model’s understanding the question not as a whole but in pieces. For example, the question “What state is Mount St. Helens in?” is classified as “geography.mountain.mountain\_type” relation, instead of the gold relation “location.location.contained\_by”. The model

understands the concept “mount”, but wrongly recognizes that the question is asking the type of mountain instead of its location. Another example is the question “What town did Justin Bieber grow up in?” being classified to the relation “people.person.place\_of\_birth”, while the correct relation should be “people.person.places\_lived people.place\_lived.location”. To avoid this kind of error, a more sophisticated language encoding model may help such as the deeper CNN model [28,29], the attention mechanism [30,31], or the residual network [32].

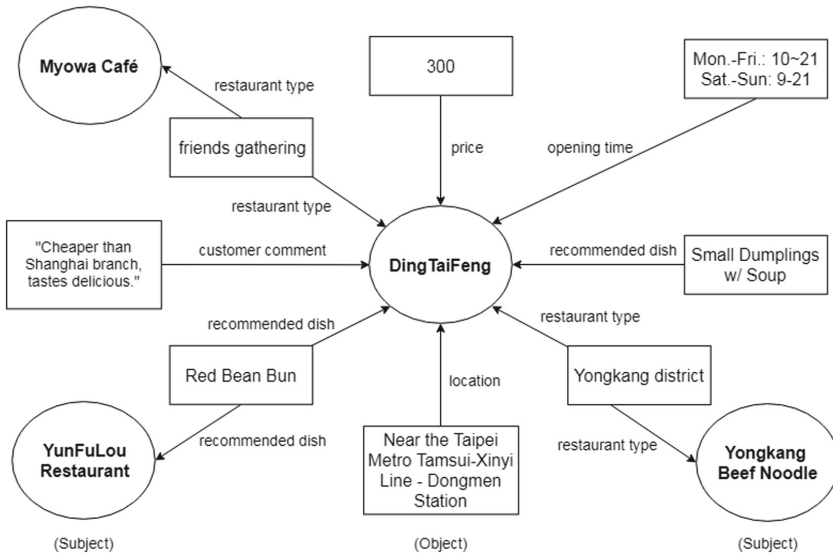
## 5 KBQA on a Catering Knowledge Base

### 5.1 Constructing CaterKB from iPeen

CaterKB is a Freebase-style knowledge base generated from iPeen<sup>1</sup>, the biggest Taiwanese restaurant ranking website. Each restaurant has its own webpage

**Table 2.** Relations of CaterKB

Chinese	English	Chinese	English
餐廳類型	Restaurant Type	消費方式	Payment
推薦菜色	Recommended Dish	聯絡方式	Contact
客人短評	Customer Comment	類似餐廳	Similar Restaurant
營業時間	Opening Time	媒體推薦	Media Recommendation
地理位置	Location	席位	Seat
消費價位	Price		



**Fig. 5.** CaterKB sample

<sup>1</sup> [www.ipeen.com.tw](http://www.ipeen.com.tw).

in iPeen from which most of the information can be collected. Each relation representation is a RDF triple in the form of <Subject> <Relation> <Object>. A total of 11 relations are defined in CaterKB, as shown in Table 2. We collected a total of 2,371,397 triples from 147,868 restaurants. Figure 5 shows a sample partial CaterKB knowledge graph.

## 5.2 Generating Questions for Catering

As people usually search for information they need on the Internet using keywords rather than complete questions, especially with catering information, we failed to collect enough questions with the Google Search Suggestion API. Therefore, we recruited 15 native Chinese speakers to generate 200 questions about restaurants and foods. As it is challenging to generate high-quality questions for all types of relations, six types of restaurant relations were selected for experiments: restaurant type, recommended dish, customer comment, opening time, price, and location.

## 5.3 Experimental Settings

We applied the CNN+masking layer model to the generated 200 questions. As for data preparing, we performed ten-fold cross validation to make sure low variance between each setting. The result reported is the average of the testing accuracy of the ten folds. The learning rate was set to 0.00005, the batch size 4, and the hidden layer size 128. We adopted the Stanford CoreNLP parser [24] for word segmentation, POS tagging and dependency parsing. To utilize the parser, we use OpenCC toolkit<sup>2</sup> to translate sentences from traditional Chinese to simplified Chinese. After parsing, we translate the parsed result from simplified Chinese to traditional Chinese, and we use Skip-Gram [33] to train on the Chinese Gigaword Second Edition (CG2) [34] as the pre-trained word embedding.

## 5.4 Results and Discussion

Results are shown in Table 3. The accuracy is unchanged in three different dropout rates and only achieves 75%. As we investigate the results and errors, we find that the relation “餐廳類型 (restaurant type)” is not easily identified by the model; questions of this type bring in much noise. These questions include various terms such as “brunch”, “Japanese”, “secret place”, “historic”, and “Taipei”, which are too diverse to learn. Hence in terms of KB construction principles, “restaurant type” may not be a good relation.

To evaluate the impact of the error-prone “restaurant type” relation, we excluded questions of this type, keeping 167 questions with other five types of relation, and conducted the experiment again with 147 questions for training, 10 for validation, and 10 for testing. The performance improved considerably: the best performance was now 89%, from which we conclude that (1)

<sup>2</sup> <https://github.com/BYVoid/OpenCC>.

**Table 3.** CNN + masking accuracy on CaterKB questions

200 questions		167 questions	
Dropout	Accuracy	Dropout	Accuracy
0.0	75%	0.0	88%
0.2	75%	0.2	88%
0.4	75%	0.4	<b>89%</b>

carefully designing the relation types in a domain-specific knowledge base may considerably improve performance (2) the 89% accuracy indicates that the proposed CNN + masking model can be used in real-world domain-specific KBQA applications.

Overall, the best performance – an accuracy of 89% – is observed when the dropout rate is set to a relatively high value of 0.4.

### 5.5 Error Analysis

From the error results of the CNN + masking model with a 0.4 dropout rate, we noticed that the questions for the “location” relation which contain the Chinese word “開 (open/locate/drive)” tended to be classified as questions for the “opening time” relation, such as “鼎泰豐開在哪？(Where is DingTaiFeng located?)”. This error is due to polysemy in the questions, in this case, 開 (open/locate/drive). As we did not perform word sense disambiguation before relation extraction, the most commonly used sense “開 (open)” is always adopted by the model. Other questions using this word include “鼎泰豐每天都有開嗎？(Is DingTaiFeng open everyday?)” and “鼎泰豐早上9點開了嗎？(Is DingTaiFeng open at nine o’clock?)”, where 18 of 19 appearances of the word are of the sense “open”. This shows that word sense disambiguation is relatively important for domain-specific KBQA.

Questions for “recommended dish” relation such as “鼎泰豐要怎麼點菜最好？(How to make the best order in DingTaiFeng?)” are also challenging for the proposed model. These questions involve the question word “怎麼 (how)”, which is also a challenging question type in the conventional question answering task. The results showed that the model confused these questions with those for the “customer comment” relation, as the latter are common connected with the question word “怎麼 (how)”. Usually when one word is found in questions for different relations, its context can aid disambiguation. However, as how-type questions are expressed using a wide variety of words, the context of the question word “how” may not help in relation extraction.

Finally, we note that the vocabulary of the specific domain is small; hence some words are common in questions. For example, in catering questions words such as “菜 (dish)”, “價錢 (price)”, “營業 (open)”, “評價 (comment)”, and “哪裡 (where)” are often seen, and are strong features for relations “recommended

dish”, “price”, “opening time”, “customer comment”, and “location” respectively. We believe this is a worthy direction for improving domain-specific relation extraction.

## 6 Conclusion and Future Work

Relation extraction plays an important role in KBQA. However, the greatest challenge comes from the large search space of relations. In this paper, we propose three models to extract relations for KBQA, as well as a masking mechanism to reduce the search space. Results show that it is comparable to the state of the art in the general domain and yields superior performance for domain-specific KBs.

In the future, we will investigate automatic question collection for relations in domain-specific KB and useful features for domain adaptation. We believe the proposed model can serve as a simple but strong tool for real-world applications.

**Acknowledgements.** This study was conducted under the “Big Data Technologies and Applications Project (3/4)” of the Institute for Information Industry which is subsidized by the Ministry of Economic Affairs of the Republic of China.

## References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC - 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-76298-0\\_52](https://doi.org/10.1007/978-3-540-76298-0_52)
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: 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. ACM (2008)
3. Miller, G.A.: WordNet: a lexical database for English. *Commun. ACM* **38**(11), 39–41 (1995)
4. Liang, J., Zhang, Y., Xiao, Y., Wang, H., Wang, W., Zhu, P.: On the transitivity of hypernym-hyponym relations in data-driven lexical taxonomies. In: AAAI, pp. 1185–1191 (2017)
5. Agrawal, R., Shekhar, M., Misra, D.: Integrating knowledge encoded by linguistic phenomena of indian languages with neural machine translation. In: Ghosh, A., Pal, R., Prasath, R. (eds.) MIKE 2017. LNCS (LNAI), vol. 10682, pp. 287–296. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-71928-3\\_28](https://doi.org/10.1007/978-3-319-71928-3_28)
6. Yih, S.W., Chang, M.W., He, X., Gao, J.: Semantic parsing via staged query graph generation: question answering with knowledge base (2015)
7. Yao, X., Van Durme, B.: Information extraction over structured data: question answering with freebase. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 956–966 (2014)

8. Dong, L., Wei, F., Zhou, M., Xu, K.: Question answering over freebase with multi-column convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 260–269 (2015)
9. Yu, M., Yin, W., Hasan, K.S., dos Santos, C., Xiang, B., Zhou, B.: Improved neural relation detection for knowledge base question answering. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 571–581 (2017)
10. Yih, W., Richardson, M., Meek, C., Chang, M.W., Suh, J.: The value of semantic parse labeling for knowledge base question answering. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, pp. 201–206 (2016)
11. GuoDong, Z., Jian, S., Jie, Z., Min, Z.: Exploring various knowledge in relation extraction. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Stroudsburg, PA, USA, ACL 2005, pp. 427–434. Association for Computational Linguistics (2005)
12. Sun, A., Grishman, R., Sekine, S.: Semi-supervised relation extraction with large-scale word clustering. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 521–529. Association for Computational Linguistics (2011)
13. Nguyen, T.H., Grishman, R.: Employing word representations and regularization for domain adaptation of relation extraction. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, pp. 68–74 (2014)
14. Gormley, M.R., Yu, M., Dredze, M.: Improved relation extraction with feature-rich compositional embedding models. arXiv preprint [arXiv:1505.02419](https://arxiv.org/abs/1505.02419) (2015)
15. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335–2344 (2014)
16. dos Santos, C.N., Xiang, B., Zhou, B.: Classifying relations by ranking with convolutional neural networks. arXiv preprint [arXiv:1504.06580](https://arxiv.org/abs/1504.06580) (2015)
17. Vu, N.T., Adel, H., Gupta, P., Schütze, H.: Combining recurrent and convolutional neural networks for relation classification. arXiv preprint [arXiv:1605.07333](https://arxiv.org/abs/1605.07333) (2016)
18. Wang, L., Cao, Z., de Melo, G., Liu, Z.: Relation classification via multi-level attention CNNs. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 1298–1307 (2016)
19. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, pp. 207–212 (2016)
20. Bast, H., Haussmann, E.: More accurate question answering on freebase. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1431–1440. ACM (2015)
21. Xu, K., Reddy, S., Feng, Y., Huang, S., Zhao, D.: Question answering on freebase via relation extraction and textual evidence. arXiv preprint [arXiv:1603.00957](https://arxiv.org/abs/1603.00957) (2016)
22. Dai, Z., Li, L., Xu, W.: CFO: conditional focused neural question answering with large-scale knowledge bases. arXiv preprint [arXiv:1606.01994](https://arxiv.org/abs/1606.01994) (2016)

23. Golub, D., He, X.: Character-level question answering with attention. arXiv preprint [arXiv:1604.00727](https://arxiv.org/abs/1604.00727) (2016)
24. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations, pp. 55–60 (2014)
25. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder-decoder approaches. arXiv preprint [arXiv:1409.1259](https://arxiv.org/abs/1409.1259) (2014)
26. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-2010), pp. 807–814 (2010)
27. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1533–1544 (2013)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
29. Conneau, A., Schwenk, H., Barrault, L., Lecun, Y.: Very deep convolutional networks for text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, vol. 1, pp. 1107–1116 (2017)
30. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473) (2014)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 6000–6010 (2017)
32. Liao, Q., Poggio, T.: Bridging the gaps between residual learning, recurrent neural networks and visual cortex. arXiv preprint [arXiv:1604.03640](https://arxiv.org/abs/1604.03640) (2016)
33. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
34. Graff, D., Chen, K., Kong, J., Maeda, K.: Chinese Gigaword Second Edition (LDC2005t14). Linguistic Data Consortium, Philadelphia (2005)