PINGAN_AI at SemEval-2022 Task 9: Recipe knowledge enhanced model applied in Competence-based Multimodal Question Answering

Zhihao Ruan, Xiaolong Hou, Lianxin Jiang
Ping An Life Insurance Company of China
{ruanzhihao322, houxiaolong430, jianglianxin769}@pingan.com.cn

Abstract

This paper describes our system used in the SemEval-2022 Task 09: R2VQ - Competence-based Multimodal Question Answering. We propose a knowledge-enhanced model for predicting answer in QA task, this model use BERT as the backbone. We adopted two knowledge-enhanced methods in this model: the knowledge auxiliary text method and the knowledge embedding method. We also design an answer extraction task pipeline, which contains an extraction-based model, an automatic keyword labeling module, and an answer generation module. Our system ranked 3rd in task 9 and achieved an exact match score of 78.21 and a word-level F1 score of 82.62.

1 Introduction

In this paper, we discuss an approach to the Question Answering (QA) task for SemEval-2022 Task9(Tu et al., 2022). This task is structured as question answering pairs, querying how well a system understands the semantics of recipes derived from a collection of English cooking recipes and videos, which involve rich semantic annotation and aligned text-video objects.

In this task, a large proportion of answers can't be derived directly from the original recipe text and the information of these answers is hidden in annotated knowledge data. So we adopted two knowledge-enhanced methods in this model: the knowledge auxiliary text method and the knowledge embedding method. The knowledge auxiliary text method incorporates the hidden-roles knowledge and co-reference knowledge in generating auxiliary text. The knowledge embedding method encodes u-pos knowledge and entity knowledge into knowledge embedding.

A key evaluation measure in this task is the exact match score. Because the extraction-based model is more robust than the generative-based model, we design an answer extraction task pipeline. The pipeline contains an extraction-based model, an automatic keyword labeling module, an answer restructures module. In the training phase, we locate keywords of answers in recipe text and provide training data by labeling the keywords. In the prediction phase, we collect output keywords of the model and generate answer text by the keywords.

Our system ranked 3rd in task 9 and achieved an exact match score of 78.21 and a word-level F1 score of 82.62. We make our code publicly available on Github[1].

2 Related Work

Question Answering (QA) Liu et al. (2019) described the Span Extraction task in MRC (Machine Reading Comprehension) and Mervin (2013) described the extraction-based question answering task. BERT(Devlin et al., 2019) displayed a general extraction-based approach for the QA task.

Knowledge enhanced CoLAKE(Sun et al., 2020), ERNIE 3.0(Sun et al., 2021) and K-BERT(Liu et al., 2020) shown the method of constructing a new context by using knowledge information. Know-BERT (Peters et al., 2019) used the knowledge embedding method and Zhang et al. make some improvements.

Prompt In generating knowledge auxiliary text, we are inspired by prompt(Liu et al., 2021) learning. We focus on prompt engineering in this paper. Yuan et al. (2021) rewrite the context by replacing phrase. Davison et al. (2019) use a unidirectional LM to score the prompt patterns. Gao et al. (2021) use the T5 model to generate a template.

3 System overview

In this paper, we discuss an approach to the Question Answering (QA) task for SemEval-2022 Task9.

[1] https://github.com/archfool/SemEval2022_Task09

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1274 - 1279
July 14-15, 2022 ©2022 Association for Computational Linguistics

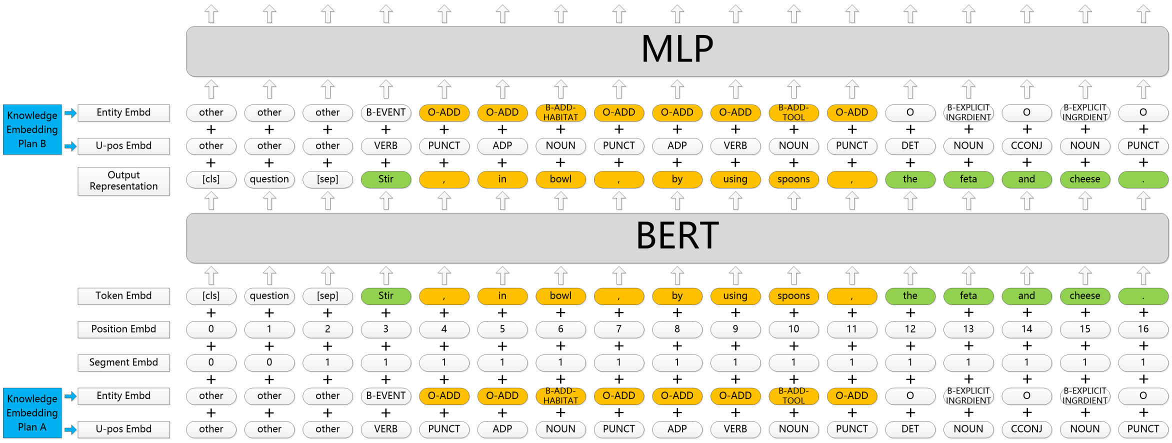
MLP
Knowledge Embedding Plan B
Entity Embd
other other other B-EVENT O-ADD O-ADD B-ADD-HABITAT O-ADD O-ADD O-ADD B-ADD-TOOL O-ADD O B-EXPLICIT INGRDIENT O B-EXPLICIT INGRDIENT O
U-pos Embd
other other other VERB PUNCT ADP NOUN PUNCT ADP VERB NOUN PUNCT DET NOUN CCONJ NOUN PUNCT
Output Representation
[cls] question [sep] Stir , in bowl , by using spoons , the feta and cheese .
BERT
Token Embd
[cls] question [sep] Stir , in bowl , by using spoons , the feta and cheese .
Position Embd
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Segment Embd
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Knowledge Embedding Plan A
Entity Embd
other other other B-EVENT O-ADD O-ADD B-ADD-HABITAT O-ADD O-ADD O-ADD B-ADD-TOOL O-ADD O B-EXPLICIT INGRDIENT O B-EXPLICIT INGRDIENT O
U-pos Embd
other other other VERB PUNCT ADP NOUN PUNCT ADP VERB NOUN PUNCT DET NOUN CCONJ NOUN PUNCT
+

Figure 1: knowledge enhanced model

We divided the samples into 18 categories after analyzing the patterns of QA pairs. We predict answers of 6 categories by rule-based method and 12 categories by deep-learning model.

We chose BERT(Devlin et al., 2019) as the backbone of our model and chose the answer extraction method for this QA task.

Knowledge information is significant in this task, so we adopted two knowledge-enhanced methods: the knowledge auxiliary text method and the knowledge embedding method.

Knowledge auxiliary text method generating expanded text that contains original text and knowledge auxiliary text. The knowledge auxiliary text incorporates the hidden roles and co-reference knowledge.

Knowledge embedding method encoding u-pos (universal-part-of-speech) knowledge and entity knowledge into knowledge embedding.

3.1 QA datasets analysis

Following the design ideas of the FAQ task, we analyzed the patterns of QA pairs first. And we summarize QA pairs into 18 pattern categories.

With the help of divided pattern categories, we fed certain categories of data into a rule-based module. For example, some questions are about counting tasks, the rule-based method is better for these tasks than the deep-learning model method. We fed the rest QA categories into the deep-learning model module. The proportion of samples using the rule-based method was 39.87%, and the proportion of samples using the deep-learning model method was 60.13%.

The QA-pair pattern categories' example is shown in Table 2 of Appendix A.

Some more dataset information is shown in Section 4.1.

3.2 Knowledge auxiliary text

By analyzing the QA cases of the datasets, we found that a large proportion of answers can't be derived directly from the original recipe text. And the information of the answers is in the Cooking Role Labeling (CRL) annotations. So we generate knowledge auxiliary text by introducing the knowledge of the co-reference column and hidden-role column in CRL annotations. We add knowledge auxiliary text to the original text as the input of the model.

Co-reference knowledge can be regarded as a kind of entity link between the current entity token and the token where this entity is mentioned the first time. In this way, readers can search and locate the item unambiguously even though different items have the same name or one item uses different names. Knowledge auxiliary text of co-reference information is generated by easily adding the alias in the co-reference column within a couple of brackets.

For example, if the original text is "mixture" and the co-reference column has the value "small balls". Then we generate knowledge auxiliary text "(small balls)". Combined knowledge auxiliary text to the original text, and get the final text: "mixture (small balls)".

Hidden-role knowledge is a supplement to the action token that appears in the original text when some elements of the action are hidden. Inspired by prompt(Liu et al., 2021) learning, we generate knowledge auxiliary text following the pattern of the answer text from the QA-pairs. For example, we use ‘by using a knife’ as knowledge auxiliary text when the hidden role column has a key value of ‘TOOL: knife’.

The knowledge auxiliary text method can be shown in Figure 1. In the token embedding layer, the tokens of green grids are the original text of the recipe, and the tokens of orange grids are the knowledge auxiliary text. We combined the original text and the knowledge auxiliary text as the input of the model.

3.3 Knowledge embedding

Entity knowledge and u-pos knowledge is useful in predicting answers in QA task. For example, a QA pair is: "How do you cut the carrots? by using a knife". In this case, the token of a knife is labeled as B-TOOL in the entity column. It would help predict the answer if the model gets the token’s entity type information.

In our model, we adopt the knowledge embedding method to incorporate the knowledge of the u-pos column and entity column. Similar to the token embedding representation, the knowledge embedding method maps values in the u-pos column to embeddings. The size of the knowledge embedding is the same as token embedding. The entity column’s knowledge embedding method is the same.

As shown in Figure 1, we have two places options to apply knowledge embedding: the embedding layer (knowledge embedding plan A in Figure 1) or the header (knowledge embedding plan B in Figure 1).

In plan A, we add knowledge embedding to the embedding layer of the model. Compared with BERT’s embedding layer whose output embedding is the sum of token embedding, position embedding, and segment embedding, our model adds entity knowledge embedding and u-pos knowledge embedding to the original BERT’s embedding layer.

In plan B, we add knowledge embedding to the header of the model. We first sum up the output representation of BERT’s last layer, the entity knowledge embedding, and the u-pos knowledge embedding. Then fed it into a small network such as MLP(Rosenblatt, 1957).

3.4 Answer extraction method for QA task

Two common methods of predicting answers in QA tasks are the extraction-based method and the generative-based method. In our model, we chose the answer extraction method for the QA task. Here are the steps to label the data for extracting answers:

First, we filtered out keywords from the questions and answers by deleting words whose u-pos are article, preposition, pronoun, and so on.

Second, we located the keywords in the text. According to the statistics, 88.5% of the QA samples can be located successfully in one sentence, and 0.5% in two adjacent sentences. 11% samples’ keywords can’t be located in texts, these samples were regarded as low confidence samples and would not be used.

Third, we labeled the answers’ keywords in the recipe texts. Concatenate the question and the labeled recipe texts as the input for the model.

Forth, predicting answer’s keywords by model.

Fifth, generate answers based on the keywords.

3.5 Rule-based answer generation method

Some QA categories in Table 2 are more suitable for predicting using rule-based methods. For example, some categories are about counting tasks.

By analyzing the QA patterns, we assigned 6 categories to the rule-base module: cat 1, cat 6, cat 7, cat 8, cat 9, and cat 10. These samples make up 44.36% of the total samples.

A simple rule description of the 6 categories is shown in Appendix B.

4 Experimental setup

4.1 Dataset

The organizer of SemEval-2022 Task9 provided R2VQ (Recipe Reading and Video Question Answering) dataset.

R2VQ dataset is a collection of cooking recipes and videos. The R2VQ dataset provided to us has 1000 samples, and the organizer split the dataset into 3 parts: training dataset (800 samples), validation dataset (100 samples), and testing dataset (100 samples). The training dataset and validate dataset have corresponding answers to the questions while the test dataset hasn’t.

Data of each sample in the R2VQ dataset is from two sources: cooking recipe text and screenshots from the videos. Each context of the recipes is consist of ingredients and directions whose labels have two annotation layers: Cooking Role Labeling (CRL) and Semantic Role Labeling (SRL). In the model of this paper, we only use the directions of recipes as the model’s input.

position	entity	upos	f1
None	False	False	88.78
embed_layer	False	True	88.93
embed_layer	True	False	89.00
embed_layer	**True**	**True**	**89.23**
header	False	True	88.47
header	True	False	88.76
header	True	True	88.03
embed_layer	⋆	⋆	89.05
header	⋆	⋆	88.42
⋆	True	⋆	88.76
⋆	⋆	True	88.66

Table 1: Squad f1 for different knowledge-enhanced methods. Column Position means which place we add knowledge embedding to. Column Entity means whether we use entity knowledge or not. Column Upos means whether we use universal-pos knowledge or not. Column F1 means the word-level F1 score of the method.

4.2 Training Details

We implemented our model using the pre-trained language model BERT(Devlin et al., 2019) as the backbone and chose Adam-W(Loshchilov and Hutter, 2018) as the optimizer. In train phrase, the batch size is 8 and the optimizer's learning rate is 3e-5. We ran the experiment on an NVIDIA GeForce RTX 3090 GPU.

As like in the MRC task, the article is a long text while the question is a short text. We spilt the long recipe text into short texts. Concatenate the short recipe text and the question by the special token [SEP] as a tokens sequence. And added [CLS] token to the beginning of the tokens sequence. Every tokens sequence is an input of the model. We set the max sequence length of the tokens sequence as 512. We set the configuration of doc stride as 128 which indicates the token length of adjacent sentences' overlap.

5 Results

We ranked 3rd in the competition of SemEval-2022 task 9. Our model's word-level F1 score is 82.62 and the exact match score is 78.21.

In our system, parts of questions got answers by rule module while others by deep-learning model module, which is mentioned in Section 3. In this paper, we mainly focus on the deep-learning model module and word-level F1 score. The oblation experiment is about knowledge enhanced method.

The baseline of our original model is 61 of squad word-level F1 score, without knowledge auxiliary text method and knowledge embedding method. Its corresponding model is BERT and its input is the original text of the recipes without any annotation of SRL and CRL.

Then we adopt the methods of knowledge enhancement including knowledge auxiliary text and knowledge embedding. We got a squad word-level F1 score of 88.78 when we used the knowledge auxiliary text method and an F1 score of 89.23 when using both of the two knowledge enhanced methods. The ablation experiment results are shown in Table 1.

The record of knowledge auxiliary text method is shown in the first line (so as the first section) of Table 1, and the squad word-level f1 score is 88.78. In this method, we used a text which contains both original text and knowledge auxiliary text as the model's input data.

The second part of Table 1 is the records of the complete knowledge enhanced method containing both knowledge auxiliary text and knowledge embedding. We have done 6 groups of experiments. They are different in the place we put the knowledge embedding in and the specific embedding terms (entity knowledge or upos knowledge).

The best score for the second part is 89.23, its corresponding parameters are: adding knowledge embedding in the embedding layer, using both entity knowledge and upos knowledge.

The third part of Table 1 is a summary pivot table of the second part.

Some conclusions could be drawn from Table 1. 1) Knowledge embedding in the embedding layer improves the model's performance. 2) Both entity knowledge and upos knowledge benefit the performance and entity knowledge is a little more important. 3) The performance dropped if we place the knowledge embedding in the header of the model. Perhaps the parameters we set in experiments are inappropriate or the model's header we designed is too simple.

6 Conclusion

We present two knowledge-enhanced methods in this model: the knowledge auxiliary text method and the knowledge embedding method. We design an answer extraction task pipeline to accommodate SemEval-2022 Task 09. Future work will involve incorporating the video data and predicting the answer of all pattern categories by model.

References

Joe Davison, Joshua Feldman, and Alexander M Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1173–1178.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *ACL/IJCNLP (1)*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Shanshan Liu, Xin Zhang, Sheng Zhang, Hui Wang, and Weiming Zhang. 2019. Neural machine reading comprehension: Methods and trends. *Applied Sciences*, 9(18):3698.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

R Mervin. 2013. An overview of question answering system. *International Journal Of Research In Advance Technology In Engineering (IJRATE)*, 1:11–14.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54.

Frank Rosenblatt. 1957. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. In *COLING*.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Jingxuan Tu, Eben Holderness, Marco Maru, Simone Conia, Kyeongmin Rim, Kelley Lynch, Richard Brutti, Roberto Navigli, and James Pustejovsky. 2022. Semeval-2022 task 9: R2VQ – competence-based multimodal question answering. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34.

Ningyu Zhang, Shumin Deng, Xu Cheng, Xi Chen, Yichi Zhang, Wei Zhang, Huajun Chen, and Hangzhou Innovation Center. Drop redundant, shrink irrelevant: Selective knowledge injection for language pretraining.

Appendix

A QA-pair pattern categories

The QA pattern categories are shown in Table 1.

B Rule-based method descriptions

Follows are brief descriptions of the some categories' rule-based methods. The QA-pair pattern categories are described in Table 1.

Cat 1: locate two sentences in the text and judge whose position is in front.

Cat 6: count how many times the tool or ingredient, is mentioned in the question, appears in the text.

Cat 7: get ingredient and action information from the question, locate the ingredient before the action occurs, and get the habitat of the ingredient.

Cat 8: count how many times the habitat, which is mentioned in the question, appears in the text.

Cat 9: locate the RESULT target in the context and extract the whole sentence. Reorganize the text by the components from original text and knowledge auxiliary text.

Cat 10: locate the HABITAT target in the context and extract relevant ingredients.

ID	Question	Answer	Pct.
01	Pouring batter in and baking other side, which comes first?	the first event	15.9
02	How do you cut the stalks?	by using a knife	13.7
03	What should be added to the pan?	the kale	12.9
04	Where should you divide the dough?	floured surface	10.8
05	Where do you transfer the garlic?	to a paper towel	8.5
06	How many times is the spoon used?	2	8.3
07	Where was the quinoum before it was mixed into the wok?	bowl	5.8
08	How many teaspoons are used?	2	5.6
09	How did you get the aromatic mixture?	by adding the shallot and garlic	4.5
10	What's in the gratin?	the cheese	4.2
11	For how long do you add diced mushroom pieces?	after a few minutes	3.5
12	To what extent do you stir the pie	til syrup thickens	3.4
13	Why do you whip the egg?	to mix well	1.3
14	From where do you drain water?	potatoes	0.8
15	What do you mix sweetener with?	with 3/4 cup water	0.6
16	By how much do you cover the beans with water in a pot?	by 2 inches	0.1
17	What do you cut the rectangle into?	into 6 squares	0.05
18	How would you reduce oven temp?	slightly	0.01

Table 2: QA-pair pattern categories