ISSN 2007-9737

Question Classification and Answer Extraction for Developing a Cooking QA System

Abdullah Faiz Ur Rahman Khilji[1], Riyanka Manna[2], Sahinur Rahman Laskar[3], Partha Pakray[1], Dipankar Das[2], Sivaji Bandyopadhyay[1], Alexander Gelbukh[3]

[1] Department of Computer Science and Engineering, National Institute of Technology Silchar, Assam, India

[2] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

[3] CIC, Instituto Politécnico Nacional, Mexico City, Mexico

{abdullahkhilji.nits, riyankamanna16, sahinurlaskar.nits, parthapakray, dipankar.dipnil2005, sivaji.cse.ju}@gmail.com, gelbukh@gelbukh.com

Abstract. In an automated Question Answering (QA) system, Question Classification (QC) is an essential module. The aim of QC is to identify the type of questions and classify them based on the expected answer type. Although the machine-learning approach overcomes the limitation of rules as is the case with the conventional rule-based approach but is restricted to the predefined class of questions. The existing approaches are too specific for the users. To address this challenge, we have developed a cooking QA system in which a recipe question is contextually classified into a particular category using deep learning techniques. The question class is then used to extract the requisite details from the recipe obtained via the rule-based approach to provide a precise answer. The main contribution of this paper is the description of the QC module of the cooking QA system. The obtained intermediate classification accuracy over the unseen data is 90% and the human evaluation accuracy of the final system output is 39.33%.

Keywords. Question classification, answer extraction, cooking QA, BERT.

1 Introduction

Question Answering (QA) is a well-defined task of Natural Language Processing (NLP), where precise answers are extracted for a question posed by the user. There are two main types of QA systems: open-domain and restricted-domain ones. An open-domain system can process questions from any domain, whereas a restricted-domain system processes domain-specific questions, such as questions on jobs, medicine, agriculture, railways, or automobiles. In this paper, we focus on a domain-specific QA system.

To obtain accurate answers or proper information, question classification (QC) plays a vital role in a QA system: it decids on types of questions and the corresponding answer [2]. There are various types of questions that start with *when*, *what*, *where* and *who*, which are known as factoid type and can be responded in a sentence or a single phrase. On the other hand, *how* and *why* belong to non-factoid questions that require a procedure, reasoning, and a suitable explanation in the answer. Apart from this, there are various categories of questions that depend on a specific context, which needs to be identified for domain-specific systems.

In this paper, we focus on the cooking recipe domain. In this domain, a question such as *"What*

are the ingredients required for garnishing milk rabdi?" might be asked by the user. To answer these questions, the QA system needs to extract information about ingredients only. Hence, the system must categorize the questions which fall under the category of "Ingredients".

To achieve this goal, we have developed a cooking recipe dataset and proposed a cooking QA system. In our system, a deep learning method is deployed to identify the category of questions and a rule-based approach is used to extract keywords from the question for item identification. And finally, the system returns a relevant answer to the user.

Our system will benefit such professions as food service managers, nutritionists, bakers, and as a personal assistant for amateur cooks.

The rest of the paper is structured as follows: Section 2 briefly explains related works. The preparation of the dataset and the architecture of our system are briefly described in Sections 3 and 4, respectively. Section 5 presents experiment result and error analysis. Section 6 concludes the paper and outlines some future work directions.

2 Related Works

In the QA system, there are two different approaches for QC: rule-based and machine-learning-based. Besides these, hybrid approaches that merge both rule-based and machine-learning-based approaches also exist [5, 12]. The rule-based approach [11] finds the identical question based on manual rules and shows good performance for a specific dataset. One such work is [9] wherein exhaustive experiments have been performed and analysis made to display the prowess of Information Retrieval (IR) using rule-based approaches.

The work [1] further reinforces the idea of IR using rule-based approaches and Answer Validation (AV) machine reading techniques. Nevertheless, the performance of such approaches decreases when applied in a different contextual setting. Furthermore, too many rules need to be built. To handle such issues a machine-learning-based approach is introduced [8], wherein question features are extracted to train a classifier model. And then, using the trained classifier predicts the class label over unseen data. There are different types of question classes known as question taxonomy or question ontology [6]. These classes are domain-specific.

For the cooking recipe domain, the QA system has not been advanced [15]. However, the literature survey finds a similar work based on Information Retrieval (IR) [7], in which cooking ontology is used for capturing key concepts such as actions, food, recipes, and utensils to answer a question. Our work aims to develop a cooking QA system that classifies the various types of recipe questions using a deep learning technique and maps the question class with the keywords extracted from the user's question through a rule-based approach.

Accuracy estimation at intermediate levels of the system can be easily evaluated through Accuracy metric in the case of machine-learning-based classification approaches and similarity scores in the case of keyword searching. For calculating the accuracy of the final output of the model [9] and [1] have used C@1 [10] in their evaluation campaign. Hence, we have used C@1 [10] along with a suitable metric proposed in this work.

3 Dataset Preparation

For unsupervised training, we require training our model on the domain-specific corpus. Thus, we crawled websites containing food recipes[1] and scraped components such as 'Title', 'Description', 'Ingredients', 'Cook Time', etc., totaling 225,602 individual food recipes.

For the supervised classification component, we have developed our own dataset containing 2175 questions and split into train, validation, and test set as shown in Table 1. This dataset contains 15 classes as described in Table 2. For simplicity, we assume each recipe question is unambiguous, which belongs to a unique class.

[1] https://www.allrecipes.com/

Table 1. Dataset statistics

Dataset	Instances
Train	1934
Validation	22
Test	219

4 Our Cooking QA System

Our cooking QA system consists of two main approaches, one being the advanced deep learning techniques and the other, contemporary rule-based approach. When a user enters a question, it passes through the steps discussed in the subsequent Sections.

4.1 Transformer Model

In the deep learning approach, the question is processed by the Bidirectional Encoder Representations from Transformers (BERT) [4]. Pre-training and fine-tuning are the two major steps of BERT. The pre-training step is unsupervised learning which trains the model on unlabeled data over different pre-training tasks. The pre-training is performed in two ways. First, the pre-trained Wikipedia corpus model is acquired following default settings of [4]. Secondly, pre-training the baseline BERT model on our corpus to learn domain-specific embeddings utilizing experimental settings as [4]. Then we have fine-tuned our pre-trained model for narrowing down the domain of contextual embeddings to accommodate the context of the recipes for the classification task on our self curated dataset. We then fine-tune the model on labeled data for learning supervised embeddings after the initialization of the pre-trained weights. As a result of this approach, the question is classified into a particular category.

4.2 Rule Based Pipeline

For the rule-based approach, keywords are first extracted which are passed as a query to the Mongo Database (MongoDB) [3]. This database is used for indexing and searching the scraped data of recipes exported in JSON format. The stop words are removed and the data is stemmed to facilitate easier indexing. Based on parameters such as similar words, similar sounding words, and similarly spelled words, a score is then generated and the search results are sorted based on these parameters. Finally, the recipe with the maximum score is displayed, which is further analyzed with the output of the deep learning approach to extract an answer.

4.3 Summarized Illustration

For the input question "what are the ingredients present in rajma" shown in Figure 1, our system passes this question into two different pipelines.

The deep-learning pipeline classifies the question as "ING" class, i.e., ingredient category and the rule-based approach extracts the keyword "rajma", and passes it as a query to the MongoDB.

The recipe generated from the query is then used to map the "ING" class from the answer extraction module to return an answer.

5 Experiment Result and Error Analysis

In this section, we will discuss in detail the classification and mapping approaches utilized.

5.1 Transformer Model: Experimental Setup

Previous approaches before BERT [4] had restricted domain knowledge as each word had the same meaning in any context. Moreover, these models lack bidirectional context understanding as in OpenAI GPT [14] making it difficult to learn domain-specific information. Thus, to leverage the contextual information we utilized BERT [4] to learn contextual embeddings that overcome the unidirectional challenge by using the Cloze task [13]. The unsupervised task is to learn to predict the randomly masked tokens from the input based on its left and right context by training a deep bidirectional transformer using cross-entropy loss. The next sentence prediction task is also performed in order to enhance the embeddings. Masking involves randomly concealing 15% of the tokens in about 80% of the pre-training data. The

Table 2. Example of recipe questions of various class

No.	Class (Label Name)	Example	No. of statements
1	ADV (Advice)	What are the tips to prepare Dahi or Yoghurt quicly?	132
2	DIFF (Difference)	What is the difference between Barfi and Kalakand recipe?	75
3	DIR (Direction)	Guide me in cooking Mashroom Chilli.	212
4	EQUIP (Equipment)	What are the required utensils for cooking Punjabi Kadai Paneer?	44
5	ING (Ingredients)	What are the ingredients for rice kheer?	138
6	JUST (Justification)	When could we use less sugar in Milk Barfi preparation?	100
7	NAME (Name)	What are great antioxidants?	119
8	OBJ (Objective)	What is Besan Parantha?	276
9	PREP (Preparation)	How to prepare Punjabi Style Dam Aloo?	85
10	PRER (Prerequisite)	What are the prerequisites for diets with low fat?	56
11	QTY (Quantity)	How much lemon juice do we use for making Paneer Tikka?	95
12	SPLINFO (Special information)	Why is rose water added in Instant Mango Mouse Recipe?	312
13	TIME (Time)	What is the cook time for Masala Beans?	184
14	WRN (Warning)	what are the precaution should be taken to preServe Achari Meat?	59
15	YESNO (Yes or No)	Is it compoulsary to add corriender leaves?	280

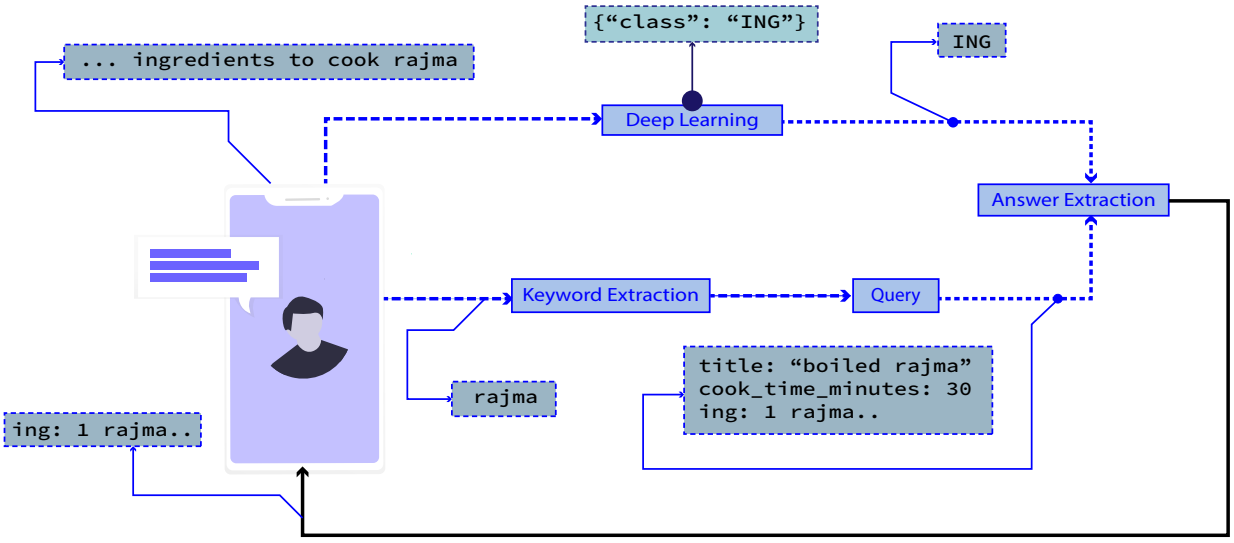

Fig. 1. Our system architecture

Table 3. Sample output of our system

SN	Question (Input)	Class	Answer (Output)
1	How can I make a super-thin, yet strong, calzone crust?	DIR	... It can be left a bit chunky (my preference) or smashed until completely smooth ... parchment paper to be the foundation so you can move the salad ...',
2	What is the most common mozzarella used in Italian pizza	OBJ	... Crumbled bacon, pineapple chunks, bell pepper strips and prepared pizza sauce make a quick and tasty pizza ...',
3	How do I know when my wine is properly reduced?	SPLINFO	'... I recommend using Bosc or Anjou pears.', 'Use Zinfandel, Shiraz, or Merlot for the red wine ...'

rest is suitably divided into a random and an unchanged token.

For our purpose, we have used a BERT base with 12 transformer layers having about 110 million parameters pre-trained on the BookCorpus[2] dataset and the Wikipedia corpus. These pre-trained weights are then initialized to learn recipe specific domain embeddings. We ran the unsupervised model for 100 epochs to obtain the next sentence prediction accuracy of 94.25%. Since BERT involves the usage of a self-attention mechanism, it is easier to accommodate many NLP tasks including multi-classification in which we take an interest.

For the classification component of BERT [4], we have used the multi-classification version of the Corpus of Linguistic Acceptability (CoLA) classifier task. We have used 15 labels for our classification task as described earlier in the dataset description in Section 3. The new parameters introduced for classification are the classification layer weights. The classification loss is computed using the standard logarithmic loss.

5.2 Similarity Score

As discussed in Section 4.2 the scraped data is stored in MongoDB and indexed to enable real-time keyword searching. The indexing is done on all the data fields to enable an exhaustive search on the dataset. This is then embedded into the system pipeline wherein the suitable search results are extracted and ordered based on similarity scores (as mentioned in Section 4.2) for retrieving the final answer as per the classification output of the Deep Learning model.

[2] http://yknzhu.wixsite.com/mbweb

5.3 Evaluation and Results

Classification of different questions into different classes resulted in an overall test accuracy of 90%. The output of the Deep Learning model is then fed into the rule-based system for final output. A sample is shown in the Table 3.

Table 4. Statistics for human evaluation

Total no. questions (n)	50
Total no. of answered questions (n_R)	37
Total no. of unanswered questions (n_U)	13

Table 5. Rating score description

Score	Description
3	Best answer
2	Average answer
1	Out of domain
0	No answer

To examine the performance of our system, we have used two evaluation measures. The first one is C@1 [10], which measures the proportion of questions that are correctly answered:

$$C@1 = \frac{1}{n}(n_R + n_U \frac{n_R}{n}), \quad (1)$$

Table 6. Human-evaluated results

HE1	HE2	HE3	Avg Score (n_{AS})	Accuracy (%)	C@1
66	56	55	59	39.33	0.93

Table 7. Misclassification of questions generated by our system

SN	Question	Gold class	System generated class
1	When should or shouldn't you toss pasta with sauce?	TIME	ADV
2	Is my ragu missing an ingredient?	SPLINFO	YESNO
3	Traditional Italian pasta: with or without eggs?	SPLINFO	YESNO
4	What are techniques to make homemade pasta without a pasta machine?	SPLINFO	NAME

where n_R is the number of questions correctly answered, n_U is the number of questions unanswered and n is the total number of questions. In the second one, we define accuracy by considering the rating of evaluators:

$$\text{Accuracy} = \frac{n_{\text{AS}}}{n_{\text{TBS}}} \times 100\%, \tag{2}$$

where n_{AS} is the average rating score and n_{TBS} is the total best rating score. Table 4 presents the statistics for human evaluation and the Table 5 depicts the rating score. The evaluated result is presented in Table 6 over the selected 50 questions that are generated by our system. The selected answers are evaluated manually by three human evaluators namely, Human Evaluator-1 (HE1), Human Evaluator-2 (HE2) and Human Evaluator-3 (HE3). The human evaluators are the research scholars. Here, n_{TBS} is calculated by multiplying best answer score with total number of questions, i.e., $3 \times 50 = 150$.

5.4 Error Analysis

Although our system acquires 90% accuracy in QC, it still suffers from misclassifications on a few test questions presented in Table 7. There are certain instances where the system wrongly predicts the question class, even though the sentence might seem quite trivial for a human evaluator. In our future works, we would increase the number of test sentences to examine broad classification categories were our system generally does not perform up to the mark.

6 Conclusion and Future Work

This paper presents a QA system in the cooking recipe domain, where our main focus is the contextual classification of recipe questions. The same has been performed using a state-of-the-art deep learning technique BERT and achieved remarkable performance on intermediate QC and has shown good accuracy on the final system output based on the evaluation metrics considered. To the best of our knowledge, there has been no work done in the contextual classification of questions for such a domain-specific QA system. Moreover, a rule-based approach is added to the our system, which yields a set of queries by the combination of all the keywords present in the user's question.

In the future, we shall increase the size of the dataset and try to implement the cooking QA system on a multi-model dataset.

Acknowledgement

We would like to thank Department of Computer Science and Engineering and Center for Natural Language Processing (CNLP) at National Institute of Technology Silchar and Department of Computer Science and Engineering at Jadavpur

University, Kolkata for providing the requisite support and infrastructure to execute this work. The last author acknowledges support of the Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional, Mexico, via the grants 20200859 and 20201948, and of the CONACYT, Mexico, via the grant A1-S-47854.

References

1. **Bhaskar, P., Pakray, P., Banerjee, S., Banerjee, S., Bandyopadhyay, S., & Gelbukh, A. (2012).** Question answering system for QA4MRE@ CLEF 2012. *CLEF (Online Working Notes / Labs / Workshop)*.
2. **Breja, M. & Jain, S. K. (2017).** Why-type question classification in question answering system. *FIRE (Working Notes)*, pp. 149–153.
3. **Chodorow, K. (2013).** *MongoDB: The definitive guide: powerful and scalable data storage*. O'Reilly Media Inc.
4. **Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018).** BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
5. **Huang, Z., Thint, M., & Qin, Z. (2008).** Question classification using head words and their hypernyms. *Proceedings of the 2008 Conference on empirical methods in natural language processing*, pp. 927–936.
6. **Loni, B. (2011).** A survey of state-of-the-art methods on question classification. Technical report, Delft University of Technology.
7. **Manna, R., Pakray, P., Banerjee, S., Das, D., & Gelbukh, A. (2016).** CookingQA: A question answering system based on cooking ontology. *Mexican International Conference on Artificial Intelligence*, Springer, pp. 67–78.
8. **Moschitti, A., Quarteroni, S., Basili, R., & Manandhar, S. (2007).** Exploiting syntactic and shallow semantic kernels for question answer classification. *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 776–783.
9. **Pakray, P., Bhaskar, P., Banerjee, S., Pal, B. C., Bandyopadhyay, S., & Gelbukh, A. (2011).** A hybrid question answering system based on information retrieval and answer validation. *CLEF (Notebook Papers / Labs / Workshop)*.
10. **Peñas, A., Forner, P., Sutcliffe, R., Rodrigo, A., Forundefinedscu, C., Alegria, I., Giampiccolo, D., Moreau, N., & Osenova, P. (2009).** Overview of ResPubliQA 2009: Question answering evaluation over European legislation. *Proceedings of the 10th Cross-Language Evaluation Forum Conference on Multilingual Information Access Evaluation: Text Retrieval Experiments*, CLEF'09, Springer-Verlag, Berlin, Heidelberg, pp. 174–196.
11. **Prager, J., Radev, D., Brown, E., & Coden, A. (1999).** The use of predictive annotation for question answering in TREC8. *In NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8*.
12. **Silva, J., Coheur, L., Mendes, A., & Wichert, A. (2011).** From symbolic to sub-symbolic information in question classification. *Artif. Intell. Rev.*, Vol. 35, pp. 137–154.
13. **Taylor, W. L. (1953).** "Cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, Vol. 30, No. 4, pp. 415–433.
14. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017).** Attention is all you need. *Advances in neural information processing systems*, pp. 5998–6008.
15. **Yagcioglu, S., Erdem, A., Erdem, E., & Ikizler-Cinbis, N. (2018).** RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Brussels, Belgium, pp. 1358–1368.

Article received on 09/12/2019; accepted on 23/02/2020. Corresponding author is Partha Pakray.