ISSN 2007-9737

Question-Answering and Recommendation System on Cooking Recipes

Riyanka Manna[1], Dipankar Das[1], Alexander Gelbukh[2]

[1] Jadavpur University,
Computer Science and Engineering,
India

[2] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{riyankamanna16, dipankar.dipnil2005}@gmail.com, www.gelbukh.com

Abstract. Question answering (QA), one of the important applications of Natural Language Processing (NLP) aims to take the user questions and returned to the user with the answers. An open domain QA system deals with a set of questions that can be of any domain. The other type of QA is close-domain where it deals with the questions under a specific domain e.g., agriculture, medicine, education, tourism, etc. Our cooking question answering system is an example of a closed domain QA system. Here, users can ask the cooking related questions and the system returns the actual answer to the user. In this paper, we present different modules of a cooking QA system. In addition to dataset preparation, the development of a cooking ontology, the classification of questions as well as the extraction of candidate answers are also treated as other important aspects, which are discussed in this paper in details. In the cooking QA system, automatic evaluation metrics such as precision, recall, F-score, and C@1 were used for the evaluation of precise answers. In addition, human evaluation is used based on a rating scale. Moreover, the recommendation of recipes has also been attempted and the evaluation metrics show satisfactory performances of the systems.

Keywords. Natural language processing, question answering, cooking recipe, question classification, recommendation.

1 Introduction

Question Answering is a developing research area that combines the research from Information Retrieval (IR), Information Extraction (IE) and NLP. It can be also considered as the next step in IR. One of the aims of a QA system is to generate automatically answers to natural language questions provided by the users. The questions can be in natural language and the generated answers are also in some target natural languages. Cooking is an interesting and challenging domain of Question-Answering system.

The development of the cooking field in QA domain was started a few years back. A number of well-known works have been done depending on recipes images [18]; a few other models in [19][20] have been done depending on different recipes. However, to understand fully Cooking QA systems, how it has been developed to serve its current QA needs, a broader survey of Cooking QA systems is required. In this paper, we present a detailed study on the different aspects of cooking QA system. To make the recommendation model important it is important to display only those recommendations that have a best probability to be fit for the user questions.

The main contributions of the paper are as follows. First, the development of cooking ontology was done. Second, different classification models were used for user question classification. Third, entailment based approach and IR based approach has been used for answer extraction and lastly recipe recommendation has been developed.

The rest of the paper are organized as follows. Section 2 reviews background works on cooking

doi: 10.13053/CyS-25-1-3899

QA systems. Cooking ontology is discussed in Section 3. Question classification is discussed in Section 4. Section 5 presents the answer retrieval and answer extraction. Recipe recommendation is discussed in Section 6. Finally, Section 7 concludes the paper and point out directions for future research.

2 Related Work

In general question answering systems, we combine the Information Retrieval (IR) with extraction techniques to detect a set of candidate answers and then use some selection strategy to generate the final answers. The most popular classes of techniques employed for QA are open-domain and restricted-domain. These two domains also use thesauri and lexicons in classifying documents and categorizing the questions. Open domain question answering (ODQA) [1] deals with questions about nearly everything and can only rely on general ontology.

To answer unrestricted questions, a general ontology or common sense knowledge would be useful. Restricted-domain question answering (RDQA) [2] closed domain deals with questions under a specific domain like tourism, medicine, etc. Over the years, many question-answering systems have been developed, for a variety of purposes. Some systems are intended to provide database access to very specific domains, while others are more open- domain, aiming to answer general trivial questions.

The context in which a QA system is used, i.e., the anticipated user, the type of questions, the type of expected answers, and the format in which the available information is stored, determines the design of the overall system. Two basic types of question answering systems can be distinguished: systems that try to answer a question by accessing structured information contained in a database, and systems that try to answer a question by analyzing unstructured information such as plain texts.

Since 1992, the annual Text Retrieval Conference (TREC)[1] organized by the National Institute of Standards and Technology (NIST) provides a forum for researchers to compare the effectiveness of their systems in information retrieval related tasks. In 2002, 34 research groups [3] participated in the question-answering track of TREC, each group having implemented their own system. These systems cover a wide spectrum of different techniques and architectures, and it is impossible to capture all variations within a single architecture. Nevertheless, most of the systems also have a number of features in common, which allows us to give a general architecture of a prototypical question answering system.

Xia et al. [24] proposed an approach to answer generation for cooking question-answering systems. They introduced an annotation scheme for knowledge database. Finally, they have presented the answer planning based approach for generating an exact answer in natural language.

Lukovnikov et al. [15] followed a quite different approach: they trained a neural network for answering simple questions in an end-to-end manner, leaving all decisions to the model. It learns to rank subject-predicate pairs to enable the retrieval of relevant facts given a question.

Zeyen et al. [16] presented a new approach for describing a collection of cooking recipes represented as cooking workflows with the help of a conversation. They have provided a method to manage a conversation with the user to find desired cooking recipes. They concentrated on the structural features of recipes that signifies as workflows.

Chen et al. [13] used encoding of a recipe into a vector for capturing cooking procedure that indicates causality effect between ingredients and actions. They model the attention of words and sentences in a recipe and align them with its image feature such that both text and visual features share high similarity in multi-dimensional space.

QA over Knowledge Bases (KBs) research performed by Veron et al. [14] used the translation of Natural Language (NL) questions into formal queries, and the detection of missing knowledge that impact the way a question is answered. Cookpad[2] recommend recipes to users but do not reflect the user’s specific needs.

[1] https://trec.nist.gov/

[2] https://cookpad.com

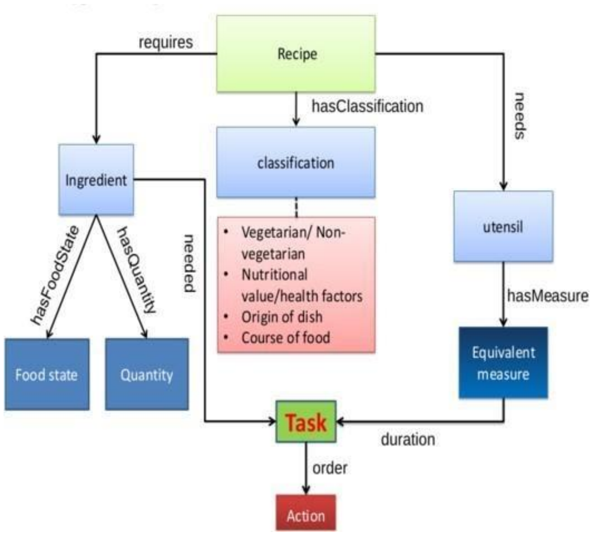

Fig. 1. Cooking Ontology System Architecture

Table 1. Different competency question examples

CQ1: Method-oriented
CQ1.1: How to cook the dish <recipe name>
CQ1.2: Help me to prepare the dish <recipe name>
CQ1.3: What are the tips to cook <recipe name>
CQ2: Time-oriented
CQ2.1: What is the preparation time to cook <recipe name>
CQ2.2: What is the cooking time of <recipe name>
CQ2.3: What is the total time make the dish <recipe name>
CQ3: Ingredient-oriented
CQ3.1: What are the ingredients to make <recipe name>
CQ3.2: What is the quantity of a particular ingredient to make <recipe name>
CQ3.3: What are the quantities to use when making <recipe name> for 4 persons?
CQ4: Utensil-oriented
CQ4.1: Which utensils can we use to make the <recipe name>
CQ4.2: Which recipes can be made using the oven?

Apart from this, schedule-based recommendation was considered by Mino and Kobayashi [21]. The three strategies for a recommendation system, viz. content-based, collaborative and hybrid were explored by Freyne and Berkovsky [22]. A recommendation system based on the availability of ingredients was reported by Müller et al. [23].

3 Cooking Ontology

Ontologies are often considered as one of the essential components to build any intelligent system. An ontology is an explicit specification of a conceptualization [4]. It is a symbolic vocabulary for a discourse, definitions of classes, relations, functions, and other objects.

In the present task to build a cooking QA system, an ontology model on the cooking domain has been developed. The user queries are processed with the help of the ontology knowledge and returned with proper answers. The cooking ontology helps not only the requirement analysis of the cooking domain; the available concepts provide a schematic view of the particulars involved in the cooking recipe as well. The relationships between the concepts or classes comprehended the real-world interaction among various modules of ontology.

Our cooking ontology model [5] describes the following process: identification of concepts and properties, concepts classification in classification trees, description of properties or attributes of classes, building relationships between the classes, instances identification and description.

In general, the concepts within these hierarchies are associated with IS-A relations whereas the attribute-based relations were also used to associate concepts from several hierarchies. In Figure 1, the ingredient class has two component class like food state and food quantity. The attributes of the food state are raw, boiled, fried, and baked.

The scope of the ontology is described through some basic questions that are called competency questions. The ontology building process was started through finding the answers to these competency questions, which is shown in Table 1.

The ontology has been designed based on the answers to user's questions. However, implicit questions were answered with the help of cooking ontology. The method of ontology building finds the semantic relationship between entities or concepts, attributes, and relationships. Our cooking ontology model has mainly three components, namely ingredients, classification, and utensils.

Ingredients: In this class defines the type and list of ingredients that come under the recipe classes. The ingredient class has subclasses as food state and quantity. The food state subclass describes the state of food, i.e., it is raw or boiled or grated or mashed. The quantity subclass describes the required quantity of ingredients to make the dish.

Utensil: In this class similarly mentions the required utensil name and it has the subclass equivalence measure that relates with the specific ingredients.

For example, “*how many liters are in a cup?*” This measures the equivalence quantity.

Classification: In the classification that provides the type representation of recipes according to different criteria.

The first classification is whether the dish is *vegetarian or non-vegetarian*, i.e., according to the major ingredient that is used. The second type is the *origin of the dish*, i.e., it is Indian / Italian / Chinese / Mexican, etc. The third type is according to its *nutritional value* such as it is low-calorie recipes for weight loss or recipes for high blood pressure or recipes for diabetic patients or recipes for pregnancy. The list is not at all exhaustive as different information at every stage can be added in the hierarchy during the process of ontology development.

The ontology has several relations that exist among its classes/concepts. There is a relation ‘requires’ that exists between recipe and ingredient and it describes the required ingredients for preparing a specific recipe. The recipe also maintains a relationship with a utensil, namely ‘*needs*’, which provides the needed utensil name. The recipe and classification maintain a relation called ‘*hasClassification*’ that holds the different classification criteria. There is a relation between the ingredient and food state is named ‘*hasFoodstate*’ that describes the state of the food-material. Another relation between ingredient and quantity is ‘*hasQuantity*’ that specifies the required quantity to make the dish. The utensil has a relation with the equivalent measure is ‘*hasMeasure*’ that defines the measurement of utensil. The needed ingredient and required equivalent utensil maintains a relation ‘*duration*’ and performs the desired task. Finally, some ordered tasks form the final action.

Advantages and disadvantages of our ontology include the follows. The advantages of this ontology are:

a) Clear representation of specialized knowledge: The ontology represents the conceptual structure so that it models the basic categories of that domain.

b) Efficiency in Information Retrieval: This model is very efficient to specify different types of relations among them, so they help to describe formally the specific domain to which the terms belong.

c) Collecting the missing information: Some information are not available in the raw dataset for the experiment. The missing information can be answered from the ontology database.

Some disadvantages of the ontology are:

a) Great number of ontological languages: There are different ontological languages for developing ontologies such as RDF, RDF Schema, OIL, DAML+OIL and OWL. It makes impossible the interchange or reuse of data between systems that do not share the same languages.

b) Difficulty of turning special knowledge into ontologies: Sometimes it is difficult to transfer specialized knowledge from texts or domain experts to abstract and effective concept representations.

c) Representation of synonymy: Another drawback regarding representation is synonymy and its representation in ontologies.

d) Lack of suitable tools: Another disadvantage is the unavailability of tools for building ontologies. Standard ontology editing tools, such as Protégé. it is not always easy to adapt standard ontology editors to terminological purposes, and the work involved can be time consuming.

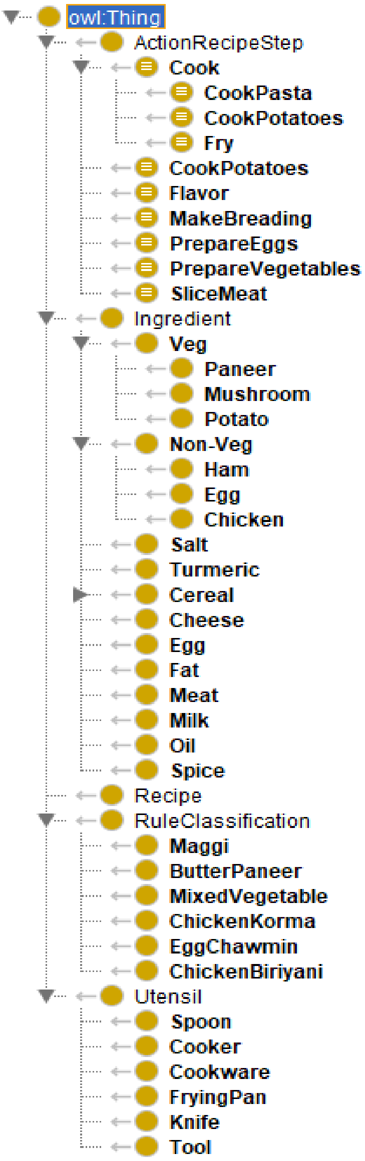

Fig. 2. Protégé OWL Architecture

Protégé [11] is a well-known tool for knowledge representation and reasoning concepts, it is used to define the knowledge concepts and their properties and instances.

We used Protégé to build our ontology by producing OWL file format as shown in Figure 2.

Not only such an evaluation of our ontology is explicitly done but also, the implicit verification of our ontology has been conducted in the stage of question classification as well.

4 Question Classification

The question classification task is one of the crucial tasks in the Question Answering system. Question classification is the task of identifying not only the question type. It also helps in extracting the required answer type of any question asked by a user. However, there are no standard text corpora available for carrying out research on specific cooking related questions.

Thus, we had to prepare experimental data ourselves [8], as discussed in the following sub-section. Based on the collected data, we conducted two different sets of experiments also.

4.1 Experiment 1

The cooking data are collected from various cooking websites: punjabi-recipes.com, www.tarladalal.com, www.allrecipes.com etc. We used Apache Nutch Crawler[3] to collect the raw data. We conducted a basic level of pre-processing to obtain clean data by using NLTK toolkit[4]. Depending on the question type, we divided the total question datasets into 14 different classes. We collected 1668 questions related to cooking recipes. We also classified these questions into 14 classes.

Table 2 gives the detailed statistics of the total number of classes, total data in each of the classes present in our cooking dataset. In contrast, for the answer selection purpose, we used another website that is Yahoo Answer[5]. With respect to more than 5,000 questions of cooking domains, the documents from Yahoo Answers were collected and 1,668 recipes were identified for human evaluators.

In order to classify the questions, we conducted two different sets of experiments as discussed below.

[3] http://nutch.apache.org/

[4] https://www.nltk.org/

[5] https://in.answers.yahoo.com/

Table 2. Question classes with examples

Class	Instances	Example
QTY	60	How much water content is required for rice?
ADV	120	Give a diet rich in vit D
ING	130	What are the ingredients for chicken biriyani?
YESNO	210	Is pasta good for health?
PREP	40	How to cook mutton biriyani?
DIR	198	How to process chicken?
WRN	50	What are the precautions should be taken to preserve lamb?
SPLINFO	200	How to check food quality?
EQUIP	40	Which utensil is required for omlette?
TIME	150	How much time us required to cook pulao?
OBJ	250	What is aloo paratha?
JUST	50	When could we use less sugar in kulfi?
DIFF	60	What is the difference between pulqao and biriyani?
NAME	110	Give recipes without grains

Table 3. Experimental results on QC task

Sl no	Learning Algorithm	No of questions	Accuracy (%)
1	SVM	1668	81.70
2	Naïve Bayes	1668	83.44
3	Bidirectional Encoder Representation from Transformer (BERT)	1668	87.02

Table 4. System-generated results

SL no	Question	Gold class	System generated class
1	When should or should not you toss pasta with sauce?	Time	ADV
2	Is my ragi missing an ingredient?	SPLINFO	YESNO
3	Traditional Italian pasta with or without eggs?	SPLINFO	YESNO
4	What are techniques to make homemade pasta without pasta machine?	SPLINFO	NAME

We performed the classification task with two state of the art machine-learning algorithms like Naïve Bayes classifier and Support Vector Machine (SVM) along with Deep Neural Network for question classification task. We used Bidirectional Encoder Representations from Transformers (BERT) [12] for pre-training the model on the corpus using the cloze task. It gives around 87.02% accuracy [12]. The experiment result is shown in Table 3.

4.2 Experiment 2

In this set up, we aimed for unsupervised training and the model was required to be trained on the domain-specific corpus [10]. Thus, we crawled

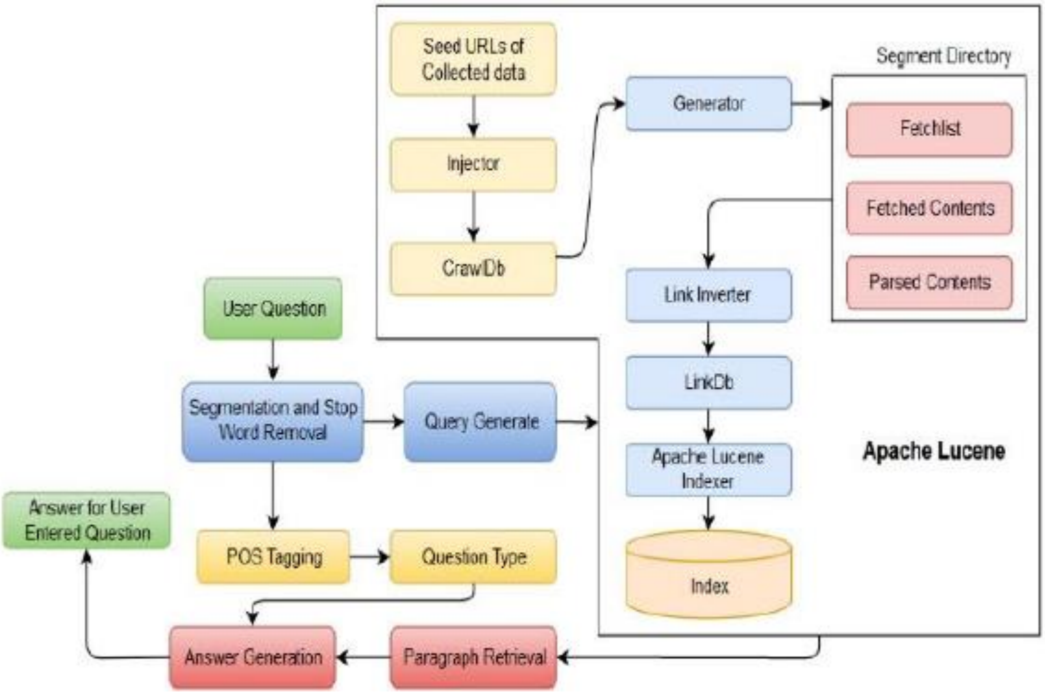

Fig. 3. System architecture of the QA system

various websites containing food recipes and scraped components such as 'Title', 'Description', 'Ingredients', 'Cook Time', etc., for a total of 225,602 individual food recipes. For the supervised classification component, we developed our own dataset containing 2175 questions and split it train (1934), validation (22), and test set (219).

In this system, we used a BERT base with 12 transformer layers having about 110 million parameters pre-trained on the BookCorpus dataset[6] and the Wikipedia corpus. These pre-trained weights are then initialized to learn embedding specific to recipe domain.

We carried out 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 our job on multi-classification of question types.

Although our system acquires 90% accuracy in classification results, it still suffers from misclassifications on a few test questions presented in Table 4. There are certain instances where the system wrongly predicts the question class, even though the sentence might seem quite trivial for a human evaluator.

However, the performance of QC module and the identified question types were re-evaluated while retrieving the correct answers from recipes.

5 Answer Retrieval

We have a standard dataset on recipes and foods from 20 famous cities in India [7]. It was collected from various Indian recipe websites. We used the open source Apache Lucene [9] for building information retrieval framework.

We made the XML documents for each city and added these XML files to the search engine. We considered the recipe name, time to cook, level of difficulty, ingredients, method.

As per our system requirement, we converted the data into our own format, so that it could be easily indexed by a search module.

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

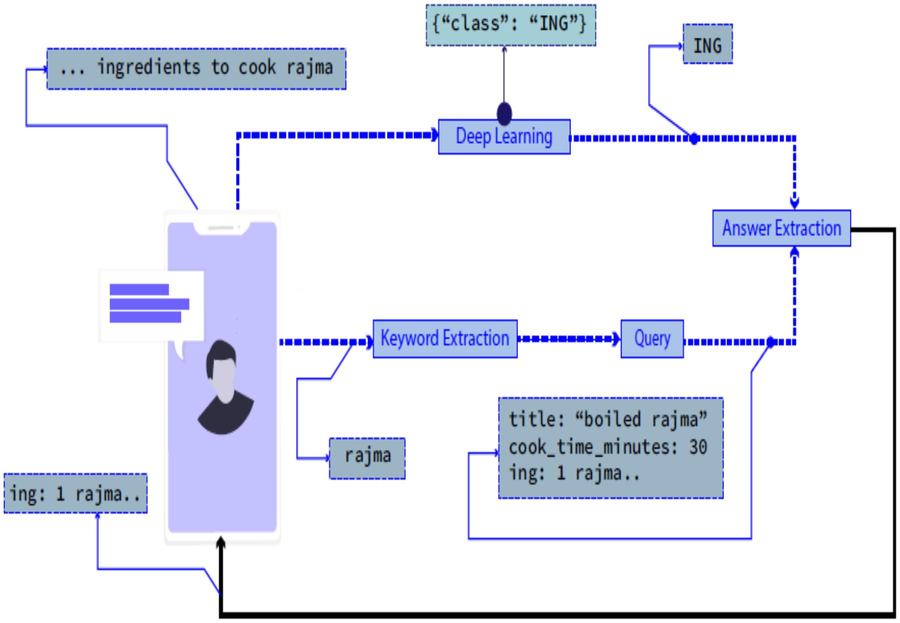

Fig. 4. System architecture for QA and QC

The evaluation of retrieved answers is conducted based on the hypothesis on which the QA system is built on. Thus, we used three different varieties with respect to the overall QA architecture. Based on the performance of different QC modules individually, we selected these three sets of experiments also. The evaluation of answers has also been done with respect to various matrices as described in the following sections.

5.1 Experiment 1

The first type of our QA system comprises of four modules. The system architecture is shown in Figure 2. These four main modules are as follows: A. Apache Lucene module B. Query Processing module C. Document Processing module D. Answer Processing module.

5.1.1 Apache Lucene

Apache Lucene [9] is a java library, which builds an index table, which is easily searchable for the retrieval. Furthermore, Apache Lucene has two important aspect: first one is the way how the data is crawled/stored and the way how to search that indexed data.

5.1.2 Query Processing module

The purpose of the query processing module is to examine the user question and process the input question and remove the stop words and formed user query certain formats.

5.1.3 Document Processing module

This module contains revised queries, which are fed into the IR system, and retrieves the bunch of documents in a ranked order. The key role of this module is to obtain relevant information from one or more systems and stored documents.

5.1.4 Answer Processing module

In order to give precise answers in a complete sentence, we prepared a set of template answers for every type of questions. Here, the answers for both the factoid and non-factoid questions could be obtained. The template is presented in the front part of the answers whereas end data appears in the rear part is extracted from the dataset so that the system could give more specific answers in a concise manner.

Table 5. Experiment result

QT	Q	CA	ICA	NA	COV	ACC
Easy type	21	18	0	3	85.71	85.71
Complex type	29	17	0	12	58.62	58.62
Total	50	35	0	15	70.00	70.00

Table 6. Performance of the system

Total number of question (n)	50
Total number of answered question (n_R)	37
Total number of unanswered questions (n_U)	13

Table 7. Human evaluation score

Score	Description
3	Best Answer
2	Average Answer
1	Out of Domain
0	No Answer

Table 8. Human evaluation results

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

Table 9. Dataset Statistics

Type	# Instances	Sources
KB	1223	Cooking websites and blogs
Test Data	50	Manual

Table 10. Dataset Statistics for Recipe Recommendation

Description	Data
Train	1903
Validation	57
Test	219

We categorized all the 50 different questions into two categories: 'easy' and 'complex' depending on the complexity of answer processing. The question sets contains 21 easy type and 29 complex type questions and we checked each of the questions and prepared the results given below. The result of evaluation is shown in Table 5. Where, QT: the type of question; Q: the number of questions; CA: the number of correct answers; ICA: the number of incorrect answers; NA: the number of no answer; Cov: coverage; Acc: accuracy.

5.2 Experiment 2

The second type of our cooking QA system [10] consists of two main approaches: one being the advanced deep learning techniques for question classification and the other, contemporary rule-based approach for answer extraction. The output of the Deep Learning model is finally fed into the rule-based system for generating the final output.

As shown in the following example in Figure 4, the deep-learning pipeline classifies the question as "ING" class, i.e., ingredient category and rule-based approach extract the keyword "rajma", and passes it as a query to the data. The recipe generated from the query is used to map the "ING" class from the answer extraction module to return an answer.

On the other hand, to examine the performance of our system, we used two evaluation measures.

The first one is C@1 [10], which measures the proportion of questions that are correctly answered:

$$C@1=\frac{1}{n}\left(n_R+n_u\frac{n_R}{n}\right),$$

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:

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

where n_{AS} is the average rating score and n_{TBS} is the total best rating score.

The evaluated results are presented in Table 6 over the selected 50 questions that are generated by our system and human evaluation score is

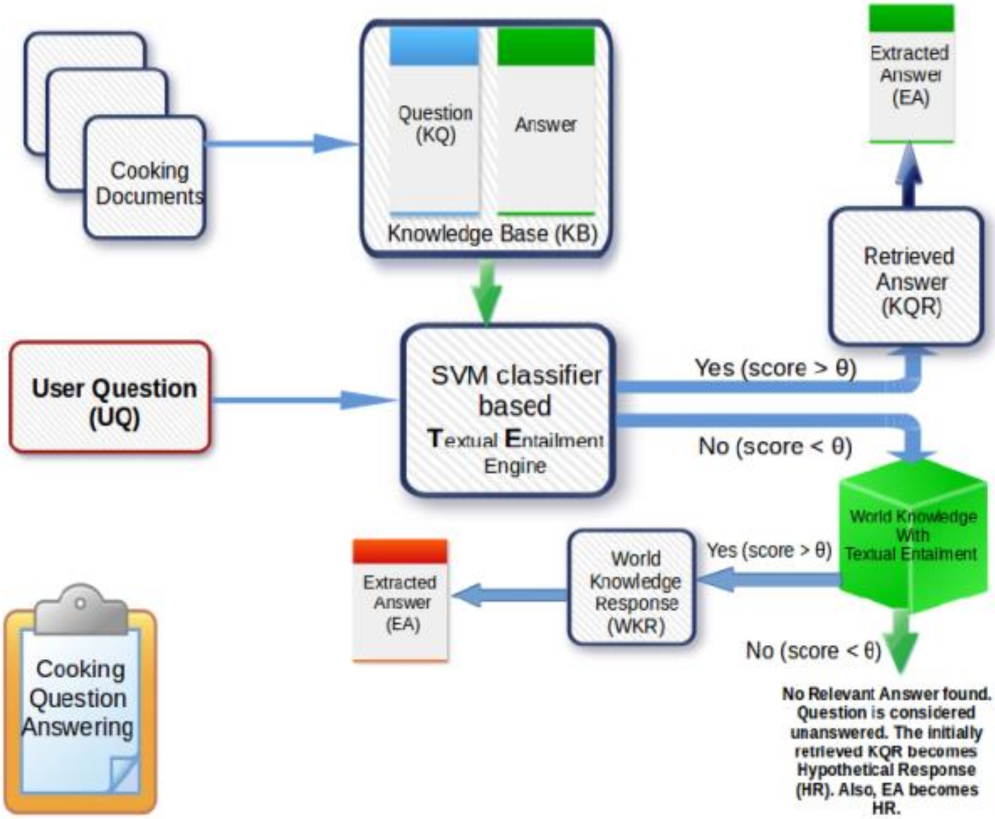
Cooking Documents
Question (KQ)
Answer
Knowledge Base (KB)
Extracted Answer (EA)
Retrieved Answer (KQR)
User Question (UQ)
SVM classifier based Textual Entailment Engine
Yes (score > θ)
No (score < θ)
World Knowledge With Textual Entailment
Yes (score > θ)
World Knowledge Response (WKR)
Extracted Answer (EA)
No (score < θ)
No Relevant Answer found. Question is considered unanswered. The initially retrieved KQR becomes Hypothetical Response (HR). Also, EA becomes HR.
Cooking Question Answering

Fig. 5. Text entailment based system architecture

shown in Table 7. 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' score is shown in Table 8.

5.3 Experiment 3

An automated system [9] using entailment for QA in cooking domain is a rather unexplored topic. It has been observed that the SVM classifier detects entailment between end user questions and the questions contained inside a Knowledge Base (KB), followed by retrieval of the answer corresponding to eminently entailed KB question.

Owing to unavailability of standard cooking dataset, and to promote further research on use of TE for QA in cooking domain, a KB comprising of tab-separated intriguing cooking questions and corresponding answers has been prepared. The KB contains 1223 instances, which are collected from authentic cooking websites, e.g., allrecipes.com, tarladalal.com, and blogs e.g., Cookbook, food52, etc. In addition, the test dataset, characterized by a mix of 50 easy and hard questions, has been authored independent of the instances in KB, which is shown in Table 9.

Based on whether the answer to a question is explicitly stated in the KB or the answer requires intricate reasoning, the test questions are designated "easy" and "hard" labels. The system architecture is shown in Figure 5.

The average Entailment Fraction (*Efrac*) measure for our system is 0.708.

Rather high value of *Efrac* designates that the KB contains questions similar to the user questions.

The accuracy measure is a fraction of the total questions whose answers are correct, partially correct or right.

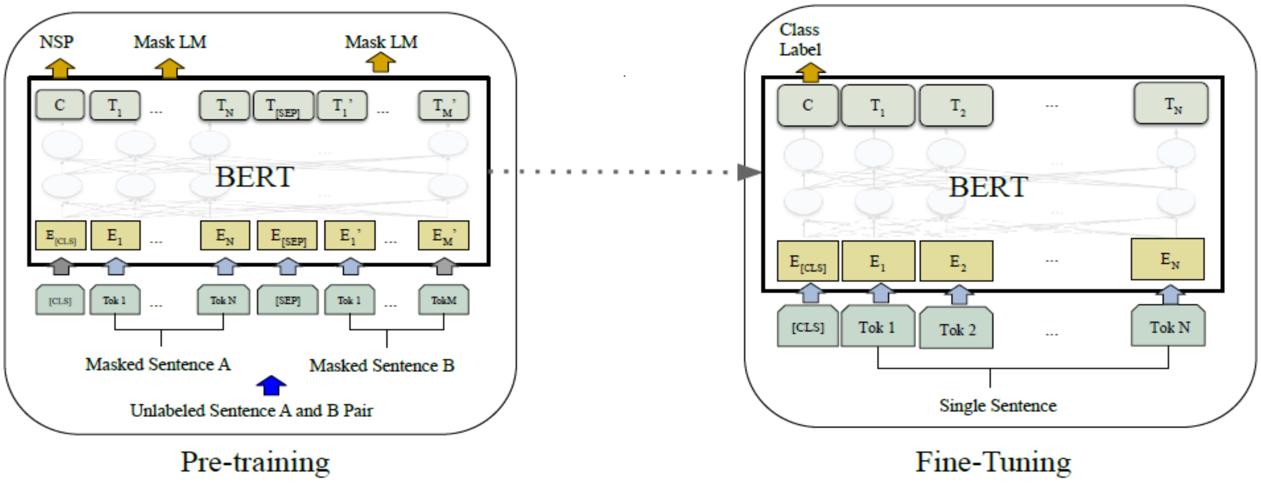

Fig. 6 Multi-classification architecture as adopted from the classification task of BERT

Average accuracy of our system is 0.588. c@1 measure accounts for unanswered questions while measuring the accuracy. Average c@1 score for our system is 0.506.

Here we give examples of system-generated output:

Example 1:

User Question: Help me to cook Amritsari paneer kulcha.

Knowledge based question: How to prepare Amritsari Kulcha?

Extracted Answer: Mix *Maida*, Sugar, Salt, Baking Powder, Baking Soda. Pour some oil in this mixture. Add Dahi (yoghurt) and kned the dough add water if necessary. The dought should not be very tight.

Discussion: Using entailment, the implemented system correctly infers that the user has requested steps of preparation for “Amritsari paneer kulcha”. System, therefore, provides the most relevant and correct answer.

Example 2:

User Question: What are the tips to cook vegetable pulav?

Knowledge based question: What is the cooking time for Vegetable Pulav?

Extracted Answer: 15 *Minutes*

Discussion: Though the KB *contains* a more relevant question (“*How to prepare Vegetable Pulav?*”) and its corresponding answer, system erroneously detects “*What is the cooking time for Vegetable Pulav?*” as the closest match for UQ. Thus, the failure to detect entailment between UQ and KB questions has led to retrieval of irrelevant/incorrect answers.

6 Recipe Recommendation

One of the aims of our recipe recommendation system [6] is to suggest recipes to the users based on their questions. We also used both the question classification (QC) and the question answering (QA) module. The QA module is used to extract the actual answers from the suggested recipe that is obtained from QC.

We used a recommendation approach by analysis of threshold estimation. The final result of the recommendation system gives benchmark results on the human evaluation (HE) metric. We scrapped the data from the cooking domain and obtained 87,730 unique recipes. For the supervised classification component, we used the dataset. The dataset contained 2175 cooking related questions and it was divided into 15 labeled classes, as shown in Table 10. Data preprocessing is required for the all recipes dataset for the BERT pre-training step for the QC task. It is also required to extract each individual ingredient from a recipe, which is shown in Figure 6.

For recommending the recipes to the users, we used a dedicated engine to suggest a suitable recipe from the available dataset. When the user asks any question as an input, the recommender system comes into force and suggests the user's different recipes with respect to the corresponding generated answer based on the QC and QA modules. For recommendation, we considered two main parameters i.e. the total number of matching occurrences and the total number of words present after stemming. The score was then calculated using the following equation:

$$S = \left(0.5 \times \frac{M}{n}\right) + 0.5\ .$$

The score is calculated based on whether the keyword is present in the recipe once or occurs frequently. Since the search-generated score is based on "and", it is able to recommend recipes based on similar words and similar-sounding words. The system gives the recommended answers on each question using our recommendation model both when the answer is the best fit and when it is not.

Thus, to clarify the wrong answers, we leveraged the help of human evaluators (HE). Based on the generated scores, a threshold is set based on human evaluation wherein they evaluate the answers but have no knowledge of the confidence score of that particular answer.

Based on user evaluation and the generated score, a threshold is set such that no answer below that threshold receives a bad evaluator rating.

7 Conclusion and Future Work

This paper demonstrates a Question Answering system in the cooking recipe domain, when our main focus is the contextual classification of recipe questions, answer retrieval and extraction, recipe recommendation.

The QA system has been performed using a state-of-the-art deep learning technique BERT for question classification and achieved remarkable performance and has shown good accuracy on the final system output based on the evaluation metrics considered, as well as good performance on answer retrieval and recipe recommendation.

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

Acknowledgments

The work was done with partial support to the third author from the Mexican Government through the grant A1-S-47854 of the CONACYT and grant 20211884 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The third author thanks CONACYT for the computer resources provided through the INAOE Supercomputing Laboratory's Deep Learning Platform for Language Technologies. The authors would also like to thank Mr. Amarnath Pathak, Mr. Sahinur Rahman Laskar and Mr. Abdullah Faiz Ur Rahman Khilji for providing the foundation, encouragement, and many fruitful discussions on this work.

References

1. **Yang, H., Chua, T.S. (2003).** QUALIFIER: Question answering by lexical fabric and external resources. Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03), pp. 363−370. DOI: 10.3115/1067807.1067855.
2. **Diekema, A.R., Yilmazel, Chen, J., Harwell, S., He, L., Liddy, E.D. (2003).** Finding answers to complex questions. **Maybury, M. (Ed.)** New directions in question answering. AAAI-MIT Press.
3. **Voorhees, E.M. (2001)**. The philosophy of information retrieval evaluation. Evaluation of Cross-Language Information Retrieval Systems. Proceedings of (CLEF´01), No. 2406, Lecture Notes in Computer Science, pp. 355–370.
4. **Gruber, T.R. (1993)**: A translation approach to portable ontology specifications. Knowledge Acquisition, Vol. 5, No. 2, pp. 199–220. DOI: 10.1006/knac.1993.1008.
5. **Manna, R., Pakray, P., Banerjee, S., Das, D., Gelbukh, A. (2016).** CookingQA: A question answering system based on cooking ontology. Lecture Notes in Artificial Intelligence, Vol. 10061. pp. 67–77.
6. **Khilji, A., Manna, R., Laskar, S., Pakray, P., Das, D., Bandyopadhyay, S., Gelbukh, A. (2020).** CookingQA: answering questions and recommending recipes based on ingredients. Arabian Journal for Science and Engineering.
7. **Manna, R., Das, D., Gelbukh, A. (2020).** Information retrieval-based question answering

system on foods and recipes. Lecture Notes in Computer Science, Vol. 12469, pp. 260−270.

8. **Manna, R., Das, D., Gelbukh, A. (2020)**. Question classification in a question answering system on cooking. Lecture Notes in Computer Science, Vol. 12469, pp. 103−108.
9. **Pathak, A., Manna, R., Pakray, P., Das, D., Gelbukh, A., Bandyopadhyay, S. (2020).** Scientific text entailment and a textual entailment based framework for cooking domain question answering. Sādhanā, No. 24. DOI: 10.1007/s12046-021-01557-9.
10. **Khilji, A., Manna, R., Laskar, S., Pakray, P., Das, D., Bandyopadhyay, S., Gelbukh. A. (2020).** Question classification and answer extraction for developing a cooking QA system. Computación y Sistemas, Vol. 24, No. 2. DOI: 10.13053/CyS-24-2-3445.
11. **Research, S. (2017)**. Protege.stanford.edu. http://protege.stanford.edu.
12. **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.
13. **Chen, J.J., Ngo, C.W., Feng, F.L., Chua, T.S. (2018).** Deep understanding of cooking procedure for cross-modal recipe retrieval. Proceedings of the 26th ACM international conference on Multimedia, pp. 1020−1028. DOI: 10.1145/3240508.3240627.
14. **Veron, M., Peñas, A., Echegoyen, G., Banerjee, S., Ghannay, S., Rosset, S. (2020).** A cooking knowledge graph and benchmark for question answering evaluation in lifelong learning scenarios. International Conference on Applications of Natural Language to Information Systems, pp. 94−101. Springer, Cham.
15. **Lukovnikov, D., Fischer, A., Lehmann, J., Auer, S. (2017).** Neural network-based question answering over knowledge graphs on word and character level. Proceedings of the 26th International Conference on World Wide Web, pp. 1211−1220. DOI: 10.1145/3038912. 3052675.
16. **Zeyen, C., Müller, G., Bergmann, R. (2017).** Conversational retrieval of cooking recipes. Computer Science, (ICCBR´17), pp. 237−244.
17. **Xia, L. (2014).** Answer planning based answer generation for cooking question answering system. Journal of Chemical and Pharmaceutical Research, Vol. 6, No. 7, pp. 474−480.
18. **Yagcioglu, S., Erdem, A., Erdem, E., Ikizler-Cinbis, N. (2018).** RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP´18), pp. 1358–1368. DOI: 10.18653/v1/D18-1166.
19. **Jermsurawong, J., Habash, N. (2015).** Predicting the structure of cooking recipes. Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP´15), pp. 781–786. DOI: 10.18653/v1/D15-1090.
20. **Malmaud, J., Wagner, E., Chang, N., Murphy, K. (2014).** Cooking with semantics. Proceedings of the Workshop on Semantic Parsing (ACL´14), pp. 33–38. DOI: 10.3115/v1/W14-2407.
21. **Mino, Y., Kobayashi, I. (2009).** Recipe recommendation for a diet considering a user’s schedule and the balance of nourishment. IEEE International Conference on Intelligent Computing and Intelligent Systems, Vol. 3, pp. 383–387. DOI: 10.1109/ICICISYS.2009.5358168.
22. **Freyne, J., Berkovsky, S. (2010).** Intelligent food planning: personalized recipe recommendation. Proceedings of the 15th International Conference on Intelligent User Interfaces, (IUI´10), pp. 321–324. DOI: 10.1145/1719970.1720021
23. **Müller, M., Harvey, M., Elsweiler, D., Mika, S. (2012).** Ingredient matching to determine the nutritional properties of internet-sourced recipes. International Conference on Pervasive Computing Technologies for Healthcare, Pervasive Health 2012 and Workshops, pp. 73–80. DOI: 10.4108/icst.pervasivehealth.2012.248681.
24. **Xia, L., Teng, Z., Ren, F. (2009).** Answer generation for Chinese cuisine QA system. International Conference on Natural Language Processing and Knowledge Engineering, pp. 1−6. DOI: 10.1109/NLPKE.2009.5313813.

Article received on 07/09/2020; accepted on 27/11/2020.
Corresponding author is Alexander Gelbukh.