Question Classification and Answering from Procedural Text in English

Somnath Banerjee   Sivaji Bandyopadhyay
Department of Computer Science and Engineering
Jadavpur University, India
s.banerjee1980@gmail.com, sivaji_cse_ju@yahoo.com

ABSTRACT

Linguistic patterns reflect the regularities of Natural Language and the applicability of such linguistic patterns is acknowledged in several Natural Language Processing tasks. Many question classification systems depend on patterns that are extracted from already framed questions. In this paper, we have investigated possible question categories and question patterns for procedural text documents in English and proposed seven question classes. More than six thousands questions of different domains, e.g., cooking recipes, electronics, home and maintenance, medical etc have been collected from Yahoo answers as experimentation corpus. Annotators reached almost perfect agreement of 94.6% at kappa scale. A procedural question answering system has been developed to verify the proposed question classes. The evaluation reveals that the proposed classes are a good approach to deal with Question Answering for procedural text questions. The procedural question answering system has achieved overall 95.08%, 86.95% and 90.84 precision, recall and F-measure value respectively.

KEYWORDS: Question Answering, Question Classification, Procedural Text

Proceedings of the Workshop on Question Answering for Complex Domains, pages 11–26,
COLING 2012, Mumbai, December 2012.

1 Introduction

Automated question answering (QA) has been a hot topic of research and development since the earliest AI applications (Turing, 1950). Many international question answering evaluation tracks have taken place at conferences and workshops, such as TREC[1], CLEF[2], and NTCIR[3] to improve question-answering systems. An important component of question answering systems is question classification. The task of a question classifier is to assign one or more class labels, depending on classification strategy, to a given question written in natural language. For example, for the question "What is the capital of India?", the task of question classification is to assign label "Location" to this question. Since we predict the type of the answer, question classification is also referred as answer type prediction. Common classification strategies include semantic categorization and surface patterns identification.

Surface pattern identification methods classify questions to sets of word-based patterns. Answers are then extracted from retrieved documents using these patterns. Without the help of external knowledge, surface pattern methods suffer from limited ability to exclude answers that are in irrelevant semantic classes, especially when using smaller or heterogeneous corpora. The amount of supported classification types greatly influences the performance of QA systems.

Question classification has been studied by using different type of classifiers. Most of the successful studies on this task used support vector machines (SVM) (Zhang and Lee, 2003; Huang et al., 2008; Silva et al., 2011; Loni et al., 2011). SVMs are very successful on high dimensional data since they are more efficient especially when the feature vectors are sparse. Question classification has also been done by Maximum Entropy models (Huang et al., 2008; Blunsom et al., 2006), Sparse Network of Winnows (SNoW) (Li and Roth, 2004) and language modeling (Merkel and Klakow, 2007).

As per Wikipedia (*en.wikipedia.org*), the term *procedure* is being used in diverse domains with different meanings-

- *Organization*: A *procedure* is a document written to support a "Policy Directive".
- *Medical:* A *procedure* is a course of action intended to achieve a result in the care of persons with health problems.
- *Mathematics and Computing*: A *procedure* is a set of operations or calculations that accomplish some goal.
- *Cooking*: A *procedure* is a set of commands that show how to prepare or make something.
- *Industry and Military*: A *procedure* is a step-by-step instruction to achieve a desired result.
- *Legal:* A *procedure* is the law and rules used in the administration of justice in the court system.
- *Computer science*: A *procedure* is a part of a larger computer program that performs a specific task.

So, in general a *procedure* is a specified series of actions or operations or a set of commands which have to be executed in order to obtain a goal. Less precisely speaking, the word

[1] http://www.trec.com
[2] http://www.clef.com
[3] http://www.ntcir.com

‘procedure’ can indicate a sequence of activities, task, steps, decisions, calculations and processes, that when undertaken in the sequence laid down produces the described results, product or outcome. So, procedural texts consist of a sequence of instructions in order to reach a goal and range from apparently simple cooking recipes to large maintenance manuals. They also include documents as diverse as teaching texts, medical notices, social behaviour recommendations, directions for use, assembly notices, do-it-yourself notices, itinerary guides, advice texts, savoir-faire guides etc (Aouladomar and Saint-Dizier, 2005). So, the questions of procedural text are as diverse as its range of diversity. In our perspective, procedural questions will be of much growing interest to the non-technical as well as technical staff. Statistics also showed that procedural questions is the second largest set of queries formed to web search engines after factoid questions (de Rijke, 2005). This is confirmed by another detailed study carried out by (Yin, 2004).

While the first QA systems (Simmons, 1965) mainly dealt with factoid questions, a number of systems in the last decade have appeared with the aim of addressing non-factoid questions (E. M. Voorhees. 2003). Procedural questions, sometimes called ’How-questions’, are questions whose induced response is typically a fragment, more or less large, of a procedure, i.e., a set of coherent instructions designed to reach a goal. Answering procedural questions thus requires being able to extract well-formed text structure unlike factoid question and analyzing a procedural text requires a dedicated discourse analysis, e.g. by means of a grammar (Delpech et al., 2008). Though less research has been conducted so far on other types of non-factoid QA, such as why-questions (Verberne et al.,2007; Pechsiri et al,2008) and procedural (how-to) questions (Yin, 2006; Delpech et al., 2008), during the last decade challenges of procedural text and argument extraction have been addressed (Fontan et al., 2008; Adam et al., 2008).

In this work, we have focused on question classification and answering from the procedural text in English and building a generic domain independent procedural question answering system.

The remainder of the paper is organized as follows: in the next section, we review the related works. Corpus preparation and system description are elaborated in third section and fourth section respectively; Corpus for procedural text and evaluation are described in fifth section and sixth section respectively; and finally seventh section describes the conclusions of our study and outlines directions of our future work.

2 Related Work

Question classification in TREC QA has been intensively studied during the past decade. Many researchers have employed machine learning methods (e.g., maximum entropy and support vector machine) by using different features, such as syntactic features (Zhang et al., 2003; Nguyen et al, 2008) and semantic features (Moschitti et al, 2007). However, these methods mainly focused on factoid questions and confined themselves to classify a question into two or a few predefined categories (e.g., "what", "how", "why", "when", "where" and so on). However, question classification in procedural text is dramatically different from factoid question classification. Therefore, traditional methods may fail to achieve the satisfactory results.

Research on procedural texts was initiated by works in psychology, cognitive ergonomics, and didactics, (Mortara et al., 1988), (Greimas, 1983), (Kosseim, 2000) to cite just a few. The issues of title identification, tagging and reconstruction via a learning mechanism in a large variety of types of procedural texts have been addressed (Adam et al., 2008). A way to retrieve the missing elements in particular predicates for incomplete title has also been proposed. The conceptual notion of instructional compounds, recognition of titles, instructions and instructional compounds has been focused by Delpech et al., 2008. A simple text grammar system that accounts for the overall text structure with respect to procedures has also been modelled and implemented. They also identified that the complexity of annotations make the task much more difficult and proposed that design domain dedicated recognizers with specific patterns might improve the low level instruction recognition results for particular domain.

The challenges of answering procedural questions from procedural text have been investigated (Saint-Dizier P, 2008) and procedural title identification and tagging, instructions and instruction arguments have also been investigated and processed. Parsing and analyzing argumentative structures in procedural texts have been addressed successfully (Fontan et al., 2008). A conceptual categorization of procedural questions based on verb categories has also been addressed for French (Aouladomar et al., 2005).Also, identification of advice and warning structures from procedural texts has been investigated (Fontan and Saint-Dizier, 2008). A quite large corpus (about 1700 texts) from several domains (basic: cooking, do it yourself, gardening, and complex: social relations, health) and a large number of web sites have been constructed for experiment and it has been found that warnings are basically organized around an ‘avoid expression’ combined with a proposition.

During the last decade, a number of researches have been done on addressing procedural text structure for various domains. But, those investigations were only carried out for French language and unfortunately, so far fewer researches have been carried out for classifying procedural text questions in any language.

3 Corpus Preparation

3.1 Corpus Collection

To our knowledge, similar to *procedural text* no standard corpora for English *procedural questions* are available for research. So, we had no choice to use any standard data and we had to prepare experimental data for our own. Due to broad coverage and authenticity, we have selected *Yahoo Answers*[4] for data collection. More than six thousand questions (6,230) of four different domains (cooking recipes, electronics, home and maintenance, medical) from *Yahoo Answers* have been collected and approximately six thousand questions (6,081) have been identified as valid procedural questions under human supervision. This rigorous manual work took almost 32 hours. The rejected questions were either not formed grammatically correct or posted in wrong category. Out of 6,081 identified valid questions, 4257 questions (70%) of the tagged corpus has been investigated to identify the patterns for proposed questions and rest 1824 questions (30%) corpus has been used to verify the identified patterns.

Domain	**$Q_{training}$**	**Q_{test}**
Cooking Recipe	1162	498
Electronics	1146	491
Home and Maintenance	966	414
Medical	983	421

Table1: Statistics of Procedural Questions in Corpus

[4] http://answers.yahoo.com

3.2 Pre-annotation

Collected questions have been POS tagged for the initial work of corpus preparation. Stanford Parser (Toutanova et al., 2003) has been used as the POS tagger. Then, Stanford named (Finkel et al., 2005) entity recognition (NER) tagger has also been used to identify named entities.

3.3 Annotation

Eleven patterns have been identified and used by the two human annotators. The inter-annotator agreement on question annotation has been measured by kappa statistics. The identified patterns are shown in the table below.

Rules	Patterns	Category	Kappa-Statistics
R1	<WH><Prerequisites><X>	PA	87.70%
R2	<WH><ITEM><VPP><X>	PA	89.60%
R3	<WH><ITEM><VPP><X><NUMBER>	PA	93.40%
R4	<WH> TO <*GOAL*>	DA	92.87%
R5	<WH><V><STEP> TO <GOAL>	DA	88.76%
R6	<WH> <Special Information> <X>	SpIA	90.90%
R7	<WH><ACTION><X>	JA	95.70%
$R7^+$	<WH><V><NP><X>	JA	95.70%
R8	<WH><ADV VERB><*X*>	AA	91.70%
R9	<WH><PREF VERB><*X*>	AA	94.50%
R10	<WH><WARN VERB><X>	WA	93.70%
R11	<WH><PREV VERB><X>	WA	94.60%
R12	<WH><ACTION VERB><ITEM><X>?	SIA	92.97%

Table2: Question Classification Patterns

3.3.1 Proposed Question Types Description and Identification

The objectives of question answering (QA) systems is to take a user's question of an information need expressed in natural language and seek an answer from the document collection. If a QA system is to answer questions accurately, it must accurately classify the question. The reason is intuitive: a question contains all the information to retrieve the answer. The question patterns have the ability of deciding the question type. We have proposed the following seven question classes for procedural text:

- Prerequisites Associated (PA)
- Direction Associated (DA)
- Extra or Special Information Associated (SpIA)
- Justification Associated (JA)
- Advice Associated (AA)
- Warning Associated (WA)
- Simple Instruction Associated (SIA)

Prerequisites Associated (PA) Questions Identification**:** Every procedure needs to meet some criteria in advance or collect some ingredients to follow the instructions. These pre-criteria or ingredients are called prerequisites for a procedure. Every procedural text contains some prerequisites to follow the instructions. So, there should be a prerequisites question for a procedure. For example, in cooking recipe ingredients are the prerequisites; in Voter I-Card

Application procedure the voter should be the citizen of that nation is the prerequisite; in changing a wheel of a car puncture repair kit, e.g., needed tools are the prerequisites. So, prerequisites describe all kinds of equipments needed to realize the action and preparatory actions. Generally this type of question appears in pattern "[what|which] are the <Prerequisites> for <X>?" Where, Prerequisites= "criteria" or "ingredients" or "tools" and X= "goal or sub-goal". For example, "what are the ingredients for cooking chilly chicken?"; "What are the criterion for making Voter I-Card?"; "What are the tools for changing wheel of a car?" . So, for this type of questions the following general pattern may be considered-
(i) R1: <WH><Prerequisites><X>

(ii)R2: <WH><ITEM><VPP><X>

(iii)R3: <WH><ITEM><VPP><X><NUMBER>

For example, "*What are the ingredients for cooking chilly chicken?"*
Where, WH= "what", Prerequisites= "ingredients" ,X= "cooking chilly chicken*"*
"What are the criterions for making Voter I-Card?"
Where, WH= "what",Prerequisites= "criterions", X= "making Voter I-Card*"*
"What are the tools for changing wheel of a car?"
Where, WH= "what", Prerequisites= "tools",X= "changing wheel of a car*"*
"*how much oil is required/needed to cook/prepare/make chilly chicken?*"
Where, WH= "how much", ITEM= "oil", VPP= "is required", X= "cook chilly chicken"
"*how much oil is required/needed to cook/prepare/make chilly chicken for three heads?*"
Where, WH= "how much", ITEM= "oil", VPP= "is required", X= "cook chilly chicken",
NUMBER= "three heads"

Direction Associated (DA) Questions Identification: Every procedure is an ordered set of instructions, Proc(X) = { I_1,I_2,I_3…I_n };where X=procedure name, I_i=i^{th} instruction in the instruction set. So, user query may be on the ordered instructions associated in procedural text to reach the goal. Questions of this type appear in the pattern:

R4: <WH> TO <GOAL>

R5: <WH> <STEP> TO <GOAL>

Where, *GOAL* = "ACTION VERB" + "NOUN PHRASE"

For example, *How to prepare tea?*
Where, GOAL ="prepare tea"; ACTION VERB="prepare", NOUN PHRASE="tea"
How to assemble a computer?
What are the steps to assemble a computer?
Where, GOAL="assemble a computer",
ACTION VERB= "assemble", NOUN PHRASE="a computer"

So, the general pattern may be - <WH><GOAL> which implies: <WH><V><NP> (R4 and R5 can be generalized to R45)

Special Instruction Associated (SIA) Questions Identification: A procedural text often contains some optional information that may be very useful to the reader. This information serves a special role in the procedure. So, this information is considered as special information. For example, in cooking recipe "*preparation time*", "*cooking time*", "*servings*", "*serve with*" are the extra or special information which give the reader valuable information. In do-it-yourself domain "*difficulty*", "*time required*", "*cost*" are the extra information. Often procedural text contains one or more tips that may be helpful to the performer. For example, in cooking recipe "*serve hot with rice*" gives serving instruction to the cook. The writer of procedural text may or may not provide this sort of information. So, question may be formed to retrieve this type of information. For example, "*how long it will take to prepare tea?*" It is quite possible to find common patterns for a particular domain, but it is very difficult to form any domain independent general pattern for this type of question because special information and its questions patterns are very much domain dependent. For cooking recipe, the pattern *R6*: <WH> <Special Information> <X> may be used to classify the questions-"what is the *preparation time* for cooking fish fry?", "what is the *cooking time* for cooking fish fry?". "How long does it take to defrost a 20lb turkey?", e.g. <how long><V=defrost><X>;

Justification Associated (JA) Questions Identification: An instruction in the form: A_j because S_j, means an action instruction Aj paired with a support S_j that stresses the importance of A_j (Fontan and Saint-Dizier, 2008). For example, "Add about three cups of chilled water *to adjust the consistency*." "Carefully plug in your mother card vertically; *otherwise you will damage the connectors*." In these sort of instructions, the support part justifies the action part. So, this sort of instruction justifies an action to the performer. This information provides the performer the outcome or the risk factor of the action associated with the procedure. Justification associated question may be formed in the pattern-"Why to <*Action*> in <*X*>?" ;Where, ACTION="ACTION VERB" + "NOUN PHRASE", *X*="Procedure name"

For example, Why to *add three cups of child water* in cooking rice?; Why to *add child water* in cooking rice?; Why to *add water* in cooking rice?

In three example questions above, "three cups of chilled water", "add child water" and "add water" are the ACTION respectively, where "*add*" is the action verb for all examples and "*three cups of child water*", "*child water*" and "*water*" are the NOUN PHRASEes respectively. So, the general pattern can be-

R7: <WH><ACTION><X>; we can derive from R7 that $R7^+$: <WH><V><NP><X>.

Advice Associated (AA) Questions Identification: Procedural text also contains some advice or suggestion instructions. This instruction is identified by preference expression which may be a verb, e.g. prefer or an expression, e.g. "is advised to", "it is better", "preferable to", etc. For example, "C*ook on low heat till the rice gets heated through*", "*Cook for 4-5 minutes or till the spinach is soft*", "*choose specialized products dedicated to furniture*". These instructions should follow for better outcome of the procedure. So, a query could be: *R8:* <WH><ADV VERB><*X*>? and R9*:* <WH><PREF VERB><*X*>?

Where, ADV VERB=Advice or
PREF VERB= Preference verb e.g. "suggestion", "preferable", "recommendation";
X= "Procedure name"
For example, "*what* are the *suggestion*s for *cooking chilly chicken*?"

Warning Associated (WA) Questions Identification: Procedural text often contains some action instructions that must be followed carefully to reach the goal or to avoid risk factors. Warnings are basically organized around a unique structure composed of an 'avoid expression' combined with a proposition (Fontan and Saint-Dizier, 2008). The propositions are identified by various marks- via connectors, e.g. otherwise, under the risk of, etc.; via negative expressions, e.g. in order not to, in order to avoid, etc.; via risk verbs e.g. break; via negative terms, e.g. death, disease, etc. The outcome of the procedure highly depends on this action instruction and unsuccessful action may lead to damage or harm. For example, "*Carefully plug in your mother card vertically*, *otherwise you will damage the connector*".

So, performer of a procedure pay much attention about this type of instructions and forms query: "What are the warnings for <*X*>?" or "What are the instructions must follow for <*X*>?"; where *X*= "procedure name" . The pattern could be- *R10:* <WH><WARN/PREV VERB><X> and *R11:* <WH><PREV VERB><X>

Where, WARN VERB=Warning; PREV VERB= Prevention verb, e.g., "risk", "avoid", "damage" etc., *X*= "Procedure name"

Simple Instruction (SI) Associated Questions Identification: More often an instruction has no support and considered as simple instruction or instruction with empty support (Delpech and Saint-Dizier, 2008). For example, "*Add the chili powder, salt and tomatoes.*", "*Heat oil in pan, fry the onions and green chilies.*"

So, queries on these action instructions are aimed to extract the timing of action. For example, in cooking recipe, the query could be *"When to add chili powder in cooking chilly chicken?*

Most of the cases, the answer may be after completion of the preceding action instruction or before completion of the following action instruction. So, the query of this type often is in the form: *R12:* <WH><ACTION VERB><ITEM><X>?

Where, ACTION VERB= Action verb, e.g. "do", "perform", "add", "start" etc., ITEM= an item e.g., ingredient, tool, criteria etc., X= "procedure name".

4 System Description

We also built a QA system to verify our proposed question classes and identified patterns. This involves storing procedures from procedural web page collected from the web. System description includes storing procedures, question classification and answer extraction.

4.1 Building Repository for Procedure

4.1.1 Title and Keyword Extraction

This process involves extraction of the title of the procedural text, prepares a list of valid keywords. Title of the recipe is determined in three phases.

First phase, extracts the title of the web file (xml, html) included in the TITLE tag. *Second phase,* extracts the text enclosed within the <Hn> tag {where n=1, 2, 3}. In the *final phase,* the extracted title texts (1st and 2nd phase) are compared. If they are matched, then one of them is taken as title, otherwise the most relevant title is taken. It has been observed from experiment that most the most relevant title is found in the first phase. Two strategies are used to determine title relevancy. In the first strategy, number of words i.e. length of the title text is used as relevancy parameter.

Title text with less than 10 words is considered as valid title. In contrast, second strategy uses stop list (e.g. click, see, buy, recommendation, advice etc) of 100 words to reject a title text as invalid title. The system uses both strategies to validate title text. The strategies are included in the present work after manual experimentation on 200 documents of the development set.

If the *title text* is “a_1 a_2 a_3 …… a_n”, then the *keyword list* will be {“a_1 a_2 a_3 ….a_n”, “a_1”, “a_2”…“a_n”} means that the title text with each word appears in title text. As the title text may contain preposition (e.g., the, at etc.) and some words (e.g., com, www etc.) that cannot be considered as keyword, so each keyword is considered as a valid keyword after verifying with ***stop word list***. So, the maximum size of the keyword list is n+1 if the title text contains n words and it may be less than maximum size if the title text contains invalid keywords (*stop words*). This keyword list will be processed in the next step to generate inverted index for searching.

4.1.2 Constructing Procedure Structure

The basic idea of data organization in the document is taken from (Fontan et al., 2008). Also, additional idea is introduced in data organization in order to meet the requirements of the designed system. Each identified relevant document is stored according to the structure depicted in FIG-1.Each tag term used in the structure are described below-

```
<Procedure ID= Proc_id>
<title> title of the procedure </title>
<keyword> title, each valid word in title </keyword>
<prerequisites> prerequisites list </prerequisites>
<method>
    <instructional-compound>
        <instruction> simple instruction <support> support text</support></instruction>
        <advice> advice instruction <support> support text </support></advice>
        <warning> warning instruction <support> support text</support></warning>
    </instructional-compound>
    <instructional-compound>
     .
     .
   </instructional-compound>
    .
    .
    .
   </method>
```

Fig-1: Procedure Structure

Proc ID: The system needs a unique identification number to distinguish each procedure. So, each procedure is assigned a unique integer value by the system in the first sub-module of this module.

Title: Every procedure has a name which suggests what to achieve or what to produce. For example, in recipe domain “Egg Roll”, the title text describes that the step by step instruction will produce Egg roll.

Keywords: Each keyword of a procedure relates that procedure to another procedure in terms of some common matter. For example, “Chicken Roll” and “Chicken Kasa” are different preparation of item “Chicken”. The “Chicken” keyword in both titles relates two procedures and describes that both item are prepared using the item “Chicken”.

Prerequisites: Every procedure needs to meet some criteria in advance or needs ingredients to follow the instructions in order to reach a goal or sub-goals. These pre-criteria or ingredients are called prerequisites for a procedure. Every procedural text contains some pre-requirements to follow the instructions. So, the text attached with this tag describes the pre-criteria or ingredients for the describing procedure.

Method: Every procedure is an ordered set of instructions. So, to reach the goal those instructions must be processed in the prescribed order. The ordered instructions are described within the scope of this tag.

Instructional-Compound: Each sentence in the describing procedure is considered as an instructional-compound. So, a method is composed of instructional-compounds. An instructional-compound may contain a single or multiple instructions. The type of the instructions may be of three types- (i) Simple instruction with or without support, (ii) Advice instruction with or without support (iii) Warning instruction with or without support

Support: An instruction may be in form: Aj because Sj, which means an action instruction Aj paired with a support Sj stresses the importance of *Aj* (Fontan et al., 2008). For example, “Add about three cups of chilled water *to adjust the consistency*.” “Carefully plug in your mother card vertically; *otherwise you will damage the connectors*.” This sort of instructions, one part justifies the other part action. This type of instructions Sj is considered as support instruction. Support instruction may appear with simple instructions or advice or warning instructions.

Instruction: A simple instruction is stored within the instruction tag. It may or may not contain support instruction.

Advice: Often an instruction expresses an advice, suggestion or preference. For example, “Cook on low heat till the rice gets heated through”, “Cook for 4-5 minutes or till the spinach is soft”, “You should better let a 10 cm interval between the wall and the lattice”. The advice, suggestion or preference expressions are considered as advice instruction and included within the scope of advice tag. Sometimes, an advice instruction is justified with a support part. So, an advice instruction may contain support instruction.

Warning: Procedural text often contains some action instructions that must be followed carefully to reach the goal or to avoid risk factors. For example, “Carefully plug in your mother card vertically, *otherwise you will damage the connectors*.” These instructions are considered as warning instructions and included within the scope of the warning tag. In the said example instruction, the action is justified with a support instruction. So, a warning instruction may contain support instruction.

4.1.3 Instruction Categorization

Initially, all the instructions within *Instructional-Compound* are considered as simple instructions. We need to process each instruction text in order to achieve three categories described above. Three lists of cue words and phrases have been prepared manually and those are used to check each instruction. For examples,

Advice List: {*if needed, at least, if necessary, so that, allow, better… etc.*};
Warning List: {*should be, do not, must… etc.*};

Advice list and warning list have been used to separate simple instruction, advice and warning instructions. For examples,

Simple Instruction:

Advice: <advice>Cook on low heat ***till*** the rice gets heated through. *</advice>*

Warning: <warning>The paranthas ***should be*** as thin as a papad. </warning>

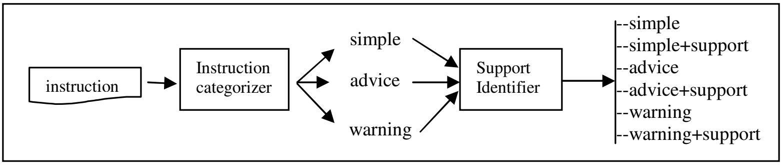

Fig 2: Processing of Instructions

Then support list has been applied to check whether the instruction includes support or not. The support portion appears in the instruction after the support list phrase. Then the input instruction is tagged properly. Support List: {*to make, to adjust, to remove… etc.*}

For example,

Instruction + Support (justification): <instruction>squeeze <support> ***to remove*** *all the oil.* </support> </instruction>

Advice + Support (justification): <advice> Add about three cups of chilled water <support> ***to adjust*** *the consistency*. </support></advice>

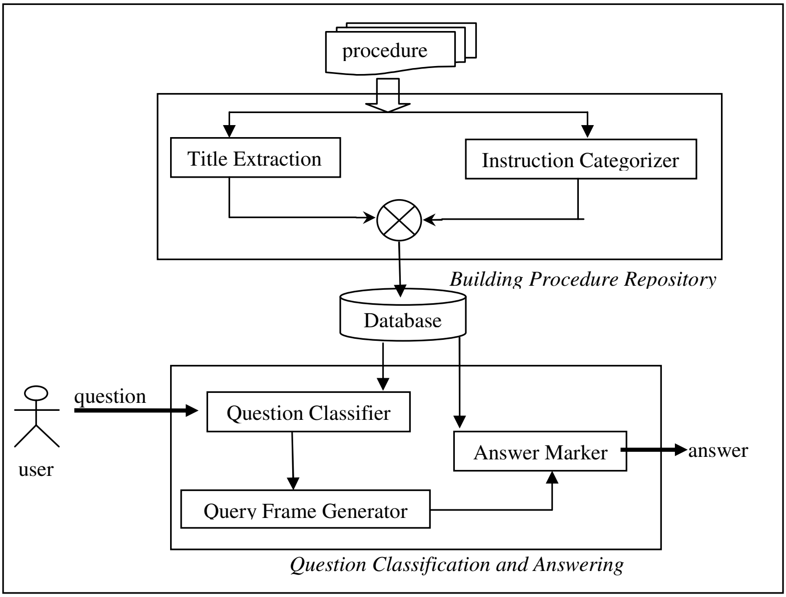

Fig 3: System Diagram

The three lists have been prepared after manually tagging 200 procedural documents. It has been observed that advice and warning cue words and phrases contains *modal verbs* (e.g., can, could, may, might, must, ought to, should, would etc.) as well as not modal verbs (e.g., had better**,** have to, have got to). It has been also observed that support list elements are *infinitives (to adjust, to remove etc.).*

So, if we know the syntactic structure of a language, then the model may support that language with minimal changes in the lists (Advice List, Warning List, and Support List).

If we consider the recipe domain, then prerequisites are the ingredients for the recipe. So, prerequisites list contains item with quantity. In web page they appear under ingredients header with pattern [<no>] <item> [:] <quantity> OR [<no>] <quantity> <item>, where [ ] denotes optional pattern. They can be easily extracted from the web documents.
<prerequisites> (1)5 to 5 1/2 cups flour (2)1/2 cup sugar ... </prerequisites>
<prerequisites> (1)Maida : 500 gms (2)Oil : 200 gms ... </prerequisites>

The *method tag* contains the instructions to prepare recipe. They appear in the web page under Instruction/Direction/How to make <recipe title> header. The instructions in this domain also can be identified by the instruction categorizer using the manually list of cue words and phrases. So, Advice, warning and support lists are used for recipe domain to check each sentence.

4.2 Question Classification and Answering

User forms the natural language question and submits to the system via an interface. Question Classifier module classifies the question according to the proposed question classes. This question answering system generally does not need all the information from the user input query, so a partial or shallow parsing of the input sentence is more accurate and more robust than deep or full parsing. Shallow parsing provides the structural basis for natural language questions. In the describing QA system, shallow parsing technique has been used at the syntactic level and not at the semantic level. Swallow parser generates a query parse tree (QPT) for the input question using the algorithm below-

```
Algorithm ParseQuestion(NL question)

    Step1: node1:=find_WH( )
    Step2: If ( find_Preq( ) !=null)
               node2=find_Preq( ) and go to Step3
               If (find_Action() !=null)
                       node2:=find_Action()
               If ( find_Verb( ) !=null)
                       node21:=find_Verb()
               If ( find_NP( ) !=null)
                       node22:=find_NP()
               Else
                       node2:=null
    Step3: If (find_X( ) !=null)
                      node3:=find_X( )
               Else
                      node3:=null
    Step4: Stop
```

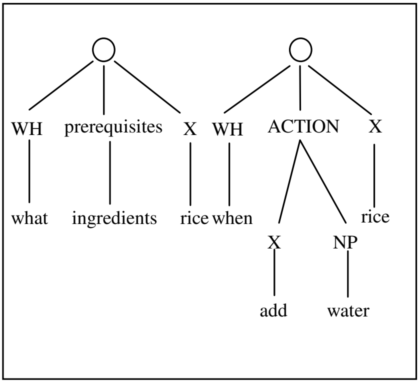

Fig-4: QPT Construction

The QPT is used to generate intermediate question pattern. This question pattern is used to classify the question according to proposed question classes. Fig-4 shows the parsing tree for the questions: "*what are the ingredients for cooking rice?*" and "*when to add water in cooking rice?*".

Now, question pattern information is used to retrieve the documents along with the answer. For example, if question class is identified as PA question, then prerequisites tag has been marked for identified procedure.

1 Corpus for Procedural Text

We have collected 50 cooking recipes from the BBC recipe website[5], 50 electronics maintenance instructions from eHow[6]. 50 home and maintenance procedures from Home Repair[7],and 50 medical procedure descriptions from Health.Com[8]. The instructions in the home and maintenance domain are more complicated since they often involve multiple sub-procedures. For simplicity procedures with sub-procedures have not been taken. On average, each procedure contains approx 9 instructional-compounds, approx 6 simple instructions, approx 1 warning instructions and approx 2 advice instructions. Each instructional-compound is containing an average of 13 tokens (e.g., words and symbols separated by spaces).

In order to evaluate the automatic extraction system, we ask human annotators to create a gold standard against which the automatically generated content is compared. Since the system automatically identifies the instructions and classifies them into the one of the three categories (simple instruction, advice instruction and warning instruction) with or without support instruction described in the system description section, human annotators are requested to do the same by annotating the instructions using an annotation tool. For each domain, three annotators are invited to perform the task, and a subset (25%) of the corpus is used for studying Inter-Annotator-Agreement following the approach in Hripcsak and Rothschild (2005).

Domain	Instructional Compound	Simple Instruction	Advice Instruction	Warning Instruction
Cooking Recipe	510	360	102	48
Electronics	460	312	98	50
Home & Maintenance	446	281	112	53
Medical	386	230	105	51

Table 3: Procedure statistics for different domains

6 Evaluations

The evaluation set composed of 1824 questions over four domains: cooking recipes, electronics, maintenance and medical procedure. Though the test set is not very large, but it is sufficient for inductive evaluation.

We have used standard evaluation metrics precision, recall and F-measure.

[5] http://www.bbc.co.uk/food/recipes/
[6] http://www.ehow.com
[7] http://homerepair.about.com
[8] http://www.health.com

$$\text{Recall(R)} = \frac{\textit{number of questions classified correctly}}{\textit{total number of questions}}$$

$$\text{Precision (P)} = \frac{\textit{number of questions classified correctly}}{\text{number of questions classified by the system}}$$

$$\text{F-measure} = \frac{2PR}{P+R}\text{; where, } \beta = 1$$

Fig-5: Evaluation Metrics

Out of 1824 test questions, 1586 questions have been classified correctly of 1668 classified questions. Overall 95.08%, 86.95% and 90.84 are the precision, recall and F-measure value respectively. Table-4 and Table-5 show the statistics for cooking recipe, electronics, home and maintenance, and medical domains respectively.

Home and Maintenance								Medical						
Class	TQ	C	CC	P(%)	R(%)	F		**Class**	TQ	C	CC	P(%)	R(%)	F
PA	60	56	53	94.64	88.33	91.38		**PA**	75	71	69	97.18	92.00	94.52
DA	88	85	82	96.47	93.18	94.80		**DA**	80	73	70	95.89	87.50	91.50
SpIA	60	52	49	94.23	81.67	87.50		**SpIA**	54	49	46	93.88	85.19	89.32
JA	56	51	49	96.08	87.50	91.59		**JA**	66	61	58	95.08	87.88	91.34
AA	52	45	42	93.33	80.77	86.60		**AA**	44	40	38	95.00	86.36	90.48
WA	50	45	43	95.56	86.00	90.53		**WA**	54	49	45	91.84	83.33	87.38
SIA	48	42	39	92.86	81.25	86.67		**SIA**	48	43	40	93.02	83.33	87.91

Table 4: Home and Maintenance and Medical domains results

Cooking Recipe								Electronics						
Class	TQ	C	CC	P(%)	R(%)	F		**Class**	TQ	C	CC	P(%)	R(%)	F
PA	80	72	70	97.22	87.50	92.11		**PA**	75	72	68	94.44	90.67	92.52
DA	112	102	100	98.04	89.29	93.46		**DA**	108	100	96	96.00	88.89	92.31
SpIA	88	80	75	93.75	85.23	89.29		**SpIA**	86	80	75	93.75	87.21	90.36
JA	82	76	73	96.05	89.02	92.41		**JA**	80	74	72	97.30	90.00	93.51
AA	48	40	38	95.00	79.17	86.36		**AA**	50	42	39	92.86	78.00	84.78
WA	42	38	37	97.37	88.10	92.50		**WA**	48	46	42	91.30	87.50	89.36
SIA	46	42	39	92.86	84.78	88.64		**SIA**	44	42	39	92.86	88.64	90.70

Table 5: Cooking Recipe and Electronics domains results

(TQ: Test question, C: Correct, CC: Correctly Classified, P: Precision, R: Recall, F: F-Measure)

7 Conclusion

The simplicity of this approach makes it perfect for multilingual question answering. One can learn the question patterns for a new language using the syntactic structure of the natural language question text.

It has been observed that patterns of special or extra information associated (SpIA) question for procedural texts are highly domain dependent. So, domain specific prior knowledge is needed to recognize this type questions.

References

Aouladomar, F. (2005). A preliminary analysis of the discursive and rhetorical structure of procedural texts. In *Symposium on the Exploration and Modeling of Meaning.*

Aouladomar, F. and Saint-Dizier, P. (2005). An Exploration of the Diversity of Natural Argumentation in Instructional Texts. In *5th International Workshop on ComputationalModels of Natural Argument, IJCAI,* Edinburgh.

Aouladomar, F. and Saint-Dizier, P. (2005). Towards Answering Procedural Questions. In *Proceedings of Workshop KRAQ05, IJCAI05*, Edinburgh.

Moschitti, A., Quarteroni, S., Basili, R. and Manandhar, S. (2007). Exploiting syntactic and shallow semantic kernels for question/answer classification. In *ACL*, pages 776–783, 2007.

Adam, C., Delpech, E. and Saint-Dizier, P. (2008). Identifying and Expanding Titles in Web Texts. In *Proceedings of ACM*.

Pechsiri, C., Sroison, P. and Janviriyasopa, U. (2008). Know-why extraction from textual data. In *Proceedings of KRAQ*.

Zhang, D. and Lee, W. S. and Lee, S. (2003). Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '03*, pages 26–32, New York, NY, USA,ACM.

Delpech, E. and Saint-Dizier, P. (2008). Investigating the structure of procedural texts for answering how-to questions. In *Proceedings of LREC.*

Voorhees, E. M. (2003). Overview of TREC 2003. In *Proceedings of TREC.*

Greimas, A. (1983). La Soupe au Pistou ou la Conservation d'unObjet de Valeur, In *Du sens II*, Seuil, Paris.

Hripcsak, G. and Rothschild, A. (2005). Agreement, the F-measure and reliability in information retrieval: In *Journal of the American Medical Informatics Association*, pages 296-298.

Finkel, J. R., Grenager, T and Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363-370.

Toutanova, K., Klein, D., Manning, C. and Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pages 252-259.

Kosseim, L. and Lapalme, G. (2000). Choosing Rhetorical Structuresto Plan InstructionalTexts, Computational Intelligence, Blackwell, Boston.

Fontan, L. and Saint-Dizier, P. (2008). Creating and Querying a Domain dependent Know-How Knowledge Base of Advices and Warnings. In *Proceedings of KRAQ*.

Yin, L. (2006). A two-stage approach to retrieving answers for how-to questions. In *Proceedings of EACL (Student Session).*

Fontan, L. and Saint-Dizier, P. (2008). Analyzing Argumentative Structures in *Procedural Texts. GoTAL 2008*, pages 366-370

Fontan, L. and Saint-Dizier, P. (2008). Constructing a know-how repository of advices and warnings from procedural texts. ACM Symposium on Document Engineering 2008, pages 249-252

Cai, L., Zhou, G., Liu, L. and Zhao, J. () Large-Scale Question Classification in cQA by Leveraging Wikipedia Semantic Knowledge.

Mortara Garavelli, B., Tipologia dei Testi, In (G. Hodus et al., 1988: lexicon der romanistischen Linguistik, vol. IV, Tubingen, Niemeyer).

Simmons, R. F. (1965). Answering English questions by computer: a survey. Comm. ACM, 8(1):53–70.

S. Verberne, L. Boves, N. Oostdijk, and P. Coppen. 2007. Evaluating discourse-based answer extraction for why-question answering. In *Proceedings of SIGIR*, pages 735–737.

Saint-Dizier, P. (2008). Some Challenges of Advanced Question-Answering: an Experiment with How-to Questions. In *Proceedings of PACLIC 2008*, pages 65-73

Nguyen, T., Nguyen, L. and Shimazu, A. (2008). Using semi-supervised learning for question classification. *Journal Natural Language Processing*, 15(1):3–21.

Zhang, Z., Uren, V., Ciravegna, F. (2010). Position paper: A comprehensive solution to procedural knowledge acquisition using information extraction. In *Proceedings of KDIR2010, Valencia*.

Li, X. and Roth, D. (2004). Learning question classifiers: The role of semantic information. In *Proceedings of International Conference on Computational Linguistics (COLING)*, pages 556–562.

Huang, Z., Thint, M. and Qin. Z. (2008). Question classification using head words and their hypernyms. In *Proceedings of EMNLP*, pages 927–936.

Silva, J., Coheur, L., Mendes, A. and Wichert, A. (2011). From symbolic to sub-symbolic information in question classification. Artifcial Intelligence Review, 35(2):137–154.

Merkel, A. and Klakow, D. (2007). Improved methods of language model based question classification. In *Proceedings of Interspeech Conference*.

Blunsom, P., Kocik, K. and Curran, J. R. (2006). Question classification with log-linear models. In *Proceedings of SIGIR '06*, pages 615–616, NY, USA, ACM.

De Rijke, M. (2005). Question Answering: What's Next?, In *Sixth International Workshop on Computational Semantics*, Tilburg.

Yin, L. (2004). Topic Analysis and Answering Procedural Questions, Information Technology Research Institute Technical Report Series, ITRI-04-14, University of Brighton, UK.

Loni, B., Tulder,G., Wiggers, P., Loog, M. and Tax, D. (2011).Question classification with weighted combination of lexical, syntactical and semantic features. In *Proceedings of the 15th international conference of Text, Dialog and Speech.*