AGRI-QAS Question-Answering System for Agriculture Domain

Sharvari Gaikwad
Department of Computer Engineering
College Of Engineering, Pune, India
Email: gaikwadst11.comp@coep.ac.in

Rohan Asodekar
Department of Computer Engineering
College Of Engineering, Pune, India
Email: asodekarru11.comp@coep.ac.in

Sunny Gadia
Department of Computer Engineering
College Of Engineering, Pune, India
Email: gadiyasa11.comp@coep.ac.in

Vahida Z. Attar
Associate Professor
Department of Computer Engineering
College Of Engineering, Pune, India
Email: Vahida.comp@coep.ac.in

***Abstract*—In this paper, we focus on the need for a robust domain specific question answering system targeting agriculture domain. It aims to help farmers get information and resolve their queries related to agriculture and thereby improving agriculture literacy. The system is based on the principles of natural language processing and information retrieval. Most of the currently available information retrieval tools return ranked list of documents instead of precise answers and do not support runtime answer retrieval. Thus we focus on developing a system which processes unstructured data and returns actual answer for FACTOID questions such as 'which', 'what', 'who', 'where'. For example, "which diseases affect the wheat crop?", "what are the prevalent diseases in North-America region?" etc.**

I. Introduction

There has been significant research in the field of QA systems but there isnt any agriculture domain specific QA system which returns actual answers by analyzing unstructured data to the questions posed by the farmers. To address this limitation, we have developed a system which gives answers to domain specific questions and evaluates them. The input for the system is a corpus of agriculture related documents (news articles, blogs etc. which are easily available on net) and a set of predefined question templates. For this purpose, we have worked upon an open source generic QA framework, QANUS (Question Answering System by National University of Singapore)[6] and modified it to make agriculture domain specific QA system. QANUS adopts a pipelined approach to QA, dividing QA task into several sub-tasks including information base preparation, question processing, answer retrieval, and evaluation. In the Information Base Preparation (IBP) stage, an information source from which answers are to be derived can be set-up. The eventual information source is a LUCENE index of a corpus of documents given in XML format. Pre-processing of the documents that will make up the eventual information source is done here.

The next stage is the Question Processing stage. Typically, questions posed to the system need to be parsed and understood before answers can be found. Necessary question processing is carried out here. Typical operations here include forming a query from the posed questions to the knowledge base, question classification to determine the expected answer type, as well as possibly parsing and part-of-speech tagging. The outputs of these various operations are stored so that they can be subsequently used by the next stage of the QA pipeline.

Finally, the Answer Retrieval stage makes use of the annotations from the question processing stage, and looks up the information source for suitable answers to the posed questions. Proper answer strings that can answer the questions are extracted in this stage.

With the three stages above, QANUS provides the support necessary for a fully functional QA system. The Evaluation stage is introduced to complement the earlier stages and make it easy to verify the performance of the developed QA system. The evaluation stage cross- checks the answers computed previously by the answer retrieval stage with a set of gold-standard answers. The results of the evaluation are then output for easy review.

We have modified the source code for each stage so as to make it compatible for agriculture related queries. To ensure the availability of our system to the community, we have made the system to act as reproducible baseline where the system can be used for other domains by minor modifications in the system.

II. Literature Survey

Currently, there is a Question-Answering Services System[5] for farmer though SMS in Thai language which focuses on similar technique previously applied to language generation, thematic roles, and primitive systems of the Lexical Conceptual Structure. The annotation emphasizes on the semantic model of "What" and "How" queries, lexical inference identification and semantic role, for the answer. It works upon a corpus composed of questions raised by farmers (about 1000 questions), the responses which have been provided by experts, based on existing documents and the texts they originate from.

LUCENE[7] is basically an Information Retrieval tool by Apache. It is a flexible, extensible and very fast tool which empowers full-text search for high-traffic websites such as Twitter and LinkedIn. It also lets user control over low-level data, storage engine and scoring. A query fired to LUCENE is

978-1-4799-8792-4/15/$31.00

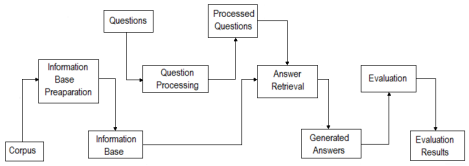

Fig. 1. Overview of QANUS

broken into terms and operators. It supports variety of queries such as wildcard searches, fuzzy searches, proximity searches, range searches, boosting a term, grouping strategies, and Boolean operators. LUCENE gives a ranked list of documents for a query by using inverted indexing on the corpus given as input. It stores statistics about the terms in order to make term-based search more efficient. The scoring is done using a combination of the Vector Space Model and Boolean model to determine the relevance of a given document to users query.

QANUS is an open-source QA framework built with extensibility as the main aim. It is powerful enough to act as a baseline, upon which new algorithms and modifications can made and domain specific QA system can be built easily. It internally uses LUCENE for indexing the input corpus and retrieving the ranked results for the user query.

The QANUS QA framework enables rapid prototyping of new QA systems. Thus it can be used to build domain specific QA system using QANUS as the baseline. The input to this system is a corpus of data and questions which are preprocessed in Information Base Preparation stage and Question Processing stage. Then, In Answer Retrieval stage, it uses nouns and verbs from the question string to form the LUCENE query. This query is then fired to Information base which is built in the Information Base Preparation stage to generate a ranked list of documents. Each document is then processed to find probable answer to the question as well as the score for each answer is calculated. The scoring strategies used are proximity scoring (the proximity between the target of the question and candidate answer within the source passage), coverage scoring (the number of words of the target appearing within the passage), repeated term penalty (penalty if the answer candidate consists of repeated words compared to question target), and sentence scoring (score derived from rank of source passage). The answer with the highest score is given as the output.

There are certain limitations in these systems. The SMS QA system stores answers for already existing questions in the system and it does not generate dynamic answers by traversing through the input documents. In QANUS, the technique used for identifying the expected answer type for a question is naive. The answer retrieval stage is not reliable as well, because it does not consider the subtype of expected answer while retrieving the answer candidate. Also, both the systems do not support multiple-answer FACTO-ID questions (e.g. the systems do not give complete answer for the question ”which crops are used for ethanol?”, if expected answer is ”corn, grain sorghum, wheat, sugarcane”). To address such limitations, we propose the following system design.

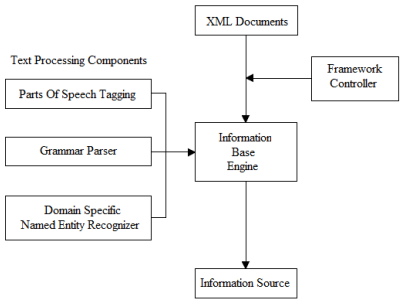

Fig. 2. Information Base Preparation Stage

III. System Description

AGRI-QAS mainly focuses on FACTO-ID questions such as Which, What, Who, Where. We have modified the available QANUS system to make it more accurate, robust, and reliable and domain specific.

The system is implemented in following stages:

A. Information Base Preparation

In this stage, QANUS system takes XML documents as input and implements POS tagging, named entity recognizer, grammar parser etc. We have added a domain specific named entity recognizer at this stage which indexes the documents according the domain specific terms instead of tagging the words as the part of speech. We have modified the IBP of QANUS to make it recognize agriculture related entities like crops, diseases, treatments, pesticides, weeds, fungi, fungicides, variety of wheat, peanut, oil etc. by modifying the output of the parser used in QANUS.

For example, consider the following sentence:

”Leaf rust is a fungal disease that reduces the size of the leaf, ultimately resulting in yield loss. Many wheat varieties in a University of Arkansas performance test at Lewisville in Lafayette County are already infected.”

QANUS tagged this sentence as follows which includes only part of speech tagging:

”Leaf/NNP rust/VBZ is/VBZ a/DT fungal/JJ disease/NN that/WDT reduces/VBZ the/DT size/NN of/IN the/DT leaf,/NN ultimately/RB resulting/VBG in/IN yield/NN loss./VBP Many/JJ wheat/NN varieties/NNS in/IN a/DT University/NNP of/IN Arkansas/NNP performance/NN test/NN at/IN Lewisville/NNP in/IN Lafayette/NNP County/NNP are/VBP already/RB infected./JJ”

AGRI-QAS tags the same sentence as follows which includes domain specific named entity recognizing (FOOD, DISMED, etc.) in addition to parts of speech tagging:

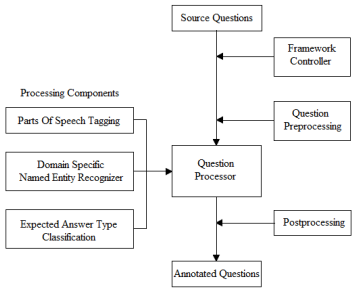

Fig. 3. Question Processing Stage

”Leaf/DISMED rust/DISMED is/VBZ a/DT fungal/JJ disease/NN that/WDT reduces/VBZ the/DT size/NN of/IN the/DT leaf,/NN ultimately/RB resulting/VBG in/IN yield/NN loss./VBP Many/JJ wheat/FOOD varieties/NNS in/IN a/DT University/NNP of/IN Arkansas/LOCATION performance/NN test/NN at/IN Lewisville/ LOCATION in/IN Lafayette/LOCATION County/LOCATION are/VBP already/RB infected./JJ”

Also, to make AGRI-QAS more flexible to adapt to different domains, we separated the domain independent code from the domain dependent part and thereby, made it easier to port the system to other domains. The lists of crops, diseases and treatments used are automatically extracted by using the algorithm proposed by Patil et al. [10].

B. Question Processor

The question processing stage predicts the type of expected answer for the posed question. The original question classifier of the system was not completely suitable for the domain.

In order to make the question tagger more accurate, we have defined the rules to process the question in two phases:

1) Pre-processing: In this phase, the expected answer type of the given question is predicted more accurately by modifying the question before passing it to the original QANUS question classifier.

Following methods are used for this purpose:

- The distance between the ’wh’-word and the subject of the question is reduced by removing articles, adjectives and supporting verbs present in between the ’wh’-word and the subject.
  For example, the predicted answer type given for the question
  ”What are the prevalent crops in South Asia region?” by QANUS is ”HUM:gr”.
  AGRI-QAS skips the stop words and adjectives (”are”, ”the” and ”prevalent”) and correctly predicts the expected answer type as ”ENTY:food”.
- The question is also reformulated by substituting the phrases in the question by synonymous words
  For example, replacing ”leads to” by ”causes” which is more efficiently processed by the question classifier used in QANUS
  Question: ”What leads to excessive sprouting?”
  Predicted answer type by QANUS: ”DESC:desc”
  Automatically reformulated question by AGRI-QAS: ”What causes excessive sprouting?”
  Predicted answer type by QANUS: ”DESC:reason”
- The question is reformulated by adding hyphen between the consecutive names of directions
  For example, replacing ”south east” by ”south-east” which increases the confidence of the predicted answer type as well as shifts the focus of the question from the names of multiple directions to the actual subject
  Original question: ”Which plant is most widely grown in South East America?”
  Predicted answer type by QANUS: ”LOC:other”
  Automatically reformulated question by AGRI-QAS: ”Which plant is most widely grown in South-East America?”
  Predicted answer type by QANUS: ”ENTY:food”

2) Post-processing: In this phase, the question is processed based on the output of the question tagger used in QANUS.

Following methods are used in this stage:

- The categories ’food’ and ’crop’ are collaborated as the two are synonymous for the agriculture domain. Hence, if the question type is ’crop’ then it is changed to ’food’.
- The confidence of the answer type given by the output of the question tagger is between ’0.0’ to ’1.0’. If this value comes out to be less than 0.5 then, the question is passed to Stanford parser which generates the typed dependencies between the words in the question string. The subject of the question is predicted by tracing these typed dependencies. For example,
  Question: ”Which is the most common type of stalk rot?”
  Predicted answer type by QANUS: ”HUM:ind” with confidence of 0.4826
  Predicted answer type AGRI-QAS: ”ENTY:dismed”

Getting the appropriate answer type for a question is one of the crucial steps for a QA system because correct answer candidates can be found only when the system knows what the question expects.

C. Answer Retrieval

In this stage, the question terms are used to formulate the query for the information source. The system formulates the ’LUCENE’ query by considering the verbs and nouns of the question text. This query is fired on the information source developed in the first stage which gives a list of probable answer strings for the question. Each answer string is then analyzed based on the predicted type of the answer in the second stage and the answer with highest score is given as the output.

Entity typed questions are the ones whose expected answer is a name or list of names of a particular entity such as crop, disease or treatment etc. for the agriculture domain. For these questions, the original QANUS system does not consider the subtype of the question (e.g. it only considers ”ENTY” from

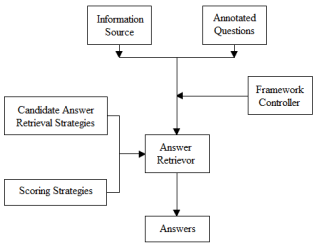

Fig. 4. Answer Retrieval Stage

”ENTY:food” instead of considering ”food”). Also, QANUS directly gives the first occurrence of noun from the retrieved probable answer strings as the final answer. It also does not support the questions which expect multiple entity names.

We have modified the system to accurately give the results for the questions considering the type as well as the sub-type of expected answer.

For example,

Question: ”Which type of stalk rot is the most common?”

Answer given by QANUS: ”types”

Answer given by AGRI-QAS: ”anthracnose”

Answer string: ”There are several types of stalk rot but anthracnose is one of the most common. It is caused by a fungus called Colletotrichum graminicola. It is unique from other types of stalk rot because its symptoms are visible through discolorations on the leaves of the plant as well as the stalk. Once a plant is infected with anthracnose, its cells begin to rot and die, which can lead to weakened cornstalks and lodging.”

Though, the answer strings retrieved by both the systems are same, QANUS gives incorrect answer because of incorrect question classification whereas, AGRI-QAS classifies the question correctly as well as considers the sub-type and hence, gives the correct answer.

We also developed the grammar for the answer retriever to support the multiple entities so that the system gives a list of expected entity names, if present, instead of only the first single entity name.

For example,

Question: ”Which crops are used for ethanol?”

Answer given by QANUS: ”corn”

Answer given by AGRI-QAS: ”corn, grain sorghum, wheat, sugarcane”

Answer string: ”Presently, Louisiana crops used for ethanol are corn grain sorghum wheat and sugarcane”

Due to these modifications, the final answer strings are more accurately generated in AGRI-QAS than QANUS thereby enhancing performance of the system.

D. Evaluation

When the data required to appropriately address the question is present in the input corpus and the answer suggested by the data is the output given by the system, then the output is considered to be the correct answer. Evaluation stage counts the number of such correctly generated answers, and outputs the accuracy achieved by the system. Accuracy in this case is the ratio between the number of correctly generated results to the total number of FACTO-ID questions given to the QA system. For this stage, a set of standard answers corresponding to every input question is provided to the system. The final result of answer retrieval stage for these questions is then compared with the standard answers. If the two entities match, the answer is counted as correctly generated answer.

To make the system easier to compatible to other domains, all the domain related information is stored in a single file ’constants.java’. The domain can be changed by only modifying this file.

IV. Implementation and Results

AGRI-QAS is a fully functioning QA system developed to run on any source document which is in the XML format. The input to the system is corpus of data which includes any number of XML files. The sample data set used as input for AGRI-QAS is obtained from the news articles or blogs by experts in agriculture field, which are then indexed and converted to XML format. Similarly questions posed by farmers are stored in a XML file as well. Then, AGRI-QAS makes use of IR-based techniques as explained earlier to perform the QA task. The system is designed to appropriately address the user needs of finding out information regarding diseases, pests, weeds, fungi, crops affected by them, fertilizers etc.

Information Base Preparation stage makes use of the Aquaint-2 corpus which is stored in XML format. The corpus that we have referred is articles from Delta farm press[11] and South west farm press[12]. AQAINTXMLParser is used to interface the corpus with AGRI-QAS. Then LUCENE is used to build an index of the input corpus. This index will be used for retrieving documents relevant to posed questions in the later stages of the system.

In Question Processing stage, AGRI-QAS classifies the expected answer type of the input questions using Question Classifier. We have built the classifier by modifying the Stanford Classifier so as to make it suitable for Agriculture domain. The classification assigned to each question is stored and passed on to the Answer Retrieval stage. The increased accuracy in predicting the expected answer type increases the final accuracy of AGRI-QAS significantly as compared to QANUS.

Answer Retrieval- To look up for the answers for the posed questions, AGRI-QAS forms a query by eliminating stop-words in the question. LUCENE query uses this query to search through the LUCENE index which was built in Information Base Preparation stage. Documents retrieved by LUCENE are then scored by the Answer Retrieval and then answer candidates are obtained depending upon the expected answer type determined in the Question Processing stage.

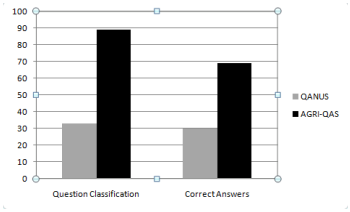

Fig. 5. Accuracy Comparison Graph

e.g. For a question seeking a name of crop, a named entity recognizer is used to find candidate crop names from the ranked documents. Finally, the answer candidates are again ranked based on techniques such as proximity scoring, coverage scoring, repeated term penalty, and sentence scoring. Highest ranked candidate is returned as the final answer.

Figure 5 shows the improvisation in the accuracy of answers given by AGRI-QAS as compared to QANUS.

For a sample set of 100 questions, QANUS classified only 33 questions correctly, whereas AGRI-QAS classified 89 questions correctly. Also, in Answer Retrieval stage, QANUS answers 30 questions correctly while AGRI-QAS answers 69 questions correctly. Thus, the outputs given by QP and AR stages are improved significantly.

V. Conclusion and Future Work

The Lack of domain specific QA system for unstructured data motivated our work for an Agriculture domain specific QA system AGRI-QAS. So we developed a platform to bring agriculture related information under one single system and to provide easy access to the farmers.

In our system, we have successfully tackled the issues of the QA systems such as runtime answer retrieval for the question, development of accurate and reliable answer type predictor, correct retrieval of multiple-entity answers and displaying precise answer instead of a ranked list of documents.

AGRI-QAS is carefully designed to ensure flexibility. It separates the domain-dependent and independent part of the code which makes it much easier to change the domain of the system without changing the functioning of the QA system.

In our future work and research, we will engage in the work of the following several aspects:

- To improve the answer retrieval stage by covering FACTO-ID questions such as When, Yes/No type of questions
- To extend AGRI-QAS using a larger dataset to train additional classifiers for the answer types that are beyond the scope of IR classifiers. A larger dataset will also enable further analysis, for example to identify any common features of questions which are hard to categorise
- Answer Retrieval stage can be made more reliable by identifying the answers using type dependencies given by Stanford Parser.
- Generalising the QA system to work for various type of input/output modules, so that it would be easy to customise the data formats that can be used with the framework.
- Reworking in Answer retrieval stage by using proximity algorithms on query terms so as to improve the accuracy of the QA system.
- Currently, the system does not handle list /questions. So, it will also be useful to improve the functionalities of QA system by handling list type of questions (e.g. give list of the symptoms for AA disease)
- To port the system into the other domains such as medical, biology, etc. to check the reliability of domain independent code of the system

Acknowledgment

We would like to thank Mr. Girish Palashikar, Mr. Sachin Pawar, Mr. Swapnil Hingmire, Mr. Nitin Ramrakhiyani from Tata Research Development and Design Centre, Pune for their valuable guidance.

References

[1] Stanford typed dependencies manual by Marie-Catherine de Marneffe and Christopher D. Manning, Sempetmber 2008, Revised for Stanford Parser v 3.3 in December 2013

[2] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[3] Oleksandr Kolomiyets, Marie-Francine Moens, a survey on question answering technology from an information retrieval perspective, Information Sciences, Volume 181, Issue 24, 15 December 2011, Pages 5412-5434

[4] Sofia J. Athenikos, Hyoil Han, Biomedical question answering: A survey, Computer Methods and Programs in Biomedicine, Volume 99, Issue 1, July 2010, Pages 1-24

[5] Mukda Suktarachan, Patthrawan Rattanamanee and Asanee Kawtrakul, The development of a Question-Answering services System for the Farmer through SMS:Query Analysis, Proceedings of the 2009 Workshop on Knowledge and Reasoning for Anwering Questions, ACL-IJCNLP 2009, pages 3-10, Suntec, Singapore

[6] QANUS: Question-Answering (QA) System, http://www.qanus.com/about-qanus/

[7] LUCENE : http://lucene.apache.org/

[8] Ephyra: Question Answering System, http://www.ephyra.info/

[9] Stanford Typed Dependencies: nlp.stanford.edu/software/dependencies_manual.pdf

[10] Sangameshwar Patil, Sachin Pawar, Girish K. Palshikar, "Named Entity Extraction using Information Distance" IJCNLP 2013: 1264-1270

[11] www.deltafarmpress.com

[12] www.southwestfarmpress.com

[13] L. Hirschman and R. Gaizauskas. 2001 Natural Language Question Answering: The View From Here. Natural Language Engineering, 7, Issue 4: Pages 275-300, December