Abstract

A Question Answering system constructs its answers by querying a data resource. In this thesis, we develop an architecture for a multi-lingual dialog based QA system using PurposeNet based Ontology as the data resource. We have handled two unrelated domains, one of which is an open domain (recipe) while the other is a closed domain (MMTS) and the QA system is reusable with few domain based changes. In this cross-lingual setup we have considered Hindi as the user language (question-input and answer-output are in Hindi) while the Ontology is written in English language.

We discuss the creation of Ontological resource in both the domains. The Ontological structures for both domains was highly influenced by PurposeNet architecture. Data for both the domains exhibit a different characteristic. While a major chunk of the MMTS domain data could be easily extracted by automated means, extracting recipe domain data was a layered approach. The action and artifact Ontology of the MMTS domain are created separately, artifact Ontology is created using automated means while action Ontology is manually populated. For recipe domain, we populated the resource using recipe blog data and then executed a post-processing step, to bring a more unified structure to the Ontology.

Question analysis is a very important aspect of a QA system. We have tried to semantically and structurally classify questions in Hindi. Structural classification helps us in parsing the questions correctly and hence identify the question arguments correctly. Semantic classification helps us in mapping the question arguments to the Ontology and hence extract the answers from the data resource. We also study polysemous question types (*kya* and *kaise*) in Hindi. These questions are partly tackled by the dialog system which is integrated in the QA system.

We give an architecture of a QA system using PurposeNet based Ontology created in OWL, with Pellet reasoner (for data inference). The design of the QA system ensures a multilingual architecture and domain independence. After understanding the question type and its requirements, we create Sparql queries, answers of which are extracted from the knowledgebase. We have handled all kinds of question types giving a major emphasis on non-factoid questions including *why* and *how*. The structure of PurposeNet helps in identifying the answers to these questions effectively. We provide a graph based algorithm for answering these question types efficiently.

As said earlier we also support the QA system with a dialog system which can interact with the user before, during or after a question answering cycle. If a QA system finds an ambiguity, incompleteness or incorrectness in a question, it asks for user interference using the dialog module. We maintain the state of the question and start processing from this state on user's input to efficiently process questions faster.

To test the premise of domain-independence of the QA system, we first create it for MMTS domain and then extended to recipe domain. Testing against gold data gives us a decent accuracy of 76.2% and 66.25% for MMTS and recipe domains respectively. The answers we receive from the QA system is cross-verified by human evaluators on correctness and readability. The results are extremely promising with scores of 2.32/3.00 and 2.46/3.00 respectively for both the domains.