WIKIPEDIA

Question answering

Question answering (**QA**) is a computer science discipline within the fields of information retrieval and natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language.[1]

Contents

Overview

A question answering implementation, usually a computer program, may construct its answers by querying a structured database of knowledge or information, usually a knowledge base. More commonly, question answering systems can pull answers from an unstructured collection of natural language documents.

Some examples of natural language document collections used for question answering systems include:

- a local collection of reference texts
- internal organization documents and web pages
- compiled newswire reports
- a set of Wikipedia pages
- a subset of World Wide Web pages

Question answering research attempts to deal with a wide range of question types including: fact, list, definition, *How*, *Why*, hypothetical, semantically constrained, and cross-lingual questions.

- *Closed-domain* question answering deals with questions under a specific domain (for example, medicine or automotive maintenance), and can exploit domain-specific knowledge frequently formalized in ontologies. Alternatively, *closed-domain* might refer to a situation where only a limited type of questions are accepted, such as questions asking for descriptive rather than procedural information. Question answering systems in the context of machine reading applications have also been constructed in the medical domain, for instance related to Alzheimers disease[2]
- *Open-domain* question answering deals with questions about nearly anything, and can only rely on general ontologies and world knowledge. On the other hand, these systems usually have much more data available from which to extract the answer.

History

Two early question answering systems were BASEBALL[3] and LUNAR.[4] BASEBALL answered questions about the US baseball league over a period of one year. LUNAR, in turn, answered questions about the geological analysis of rocks returned by the Apollo moon missions. Both question answering systems were very effective in their chosen domains. In fact, LUNAR was demonstrated at a lunar science convention in 1971 and it was able to answer 90% of the questions in its domain posed by people untrained on the system. Further restricted-domain question answering systems were developed in the following years. The common feature of all these systems is that they had a core database or knowledge system that was hand-written by experts of the chosen domain. The language abilities of BASEBALL and LUNAR used techniques similar to ELIZA and DOCTOR, the first chatterbot programs.

SHRDLU was a highly successful question-answering program developed by Terry Winograd in the late 1960s and early 1970s. It simulated the operation of a robot in a toy world (the "blocks world"), and it offered the possibility of asking the robot questions about the state of the world. Again, the strength of this system was the choice of a very specific domain and a very simple world with rules of physics that were easy to encode in a computer program.

In the 1970s, knowledge bases were developed that targeted narrower domains of knowledge. The question answering systems developed to interface with these expert systems produced more repeatable and valid responses to questions within an area of knowledge. These expert systems closely resembled modern question answering systems except in their internal architecture. Expert systems rely heavily on expert-constructed and organized knowledge bases, whereas many modern question answering systems rely on statistical processing of a large, unstructured, natural language text corpus.

The 1970s and 1980s saw the development of comprehensive theories in computational linguistics, which led to the development of ambitious projects in text comprehension and question answering. One example of such a system was the Unix Consultant (UC), developed by Robert Wilensky at U.C. Berkeley in the late 1980s. The system answered questions pertaining to the Unix operating system. It had a comprehensive hand-crafted knowledge base of its domain, and it aimed at phrasing the answer to accommodate various types of users. Another project was LILOG, a text-understanding system that operated on the domain of tourism information in a German city. The systems developed in the UC and LILOG projects never went past the stage of simple demonstrations, but they helped the development of theories on computational linguistics and reasoning.

Specialized natural language question answering systems have been developed, such as EAGLi for health and life scientists, and Wolfram Alpha, an online computational knowledge engine that answers factual queries directly by computing the answer from externally sourced curated data.

Architecture

As of 2001, question answering systems typically included a *question classifier* module that determines the type of question and the type of answer.[5] A *multiagent* question-answering architecture has been proposed, where each domain is represented by an agent which tries to answer questions taking into account its specific knowledge; a meta–agent controls the cooperation between question answering agents and chooses the most relevant answer(s).[6]

Question answering methods

Question answering is very dependent on a good search corpus—for without documents containing the answer, there is little any question answering system can do. It thus makes sense that larger collection sizes generally lend well to better question answering performance, unless the question domain is orthogonal to the collection. The notion of data redundancy in massive collections, such as the web, means that nuggets of information are likely to be phrased in many different ways in differing contexts and documents,[7] leading to two benefits:

1. By having the right information appear in many forms, the burden on the question answering system to perform complex NLP techniques to understand the text is lessened.
2. Correct answers can be filtered from false positives by relying on the correct answer to appear more times in the documents than instances of incorrect ones.

Some question answering systems rely heavily on automated reasoning.[8][9] There are a number of question answering systems designed in Prolog,[10] a logic programming language associated with artificial intelligence.

Open domain question answering

In information retrieval, an open domain question answering system aims at returning an answer in response to the user's question. The returned answer is in the form of short texts rather than a list of relevant documents.[11] The system uses a combination of techniques from computational linguistics, information retrieval and knowledge representation for finding answers.

The system takes a natural language question as an input rather than a set of keywords, for example, "When is the national day of China?" The sentence is then transformed into a query through its logical form. Having the input in the form of a natural language question makes the system more user-friendly, but harder to implement, as there are various question types and the system will have to identify the correct one in order to give a sensible answer. Assigning a question type to the question is a crucial task, the entire answer extraction process relies on finding the correct question type and hence the correct answer type.

Keyword extraction is the first step for identifying the input question type.[12] In some cases, there are clear words that indicate the question type directly. i.e. "Who", "Where" or "How many", these words tell the system that the answers should be of type "Person", "Location", "Number" respectively. In the example above, the word "When" indicates that the answer should be of type "Date". POS (part-of-speech) tagging and syntactic parsing techniques can also be used to determine the answer type. In this case, the subject is "Chinese National Day", the predicate is "is" and the adverbial modifier is "when", therefore the answer type is "Date". Unfortunately, some interrogative words like "Which", "What" or "How" do not give clear answer types. Each of these words can represent more than one type. In situations like this, other words in the question need to be considered. First thing to do is to find the words that can indicate the meaning of the question. A lexical dictionary such as WordNet can then be used for understanding the context.

Once the question type has been identified, an information retrieval system is used to find a set of documents containing the correct key words. A tagger and NP/Verb Group chunker can be used to verify whether the correct entities and relations are mentioned in the found documents. For questions such as "Who" or "Where", a named-entity recogniser is used to find relevant "Person" and "Location" names from the retrieved documents. Only the relevant paragraphs are selected for ranking.

A vector space model can be used as a strategy for classifying the candidate answers. Check if the answer is of the correct type as determined in the question type analysis stage. An inference technique can also be used to validate the candidate answers. A score is then given to each of these candidates according to the number of question words it contains and how close these words are to the candidate, the more and the closer the better. The answer is then translated into a compact and meaningful representation by parsing. In the previous example, the expected output answer is "1st Oct."