

Title: Question answering

URL: https://en.wikipedia.org/wiki/Question_answering

PageID: 360030

Categories: Category:Computational linguistics, Category:Deep learning, Category:Information retrieval genres, Category:Tasks of natural language processing

Source: Wikipedia (CC BY-SA 4.0).

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

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 [clarification needed] collection of reference texts

internal organization [ambiguous] documents and web pages

compiled newswire reports

a set of Wikipedia pages [2]

a subset of World Wide Web pages

Types of question answering

Question-answering research attempts to develop ways of answering a wide range of question types, including fact, list, definition , how, why, hypothetical, semantically constrained, and cross-lingual questions.

Answering questions related to an article in order to evaluate reading comprehension is one of the simpler form of question answering, since a given article is relatively short compared to the domains of other types of question-answering problems. An example of such a question is "What did Albert Einstein win the Nobel Prize for?" after an article about this subject is given to the system.

Closed-book question answering is when a system has memorized some facts during training and can answer questions without explicitly being given a context. This is similar to humans taking closed-book exams.

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 [vague] machine reading applications have also been constructed in the medical domain, for instance related to [vague] Alzheimer's disease. [3]

Open-domain question answering deals with questions about nearly anything and can only rely on general ontologies and world knowledge. Systems designed for open-domain question answering usually have much more data available from which to extract the answer. An example of an open-domain question is "What did Albert Einstein win the Nobel Prize for?" while no article about this subject is given to the system.

Another way to categorize question-answering systems is by the technical approach used. There are a number of different types of QA systems, including

rule-based systems ,

statistical systems, and

hybrid systems .

Rule-based systems use a set of rules to determine the correct answer to a question. Statistical systems use statistical methods to find the most likely answer to a question. Hybrid systems use a combination of rule-based and statistical methods.

History

Two early question answering systems were BASEBALL [4] and LUNAR. [5] BASEBALL answered questions about Major League Baseball over a period of one year [ambiguous] . LUNAR 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. LUNAR was demonstrated at a lunar science convention in 1971 and it was able to answer 90% of the questions in its domain that were 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 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. 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 [clarification needed] 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 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. [6]

Applications

QA systems are used in a variety of applications, including

Fact-checking if a fact is verified, by posing a question like: is fact X true or false?

customer service,

technical support,

market research,

generating reports or conducting research.

Architecture

As of 2001 [update] , question-answering systems typically included a question classifier module that determined the type of question and the type of answer. [7]

Different types of question-answering systems employ different architectures. For example, modern open-domain question answering systems may use a retriever-reader architecture. The retriever is aimed at retrieving relevant documents related to a given question, while the reader is used to infer the answer from the retrieved documents. Systems such as GPT-3 , T5, [8] and BART [9] use an end-to-end [jargon] architecture in which a transformer-based [jargon] architecture stores large-scale textual data in the underlying parameters. Such models can answer questions without accessing any external knowledge sources.

Question answering methods

Question answering is dependent on a good search corpus ; without documents containing the answer, there is little any question answering system can do. Larger collections generally mean better question answering performance, unless the question domain is orthogonal to the collection. 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, [10] leading to two benefits:

If the right information appears in many forms, the question answering system needs to perform fewer complex NLP techniques to understand the text.

Correct answers can be filtered from false positives because the system can rely on versions of the correct answer appearing more times in the corpus than incorrect ones.

Some question answering systems rely heavily on automated reasoning . [11] [12]

Open domain question answering

In information retrieval , an open-domain question answering system tries to return 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. [13] The system finds answers by using a combination of techniques from computational linguistics , information retrieval , and knowledge representation .

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?" It then transforms this input sentence into a query in its logical form . Accepting natural language questions makes the system more user-friendly, but harder to implement, as there are a variety of 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 in identifying the input question type. [14] In some cases, words clearly indicate the question type, e.g., "Who", "Where", "When", or "How many"—these words might suggest to the system that the answers should be of type "Person", "Location", "Date", or "Number", respectively. POS (part-of-speech) tagging and syntactic parsing techniques can also determine the answer type. In the example above, 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 correspond to unambiguous answer types: Each can represent more than one type. In situations like this, other words in the question need to be considered. A lexical dictionary such as WordNet can be used for understanding the context.

Once the system identifies the question type, it uses an information retrieval system to find a set of documents that contain the correct keywords. A tagger and NP/Verb Group chunker can verify whether the correct entities and relations are mentioned in the found documents. For questions such as "Who" or "Where", a named-entity recogniser finds relevant "Person" and "Location" names from the retrieved documents. Only the relevant paragraphs are selected for ranking. [clarification

needed]

A vector space model can classify the candidate answers. Check [who?] if the answer is of the correct type as determined in the question type analysis stage. An inference technique can 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 by parsing into a compact and meaningful representation. In the previous example, the expected output answer is "1st Oct."

Mathematical question answering

An open-source, math-aware, question answering system called MathQA , based on Ask Platypus and Wikidata , was published in 2018. [15] MathQA takes an English or Hindi natural language question as input and returns a mathematical formula retrieved from Wikidata as a succinct answer, translated into a computable form that allows the user to insert values for the variables. The system retrieves names and values of variables and common constants from Wikidata if those are available. It is claimed that the system outperforms a commercial computational mathematical knowledge engine on a test set. [15] MathQA is hosted by Wikimedia at <https://mathqa.wmflabs.org/> . In 2022, it was extended to answer 15 math question types. [16]

MathQA methods need to combine natural and formula language. One possible approach is to perform supervised annotation via Entity Linking . The "ARQMath Task" at CLEF 2020 [17] was launched to address the problem of linking newly posted questions from the platform Math Stack Exchange to existing ones that were already answered by the community. Providing hyperlinks to already answered, semantically related questions helps users to get answers earlier but is a challenging problem because semantic relatedness is not trivial. [18] The lab was motivated by the fact that 20% of mathematical queries in general-purpose search engines are expressed as well-formed questions. [19] The challenge contained two separate sub-tasks. Task 1: "Answer retrieval" matching old post answers to newly posed questions, and Task 2: "Formula retrieval" matching old post formulae to new questions. Starting with the domain of mathematics, which involves formula language, the goal is to later extend the task to other domains (e.g., STEM disciplines, such as chemistry, biology, etc.), which employ other types of special notation (e.g., chemical formulae). [17] [18]

The inverse of mathematical question answering—mathematical question generation—has also been researched. The PhysWikiQuiz physics question generation and test engine retrieves mathematical formulae from Wikidata together with semantic information about their constituting identifiers (names and values of variables). [20] The formulae are then rearranged to generate a set of formula variants. Subsequently, the variables are substituted with random values to generate a large number of different questions suitable for individual student tests. PhysWikiQuiz is hosted by Wikimedia at <https://physwikiquiz.wmflabs.org/> .

Progress

Question answering systems have been extended in recent [may be outdated as of April 2023] years to encompass additional domains of knowledge [21] For example, systems have been developed to automatically answer temporal and geospatial questions, questions of definition and terminology, biographical questions, multilingual questions, and questions about the content of audio, images, [22] and video. [23] Current question answering research topics include:

interactivity—clarification of questions or answers [further explanation needed] [24]

answer reuse or caching [25]

semantic parsing [26]

answer presentation [further explanation needed] [27]

knowledge representation and semantic entailment [28]

social media analysis [further explanation needed] with question answering systems

sentiment analysis [29]

utilization of thematic roles [30]

Image captioning for visual question answering [22]

Embodied question answering [31]

In 2011, Watson , a question answering computer system developed by IBM , competed in two exhibition matches of Jeopardy! against Brad Rutter and Ken Jennings , winning by a significant margin. [32] Facebook Research made their DrQA system [33] available under an open source license . This system uses Wikipedia as knowledge source. [2] The open source framework Haystack by deepset combines open-domain question answering with generative question answering and supports the domain adaptation [clarification needed] of the underlying [clarification needed] language models for industry use cases [vague] . [34] [35]

Large Language Models (LLMs) [36] like GPT-4 [37] , Gemini [38] are examples of successful QA systems that are enabling more sophisticated understanding and generation of text. When coupled with Multimodal [39] QA Systems, which can process and understand information from various modalities like text, images, and audio, LLMs significantly improve the capabilities of QA systems.

References

Further reading

Dragomir R. Radev, John Prager, and Valerie Samn. Ranking suspected answers to natural language questions using predictive annotation Archived 2011-08-26 at the Wayback Machine . In Proceedings of the 6th Conference on Applied Natural Language Processing, Seattle, WA, May 2000.

John Prager, Eric Brown, Anni Coden, and Dragomir Radev. Question-answering by predictive annotation Archived 2011-08-23 at the Wayback Machine . In Proceedings, 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, July 2000.

Hutchins, W. John ; Harold L. Somers (1992). An Introduction to Machine Translation . London: Academic Press. ISBN 978-0-12-362830-5 .

L. Fortnow, Steve Homer (2002/2003). A Short History of Computational Complexity . In D. van Dalen, J. Dawson, and A. Kanamori, editors, The History of Mathematical Logic . North-Holland, Amsterdam.

Tunstall, Lewis (5 July 2022). Natural Language Processing with Transformers: Building Language Applications with Hugging Face (2nd ed.). O'Reilly UK Ltd. p. Chapter 7. ISBN 978-1098136796 .

External links

Question Answering Evaluation at TREC

Question Answering Evaluation at CLEF

v

t

e

AI-complete

Bag-of-words

n -gram Bigram Trigram

Bigram

Trigram

Computational linguistics

Natural language understanding

Stop words
Text processing
Argument mining
Collocation extraction
Concept mining
Coreference resolution
Deep linguistic processing
Distant reading
Information extraction
Named-entity recognition
Ontology learning
Parsing Semantic parsing Syntactic parsing
Semantic parsing
Syntactic parsing
Part-of-speech tagging
Semantic analysis
Semantic role labeling
Semantic decomposition
Semantic similarity
Sentiment analysis
Terminology extraction
Text mining
Textual entailment
Truecasing
Word-sense disambiguation
Word-sense induction
Compound-term processing
Lemmatisation
Lexical analysis
Text chunking
Stemming
Sentence segmentation
Word segmentation
Multi-document summarization
Sentence extraction
Text simplification
Computer-assisted
Example-based

Rule-based
Statistical
Transfer-based
Neural
BERT
Document-term matrix
Explicit semantic analysis
fastText
GloVe
Language model (large)
Latent semantic analysis
Seq2seq
Word embedding
Word2vec
Corpus linguistics
Lexical resource
Linguistic Linked Open Data
Machine-readable dictionary
Parallel text
PropBank
Semantic network
Simple Knowledge Organization System
Speech corpus
Text corpus
Thesaurus (information retrieval)
Treebank
Universal Dependencies
BabelNet
Bank of English
DBpedia
FrameNet
Google Ngram Viewer
UBY
WordNet
Wikidata
Speech recognition
Speech segmentation
Speech synthesis

Natural language generation
Optical character recognition
Document classification
Latent Dirichlet allocation
Pachinko allocation
Automated essay scoring
Concordancer
Grammar checker
Predictive text
Pronunciation assessment
Spell checker
Chatbot
Interactive fiction
Question answering
Virtual assistant
Voice user interface
Formal semantics
Hallucination
Natural Language Toolkit
spaCy