Open-Domain Question–Answering
Open-Domain Question–Answering

John Prager

IBM T.J. Watson Research Center
Yorktown Heights
NY 10598
USA

jprager@us.ibm.com
Foundations and Trends® in Information Retrieval
Volume 1 Issue 2, 2006
Editorial Board

Editors-in-Chief:
Fabrizio Sebastiani
Consiglio Nazionale delle Ricerche
fabrizio.sebastiani@isti.cnr.it

Jamie Callan
Carnegie Mellon University
callan@cmu.edu

Editors
Alan Smeaton (Dublin City University)
Andrei Z. Broder (Yahoo! Research)
Bruce Croft (University of Massachusetts, Amherst)
Charles L.A. Clarke (University of Waterloo)
Ellen Voorhees (National Institute of Standards and Technology)
Ian Ruthven (University of Strathclyde, Glasgow)
James Allan (University of Massachusetts, Amherst)
Justin Zobel (RMIT University, Melbourne)
Karen Sparck Jones (University of Cambridge)
Maarten de Rijke (University of Amsterdam)
Marcello Federico (ITC-irst)
Norbert Fuhr (University of Duisburg-Essen)
Soumen Chakrabarti (Indian Institute of Technology)
Susan Dumais (Microsoft Research)
Wei-Ying Ma (Microsoft Research Asia)
William W. Cohen (CMU)
Editorial Scope

Foundations and Trends® in Information Retrieval will publish survey and tutorial articles in the following topics:

- Applications of IR
- Architectures for IR
- Collaborative filtering and recommender systems
- Cross-lingual and multilingual IR
- Distributed IR and federated search
- Evaluation issues and test collections for IR
- Formal models and language models for IR
- IR on mobile platforms
- Indexing and retrieval of structured documents
- Information categorization and clustering
- Information extraction
- Information filtering and routing
- Metasearch, rank aggregation and data fusion
- Natural language processing for IR
- Performance issues for IR systems, including algorithms, data structures, optimization techniques, and scalability
- Question answering
- Summarization of single documents, multiple documents, and corpora
- Text mining
- Topic detection and tracking
- Usability, interactivity, and visualization issues in IR
- User modelling and user studies for IR
- Web search

Information for Librarians
Foundations and Trends® in Information Retrieval, 2006, Volume 1, 4 issues. ISSN paper version 1554-0669. ISSN online version 1554-0677. Also available as a combined paper and online subscription.
Open-Domain Question–Answering

John Prager

IBM T.J. Watson Research Center, 1S-D56, P.O. Box 704, Yorktown Heights, NY 10598, USA, jprager@us.ibm.com

Abstract

The top-performing Question–Answering (QA) systems have been of two types: consistent, solid, well-established and multi-faceted systems that do well year after year, and ones that come out of nowhere employing totally innovative approaches and which out-perform almost everybody else. This article examines both types of system in depth. We establish what a “typical” QA-system looks like, and cover the commonly used approaches by the component modules. Understanding this will enable any proficient system developer to build his own QA-system. Fortunately there are many components available for free from their developers to make this a reasonable expectation for a graduate-level project. We also look at particular systems that have performed well and which employ interesting and innovative approaches.
Contents

1 Introduction 1
  1.1 A Brief History of QA 3
  1.2 Article Plan 7

2 Overview of Question–Answering 9
  2.1 Classes of Questions 9
  2.2 A Typical Software Architecture 16
  2.3 Terminology 20
  2.4 Building a QA System 25
  2.5 General Issues in QA 41

3 Evaluation 49
  3.1 Factoid Evaluation 50
  3.2 List Evaluation 54
  3.3 “Other” Evaluation 55
  3.4 No Answer Evaluation 57
  3.5 Confidence Evaluation 57
  3.6 Evaluation Resources 62
  3.7 Some TREC Details 62

4 Specific Approaches 65
  4.1 WEB-Based QA 65
Question–Answering (QA) is a research activity which is difficult to define precisely, but most practitioners know what it is when they see it. Loosely speaking, it is the field of study concerning the development of automatic systems to generate answers to questions in natural language. The source of the answers and the manner of generation are left open, as are the kinds of questions. However, as a first approximation, the field is currently mostly concerned with answering factual questions (questions about agreed, or at least authoritatively reported facts) by consulting one or more corpora of textual material.

This is not to say that such questions are exclusively about simple properties of objects and events (the height of Mt Everest, the birthdate of Mozart, and so on). The field is also interested in definitions (finding important unspecified characteristics of an entity); relationships (how entities are interrelated); and even opinions (how people or organizations have reacted to events). What is common between these is that QA systems are currently extractive: They just report information that is found in external resources such as newswire, without any attempt to prove the authors correct, and also without any attempt to construct answers that are only implicit in the sources.
Introduction

The kind of QA that will be the subject of this article for the most part will be that which is the subject of the annual TREC (Text Retrieval Conference) evaluation at NIST beginning in 1999. The majority of the QA systems that have been developed, both in academia and industrial research labs, have been at least partly for participation at TREC, and the majority of technical papers on the subject have used TREC corpora, question sets and metrics for evaluation, so such a focus is only natural. However, we will also address aspects of QA that TREC has avoided so far, and we will examine some of the deficiencies of the TREC-style approach.

QA draws upon and informs many of the subfields of Information Retrieval (IR) and Natural Language Processing (NLP), but is quite different from them in certain ways. QA is a very practical activity — more an engineering field than a science — and as such, at least today, is more a collection of tools and techniques than formulas and theorems. This is very understandable when one considers that at its heart, QA is concerned with matching a natural language question with a snippet of text (or in general, several snippets), an algorithmic solution to which could be said to be NLP-complete.

QA is heavily reliant on processes such as named entity recognition (NER), parsing, search, indexing, classification and various algorithms from machine learning, but as we will see it seems to be surprisingly insensitive to the particular choices made. In the author’s experience, choosing to use a state-of-the-art component over a less advanced version does not usually make much difference in the QA-system’s overall performance. What makes a difference is how the components are organized relative to each other; in other words, it is what the system is trying to do that is typically more important than how it does it. This leads to the fascinating situation where contributions usually come from the introduction of brand-new approaches, rather than the fine-tuning of parameters in, say, ranking algorithms. The top-performing systems in TREC have been of two types: consistent, solid, well-established, and multi-faceted systems that do well year after year, and ones that come

\[^1\] A play on the notion of NP-completeness from the field of computational complexity, and AI-completeness, the less formal but still widely held belief that solution to any hard Artificial Intelligence (the parent field of NLP) problem leads to the solution of any other.
out of nowhere employing totally innovative approaches and which out-
perform almost everybody else.

This article will examine both types of system in depth. We will
establish what a “typical” QA-system looks like, and cover the com-
monly used approaches by the component modules. Understanding this
will enable any proficient system developer to build his own QA-system.
Fortunately, there are many components available for free from their
developers to make this a reasonable expectation for a graduate-level
project. We will also look at particular systems that have performed
well and which employ interesting and innovative approaches, but we
will not examine every single system that has acquitted itself well.
We will not cover commercial systems that have not been forthcoming
about their internal workings.

1.1 A Brief History of QA

The field of QA, as it is currently conceived, was inaugurated in 1999
when NIST introduced a Question–Answering track for TREC-8. How-
ever, it was not for this that the first question–answering systems were
developed. For those, we need to go back to the 60s and 70s and the
heyday of Artificial Intelligence facilities such as MIT’s AI Lab. In
those days and in those locations almost all programming was done in
LISP and PROLOG and derived languages such as Planner and Micro-
Planner. These were the test-beds of pioneering AI systems, many of
which could now with hindsight be called QA systems, although they
were not at the time (see, e.g., SHRDLU [80]).

For the most part, these systems were natural-language interfaces
to databases. A question, problem or action to be taken was input in
English. This was parsed into a semantic form — a semantic represen-
tation of the “meaning” of the information need. Then either directly
or through a theorem-proving or other inferencing system, goals were
generated which could be directly translated into database queries or
robot commands. The repertoires, both in terms of actions taken or
inputs understood, were severely limited, and were either never used in
a practical system, or were only usable for the very narrow application
Introduction

for which they were designed. Such systems included LIFER/LADDER [23], LUNAR [81], and CHAT-80 [79].

What these systems had in common was that they were toy systems. They were brittle, and did not scale. They used very complex approaches (inferencing, subgoals, etc.) and did not degrade gracefully [41]. They suffered from lack of general-purpose community resources, which made them expensive to develop or extend. What is ultimately damning is that there was no easily identifiable line of evolution from those systems to the present day; they died out like dinosaurs.

Possibly the first system that can be recognized as what some might call a modern QA system, in the sense that it was open-domain and used unrestricted free text, was the MURAX system [30]. It processed natural-language questions seeking noun-phrase answers from an online encyclopedia; it used shallow linguistic processing and IR, but did not use inferencing or other knowledge-based techniques.

The next milestone came shortly after the creation of the World Wide Web: MIT’s Start system [27, 28] was the first Web Question–Answering system. This work has progressed to the present day, and is still available [2]. Ask Jeeves [3] (now Ask.com), founded in 1996, was maybe the first widely-known Web QA system, although since it returns documents, not answers, one can debate if it is a true QA system [3]. Since that time, other QA systems have come online, both academic and commercial; these include Brainboost [5] and AnswerBus [6] both of which return single sentences. There is a strong argument that even if a number or a noun-phrase, say, is the technically correct answer to a question, a sentence attesting to the subject fact is even better. Especially when typical system’s accuracy is far from 100%, as much transparency is desired as possible.

In 1992 NIST, the U.S. National Institute of Standards and Technology, inaugurated the annual Text Retrieval Conference, commonly

\[4\) Although if the answer is embedded in the document abstract presented in the hit-list, the end-user typically would not care.
1.1 A Brief History of QA

called TREC. Every year, TREC consists of a number of tracks (the exact composition usually changes a little from year to year), each one concerned with a different aspect of IR. In each track, one or more evaluations are run in which teams from around the world participate. TREC is generally now considered a hugely important factor in IR research, since it provides relevance judgments\(^7\) and allows researchers to compare methodologies and algorithms on a common testbed. Many more details about TREC can be found on its official website\(^8\) or in a recently-published book from NIST \(^76\).

In 1999, NIST added a QA track to TREC. There was a feeling that the Question–Answering problem could benefit from bringing together the NLP and IR communities. NLP techniques were, and still are, much more precise than IR techniques, but considerably more computationally expensive; NLP was typically used in closed-world domains, IR typically in open-domain. Thus using the power of IR to search many megabytes or gigabytes of text in short times, together with the refinement of NLP techniques to pinpoint an answer was expected to be a worthwhile technological challenge. The track has continued to this day, although it has changed in ways discussed later.

To emphasize the world-wide interest in QA that has arisen in recent years, we will mention here some other venues for QA. In 2001, NTCIR\(^9\) a series of evaluation workshops to promote research in IR and NLP activities in Asian languages, introduced a Question–Answering task. In 2003, the Cross-Language Evaluation Forum (CLEF)\(^10\) an European TREC-like context for cross-language IR, inaugurated a multiple-language Question–Answering track. Both of these are ongoing. Recent workshops include: Open-Domain QA (ACL 2001), QA: Strategy and Resources (LREC 2002), Workshop on Multilingual Summarization and QA (COLING 2002), Information Retrieval for QA (SIGIR 2004), Pragmatics of QA (HTL/NAACL 2004), QA in Restricted Domains (ACL 2004), QA in Restricted Domains (AAAI 2005), Inference for Textual QA (AAAI 2005), Multilingual QA (EACL

\(^7\)In some cases provided by NIST assessors, in others by the research community.
\(^8\)http://trec.nist.gov/.
\(^10\)http://www.clef-campaign.org/.
Introduction

2006), Interactive QA (HLT/NAACL 2006) and Task-Focused Summarization and QA (COLING/ACL 2006).

1.1.1 TREC Minutiae

For those interested, Table 3.2 in Section 3.6 lists the teams/systems that have placed in the top 10 in the main QA task since the TREC8 QA in 1999. However, in the remainder of this article we will not for the most part be reporting performance scores of teams since over time these wane in significance, and besides they can be discovered in the teams’ own writings and in the annual TREC proceedings. We will report, where known and of interest, how different components contribute to teams’ overall scores.

As discussed in [53], the difficulty of questions cannot be assessed independently of knowing the corpus or other resource in which the answers must be found. Up to the writing of this article, TREC has favored newswire text, which has the characteristics of being written by educated English-speaking adults and of being edited, so there is a minimum of typological errors (as compared with mail and Web documents, and especially compared with OCR (optical character recognition) or ASR (automatic speech recognition) documents). Details of the TREC datasets are given in Section 3.6.1.

1.1.2 AQUAINT

In 2001, the U.S. Government, through a small agency called ARDA (Advanced Research Development Activity) began a three-phase multi-year activity called AQUAINT (Advanced Question–Answering for INelligence). In each phase, a couple of dozen (approximately) U.S. teams, from academia primarily but also industry, were funded to perform research and development with the ultimate goal of producing QA systems that could be used effectively by intelligence analysts in their daily work. One of the goals of the program was to push research away from factoid QA (see Section 2.1.1) into questions about reasons, plans, motivations, intentions, and other less tangible quantities. These are very difficult objectives and it remains to be seen to what extent they can be achieved in the near future. One direct consequence of
this thrust, though, has been to support and influence the QA-track in TREC.

In particular, the AQUAINT program organized several pilot studies in different dimensions of QA. The Definition Pilot explored a different approach to definition questions — finding a comprehensive set of descriptive text fragments (called *nuggets*) rather than a single phrase as required for factoid questions. When in 2003 TREC started using *question series* — groups of questions about a single target — the last question in every series was “other,” meaning “return everything important that has not been asked about yet”; this was a direct outgrowth of the Definition pilot.

The Relationship Pilot, where a relationship was defined as one of eight broad ways in which one entity could influence another (organizational, familial, financial etc.) became a subtask of [75, 76]. There have also been Opinion and Knowledge-based question Pilots, which have prompted research reported elsewhere in the literature. More information about these pilots can be found on the NIST web site.[11]

### 1.2 Article Plan

This article is designed to give readers a good background in the field of Question–Answering. The goal in writing it has been to cover the basic principles of QA along with a selection of systems that have exhibited interesting and significant techniques, so it serves more as a tutorial than as an exhaustive survey of the field. We will not cover (except for occasional mentions in passing) Opinion or Relationship Questions, Interactive QA or Cross-Language QA. Further reading can be found in two recent books [42, 72], as well as the proceedings of the QA tracks of TREC. Finally, we should mention the review article by Hirschman and Gaizauskas [25], which summarizes the state of Question–Answering as of 2001. As of this writing, there is no web-site or umbrella publication from AQUAINT.

The rest of this article is organized as follows. In Chapter 2, we provide a general overview of the theory and practice of

Question–Answering (mostly practice). We look at the typical architecture of a QA system, the typical components that comprise it, the technical issues they tackle and some of the most common and successful techniques used to address these problems. In Chapter 3, we look at the different ways QA systems are evaluated. In Chapter 4, we look at some of the specific approaches that have been used by well-performing systems. We will note that three themes seem to permeate these approaches: testing of identity, use of analogy, and detection of redundancy. Because these are very high-level concepts, each of which can be achieved in a number of different ways, it should be of no surprise that different methodologies, namely linguistic, statistical, and knowledge-based, are all found in QA systems. In Chapter 5, we step back and look at the more abstract concepts of User Modeling and Question Complexity; the issues here have not to date been tackled seriously by the community, but it is asserted here that they are of significant importance, and dealing with them will be necessary for future success. We conclude in Chapter 6 with some comments about challenges for QA.


References


References


References


References


References


