Briefly Noted

Open-Domain Question Answering from Large Text Collections

Marius Paşca
(Language Computer Corporation)

Stanford, CA: CSLI Publications (CSLI studies in computational linguistics, edited by Ann Copestake) (distributed by the University of Chicago Press), 2003, xiii+149 pp; hardbound, ISBN 1-57586-427-4, $70.00; paperbound, ISBN 1-57586-428-2, $25.00.

This book is a revised version of Marius Paşca's 2001 dissertation at Southern Methodist University, supervised by Sanda Harabagiu. Paşca, Harabagiu, and Dan Moldovan are members of a Texas-based research team that has achieved considerable success in the question answering (QA) track of TREC, the major annual text retrieval conference.

Open-domain QA involves retrieving relevant passages from large text collections (e.g., newswire or the WWW) in hopes of finding answers to specific factual questions on arbitrary topics. Queries are not lists of keywords, but rather full sentences, such as "Who was Secretary of State during the Nixon administration?" Paşca focuses on extraction-based QA, in which answer strings (typically NPs or named entities) are not generated, but rather extracted verbatim from relevant text passages. Extraction-based QA is appropriate for simple "factoid" questions but, as Paşca acknowledges, usually fails on *why* and *how* questions, as well as complex but decomposable questions such as "How far is it from the largest alpine lake in North America to the largest city in Nevada?"

The book's introductory chapters present background information that newcomers to QA will find useful. Here, Paşca analyzes the extraction-based QA task and breaks it down into three main subproblems:

1. Question processing, in which the query is parsed into a dependency representation and the semantic category of the expected answer is determined.
2. Passage retrieval, which reduces the search space from the entire document collection to a smaller set of relevant document passages.
3. Answer extraction, in which potential answer strings are identified, extracted, ranked, and returned to the user.

The remainder of the book presents explicit techniques for solving the three subproblems and demonstrates how these solutions fit together into a working end-to-end QA system. The resulting system is not built from scratch—Paşca wisely reuses several existing resources, including Brill's (1995) tagger, Collins's (1996) statistical parser, the SMART information retrieval engine, and WordNet.

The "meat" of the book is in chapters 4 through 6. Chapter 4 addresses the problem of determining the semantic type of a possible answer to a question. Paşca shows that typical named-entity categories such as person, location, and quantity are too rough-grained; more-specific entity types such as city, product, speed, duration, and physical dimension are required. Furthermore, one cannot rely on the question (*wh*-) word to determine the answer type. Paşca's approach is to extract the words connected to the *wh*-word in the dependency representation of the question and look up those words in WordNet conceptual hierarchies that have been hand-mapped onto specific answer types. Thus "What is the *wingspan* of a condor?" maps to a dimension, while "What does Peugot *manufacture*?" maps to a product.

Chapter 5 is on passage retrieval. Here, straightforward keyword expansion and contraction techniques are used to retrieve a sufficient yet manageable set of text passages that are relevant to the question. Finding answers within these relevant passages is the topic of chapter 6. Answer strings are selected using a set of mainly proximity-based heuristics whose weights are set by a machine-learning algorithm. Entities matching the expected answer type are identified in the passages using the WordNet mapping approach from chapter 4. In all three chapters, variations on Paşca's techniques are evaluated in terms of precision scores against a gold-standard collection of 893 questions and answers taken from the 1999 and 2000 TREC QA tracks.

In sum, this slim volume is a lucid and explicit description of a successful extraction-based QA system. Though not a textbook, the book presents enough background

material to be useful to newcomers to QA. More-experienced hands will appreciate it as well.

My only criticism is that the book is already somewhat out of date. QA is a dynamic (and currently well-funded) field that has advanced steadily since 2001, when this dissertation was written. QA subproblems such as question decomposition and answer justification are receiving increased attention. Indeed, Moldovan, Harabagiu, and colleagues earned the top score in the TREC 2002 QA track using a technique not described in Paşca's book: automated reasoning based on lexical chains derived from WordNet glosses (Moldovan et al. 2003). Obviously the definitive book on this fast-moving field is yet to be written.—*John Fry, SRI International*

References

Brill, Eric. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–566.

Collins, Michael. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL '96)*, Santa Cruz, CA.

Moldovan, Dan, Sanda Harabagiu, Roxana Girju, Paul Morarescu, Finley Lacatusu, Adrian Novischi, Adriana Badulescu, and Orest Bolohan. 2003. LCC tools for question answering. In *Proceedings of the 11th Text Retrieval Conference (TREC 2002)*. Special Publication no. 500-251, National Institute of Standards and Technology. Available at http://trec.nist.gov/pubs.html.

John Fry is a research linguist in the AI Center at SRI International in Menlo Park, CA. His current projects are in the areas of question answering and spoken language translation. E-mail: fry@ai.sri.com.

Exploring Time, Tense and Aspect in Natural Language Database Interfaces

Ion Androutsopoulos
(Athens University of Economics and Business)

Amsterdam: John Benjamins Publishing Company (Natural language processing series, edited by Ruslan Mitkov, volume 6), 2002, ix+306 pp; hardbound, ISBN 90-272-4990-3 and 1-58811-269-1, $116.00, €116.00

The representation of time in language has long been an active area of research in linguistics, philosophy, and artificial intelligence, but only recently has it gained relevance for the database community. Although there has been extensive development of natural language interfaces to databases, incorporating linguistic representations of time has not been a concern, as conventional databases do not typically store time-dependent information. In contrast, temporal databases include support for temporal information, so entries can be evaluated with respect to associated times and the time of the query. As the cost of disk storage decreases, it is more practical to store large amounts of data and there is increasing interest in temporal databases as well as in building natural language interfaces that can handle temporal language.

This book, which is based on the author's doctoral thesis, addresses this issue with a theoretical framework that allows temporal information expressed in natural language queries to be used to retrieve information from a temporal database. Central to the framework is an intermediate representation language called TOP, an operator-based formalism designed to represent temporal linguistic phenomena such as past and present tense, a number of time adverbials, calendric reference, and temporal subordinate clauses. A table at the end of chapter 2 conveniently summarizes the phenomena that are supported as well as many that are not, such as future time reference, *when* clauses, and most forms of temporal anaphora.

Using a pipeline architecture, stand-alone natural language queries are analyzed with a modified version of HPSG (Pollard and Sag 1994) that encodes the elements needed to construct TOP expressions in feature structure representations. The TOP expressions are then extracted from the feature structures and automatically translated via a set of translation rules and a provably correct mapping algorithm into the temporal database query language TSQL2 (Snodgrass 1995) for evaluation by the database system. To demonstrate the viability of the framework, a freely available prototype interface

is implemented for a hypothetical airport database using ALE and Prolog.

A primary concern of this framework is representing the temporal contour of events both in language and in the database, and providing a reliable mapping between the two. The framework seeks to capture the expression of actual "occurrences" in language, the issue at the heart of the imperfective paradox. For example, *John was building a house* does not entail that a house was built, but *John built a house* does. To this end, verbs are classified in terms of a Vendlerian aspectual taxonomy (Vendler 1957) and represented in HPSG lexical definitions with an added ASPECT feature. Verbs that have different aspectual interpretations depending on context, as in the habitual (stative) reading for *BA737 departs from Gatwick* and the actual (eventive) reading for *BA737 departed from Gatwick 5 minutes ago*, are treated in this framework as lexically ambiguous with distinct entries for each aspectual class (see, e.g., Moens and Steedman [1988] for a more sophisticated treatment of this as aspectual type coercion). The TOP formalism includes a CULM operator that evaluates telic expressions such as *build a house* as true only if the completion of the denoted event (i.e., a status of "complete" is associated with the relevant database entry) falls within the specified event time. Related work (Nelken 2001) points out, however, that the aspectual distinctions this framework is designed to capture are not supported in actual temporal databases.

TOP is specifically designed to provide an intermediate formal representation that facilitates translating temporal linguistic expressions into a target database query language, but it lacks mechanisms for reasoning about time and events and instead relies on the underlying database system for evaluation. TOP does not have the expressive power of predicate logic. There is no handling of phenomena such as negation or universal quantification, and no inference rules are provided. These limitations restrict the interest this work has for those not working within the specific domain of application of the framework.

In spite of these limitations, this book is explicit and provides a useful resource for researchers and developers of natural language interfaces to temporal databases. The framework addresses difficult problems involved in making natural language queries to temporal databases possible, and together with its prototype interface, it constitutes a foundation from which more complex interfaces may be developed and the value of representing event structure in temporal databases may be explored.—*Mary Swift, University of Rochester*

References

Moens, Marc and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28.

Nelken, Ran. 2001. *Questions, Time and Natural Language Interfaces to Temporal Databases*. Ph.D. thesis, Department of Computer Science, Technion, Israel.

Pollard, Carl and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Snodgrass, Richard T., editor. 1995. *The TSQL2 Temporal Query Language*. Kluwer Academic.

Vendler, Zeno. 1957. Verbs and times. *Philosophical Review*, 6:143–160.

Mary Swift is a research associate in the Conversational Interaction and Spoken Dialog Research Group, Department of Computer Science, University of Rochester. Her research on temporality and event structure includes the semantic interaction between telicity and aspectual interpretation and the representation of time in language for understanding in spoken dialog systems. E-mail: swift@cs.rochester.edu.