Text Databases: One Database Model and Several Retrieval Languages

Crist-Jan Doedens
(Utrecht University)

Editions Rodopi (Language and
Computers: Studies in Practical
Linguistics, edited by Jan Aarts and
Willem Meijs, volume 14), 1994, 314 pp;
paperbound, ISBN 90-5183-729-1,
$52.50, Dfl 100.00

Reviewed by
Nancy Ide
Vassar College

The development of adequate, sufficiently general models for text-dominated databases is a sticky problem, and one that is far from solved. With the increased use of corpora for their research, computational linguists have been forced in recent years to grapple with the organization of and access to large text databases, usually by developing application-specific, ad hoc systems barely grounded in database theory. The advent of digital libraries and the prospect of distributed research environments has generated more interest in the development of application-independent text database models, but few, if any, directly address the needs of corpus linguistics and natural language processing research.

The work Doedens describes in this book is motivated by the need to accommodate the data in the Electronic Concordance Application (ECA) at the Vrije Universiteit Amsterdam. The data comprise the Hebrew Bible, together with annotation for morphology, part of speech, and syntactic structure. Consequently, although the text model and query languages he describes are generalizable to other texts, they were developed to directly address the representation and retrieval needs for linguistic data. The discussion and examples are therefore particularly relevant and accessible to computational linguists.

In Chapter 2, Doedens outlines the properties of enriched text—such as hierarchical organization, recursively nested structures, variants and alternatives, and discontinuities—that make traditional database models (e.g., relational models) inadequate for their representation. He also describes the need for user access to different "views" of a text (e.g., as physical structure, logical structure, linguistic structure, etc.), a particularly thorny retrieval problem. The discussion could be more detailed, and it could be usefully generalized to other types of richly encoded text beyond those containing linguistic annotation. Nonetheless, this is one of the few places these special properties have been described and considered in developing text database models. The author provides a useful list of "basic demands" for text database models, and uses them to evaluate four previously developed models (two of them developed at the University of Waterloo, as an outgrowth of the *New OED* project). He gives cursory treatment to SGML as a text database model; given its increasingly widespread use, a more thorough discussion—especially of its weaknesses—would be valuable.

Chapter 3 outlines the Monads dot Feature (MdF) model for text databases, and Chapter 4 describes its implementation in the ECA database. The model itself seems well-grounded in database theory, although the author fails to adequately locate it in this context. All of the examples used in describing the model are based on the ECA style of syntactic analysis, but the author demonstrates how the model can accommodate other types of analysis. The reader interested in computational linguistics will likely wish that the author provided more than a superficial overview of the parsing strategies used to construct the ECA database; however, parsing is not the focus of the book, and the author provides plenty of pointers to additional information about the ECA project.

The remainder of the book is devoted to query languages. Chapter 5 provides a rather long and belabored definition of what the author calls "topographic" languages: i.e., languages whose syntactic form mirrors the structural relations being defined. In Chapter 6, the topographic query language QL is described in detail, especially its extended features introduced to accommodate special properties of enriched text. The next chapter outlines LL, a nontopographic logical retrieval language, which, because it consists of logical operations over sets, is more elegant and natural (at least, for logicians and computer scientists) than QL. The author explains that because it is a higher-level query language, LL is appropriate as an intermediate language in a natural language query system; natural language requests can be translated to LL and then from LL to QL. The design of both languages is theoretically sound, and browsing through their descriptions provides a useful and well-founded introduction to query language design. A formal description of LL and formal transformation rules from LL to QL are included as appendices. Chapter 9 proposes a software architecture for applying MdF, QL, and LL.

This book is taken from a dissertation, and this shows everywhere in the prose and content. The prose is somewhat stilted and redundant, especially because the author repeatedly summarizes, at the beginning of each chapter and section, what is about to be covered. More seriously, the presentation of the material lacks a current context: the book was published in 1994[1] and describes work done two or three years earlier than that. It would be helpful to see Doedens's proposals in relation to more recent work on the development of text databases and query languages for corpora. For instance, Doedens probably would have had to better justify his dismissal of the well-known relational database query language SQL as a starting point, since it has been used as a basis for newer text database query languages (for example, in recent work at Waterloo—see Blake et al. [1995]). Still, I was impressed by Doedens's principled treatment of open problems in the design of text databases and query languages for linguistic corpora, and at times even considered that some recent work would have benefited from familiarity with his approach.

The most valuable and unique feature of this book is that its topic cuts across several disciplines, including database theory and design, computational linguistics, computer science, and text encoding and analysis, applying methodologies from some to address the needs of others. As a result, Doedens's approach goes considerably farther than any other work I know toward developing a principled approach to text database design that is based on a good understanding of the requirements for handling richly annotated texts. The book is advertised as representing "one step in the road" toward creating a framework for future development of computer management and use of textual information, and this is exactly the impression I was left with after reading the book. For those with an interest in text databases, it is certainly worth looking at.

[1] The book was not submitted for review until September 1997.

Reference

Blake, G. Elizabeth, Mariano Consens, Ian J. Davis, Pekka Kilpeläinen, Eila Kuikka, Per-Åke Larson, Tim Snider, and Frank Wm. Tompa. 1995. Text / Relational database management systems: Overview and proposed SQL extensions. http://solo.uwaterloo.ca/trdbms/docs/trdbms1.ps

Nancy Ide is Associate Professor and Chair of Computer Science at Vassar College. Her publications describe her work on word-sense disambiguation, computational lexicography, and database models and encoding schemes for dictionary data and linguistic corpora. She is co-editor of the Kluwer book series *Text, Speech, and Language Technology* and co-editor-in-chief of the journal *Computers and the Humanities.* She is also the founder of the Text Encoding Initiative (TEI). Ide's address is: Department of Computer Science, Vassar College, 124 Raymond Avenue, Poughkeepsie, NY 12604-0520, USA; e-mail: ide@cs.vassar.edu