1996
Exposing text: How questions for the OED led to answers for the World Wide Web

Frank W. Tompa

> This article was written in January 2002, intended to reflect some of the essence of the Friends of the Library presentation made six years earlier. Unfortunately there is no written record of the original presentation, and not even the slides have been preserved. This text is based on an earlier paper entitled "Not Just Another Database Project: Developments at UW," published in *Reflections on the Future of Text, Proc. 10th Conf. of Univ. of Waterloo Centre for the New OED* (October 20-21, 1994).

A little history

In 1983, the Oxford University Press sent out a call for proposals to convert the Oxford English Dictionary into electronic form so that it could be updated and maintained, new editions could be published, and derivative reference works could be produced. A further purpose was to make the OED available to users online.

This was not just any publishing project. The OED has been described as a national monument and a linguistic treasure house. The first edition's 12 volumes and four supplement volumes, containing 60 million words and 2.5 million quotations tracking the evolution of the language, were the work of well over half a century. When the editors began planning the second edition, they realized that traditional compiling and publishing methods would be inadequate for the task. It was time to bring the dictionary into the electronic age.

When the Oxford University Press issued the call for proposals to create an electronic OED, the University of Waterloo was the only academic institution in the target group of more than a dozen. The others were well-known North American and British companies, including IBM and International Computaprint Corporation (ICC). It was by pure chance that we even caught wind of the project. Michael Brookes, who was UW's director of planning in the early years, left in 1964 to become estates manager at Oxford University. Hearing of the OED project, he put the editors in touch with Douglas Wright, who was then UW's president.

The Press chose three partners to carry out different parts of the project. ICC would capture the text in electronic form, IBM would organize the computing and help bring out the electronic dictionary (known as OED2e), and finally all the data would go into a database that Waterloo would design. As it turned out, Waterloo got much more involved with the computing than was earlier planned. We took on responsibility for converting the dictionary into database form and creating electronic tools for storing and manipulating the contents of the dictionary in that form, as well as designing the database. We also worked with the Press to conduct a survey of potential users of the new electronic dictionary.

We were chosen for the job because our strengths and expertise complemented what the two corporations had to offer. But I'm sure it didn't hurt that when the representatives of the Press toured this campus and talked with people here—including Doug Wright, Wes Graham, my colleague Gaston Gonnet, and myself—they found so much excitement about the project. We had already begun some work and were keen to contribute new ideas. When we shared some of these ideas there was a palpable energy in the air. And that enthusiasm continued on both sides. We worked very closely with the Press until the print edition of the dictionary was published in 1989.

In 1985 we established the UW Centre for the New Oxford English Dictionary as a forum where computer scientists, lexicographers, publishers, humanists, and social scientists could discuss and experiment with prototypes of the software and preliminary versions of the data. The development group included not only software designers and implementers but also future users—staff of the Oxford University Press, people from Waterloo's academic community, and short-term visitors to the Centre.

Key faculty in Computer Science included Gaston Gonnet and Ian Munro. From Arts we had John Stubbs in History, Jack Gray, Harry Logan, and Grace Logan, all in English, and Delbert Russell in French. Our administrative directors Gayle Johannesen and Donna Lee Berg, and later Linda Jones, were key staff members; Tim Bray managed our technical group, which included Tim Snider, Darrell Raymond, Heather Fawcett, and many co-op and graduate students.

Styles of creativity

That was the key to our success—that mix of people, all working closely together from the beginning. As with most projects in computer science, the creative spark leaped from the interactions of many individuals.

This raises the question, what is creativity? It seems to mean different things for different people. While John Stubbs and I were trying to understand how to embrace the interests of people across campus, we discovered there were real differences in culture between arts and computer science.

In computer science, we like to take a problem and make it more general, so that we can concentrate on developing tools to solve several related problems. For us, the interesting thing is the method we are developing for attacking *similar* problems. In arts, researchers are interested in the particulars and in the answer to the part of the problem that makes it distinct from all others.

If I wanted to analyze the text of *Moby Dick*, for example, I would really be interested in building an innovative tool to do the analysis, and I wouldn't be excited by re-using standard techniques. In arts, they'd want to understand the text, and to that end they would want a good tool that they could use and reuse.

The computer science brand of creativity explains my first reaction to the prospect of tackling the OED. I was initially not especially excited. (That came later.) We were sure we could help the Press solve their problems. What we didn't know was whether the solutions would be exciting and innovative. Text search tools did exist (Nexis for newspaper articles; Lexis for legal documents). Text editors did exist. It was possible that off-the-shelf tools would solve

OUP's problems. If so, the project would not advance research in computer science and there would be no call for creativity on our part.

When we got into it, however, we found plenty of challenges. The highly sophisticated, yet structured nature of the text. The sheer size of it—about 600 megabytes, at least 20 times the size of most computers' main memories at the time. Multiple type fonts. How to convert from ICC's input format to a format that would bring out the structure of the text. How to handle text display. How to search for phrases of interest.

But at the beginning we were preoccupied with trying to understand the nature of the challenge.

Data design

Our first step was to create an appropriate model of the data—that is, to find the best way of thinking about it as a whole, which would suggest the best approach to handling it. Until then, when text was stored, accessed, and maintained electronically, it was done in an ad hoc fashion. There were no widely accepted standards for text-based information retrieval systems that could be used for the OED.

We found that existing text search tools like Nexis or Lexis would not work well on the OED—first, because of its size: they got bogged down. The second reason was the complexity of the dictionary's structure. Text search tools could handle the simple structure of a typical newspaper article, where you have a title, perhaps a subtitle, an author, and then the article. Searching for a word in Nexis or Lexis gave you an entire article or document. But this wouldn't work in the OED, where an entry could be one line or 25 three-column pages in length, and where you had a highly nested structure, with the header, etymology, quotations, and definitions, all in a certain order. Thirdly, those systems were based on the notion that when a user searches for a word such as "honour," the system should return all matches to variants such as "honor." This is fine for searching news or legal databases for articles discussing the concept, but it is aggravating when trying to find a specific form of a word in a dictionary.

A bright light went on when we thought more about the structure of the OED. This structure was as important to the query as was the content. When you use a dictionary you want to find words in certain contexts. You might be looking for the etymology of a word, or for a modern definition, or you might want to see all entries for all forms of a word. So it was important to tailor the data to the needs of the users.

This was something we were familiar with in databases, where you can search for different kinds of data and extract views of the data tailored to your needs. For example, you may want the names of all first-year students in Computer Science, or you may only want those with averages over 95 percent. From the OED editors' point of view, the dictionary would also have to be updated in a logical and structured manner that resembled the way you update a database more than the way you add to a text base, where you simply slide the new document in with the others.

In fact, the OED resembled a database more than it did an ordinary piece of text. We would apply database management principles: that is, we would consider the OED as a specific example of a systematic way of arranging and classifying data so that it can be easily stored, modified, and extracted. This was a very new way of looking at text, and it proved to be the key.

Unfortunately, standard relational database systems, where data looks like a collection of tables with rows and columns, were inappropriate to use. However, we adopted some fundamental characteristics of relational database systems—for example, the ability to furnish tailored views of the data—to apply to text databases. The dictionary would be a different thing for different people, depending on the needs of their search.

What we had to do was create something new: a system that incorporated the text-aware features typically found in information retrieval systems (awareness of upper and lower case, for example) into the sort of framework used in database management systems. But that framework would not take the shape of rows and columns, it would be—appropriately enough—a grammar.

"Grammar" means essentially the same thing whether we are talking of a text-dominated database or a printed book. Grammar sets out the rules governing the way we use language. When we apply these rules to show the structure of a sentence, we are "parsing." In a text database, we are dealing with parsed strings—words strung together and interspersed with pairs of matching tags that encode the start and end of each fragment of the data. A tagged string is too cluttered for easy reading on paper, but it allows you to read a text electronically. It looks like this:

Then he asked <quote>Have you read <title>Lord Jim</title>?</quote>, but I hadn't.

The first requirement of the project in 1984 was to convert the dictionary data from the format in which ICC keyed it into the computer, to a consistently tagged form, a process called transduction. We created transduction tools to recognize and expose the structure of text, much like a renovator exposing the beams of a building.

A similar approach forms the basis of the Standard Generalized Markup Language (SGML), which was becoming increasingly important in commercial text processing, and of the Extended Markup Language (XML) co-invented by our own Tim Bray, which has more recently become the basis of data interchange on the World Wide Web.

One thing that made our job easier was that the OED already had its own consistent structure. For example, we could easily recognize and tag the headword, the part-of-speech label, and so on. However, no detailed grammar was available, so we had to design ways to recognize which bits of italic text represented foreign words, which were titles of cited books and journals, which were botanical names, and so forth. Along the way we identified some idiosyncrasies that appear in the dictionary, including several that the OED's editors hadn't anticipated. Some 40 percent of the OED consisted of "typical" entries—they were arranged and formatted in a consistent way. The rest failed to conform to a simplistic grammar in some way: a different typeface used for a subheading, a different form of an abbreviation, and so on—not surprising in a work compiled by many people over many decades.

Once we had established a model of the data and developed transduction software to convert it to a consistently tagged format, we created tools to make the data usable: GOEDEL, PAT, and LECTOR.

Text editing and searching: GOEDEL and PAT

As part of the data-defining process in 1985, we developed a data manipulating language for editors to use in updating tagged text. We named the first prototype GOEDEL, after Kurt Gödel,

a famous mathematician whose name could be written to include the letters OED. GOEDEL was also used to compile a first draft of the fourth edition of the New Shorter Oxford English Dictionary, published in 1993.

In 1987 we introduced a new text indexing and retrieval engine for the use of readers and researchers. PAT (short for PATRICIA, which stands for Practical Algorithm To Retrieve Information Coded in Alphanumeric) is based on the idea of the "semi-infinite string"—that is, a string of data that starts at a certain point in the text, say at the beginning of a desired word, and continues without interruption to the end of the text. Each entry in the search index designates a semi-infinite string. We also created a separate index for each interesting set of text regions, such as definitions or etymologies.

PAT opened up the OED to a spectrum of use and scholarship that wasn't possible before. It has been used to answer queries ranging from "Which English words are derived from German?" to "Which words were first cited in *Maclean's* magazine?" One researcher spent four months at the Centre using the database to study the evolution of baby talk. PAT was also used in 1988 to help proofread the text of the OED2e to identify systematic errors (for example, the use of *obs.* or *Obs* instead of *Obs*., for obsolete) as well as dated phrases (such as "the current Austro-Hungarian Empire") within definitions.

Later on, we developed new codes for approximate string searching, which meant that users did not have to limit their search to exact terms. Finally, we developed ways to allow PAT to operate in a distributed computing environment, which gives it the potential to handle very large, possibly heterogeneous, collections of text distributed across a variety of computers.

In 1995, PAT became the technical foundation for the Open Text Index, one of the earliest search engines on the Web. What set the Open Text Index apart from its competitors was speed, the ability to search structural regions of text, and the ability to search efficiently for phrases, instead of just single words.

The ability to search for phrases was a giant step forward. Other engines had to find each word in a phrase separately, then compare the many matches to see which words were closest to each other. But because PAT treated all text as a string, it could find a phrase, no matter how long, as quickly and easily as it could find a single word.

This capability grew from the needs of the editors of the OED. For them, every word was important. This was a new problem, and we had to be creative to deal with it. The conventional way of searching text was to make a list of "stop" words that have function but no content (if, of, the, and, on…) and then ignore them. Using that approach, if you were to search for the quotation, "To be or not to be, that is the question," any other search tool would search for the word "question." (The other words carry no meaning, or so those engines seem to say.)

Which goes to show that some creativity is needed to be able to rule out common practice when it goes against your needs. We need to have insight to be able to decide which common practices to accept (tailored views of data) and which to discard (stop words).

I demonstrated the importance of stop words on one occasion when my son was doing a school assignment. He had to find the collective terms for different kinds of animals (a flock of geese, a pack of wolves, and so on) but didn't know how to go about it. So I showed him how we could use the OED database to find the answers.

We started searching for the name for a group of lions. Of course, I knew the answer to that one—but I wanted the OED to give it to us and I wanted my son to learn how to do it. So we searched on "lion" or "lions" and got 2,161 matches—too many to look through. Next we queried for those words just within definitions. That brought 288 matches, still too many. And then I had a brainwave. We were looking for the *something* of lions, so why not search for the phrase "of lions" within definitions? That search brought six matches: "a den (of lions)," "a gathering of lions," "characteristic of lions," "a group of lions forming a social unit," "(chiefly said of lions.)," and "<LB> esp. </LB> of lions." Clearly the fourth of these was what we wanted! It was in the definition for "pride."

What's the point of this story? The point is, I proved that "of" is an extremely important word. We went on to apply that capability to other texts besides the OED, including the Bible and the works of Shakespeare. Open Text's reputation was built on its claim to index "every word, every page." In 1996, other people, realizing its value, are just starting to pick it up.

Text display: LECTOR

Search and retrieval are only part of the information gathering process. You also need to be able to read the results of your search. In 1988 we implemented the LECTOR display system. LECTOR is a general-purpose browser, a program that takes a stream of tagged text (remember how unreadable it seemed?) and formats it so you can read it on the screen. Using a specially designed formatting language, it lets you choose the way you want the screen text to look. You can, for example, get on-screen text to look like the source document.

Using LECTOR, we created applications to show more than one view of a text simultaneously, including a program for cross-references. In creating these applications we were able to design programs to link different areas of text in a way that resembles—and predated—web browsers' screen formatting and hypertext linking.

In conventional hypertexts, links are connections between specific regions of a text. Wanting something more flexible, we developed an approach that treats links as the outcome of responses to user "pokes." In this approach, the location of a poke—say, clicking your mouse on a word in Shakespeare—is not the only thing that determines where a link will take you. A potential target application can often decide the meaning of a poke. If you poke at a Shakespearean quotation in the OED, the meaning of that poke will depend on whether the supporting application is a bibliographic lookup tool (in which case you're requesting full bibliographic information about the quotation) or the Shakespeare lookup tool (in which case you're requesting the quote in the context of the play).

This approach changes the notion of hypertext and its links. Instead of defining a hypertext as a document with interconnecting links, we define it as a collection of link-resolving components. Not only can individual hypertext links be stored as explicit pointers, as in most other systems, but they can also be redefined while in use. These properties predated web browsers and far exceed the capabilities of most of the browsers in existence even today.

The OED and the World Wide Web

At the same time as we were launching the New OED project, Tim Berners-Lee, quite separately, was developing the idea of the World Wide Web, a technology for supporting a hypertext of materials stored on computers connected by the Internet. He started design and

implementation in 1989 (the same year Open Text Systems was incorporated), publicized it widely starting in 1991 (the same year Open Text Corporation was formed), and formed the World Wide Web Consortium in 1994.

Darrell Raymond and I were first exposed to the idea of the Web in 1990, at a hypermedia conference. But, not being gifted with foresight, neither of us thought it would amount to anything. Why should the Web become world-wide, when Telidon and other predecessors failed? In hindsight, the elements critical to its success were a very simple-to-use yet flexible front end, and the ability to manage and support documents and files in their original form and to find remote files and bring them to your computer. The Web allowed data to be stored and exchanged in a variety of formats. One new format, Hypertext Markup Language (HTML) was a simple encoding that allowed links within and between documents to be defined and recorded. Through the development of browsers such as Mosaic, which evolved into Netscape in 1994, the dream of simple, ubiquitous access to the information universe became a reality—at least in part.

It was Tim Bray, the first president of Open Text, who in 1994 first recognized the possibilities of using PAT as a search engine for the Web. We had produced three innovative kinds of technology: transduction, search, and display. Because there were other web browsers around, our display technology would not be viable, even though it was more powerful. But Tim recognized that our search technology would have a big impact on the Web. And so it did. The tool we developed to search the OED2e grew into a search engine capable of indexing the entire World Wide Web, so that it could be searched in less time than it's taken me to write this sentence. The Open Text Index was released in 1995 as a free service, and it now handles more than a million queries a day by people around the world.

So, what did our partnership with the Oxford University Press mean in the long run? While committing ourselves to design and develop the technology that would bring the OED into the electronic age, we were already looking beyond the dictionary towards other large text databases. We were aiming to develop a database system that could be used by both text creators and text consumers. Along the way we invented many of the technologies that would eventually be used in the World Wide Web. Not because we foresaw the Web—we didn't—but because we recognized the universality of the technology. To that extent it's fair to say that we at Waterloo had a hand in revolutionizing not only the publishing industry, but the way people use and even think about information.

Frank Tompa has been a member of the Department of Computer Science at the University of Waterloo since 1974 and is a founding member of Open Text Corporation, a Waterloo-based company specializing in text management software. Educated at Brown University with degrees in applied mathematics and at the University of Toronto with a doctorate in computer science, Frank Tompa has turned Oxford University on its venerable head as director of the UW Centre for the New Oxford English Dictionary and Text Research. In 1991 he was named a "leader in Canadian Science" by the Natural Sciences and Engineering Research Council of Canada (NSERC), and in 1997 he received a University-Industry Synergy Research and Development Partnership Award from the Conference Board of Canada and NSERC.