Experiences with the *OED*

Frank Wm. Tompa

Centre for the New OED and Text Research
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

fwtompa@uwaterloo.ca
(519) 888-4675
fax: (519) 885-1208

ABSTRACT

According to the *Oxford English Dictionary*, a dictionary can be either "a book dealing with the individual words of a language..." or "a repository of knowledge, convenient for consultation." An effective dictionary database must serve both roles simultaneously; that is, it must be capable of answering precise questions about the written dictionary text as well as the language described by that text.

An effective representation for the *OED* has been based on the recent text structuring technique known as "descriptive markup," which introduces tags into a text stream. Thus, dictionary components are explicitly identified and delimited, so that, for example, an entry is marked by `<E>...</E>`, an etymology by `<ET>...</ET>`, a usage label by `<LB>...</LB>`, and a cited work by `<W>...</W>`.

The most visibly successful aspect of our research is embodied in the flexible and efficient search and display software. LECTOR™ is a general purpose browser that takes as input a stream of tagged text and formats it to the screen using typography to illustrate its structure. It uses a specially-designed formatting, or display-specification, language to accomplish this, through which the choice of typographical strategies is user-selectable. As a complementary software component, efficient retrieval is provided by the PAT™ text search engine. Each entry in the search index designates a ''semi-infinite'' string that starts at a critical point in the text (e.g., at a word start) and continues uninterruptedly to the end of the text. Text regions (e.g., those representing individual dictionary components) can be specified to limit the scope of material being searched or displayed. Used together, PAT and LECTOR form a powerful query facility for text databases.

Examples drawn from our experiences with researchers and casual visitors illustrate the application of these tools to exploring the *OED*.

1. Background

The *Oxford English Dictionary*[1] (the *OED*) is the largest dictionary of written English, including over 290,000 entries and covering 20 volumes of print or 570 Mbytes of computer storage. Unlike most other dictionaries, the *OED* is based "on historical principles," and thus includes entries for all obsolete words, extensive etymological information, all historical written forms, chronological development of all word senses, and extensive citations giving historical evidence of each sense of each word[2,3] (Figure 1).

In May 1984, the Oxford University Press announced The New OED Project. In Phase 1, the Press was to capture the text of the original twelve-volume *OED* and its four-volume *Supplement* in machine-readable form, integrate the two parts to form one unified work, and publish the resulting dictionary. For this phase, the Press tendered a contract to the International Computaprint Corporation to capture the text; the University of Waterloo assisted the Press in converting the text from its data capture form to a form more suitable for subsequent processing; IBM United Kingdom Limited donated computing equipment, software, and personnel to aid with the integration effort; new materials were added to the text and some revisions made by the Press's lexicographers, with assistance from the University of Oxford Phonetics Laboratory; and Filmtype Services Ltd. was contracted to set the type for the book, which was manufactured by Rand McNally & Company. The publication of the second edition marked the completion of this phase in March 1989.

Subsequent phases of the project include plans for the Press's lexicographers to update, revise, and enhance the data that constitutes the *OED*. The agreement struck between the Press and the University of Waterloo in 1984 promises continuing cooperation in developing the *OED* database, with Waterloo designing and implementing a suitable database system for managing the text. As a result, the Waterloo Centre for the New OED and Text Research has taken responsibility for pursuing research in text database management and its applications to a wide spectrum of text databases, including the *OED* in particular.

2. Text data modelling

Conventionally a database is interpreted as a repository of data that, taken as a whole, constitutes a model of (some aspects of) an enterprise. A text database, on the other hand, is a model of one or more texts, which in turn model some aspects of reality (Figure 2). Thus, text databases are used not only for information retrieval (e.g., "What types of monkeys are found in Brazil?"), but also for editorial work and lexical analysis (e.g., "Which words are defined using 'of or pertaining to'?"). Thus, while some queries ask about reality, others ask about the text.[4] Furthermore, as well as supporting retrieval activities, a text database must provide mechanisms for update and revision as well as for formal publication and other forms of dissemination.

We need to preserve text "as written" and to transmit such text from process to process and from machine to machine. Therefore, to indicate the significant units within a text (e.g., the textual extent of an etymology), we have chosen to represent the data using *text markup*.[5] Three distinct forms of markup are possible: presentational, procedural, and descriptive.

Presentational markup

This form of text representation, also known as "what you see is what you get" or WYSIWYG, uses typography and layout to indicate textual sub-units.

LECTOR and PAT are registered trademarks of Open Text Systems, Incorporated.

> **Macco** (mæ·ko). ? *Obs.* [? A variant spelling of MACAO.] A gambling game; = MACAO.
>
> **1809** BYRON in Moore *Life* (1875) 143 When macco (or whatever they spell it) was introduced. **1825** *Sporting Mag.* XVI. 277 A rubber of whist, or a game of Macco. **1859** THACKERAY ....

Ironically, through the adoption of standard printing conventions, this form of markup makes it difficult to distinguish types of text units. For example, within the citation to *Sporting Mag.*, where does the location information end and the text itself begin? Consider the difficulty when the last piece of location information is the roman numeral I or the first word in the text is the pronoun I. Furthermore, the string MACAO and the string BYRON have similar form, but the former is a cross-reference to a dictionary entry whereas the latter is the name of a cited author. The system would find it difficult to satisfy a user who wished to retrieve all citations for Lord Bacon without accidentally retrieving cross-references to pork.

Procedural markup

An alternative representation uses tags in the text to indicate font shifts and spacing: interpreting the procedural markup converts the tagged text into a corresponding presentational form. This form of tagging is used internally in most word processors as well as for typesetting tapes and to control mainframe typesetting systems. The following example is adapted from the keying conventions used for the *OED*.

```
+L +B Macco +R +N (m+23 +11 k+I o+R ). ?+I +0 Obs.  +OB ? A vari-
ant  spelling  of  +SC  Macao.+EB  +0  A  gambling  game;  +29  +0  +SC
Macao.  +PP +S +B 1809 +SC Byron +R in Moore +I Life +R (1875) 143
When macco (or whatever they spell it) was introduced.  +0 +B 1825
+I  Sporting  Mag.  +R  XVI.   277  A  rubber  of  whist,  or  a  game  of
Macco.  +0 +B 1859 +SC Thackeray ....
```

Unfortunately, we still have the same difficulties as before. Furthermore, although each typographically distinct field is marked at its start, the extent of the field now has to be deduced from the starting point of the next field (e.g., the end of the date field for the first citation is indicated by +SC whereas the end of the date field for the next citation is indicated by +I). This places a potentially complicated pattern-matching burden on all programs that must extract fields from the text.

Descriptive markup

As for procedural markup, the third form of text markup uses tags to delimit units of text. However, the name of each tag is chosen to indicate the *role* of each unit in the text rather than indicating how it is to appear in print.

```
<E>
<HG><HL>macco</HL> <MPR>m&ae.&sd.k<i>o</i></MPR></HG> <lab>? Obs.</lab>
<ET>? A variant spelling of <XR><XL>Macao</XL></XR>.</ET>
<S6> <DEF>A gambling game; = <XR><XL>Macao</XL></XR>.</DEF>
<QP> <EQ><Q><D>1809</D> <A>Byron</A> in Moore <W>Life</W> (1875) 143
<T>When macco (or whatever they spell it) was introduced.</T></Q></EQ>
<Q><D>1825</D> <W>Sporting Mag.</W> XVI. 277 <T>A rubber of whist, or a
game of Macco.</T></Q>
<Q><D>1859</D> <A>Thackeray</A> ...
 ...
</QP></S6></E>
```

Notice that each field is delimited at both ends, and that the uses of cross-reference tags (XR and XL) *vs.* author tags (A) distinguish the role of Macao from the role of Byron. This is the form of marckup chosen for the *OED*: a partial list of tags is given in Figure 3.

3. Support for elastic typography

One major advantage gained by adopting descriptive markup is the flexibility to format a text to suit various media and uses.[6] In particular, the tagging used in the *OED* does not bind a display engine to produce a particular form. In fact, it is left completely to the users' discretion to design text formats that will display extractions from the *OED* to best advantage. To this end, a user must create a specification file, or style sheet, that correlates tags in the text with presentational features — more typically a user requests that some pre-defined specification file is to be applied. This style sheet is then interpreted by the LECTOR text display system,[7] to produce a presentation form on a screen (through the X Windows System).

For the *OED*, we have created a file that includes style sheets for standard display (mimicking the printed style of the *OED*) (Figure 4); standard display but with quotations suppressed; one-per-line display of quotations only; standard display of definitions only or of etymologies and dates only; outline displays of the sense structure showing the upper skeleton only, the complete sense skeleton (Figure 5), or the skeleton with all lemmas shown as well; and a standard display augmented by display of all tags (Figure 6), as well as a completely uninterpreted dump of the characters (including tags) as they are stored.

A display specification for LECTOR is itself represented as a tagged text.[8] After declaring the fonts that are to be made available to the display device, the file includes a series of style sheet declarations (indicated by `<Spec>..</Spec>`). Each style sheet is assigned a name and its default display mode.

```
<Spec>
      <Name>Standard</Name>
      <Family>Times</Family>
      <Type>Roman</Type>
      <Size>14</Size>
      <PrText>on</PrText>
      <PrTag>off</PrTag>
      ...
</Spec>
<Spec>
      <Name>Quotes Only</Name>
      <Family>Times</Family>
      <Type>Bold</Type>
      <Size>12</Size>
      <PrText>off</PrText>
      <PrTag>off</PrTag>
      ...
</Spec>
```

Following that, for each tag that can appear in the text, the style sheet indicates what typesetting is to be done: change in font or style, line break, indentation, insertion of a character string, suppression of text from the file, and so forth. For example, the “Quotes Only” style sheet includes the following directives:

```
<Tag> <Name>Q</Name> <PrText>on</PrText> </Tag>
<Tag> <Name>/Q</Name> <PrText>off</PrText> <LineBreak>on</LineBreak> </Tag>
<Tag> <Name>D</Name> <Type>Bold</Type> </Tag>
<Tag> <Name>/D</Name> <Type>Roman</Type> </Tag>
<Tag> <Name>A</Name> <Type>Small Caps</Type> </Tag>
<Tag> <Name>/A</Name> <Type>Roman</Type> </Tag>
<Tag> <Name>W</Name> <Type>Italic</Type> </Tag>
<Tag> <Name>/W</Name> <Type>Roman</Type> </Tag>
<Tag> <Name>T</Name> <Type>Roman</Type> </Tag>
<Tag> <Name>SQ</Name> <String>[</String> </Tag>
<Tag> <Name>/SQ</Name> <String>]</String> </Tag>
```

4. Support for full text search

A text database system must provide an effective query language — users’ retrieval requests must be easily expressible and efficiently supported. To this end, we have developed the PAT full text search system.[9]

PAT can retrieve all occurrences of any word or phrase appearing anywhere in the *OED* in less than one second. A user may choose to combine results using boolean expressions or proximity conditions. Furthermore, PAT’s flexible field-defining facility provides a mechanism for restricting a search to one or more particular regions of text or to retrieve all regions of a particular type (e.g., all quotations) within which a member of some intermediate result set lies.

For the *OED*, we have defined a control file for PAT which declares the character mappings to be in effect for retrieval purposes and the points in the text that are to be indexed (and thus potentially retrieved in response to a query for a word or phrase).

```
{This is a Pat control file}

{CharMappings "" ""
"{ " "} " "~ " "! " "| " "$ " "% " "′ " "( " ") " "* " "+ " ", "
". " ": " "; " "= " "> " "? " "@ " "[ " "\ " "] " "ˆ " "_ " "′ "
"\b " "\t " "\n " """ "
"Aa" "Bb" "Cc" "Dd" "Ee" "Ff" "Gg" "Hh" "Ii" "Jj" "Kk" "Ll" "Mm"
"Nn" "Oo" "Pp" "Qq" "Rr" "Ss" "Tt" "Uu" "Vv" "Ww" "Xx" "Yy" "Zz"}

{WordStarters  " \P" "-\P" "\P-" "\P<" "\P&"}
```

The CharMappings statement declares which punctuation marks and special symbols should be treated as if they were blanks (and thus word delimiters) and that retrieval should be case-insensitive (all upper case letters treated as if they were lower-case). The WordStarters statement declares that any string starting with a printable character after a blank or hyphen or any string starting with a hyphen, left angle bracket or ampersand can be retrieved. Thus a search for “able as ” in the *OED* instantly returns fifteen matches including:

```
as able as any cowboy on the range..to manage anythin..
be able as days go by Always to look myself straight ..
<T>Able as he is, he has adopted a tone and style..un..
<T>Able as he proved himself, his task was one of no ..
as able As he that hight <i>Irrefragable</i>. </T></Q..
ne able as hee went along to have seene the Wood for ..
ng able, as I noted before, to see them at that dista..
---able′, as in <CF>countless</CF>, <CF>numberless</C..
an-able as it should be), it sets a-worke thousands.<..
so able as now. </T></Q><Q><D>1611</D> <A>Shaks.</A> ..
so able as now. </T></Q><Q><D>1651</D> tr. <W>Life Fa..
as able, as opportunity occurred, to secure the servi..
ng able, as the phrase is, to take the law of him. </..
ng able, as they say, to overpower and hinder its inc..
As able as yourself and as nimble too, though I mayn′..
```

We can subsequently determine that fourteen of these are within quotations, and that the fifteenth includes

```
‘not to be --ed’, ‘un---able’, as in <CF>countless</CF>...
```

within the definition for the entry for **-less**.

Unlike other text search systems, which index individual words in a text, PAT is based on the concept of *semi-infinite strings*.[10] Thus the query “one of” should not be interpreted as a search for all occurrences of this particular two-word phrase, but rather a request to retrieve all occurrences of strings that begin with the character `o` (or `O`, since that is mapped to `o`), then followed by `n`, then `e`, then one or more blanks, then `o`, then `f`. The 23899 matches in the *OED* include not only

```
ayed one of her jade tricks.</T></Q></QP></S6></S4><p><..
> <W>One of our Conquerors</W> I. xiv. 269 <T>A young w..
 was one of the principal executors of the murder [of D..
sed..one of the ten Commandments.</T></Q></QP><QP><LB>t..
```

but also

```
put (one) off <CF>with</CF>.  <LB>Obs.</LB> <LB>rare</L..
ding one offer only and this is a conditional offer thi..
 but one Office. </T></Q><Q><D>1732</D> <A>Lediard</A> ..
that one often feels..disinclined to get off. </T></Q><..
```

(To search for the two-word phrase, one would specify "one of " to insure that a blank or punctuation mark follows the `f`.) As a result, searches for prefixes of words or arbitrarily truncated phrases are as easily supported by PAT as are searches for complete words.

5. Using PAT and LECTOR together for the *OED*

We have worked with many researchers to extract information from the *OED*. One outcome has been the creation of "A Guide for Scholars," which includes outline search strategies and commentary for the following queries:[11]

1. Find the number of citations in the dictionary's quotations to a work entitled *Travels in Arabia Deserta* by the Victorian traveller and poet Charles M. Doughty.
2. Find all the words derived from Italian between the years 1650 and 1725 which were first cited in a drama.
3. Find all the words of Dutch origin contained in the *OED*.
4. Find the percentage of citations from "journals" in the dictionary within a given time period.
5. Find all the words in the dictionary that are associated with anthropology as a discipline.

Rahter than reexamine these, however, we present here an alternative example worked out in detail to illustrate the potential of the *OED* text database. Professor Delbert Russell, a colleague in the Department of French, was interested in exploring the Anglo-Norman roots of English.[12] His starting point was to find entries in the *OED* that satisfy any of the following conditions:

1. the first language named in the etymology is Anglo-French (abbreviated AFr. or AF.);
2. the first language named in the etymology is Old French (abbreviated OFr. or OF.);
3. the first language named in the etymology is French (abbreviated Fr. or F.) *and* the earliest quotation is dated prior to 1500.

Having extracted these entries, they were to be further evaluated and filtered based on a closer look at the complete etymology and the complete list of citation dates.

Figure 7 shows a display window that provides an interface to the *OED*. The box at the top contains a sequence of pull-down menus that give the user access to the operators within PAT (for combining query results, limiting queries to particular fields, for proximity searches, and so forth). The next box is an input window for entering a search string. The box in the centre of the figure shows the history of the queries in the session (transliterated from "point-and-click" actions into textual form), and the bottom window display a sample of matches from the *OED* resulting from the last query. Finally, the box on the right provides a menu for the pre-defined fields in the *OED* which can be selected for restricting searches or for limiting output. The queries listed in Figure 7 define all strings in the *OED* that begin with a language label related to Anglo-Norman (as defined above). It is interesting to note the distribution of occurrences of each of the language names in the *OED* by examining the count fields.

Query 10 (Figure 8) then finds the entries in which the language French is used. The user specifies this query by selecting “Structure Including Last” from the `Structure` pull-down menu, and then pointing at “Entry” under the list of pre-defined fields (at the top of the box labelled `Document Structure`). Next we restrict our attention to the earliest quotations within these entries and finally to the dates within just those quotations (again by selecting one of the menu options under `Structure` and pointing at the appropriate elements on the screen). To restrict ourselves to entries in which the date precedes 1500, we find all dates that are between 15.. and 19.. (Queries 13-16), and remove these dates from the set of interest (the operators for Queries 15 and 17 being found under the menu labelled `Combine`). However we then add back those dates containing the form “ante 1500”. Finally in the last two queries we redirect our attention back to the strings starting with “<L>Fr.” or “<L>F.” within this restricted set of entries.

Queries 23 and 24 (Figure 9) then collect together the occurrences of Anglo-French, Old French, and the restricted French strings. Each string is to be checked to verify that it occurs as the first language in an etymology and that it is not preceded by a cross-reference (which would represent a formation within English rather than a borrowing). We first restrict the strings to those found within etymologies. Next, a user-defined field is declared to start with these selected etymologies and end at the first closing tag for language (`</L>`) or opening tag for a cross-reference (`<XR>`). Finally Query 30 restricts the language strings to those that start within these user-defined fields.

A sample of the results is shown in the bottom window of Figure 9. By pointing at any one of these results, the corresponding entry is displayed in a separate LECTOR window. Figure 10 shows the entry for **chape** according to the style sheet “Etymology and Dates” and Figure 11 shows the same entry in standard form. Figures 12 and 13 show entries for words from Anglo-French and from Old French, respectively.

Once the strategy had been determined, this application required less than 15 minutes in total to execute on a SUN4; under six minutes of this time was spent waiting for computations to complete. In our experience, this is acceptable response time for pursuing research productively.

6. State of the software

The software we have developed for searching and displaying the *OED* is flexible enough to handle most texts with little preparation. Interested readers may wish to contact Open Text Corporation (Waterloo Town Square, Waterloo, Ontario N2J 1P2) for licensing information.

Within the Waterloo Centre for the New OED and Text Research, we are continuing to improve the software to increase its flexibility, efficiency, and functionality. For example, we are currently investigating alternative search methods more suited to CD-ROM storage, browsing and display extensions to accommodate hypertexts, and applications of the technology to managing software specifications and code *as text*. We gratefully acknowledge financial support for this ongoing work through grants from the Natural Sciences and Engineering Research Council of Canada, from IBM Canada, Ltd., and from the Information Technology Research Centre.

References

1. *The Oxford English Dictionary. Second edition*, Clarendon Press, Oxford, England, 1989.

2. Murray, K.M.E., *Caught in the Web of Words: James A.H. Murray and the Oxford English Dictionary*, Oxford University Press, Oxford, England, 1979.

3. Berg, D.L., *A User’s Guide to the OED*, Oxford University Press, Oxford, England, 1991.

4. Tompa, F.W., and D.R. Raymond, "Database Design for a Dynamic Dictionary," in *Research in Humanities Computing*, Oxford University Press, 1991, 324-336.

5. Coombs, J.H., A.H. Renear, and S.J. DeRose, "Markup systems and the future of scholarly text processing," *Communications of the ACM 30*,11 (November 1987) 933-947.

6. Fawcett, H., "Using tagged text to support online views," UW Centre for the New OED and Text Research Tech. Rept. OED-89-04 (July 1989).

7. Raymond, D.R., "`lector` — an interactive formatter for tagged text," UW Centre for the New OED and Text Research Tech. Rept. OED-90-02 (Aug. 1990).

8. LECTOR. *Creating a Display Specification File*, Open Text Systems, Inc., Waterloo, Ontario, Canada, 1990.

9. PAT *3.3 User's Guide* and PAT *3.4 Release Notes*, Open Text Systems, Inc., Waterloo, Ontario, Canada, 1990.

10. Gonnet, G.H., R.A. Baeza-Yates, and T. Snider, "Lexicographical indices for text: inverted files vs. PAT trees," UW Centre for the New OED and Text Research Tech. Rept. OED-91-01 (Feb. 1991).

11. Berg, D.L., "The research potential of the electronic OED2 database at the University of Waterloo: a guide for scholars," UW Centre for the New OED and Text Research Tech. Rept. OED-89-02 (May, 1989).

12. Russell, D.W., "The New OED Project and Anglo-Norman Lexicography," *Proc. Jubilee Colloquium of the Anglo-Norman Text Society*, (held: St. Helier, Jersey, March 25-28, 1988) Anglo-Norman Text Society, London (Occasional Publications Series) in press.