Introducing XAIRA...

An XML-aware tool for corpus indexing and searching

Lou Burnard
Tony Dodd

Research Technology Services, OUCS
Topics

- Background: from SARA to XAIRA
- Architectural issues
- What can you do that's fantastic?

http://www.oucs.ox.ac.uk/rts/xara/
Software development: the conventional wisdom

i. Assess user needs/requirements

ii. Prototype systems to fit user needs

iii. Evaluate against user performance

iv. Repeat from stage ii. until either

   a) user is happy, or
   b) money runs out
Software development: the usual practice

- Creeping featurism
  - hey, that's a cool idea, I'll bolt that on too
- The Hausmann effect
  - this is hopeless, we need to drive a few boulevards through here
- Modularity and standardized interfaces are your only friends
Historical Background (c. 1994)

- Original design goals
  - robust searching of very large (c. 1 Gb) amount of SGML data
  - re-use available indexing tools
  - usable by researchers in CL, NLP, lexicography

- Original assumptions
  - client/server architecture
  - index build once only
  - one specific corpus (the BNC) only
Historical Background (c.2002)

- **Design goals**
  - robust searching of any amount of XML data
  - offload processing to other components wherever possible
  - assume nothing about input DTD

- **Architecture**
  - client/server still valid
  - expect to re-index often
  - expect multiple interfaces
Why another search engine?

- Can't you do all this with Google?
  - Digital texts are not just for discovery and display
  - The methods of corpus linguistics have a wider relevance
- Can't you do all this with eXist?
  - Probably, but only if you have a team of programmers at your disposal!
Xaira: the key features

- Supports word search, concordance generation and manipulation, collocation, lexical analysis
- Uses XML annotation to the max
- Supports XML-aware complex queries
- Leverages existing standards
  - TEI/XCES
  - Unicode
  - CSS and XML
  - SOAP (xmlrpc)
- Uses efficient and compact indexing appropriate to small or huge corpora
Architectural issues

How do the various parts of a XAIRA system interact?
First catch your corpus…

- any collection of well-formed XML documents
  - if a DTD is supplied, the corpus must be valid
  - if no TEI header is present, one will be created
- the more you put in, the more you get out
- "texts" are defined independently of file structure, as are the relevant units within them
- all indexing information is stored in the corpus header
Building the indexes

- tokenization
  - implicit, following Unicode rules (locale-sensitive)
  - explicit, following mark up
  - supports lexical features (e.g., collocation)
- lemmatization and POS tags
  - special case of "additional key" mechanism
  - generalized to provide fast context-specific searches
- tag indexes
  - attribute values and codebooks
Next, build your index…

- Can be done simply by adding appropriate declarations to the TEI Header and running the indexer utility
- But probably easier to do with the supplied IndexTools utility which
  - organizes and validates the files you are using
  - updates (or creates) the header with
    - tokenization and indexing rules
    - tag and attribute usage, descriptive codebooks etc.
    - "bibliographic" metadata
    - default behaviour for character encoding, formats used, etc
- optionally runs and tests the indexer
Architecture

- TEI corpus
- index lexica
- server
  - xara object model
  - SOAP
  - PC client
  - Web client
Hoorah for Unicode

- All data is held internally as Unicode
  - this allows us to defer most problems (e.g. tokenization, case-folding, line-breaking, character normalization, glyph composition) to someone else!

- User interface issues
  - For output, use one or more appropriate fonts
  - For input, we provide a keyboard definition utility
Client/protocols

- The original SARA protocol
  - Corpus Query Language
  - Ad-hoc ASCII strings
- Now revised completely
  - Sara Object Model can be accessed
    - directly by the client
    - via a SOAP wrapper
    - using saraScript
- The model defines
  - CQL in XML
  - methods to manipulate CQL queries and results
- Support for web services
Corpus Query Language

- **Tokens**
  - word, punctuation mark, substring
  - word+annotation/s (e.g. POS)
  - Unicode-compliant regular expressions for words, attribute values
  - XML start- or end-tag, plus attributes

- **Boolean operations**
  - negation, optionality
  - sequence, disjunction, join

- **Scoped operations**
  - within span, within XML element
Client features

- User-configurable display
  - plain, XML, user-defined stylesheets
- User-definable keyboard mapping
- Texts, Results, Browse windows
- Results can be exported in XML
- Scripting language
What can you do that's fantastic?

A sketchy overview of Xaira's query and display facilities
Target queries

- What is the most frequent noun in this corpus?
- Find a random sample of 100 instances of "fish" followed by "chips" within 4 words
- Find sentences beginning with a conjunction.
- Show all inflected forms of the name "Winston".
- Show sentences which begin with "well" and end with a question mark.
- How often and in what contexts is the word "nature" used in different kinds of writing?
- Which verbs collocate significantly with "bosom" at different periods of history?
- Do men use colour vocabulary differently from women?
Phrase or simple query

- search word or phrase
- can be case sensitive
- can include punctuation
- can include *anyword* character
- watch out for tokenization problems
m mode Criste þenigan æt his halgm weofode, swaswa eowrum hade gerist. Forbanþe ge synd g
wunode on byssere worulde butan ælcere synne swaswa nan oper man.

On þam lendenum is, swaswa we leornigad on bocum, sco fulc galyns
and on eallum godnyssem, mannum to bysene, swaswa byrnende leothtatu.

And Iohannes gesæah, swaswa we sædon ær, bone hælend ymbgyrdne a
man and modes þæt halige husel him geoffriæ, swaswa he sylf getæhe ær his browunge.

I manæga Godes boawan on þam scotan haðum, swaswa us scgø scæan.

Man sceal mæssan mid mycelre clænmyse. Swaswa pa haligan dydan, þe we halad confessor
And mid clænmyse Criste þenidon. Swaswa pa tanories us cyðad openlice. Quod mul
ingðæ, þæt nóman syn þæs, þæt ge swa libhan swaswa laewed man.

Ac we mynqiað cow, þæt ge clænmyse healdan, swaswa Cristes þægenas on godum gehingendum, Gode to cwemynsse, swaswa pa haligan dydon, þe we her beforan ra
labo eis quod non peribit; Gode soðlice gecwæð, swaswa us sæde se witega:

Ar Crist gwaed swæheah
her cað on leden, hit synd sume men.
Word Query

* searches the lexicon for word stem or pattern
* returns matching word forms with frequencies
* can restrict by frequency
* can apply lemmatization rules
* then carries out a lookup to display hits
<table>
<thead>
<tr>
<th>Word</th>
<th>Frequency</th>
<th>Forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>sob</td>
<td>89</td>
<td>26</td>
</tr>
<tr>
<td>sobfaest</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>sobfaestnes</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>soplic</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>soplicice</td>
<td>66</td>
<td>2</td>
</tr>
<tr>
<td>sopra</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>sobra</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>sobpre</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>sobpre</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>sopsagol</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>sopsagu</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Form</th>
<th>pos</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>sod</td>
<td>JIN</td>
<td>19</td>
</tr>
<tr>
<td>sodan</td>
<td>JJD</td>
<td>10</td>
</tr>
<tr>
<td>soban</td>
<td>JJD</td>
<td>7</td>
</tr>
<tr>
<td>sodan</td>
<td>JJA</td>
<td>6</td>
</tr>
<tr>
<td>sode</td>
<td>JIN</td>
<td>6</td>
</tr>
<tr>
<td>soda</td>
<td>JIN</td>
<td>5</td>
</tr>
</tbody>
</table>
XML query

- searches for XML start- or end-tags (not elements)
- start-tags optionally qualified by attribute values
- uses predefined codebooks (value indexes) if available
- useful in combination with other queries
XML query
Building complex queries

- visual interface

- **scope node** defines where to look
  - an XML element
  - by span

- **query nodes** define what to look for
  - word, phrase, POS, pattern, XML, or AnyWord

- **link types** define sequence in which query node targets should occur
  - next, one-way, two-way
Sentences beginning with conjunctions
Display of results

- Line (KWIC) or Page mode
- Context size expandable *ad lib*
- User defined formatting
  - stylesheet mechanism based on CSS
- Export of result files
  - in XML, or tab delimited
Sample stylesheet display
Collocations of the lemma "god"
Manipulation of results

- **Sorting**
  - by left, right, or centre spans
  - by orthographic form or POS code
  - case sensitive or insensitive

- **Thinning**
  - at random
  - by selection

- Analysis and partitioning
Partitions

- A partition is a way of grouping the texts making up a corpus, according to
  - *some explicit annotation or characterization (e.g. an attribute value)*
  - *according to whether or not they match a query (a partition of two halves)*
  - *arbitrary manual classification*

- Each member of a partition is a discrete text
- Analysis shows the rate of occurrence of hits within members of the partition
- Partitions can be saved and re-used or defined dynamically
- `indextools` generates a default partition using `<catRef>` element
<table>
<thead>
<tr>
<th>Text</th>
<th>Class</th>
<th>Title</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>coaelet3</td>
<td>REL</td>
<td>Ælfric's First and Second Letter to Wulfstan</td>
<td>25</td>
</tr>
<tr>
<td>cowulf3</td>
<td>HOM</td>
<td>Wulfstan’s Homilies (O3)</td>
<td>12</td>
</tr>
<tr>
<td>coaelive</td>
<td>BLS</td>
<td>Ælfric’s Lives of Saints</td>
<td>8</td>
</tr>
<tr>
<td>coaelet4</td>
<td>REL</td>
<td>Ælfric’s Letter to Wulfsige</td>
<td>7</td>
</tr>
<tr>
<td>cowulf4</td>
<td>HOM</td>
<td>Wulfstan’s Homilies (O3/4)</td>
<td>6</td>
</tr>
<tr>
<td>colaw2</td>
<td>LAW</td>
<td>Alfred’s Introduction to Laws, Alfred’s Laws, Ine’s Laws</td>
<td>5</td>
</tr>
<tr>
<td>cochroa2</td>
<td>HIS</td>
<td>Anglo-Saxon Chronicles until 946</td>
<td>5</td>
</tr>
<tr>
<td>colaw3</td>
<td>LAW</td>
<td>Eleventh Century Laws</td>
<td>5</td>
</tr>
<tr>
<td>coorosiu</td>
<td>HIS</td>
<td>Alfred’s Orosius</td>
<td>1</td>
</tr>
<tr>
<td>cogregd4</td>
<td>BLS</td>
<td>Dialogues of Gregory the Great (MS C)</td>
<td>1</td>
</tr>
<tr>
<td>coboeth</td>
<td>PHI</td>
<td>Alfred’s Boethius</td>
<td>0</td>
</tr>
<tr>
<td>cochroa3</td>
<td>HIS</td>
<td>Anglo-Saxon Chronicles 951 – 1001</td>
<td>0</td>
</tr>
<tr>
<td>coapollo</td>
<td>FIC</td>
<td>Apollonius of Tyre</td>
<td>0</td>
</tr>
<tr>
<td>cobede</td>
<td>HIS</td>
<td>Bede’s Ecclesiastical History</td>
<td>0</td>
</tr>
<tr>
<td>header</td>
<td>header</td>
<td>Brooklyn Corpus</td>
<td>0</td>
</tr>
<tr>
<td>cogregd3</td>
<td>BLS</td>
<td>Dialogues of Gregory the Great (MS H)</td>
<td>0</td>
</tr>
<tr>
<td>colaw4</td>
<td>LAW</td>
<td>Late Laws</td>
<td>0</td>
</tr>
</tbody>
</table>
Use of nature by domain
Saving and re-using queries

- Bookmarks
- Queries are saved with thinning information
- Optional annotation
- Associated bookmarks are preserved