

An XML-aware tool for corpus indexing and searching

Lou Burnard
Tony Dodd

Research Technology Services, OUCS

#### What is XAIRA?

- XML Aware Indexing and Retrieval Architecture
- Developed from the British National Corpus SARA program
- Components include an Index Toolkit and a Windows Client
- Open Source: available from http://xaira.sf.org

# 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.2004)

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

- Indexes the PCDATA
- 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
- Uses efficient and compact indexing appropriate to small or huge corpora

## 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

# Loading a corpus into Xaira

- Put the corpus into XML
  - convert it to Unicode
  - index tools can help
    - Adding minimal tagging
    - Converting existing tagging
- Use the Index Tools wizard

The more markup you put in, the more Xaira can do with it.

#### 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



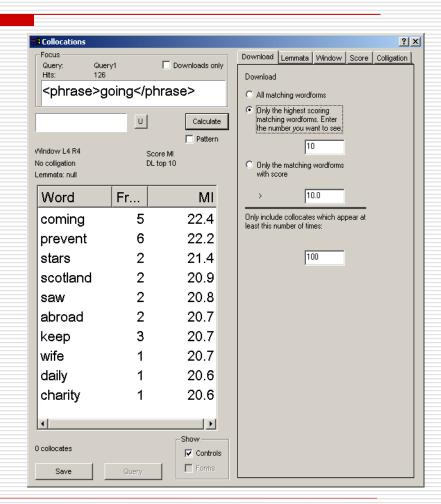
#### A word query

- Enter a stem or pattern
- See forms and frequenc
- See selected variants
- Look up hits
- Save wordlists



#### A collocation list

- After any query...
- See word forms that collocate with it
- Adjust window, filter by frequency, select score
- Look up collocates
- Save wordlists



#### A Keyword in Con

U

 $\overline{\bullet}$ 

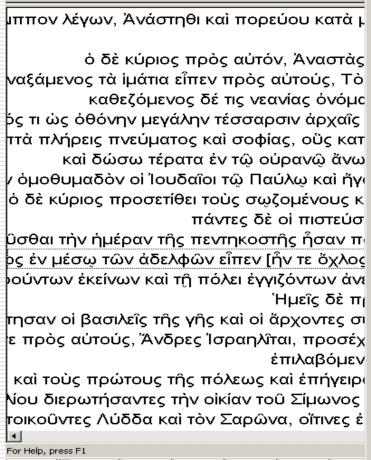
🔐 A 😰 💥 🕢

Window Help

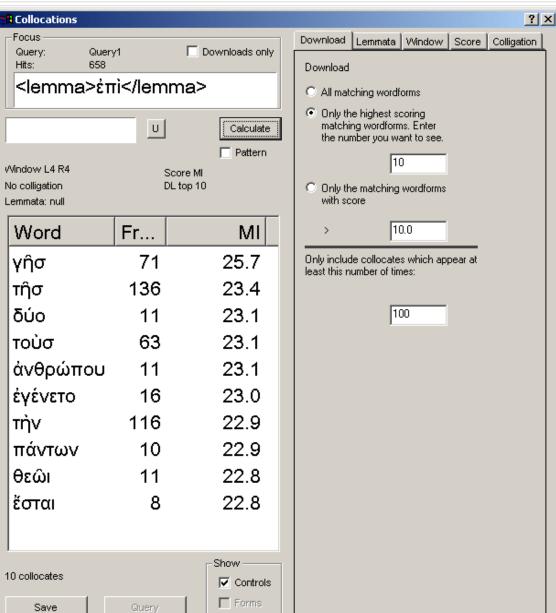
XAIRA - [Query1]

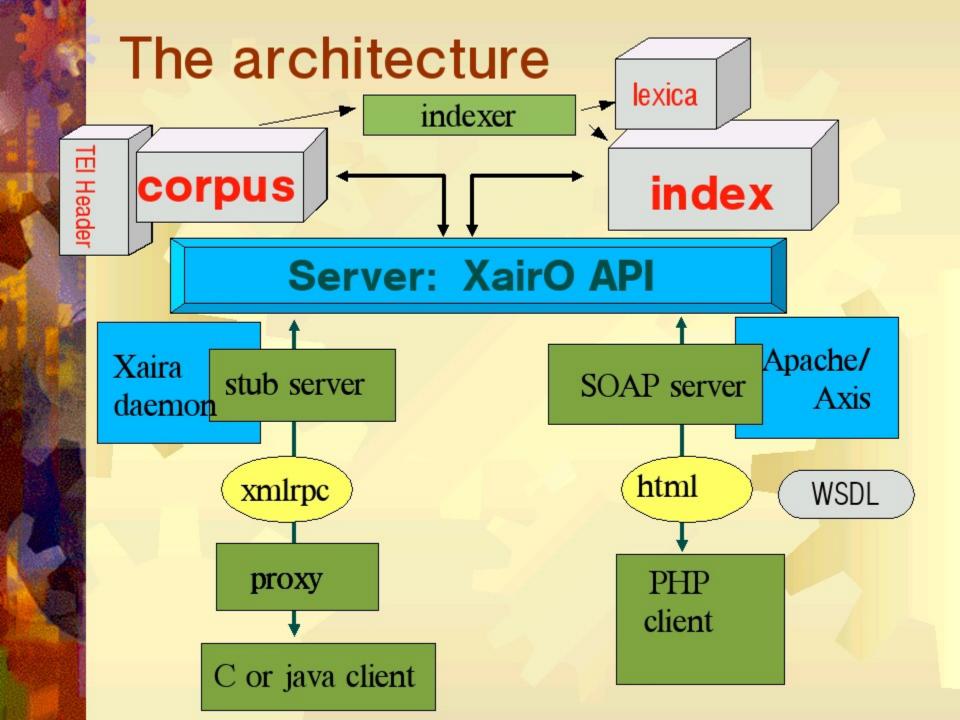
🚅 🖫 l 🞒 🔢

File Edit Ouery View



Start | | [EL... | [Exair... | 10] Mar... | 10] htt... | 11] newt | 12] goi





# XXQ: the Xaira XML Query Language

- Simple XML vocabulary for searching
  - word, punctuation mark, substring
  - word + secondary keys (e.g. POS)
  - XML start- or end-tag, plus attributes
  - Unicode-compliant regular expressions
- Facilities include
  - usual Boolean operations
  - sequence, disjunction, join
  - scoped searching, gaps
- Special lexical features

# XXQ more formally

- XXQ is a pattern matching language
- Returns manipulable streams of matching *locations* in a corpus
- XXQ query components

  - complex: <seq>, <and>, <or>
- matching may be literal or regexp

# XXQ examples

<form>fish</form>
returns locations of the form *fish* 

<lemma scheme="x">fish</lemma>
returns locations of forms which have
the lemma fish in scheme x

<addKey name="POS">VB</addKey> returns locations of forms which have an additional key of type *POS* with the value *VB* 

# Element vs pattern queries -1

- element query
  - Finds XML elements
    - Independent of their location
    - Restriction by attribute value
- pattern query
  - Restricts locations in terms of their XML context (by using Xpath)

### Element vs pattern queries -2

```
<element name="book">
  <attribute name="n">acts</attribute>
</element>
```

```
<pattern match="book[@n=Acts]">
<form>Paul</form>
</pattern>
```

#### combinations -1

<seq>: ordered sequence (possibly containing <gap>s)

```
<seq>
  <form>fish</form>
  <gap max="2"/>
  <form>chips</form>
</seq>
```

#### combinations -2

<and>: union operator

```
<and>
<form>fish</form>
<addkey name="pos">VERB</addkey>
</and>
```

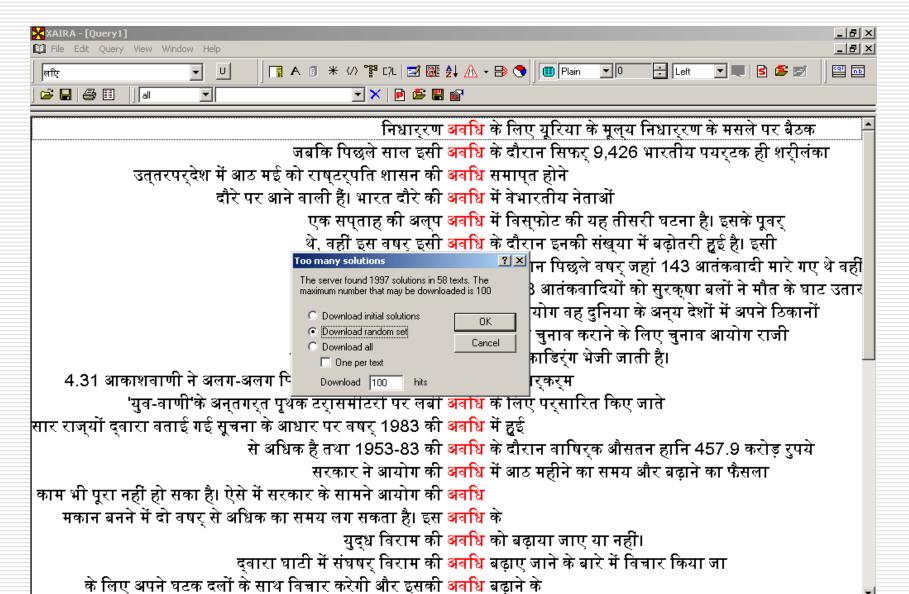
<or>: disjunction operator

```
<or>
<seq><gap/><form>fish</form></seq>
<seq><gap/><form>chips</form></seq>
</or>
```

#### This is all hidden behind the GUI, of course

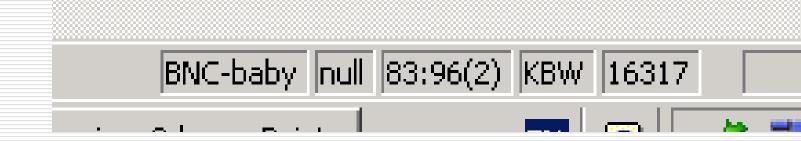
- Word and lemma query
- User-configurable display
  - plain, XML, user-defined stylesheets
- Texts, Results, Browse windows
- Results can be exported in XML
- "visual interface" for complex queries

# Exploring a text



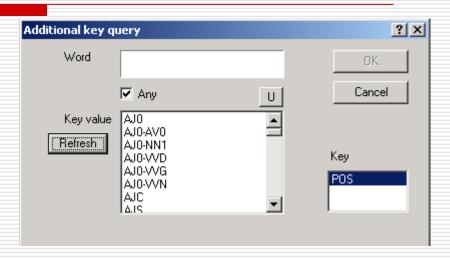
### Referencing system

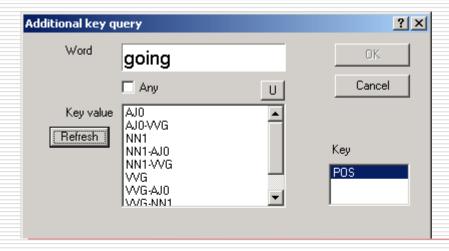
- Corpus
- Subcorpus
- Hit number : total hits (texts)
- Text identifier
- Context identifier



#### Additional keys (e.g. POS codes)

"find a POS code, whatever word it's attached to"

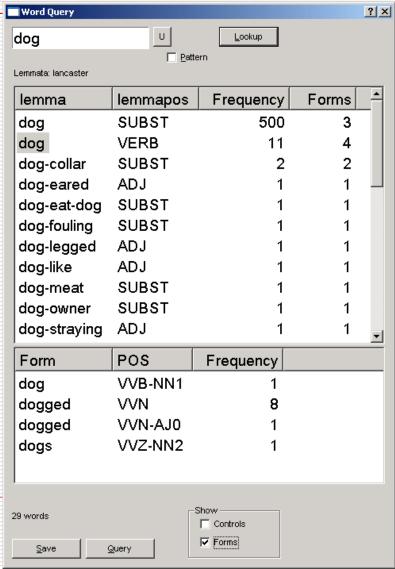




"find a POS code only when it's attached to 'going'"

#### Multi-part lemmatization

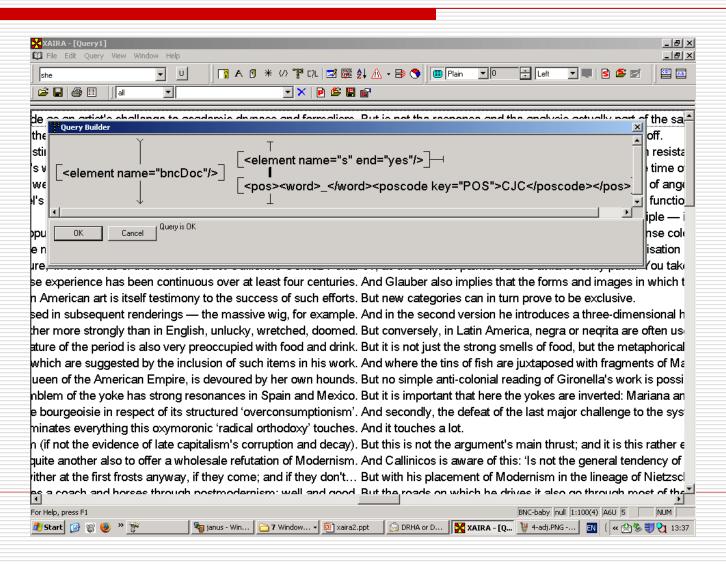
- DOG (verb)
  - Dog VVB
  - Dogged VVN
  - Dogs VVZ
  - Dogging VVI
- DOG (substantive)
  - Dog NN1
  - Dogs NN2
  - Dog NP1



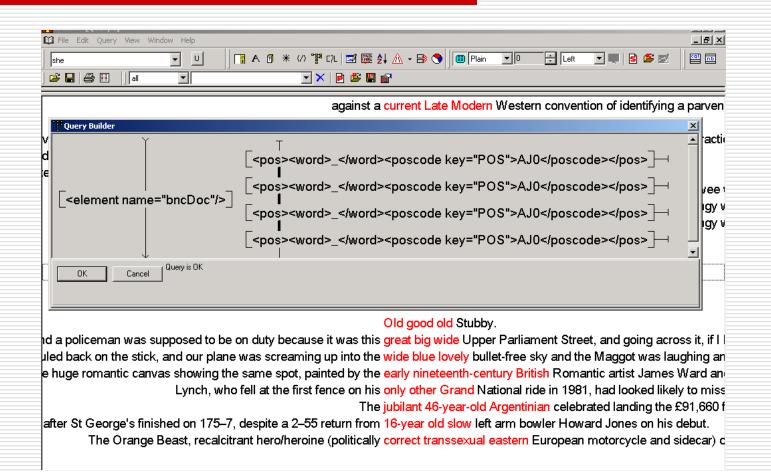
### Building complex queries

- visual interface
- scope node defines where to look
  - an XML element
  - by span
- query nodes define what to look for
  - word, phrase, addkey, pattern, XML
- link types define sequence in which query node targets should occur
  - next, one-way, two-way

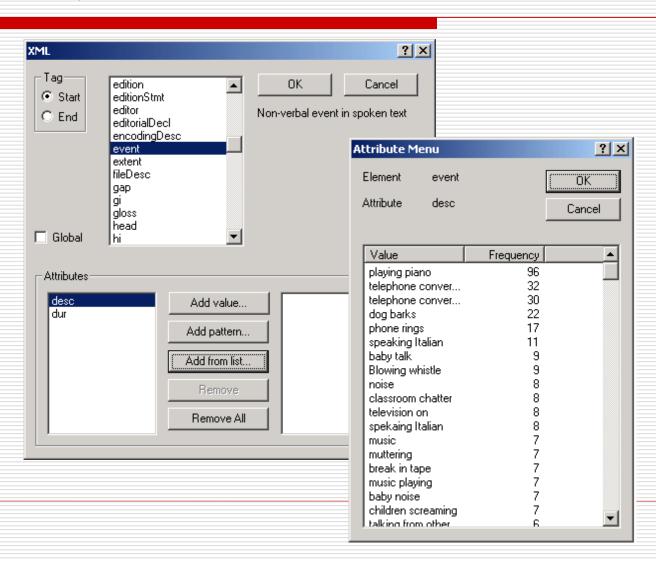
# Sentences beginning with conjunctions



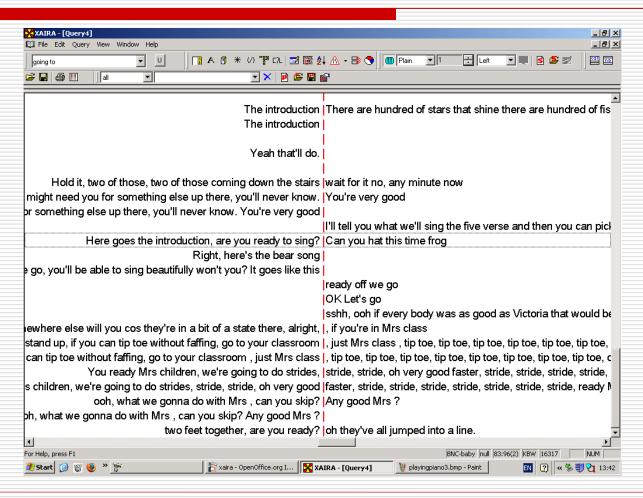
## Sequences of four adjectives



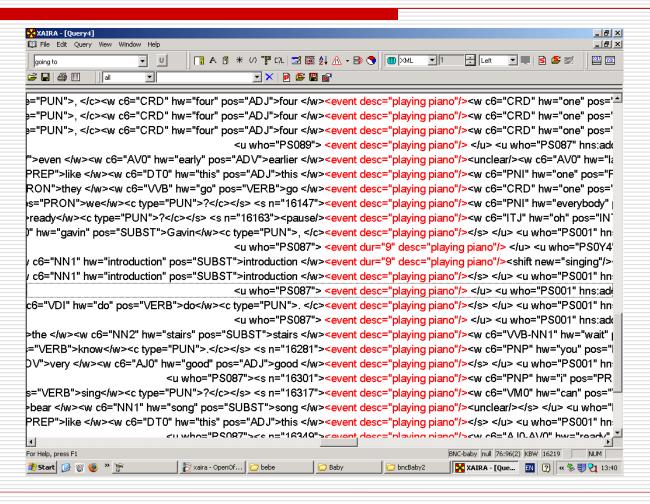
#### **XML** Queries



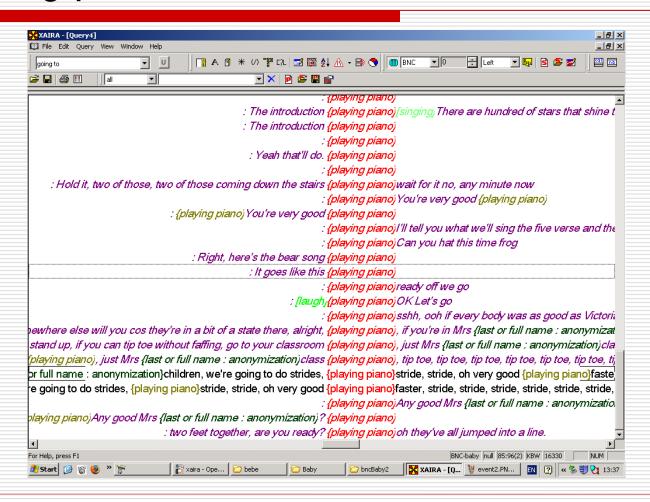
# "playing piano" – plain view



# "playing piano" – xml view



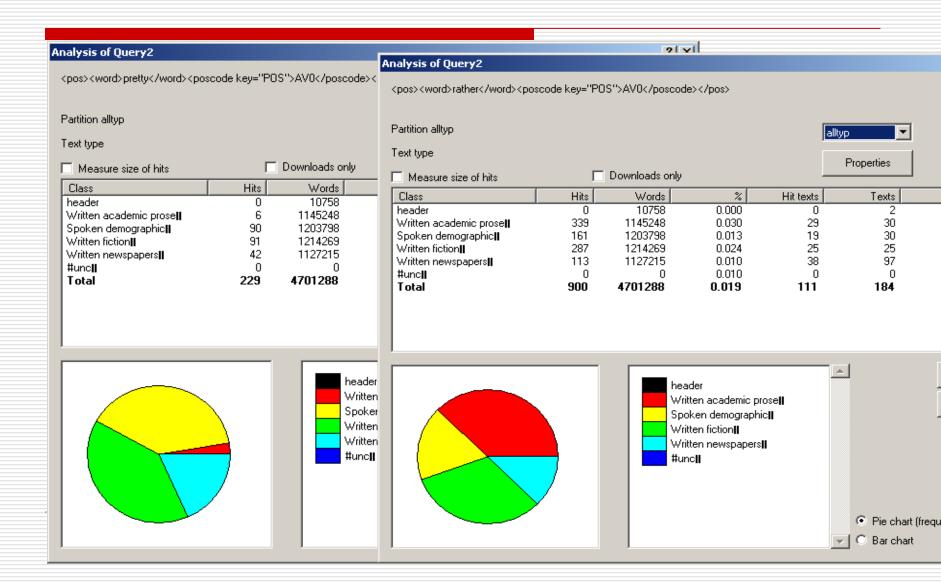
### "playing piano" – custom view



# Categorizing texts in a corpus

- A corpus can be partitioned in many ways
  - By pre-assigned category, domain, text-type, topic etc. specified in the markup
  - Manually or by running a query
- Searches can be restricted to texts of a given classification (subcorpora)
- Or results compared across the partition

# "pretty" vs "rather" in BNC Baby



#### What is XAIRA's niche?

- Web search engines
  - patchy and unknowable coverage
  - designed to recover content, not word forms
  - hard to cite, harder to process
- XML display engines
  - expensive, geared to reader not searcher
  - focus on presentation rather than content
- As a back end for your next generation web application

#### Interfaces available now

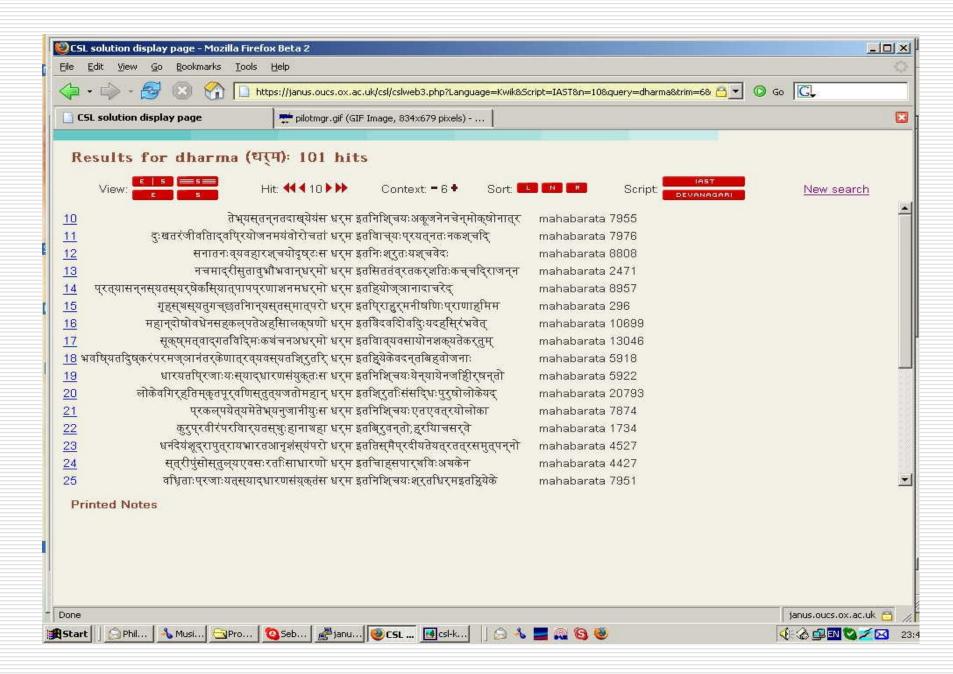
- Full-featured Windows client
  - can be run standalone or as interface to Windows,
     Unix, or Mac OSX server
  - includes simple corpus building utility
- Cross-platform
  - SOAP interface
  - Simple PHP and Java clients
  - Server and indexer
- Xpath engine

# Using Xaira as a back end

- PHP interface
  - supports all calls to the Xaira object model
  - suitable for any web programmer
  - platform independent
- For example...



**Printed Notes** 



### Frequently Asked Questions

- Does it work on a Mac?
- Does it work with non-XML text?
- How much does it cost?
- □ How do I make it do xxxx?
- Where's the documentation?