Using the XARA XML-Aware Corpus Query Tool to Investigate the METER Corpus

Robert Gaizauskas[a], Lou Burnard[b], Paul Clough[c] and Scott Piao[d]

Department of Computer Science[a]
University of Sheffield, Sheffield, S1 4DP
R.Gaizauskas@dcs.shef.ac.uk

Research Technologies Servicet[b]
University of Oxford, Oxford, OX1 2JD
lou.burnard@oucs.ox.ac.uk

Department of Information Studies[c]
University of Sheffield, Sheffield, S1 4DP
p.d.clough@sheffield.ac.uk

Department of Linguistics and Modern English Language[d]
Lancaster University, Lancaster,LA1 4YT
s.piao@lancaster.ac.uk

Abstract

The METER (MEasuring TExt Reuse) corpus is a corpus designed to support the study and analysis of journalistic text reuse. It consists of a set of news stories written by the Press Association (PA), the major UK news agency, and a set of stories about the same news events, as published in various British newspapers, some of which were derived from the PA version and some of which were written independently. The corpus has been annotated in accordance with the TEI Guidelines. The annotations include both descriptive metadata, such as the title, source, and date of publication, and human judgements about text reuse. To exploit the value of the annotations, for searching and querying the corpus, requires a tool which is designed to "understand" XML, or even, ideally, TEI. Such a tool is XARA, which has been built specifically to support corpus investigations over XML-annotated corpora. This paper reports lessons learned in using XARA to explore the METER corpus, particularly the importance of designing the annotation scheme with an understanding of the capabilities and limitations of the retrieval application.

1. Introduction

The METER (MEasuring TExt Reuse) corpus is a resource designed to support the study and analysis of journalistic text reuse. It consists of a set of news stories written by the Press Association (PA), the major UK news agency, and a set of stories about the same news events, as published in various British newspapers, some of which were derived from the PA version and some of which were written independently. The corpus has been annotated in accordance with the TEI guidelines. The annotations include both descriptive metadata, such as the title, source, and date of publication, and human judgements about text reuse. These judgements were made at two levels. Firstly, each newspaper story, as a whole, is labeled as wholly derived, partially derived, or not derived from a PA source. Secondly, for certain selected stories, every word in the text is labelled as either reused verbatim from the PA, rewritten from the PA, or new.

The corpus and the associated annotation scheme have been described elsewhere (Clough et al., 2002; Gaizauskas et al., 2001), and this description will not be repeated, beyond bare essentials, below. The current paper focuses on work we are carrying out to explore the corpus using the newest version of the SARA (SGML Aware Retrieval Application) corpus query tool, called XARA, which was originally designed to support exploration of the British National Corpus. This work has two main purposes:

(1) To verify and validate the TEI encoding scheme we have adopted for the METER corpus; i.e. have we encoded the correct things and have we encoded them correctly?
(2) To verify and validate the XARA query tool; i.e. does XARA allow us to retrieve the things we want to retrieve and does it retrieve these things correctly?

Specifically, we wanted to know whether XARA would allow us to answer queries such as: *Show me the titles of Telegraph stories wholly derived from the PA*; or, *Extract a sub-corpus consisting of murder stories with the catchline ``Axe"*; or, *Extract all sentences from the Sun stories which were rewritten from the PA*; or *Which words occur most frequently in the verbatim/rewritten/new portions of texts?*

More generally this work addresses the utility of adopting TEI encoding standards for corpus linguistic work. It also provides lessons into how to more effectively annotate texts for subsequent investigation and how to extend the capabilities of the XARA querying tool to better support corpus linguists.

2. The METER project

2.1. Aims and objectives of the project

The METER project[1] investigated the reuse of news agency (also known as newswire) sources by journalists and editors in the production of newspaper articles. While the reuse of others' text without acknowledgement is, in academic life, a cardinal sin, in the newspaper industry this is not only an accepted behaviour, but is in fact a standard business practice. In the newspaper industry, journalists depend upon "pre-fabricated" agency press releases as the source for many of their own stories: "most of what journalists actually write is a re-processing of what already exists" (Bell, 1991). Because newspapers subscribe to newswire services, they are at liberty to reuse source texts in any way they want (even verbatim), with or without attribution to the original author: the news agency.

Aims of the METER project included the following:

(1) To investigate and provide examples of text reuse by journalists writing for the British Press.
(2) To define a framework in which text reuse between the newswire-newspaper could be assessed.
(3) To build a corpus, a collection of carefully selected source and possibly derived texts for further study and evaluation of automatic methods of detection.
(4) To select and evaluate a range of algorithms to measure text reuse based on existing methods of computing similarity between natural language texts.

The METER project was supported by one of the largest UK news agencies, the Press Association, who allowed us access to the full newswire service as supplied to journalists. Being a primary supplier, the news issued by the PA is widely reused, either directly or indirectly, in British newspapers.

The study of text reuse has, aside from its intrinsic academic interest, a number of potential applications. Like most newswire agencies, the PA does not monitor the uptake or dissemination of copy they release because tools, technologies, and even the appropriate conceptual framework for measuring text reuse are unavailable. For news agencies like the PA who release vast amounts of news material everyday (on average the PA release over 1,500 stories each day), manually analyzing stories for text reuse is not only impractical, but with limited resources is also infeasible. For the PA, potential applications of being able to automatically measure reuse of their text accurately include: (1) monitoring of source take-up to identify unused or little used stories; (2) identifying the most reused stories within the British media; (3) determining customer dependencies on PA copy, and (4) new methods for charging customers based upon the amount of copy reused. This could create a fairer and more competitive pricing policy for the PA[2].

2.2. The METER corpus

To support our investigation of text reuse, we created a corpus of newswire and newspaper texts called the METER corpus. This corpus, the first of its kind as far as we are aware, provides over 700 examples of text reuse between Press Association source texts and subsequent articles published by nine newspapers in the British Press who subscribe to the PA news service. The corpus consists of 1,716 texts selected from the period between 12th July 1999 and 21st June 2000 from two staple and recurring domains in contemporary British Press: (1) law and court reporting, and (2) show business and entertainment. Newspaper and newswire texts collected for the METER corpus vary around a number of parameters which were taken into account during the corpus construction. These include the following: (1) domain, (2) source, (3) time period, (4) newspaper register, (5) length of newspaper story, (6) coverage of the news topic, (7) degree of reuse, and (8) the number of newspapers reporting the same story. Texts were selected to represent these parameters and their influence on text reuse was taken into account during analysis.

[1] METER was a 2-year UK EPSRC-funded research project – for details see: http://www.dcs.shef.ac.uk/nlp/meter.

[2] For more information on the PA and public access to a portion of their newswire, see: *http://www.ananova.com*.

As well as selecting source texts for the corpus, professional journalists also analysed the PA and newspaper text pairs and recorded the derivational relationship believed to exist between them according to a two-level, three-degree scheme for classification. At the *document level*, all 944 newspaper articles were classified as either: (1) wholly derived, (2) partially derived, or (3) non-derived, reflecting their degree of dependency on the PA source text for the provision of news material. A more fine-grained scheme, at the *lexical level*, aimed to capture the reuse of text within the newspaper text itself and classified word sequences as either: (1) verbatim, (2) rewritten, or (3) new. At this level of reuse, 445 texts only were analysed and annotated due to limitations of time and resource. These derivation relationships were captured in the corpus as pragmatic annotation – interpretive information added to the base text to capture a professional journalist's view of text reuse between a PA source text and corresponding newspaper article.

2.3. Annotating the corpus in TEI/XML

To explicitly capture characteristics of the newspaper texts, e.g. their headline, author, date of publication, catchline, domain and source, as well as the derivation relationships existing between the newswire-newspaper counterpart texts, we originally created an SGML markup scheme in a beta release of the corpus. However, we later transformed the entire collection into XML conforming to the Text Encoding Initiative (TEI) standard with the goal of making the resource compatible with international standards for corpus encoding and hence facilitating exchange with other members of the corpus linguistics community.

The corpus is stored physically in 27 separate files, one for each day on which stories were sampled. A global corpus header file contains information about the corpus as a whole, including publication information and the definition of attributes specific to the METER project (e.g. the document and lexical level annotation schemes). Within each day file, material is organized into catchlines – a group of articles from different sources all dealing with the same story, which is identified by a PA-assigned tag known as a catchline (e.g. "axe" for a story about an axe murderer). Within catchline are the individual articles as published by the PA or newspapers.

Figure 1 shows how we have captured this structure using TEI tags and attributes. The TEI scheme allows for the renaming of tags, but we chose to use standard names for simplicity; in some cases, therefore, the tag and attributes names do not necessarily provide an intuitive abbreviation of what they are intended to represent. Each file consists of a header (encapsulated within the <teiHeader> tag), followed by a collection of catchlines for both PA and newspaper texts within a <text> and <group> tag. Each catchline is defined by a <text> and <body> tag. The <text> tags are not shown in Figure 1 to simplify the diagram. Within <body> tags the <div> tag is used to indicate individual newspaper articles or pages of PA text for this catchline. At the most detailed level of annotation, each text consists of paragraphs and sentences, denoted by <p> and <s> tags. Within sentence the <seg> tag is used to indicate portions of verbatim, rewritten and new text. More information about the structure of the XML/TEI corpus can be found in Clough et al. (2002).

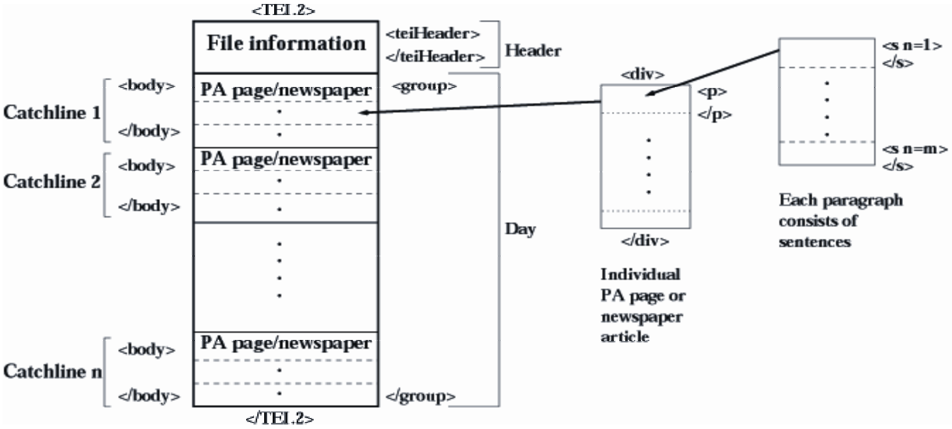


Figure 1: The TEI markup of individual days in the METER corpus

Table 1 lists, in order of their hierarchical structure, the key TEI tags and attributes which are used to encode the METER corpus, as shown schematically in Figure 1.

TEI tag	Attributes	Values	Comment
<text>	id n	e.g. "M01032000-1" e.g. "Burstein", "mother", "wagstaff"	Unique catchline ID Name of catchine
<div>	id n type ana	e.g. "A636" e.g. "pa-01032000-15" "courts" or "showbiz" "src", "wd", "pd", or "nd"	Unique ID (A=PA, M=newspaper) Reference for each news story Domain for this catchline Document-level classification
<head>	Type	e.g. "pa", "sun", "mail", "times", "guardian"	The source of the news article
<s>	N	1..N	Sentence number in the text
<seg>	Ana	"verbatim", "rewritten" or "new"	Lexical-level classification

Table 1: Some of the TEI tags and attributes used in the METER corpus

3. XARA – the XML Aware Retrieval Application

3.1. Background

SARA ("SGML-Aware Retrieval Application") was originally developed with funding from the British Academy and the British Library to meet the need for a simple searching interface to the British National Corpus (BNC), when this was first distributed in 1994. The original design brief was to survey freely-available text retrieval software libraries and to build on these a simple user interface which could also be distributed without additional licensing costs. At this time the amount of freely available SGML-aware software was small, and it was rapidly decided that it would be quicker and more cost-effective to develop our own within the BNC project.

SARA was accordingly developed with the very specific needs of the BNC in mind, incorporating a number of unusual design features unique to that resource (such as the specifics of how words are tagged, and speakers in its spoken texts are identified), while excluding a number of more generic but expensive features (such as indexing of POS tags). The system was also designed with a particular software environment in mind: one in which researchers used desktop machines to access local server machines, probably running a dialect of Unix, rather than standalone desktop systems; and in which data were counted in megabytes rather than gigabytes. Because it was intended for use with only one specific data set, little thought went into generalizing (or even distributing) the component which created the index files used by the server to optimize access to the corpus. For similar reasons, little effort was put into optimizing the structure of those indexes, or extending them to cope with user requirements beyond those already identified within the project.

Some consequences of these design decisions are readily apparent: it is designed to operate in a network environment, with the bulk of data storage and much of the processing carried out on a central server. The overall functionality of the system is constrained by the expressivity of the protocol by which clients communicate with the server. This is probably a good thing, since the availability of that protocol makes it easier to develop new clients in new environments; constraints consequent on the need to support only the BNC are less good however, since the fixed index format made it difficult or impossible to support additional requirements such as efficient identification of collocates or POS searches.

3.2. New directions

In redesigning the tool, the primary objective was to take advantage wherever possible of the availability of XML and of XML encoded data, reflected in a change of name – from SARA to XARA. This decision conveniently allows us also to leverage a number of powerful technologies: most significant of these is probably Unicode, which not only enables us to support corpora in any of the world's current languages, but also allows us to rely upon standard methods of tokenization and character classing. XARA will operate on any well-formed XML document, without the need for any DTD or detailed tagging; it will also make use of whatever DTD it is supplied with to enrich the functionality of the supplied data.

After nearly a decade of use, we were much better aware of the problems reported by users of the BNC and SARA, and felt confident that we now knew which of the many possible additional enhancements we might add would be of most general usefulness to our target user community: some of the new features are listed below. As well as relying on Unicode and XML, XARA uses other recognised standards for its communication with other systems and with the outside world. The metadata required by the indexing system is all recorded using the TEI header (though there is no requirement that the rest of the corpus follows this particular XML application); the scripting language offered for application development is based on Javascript; the formatting language is a simple subset of CSS, the W3C's Cascading Stylesheet Language; and we are also considering how best to re-express the query protocol itself in an existing XML vocabulary.

3.3. Technical Overview

The XARA system combines the following components: (1) an *indexer*, which creates inverted file style indexes to a large collection of discrete XML documents; (2) a *server*, which handles all interaction between the client programs and the data files; (3) a *Windows client*, which handles interaction between the server and the user.

In addition, an index building utility, called Indextools, is supplied with XARA, which simplifies the process of constructing a XARA database. Its chief function is to collect information about the corpus to be supplied additional to that present in any pre-existing corpus header, and to produce a validated and extended form of the corpus header. It can also be used to run the indexer and test its output.

3.3.1. Functionality for indexing

In order to process a large amount of textual data marked up in XML, XARA uses a pre-built index to optimize access to the original source texts for particular (largely lexically-motivated) kinds of query. To construct this index, the software needs information about:

- how PCDATA (element content) is to be tokenized
- how tokens are to be mapped to index terms (for example, by lemmatization or by the inclusion of additional keys)
- how index terms are to be referenced in terms of the document structure
- how and whether XML tags and attributes are to be indexed

Much, perhaps most, of this information is implicit in the XML structure for a richly tagged corpus: one could imagine a corpus in which every index term was explicitly tagged, with lemma, part of speech, representation etc. In practice however, such richly tagged corpora remain imaginary, and software performs the largely automatic task of adding value to a document marked up with only a basic XML structure. XARA allows for the specification of how that task is to be performed in a standardized way, using the existing structure of the TEI header (see http://www.tei-c.org/Guidelines/HD.htm) as the vehicle for its specification.

3.3.2. Functionality for searching

XARA inherits from SARA a rich range of query facilities, but adds to them considerably. The system allows the user to search for substrings, words, phrases, or the tags which delimit XML elements; it also supports a variety of scoped queries, searching for combinations of words etc. in particular contexts, which may be defined as XML elements, or as combinations of other identifiable entities, as further discussed elsewhere in this paper. Searches can be made using additional keys such as part of speech, or root form of a token, specified either explicitly in the tagging of the texts, or implicitly by means of a named algorithm.

Outputs from a search usually take the form of a traditional KWIC concordance, which can be displayed or organized in several different ways, under user control, or can be exported as an XML file for use with other XML-aware systems such as a word processor.

Corpora can be reorganized or partitioned in a user-defined way, using the results of any query, the values of specified element/attribute combinations, or by means of manual classification. Searches carried out across partitioned corpora can be analysed by partition: so the client can display the relative frequencies of a given lexical phenomenon in texts of all different categories identified in a corpus.

4. Using XARA to explore the METER corpus

4.1. Indexing the METER corpus

To index the METER corpus, we used the XARA Indextools utility. Given a TEI corpus header file and a group of XML/TEI texts, the indexer tool creates a searchable index from the texts, which can then be explored using XARA. XML/TEI validation is carried out during indexing.

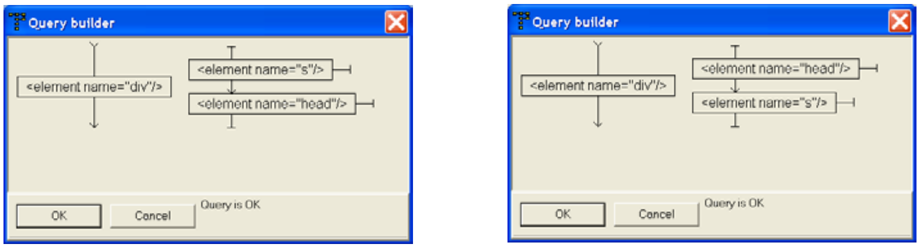
One aspect of XARA which has important implications for subsequent searching, is the requirement for the user to specify a notional *document* and *unit* during indexing. The document can be any XML element. Its significance is that partitions of the corpus, or sub-corpora, can only be defined in terms of sets of whatever element is chosen as document. The unit can also be any XML element; its function is to identify the lower level context within which hits are identified. We chose <div> to be our notional document, i.e. the individual newspaper or PA article, and <s>, or sentence, to be our notional unit. Thus we can create partitions, or sub-corpora, of articles matching various search criteria and lexical search results are displayed in terms of the sentences in which they occur.

4.2. Searching the METER corpus

Once indexed, the corpus can be searched. XARA offers a number of query options including: word, phrase, part-of-speech, pattern and XML-based searching. There are at least four specific features of XARA which are useful for querying the METER corpus: (1) the Query Builder tool which enables users to create complex queries using a visual interface, (2) the ability to define a partition of the corpus, dividing it into sections according to a number of different criteria, (3) the way in which search results may be presented to the user and (4) how search results may be saved for subsequent exploitation. We describe these capabilities in this section, then present a set of example queries in the next section. More detailed information about searching with SARA can be found in Aston and Burnard (1998).

Query Builder

To create complex queries the Query Builder tool can be used, the visual interface for which is shown in Figure 2. The left node is called the *scope node* and defines the scope or context in which a complex query is executed. For example, if the scope node is defined as the <div> annotation, the search will take place on text and annotations only within the scope of this annotation – a news story in the case of the METER corpus. To fully specify the query, one or more right-hand nodes must be specified using any of the query options, e.g. pattern, word, phrase or XML annotation. Nodes can be added both vertically and horizontally. Query nodes added vertically are logically ANDed together to narrow the context in which the search is executed. Nodes added horizontally represent alternatives, i.e. are logically ORed. For vertical connections, the *link type* between nodes can be specified as either next (i.e. adjacent), not next, one-way, or two-way, all indicating the order and proximity in which matching search terms must be found.



<element name="div"/>
<element name="s"/>
<element name="head"/>

<element name="div"/>
<element name="head"/>
<element name="s"/>

Figure 2: Ordering of nodes can affect the query: left-hand query produces no solutions; right-hand one produces 21,209 solutions

Searches based on annotations and one-way link types are sensitive to the hierarchical structure and order of the corpus annotations. For example, consider the problem of listing all sentences from news stories. Two versions of the query are shown in Figure 2. The query on the left returns no solutions, whereas the query on the right returns all 21,209 sentences in the METER corpus. The difference is the ordering of nodes in the query. The left-hand query does not work because the ordering requires matching sentences first and then the headline, but sentences always come after the headline.

Defining partitions

XARA allows the user to create sub-corpora, or *partitions*, each representing a different grouping of the documents in the whole corpus. By this means the user can select a portion of the corpus which is of interest and perform further analysis on this portion only. Partitions can be created by either selecting texts from the user interface or by using a query to partition the corpus into documents that match the query and those that do not. Once a partition has been created, it can be activated and queries are addressed to the activated partition only. By partitioning the METER corpus, queries aimed at one particular subset of the corpus, e.g. a particular newspaper or only articles derived from the PA, can be easily created, thereby reducing their complexity. This capability highlights the importance of (1) ensuring that the textual units over which one wants to aggregate results are defined as XML elements in the building the corpus, and (2) selecting the appropriate XML/TEI element as the notional "document" in indexing the corpus with XARA.

Viewing the search results

After executing a query in XARA, the results are displayed as either a listing or each result is displayed individually. The listing usually takes the form of a KWIC concordance in which results can be viewed as plain text, in XML showing the annotations, or in a custom-format using a user-defined Cascading Style Sheet (CSS). The context of the listing will depend on the search itself, but can be broadened and narrowed by the user. The listing can be exported as an XML file which can be viewed or edited with other XML-aware applications.

Saving search results

XARA enables the user to save various outputs including word frequency lists produced during a word query and the results of query analysis, in XML or comma/tab separated plain text formats. The XML results can be parsed by and viewed in any XML-aware tool.

4.3. Sample Searches

In this section we present a number of example questions that one might like to ask of the METER corpus, and discuss whether and how XARA can support these requests. These fall into three classes: those addressing the lexical content of the corpus only, those addressing the XML annotations alone, and those combining lexical content and annotations.

4.3.1. Lexical searches based on the content of the news stories: free-text searching

(1) Find me which texts contain a given word or phrase

Suppose we begin with a simple word or phrase search, for example: "find me all the texts which contain the word 'Ipswich'". This can be performed using the "word query" option which matches 30 occurrences of "Ipswich", and 1 occurrence of "Ipswich's". At this point, the word frequency table can be saved in XML, or displayed as a KWIC concordance. As well as exact match searches, a regular expression can also be defined to search for a pattern, e.g. "town*" finds the words "town", "town's", "townhill", "townhouse" and "townend". Phrases can also be searched for, e.g. "Ipswich town", although this option does not permit pattern searches.

(2) Find me which words co-occur with a given word

XARA can be used to find collocations, a common task in corpus linguistics. A Z-score or Mutual Information (MI) score indicates the "strength" of collocate and can be used to rank the results. Collocates of "Ipswich" include "CID", "borough", "town" and "crown". Again, this listing can be viewed and/or saved.

(3) Find me which words are most frequent in the METER corpus

To create a list of word frequencies, the word query option is used and the search term defined as a pattern, e.g. [a-zA-Z-]*. This calculates word frequency *based on the entire METER corpus* and the results can be saved to a file in the form <word><frequency> (e.g. comma-separated or XML tagged). It would be useful to be able to filter certain words, e.g. function words, which are invariably most frequent, from these lists, either at search or index time: XARA does not support this directly, though it is possible to order lists by frequency and specify a cut-off point.

When a partition is active, word frequencies are provided for each component of the partition, allowing one to compare the relative frequencies of the given search term across the partition.

(4) More complex lexical searches

As with most search engines, XARA can be used to generate complex queries using Boolean operators, such as "find me which texts contain the words X and Y", or "find me which texts contain the words X or Y". The Query Builder tool can be used to perform this search task, but also supports more sophisticated searching by enabling the user to limit the scope and order with which the query words match. For example, the search "Find me all texts containing Ipswich and council" can be defined in Query Builder as a search for two vertically-placed word nodes within the scope of a <div>. With the link type defined as next, the query requires the word "council" to *immediately* follow the word "Ipswich" (equivalent to a phrase search), a 1-way link means that "council" must follow "Ipswich" within the scope of <div> (i.e. ordered), and 2-way link implies the words can occur in any order, but still within the same scope.

4.3.2. Searching on the XML annotations

Using the XML markup, we are able to exploit the interpretative information added to the METER corpus to answer queries, which cannot be answered from the lexical content. As with searching over lexical content, XARA supports complex searching over annotations too, enabling us to answer questions such as "Find me all texts from the Sun newspaper derived from the PA".

What you can do with XARA depends crucially on the annotation scheme; therefore, careful design of this scheme is paramount. For example, in the initial TEI markup of the METER corpus, we did not use the "type" attribute of the <head> tag to identify the article source, but only to indicate whether the source was a newspaper or a PA text. The specific source could only be extracted from a substring of the "n" attribute on the <div> tag (e.g. "sun-01032000-3"). The version of XARA we used did not support pattern searches on the XML attribute value strings: our solution was to extract the source name from the <div> annotation, using a Perl script, and change the "n" attribute to reflect this, i.e. to modify the annotation scheme.

(1) Show me how many newspaper stories are in the METER corpus

When constructing an XML query, selecting the <head> tag and "type" attribute displays all possible attribute values and their counts, giving the number of articles from each newspaper in the corpus. This is true for all attributes, e.g. selecting the <div> tag and "ana" attribute enables the user to determine there are 205 non-derived texts, 438 partially-derived and 301 wholly derived.

(2) Extract all titles from the Sun which are wholly derived from the PA

This can be performed using Query Builder to create the query shown schematically below. Scope is defined to be within <div>, and the additional query constraints are that the "ana" attribute has be "wd" (wholly-derived) and the <head>'s "type" attribute must be "sun". The result is a list of 19 headlines.

Scope node:
<element name=”div”/>

Query term nodes:
<element name=”div”><attribute name=”ana”>wd</attribute></element>
CONTAINING
<element name=”head”><attribute name=”type”>sun</attribute></element>

As a further example, to select all wholly-derived texts from the Sun for the courts domain, the preceding query need only be modified by adding a further constraint on the scope node (“type” equals “courts”).

(3) Extract all sentences from the Sun which were taken verbatim from the PA

To extract verbatim sentences a query can be defined to match sentences containing a <seg> tag with the “ana” attribute value equal “verbatim. If during indexing the unit is specified to be <s>, as we have assumed, the results will be displayed within the context of the sentence as required. Searches such as “Show me all verbatim segments from the Sun”, or “Show me texts containing verbatim segments from the Sun” require re-indexing the corpus, defining the unit to be either <seg> or <div> respectively.

4.3.3. Combining searches

Using the Query Builder tool, search requests which involve combining both information from the annotations and free-text can be constructed, enabling us to answer questions such as the following.

(1) Show me all titles from the Sun with the word “Axe” in the headline

We define the scope node to be <head> with the attribute “type” set to match “sun”. By adding a word query node, as shown below, to match the word “axe”, two titles are returned: “Sex taunt made axe monster murder 3 in love triangle” and “Street to axe Ravi”. Note that in the first, “axe” is used to modify the noun “monster”, the context being a murder, whereas in the second “axe” is a verb which in this story is used to headline a story about the actor Ravi being removed from the British soap opera, Coronation Street. This reveals some interesting characteristics of the kind of newspaper language used by the popular press.

Scope node:

```
<element name=”head”>
  <attribute name=”type”>sun</attribute>
</element>
```

Query term nodes:

```
<or><lemma>axe</lemma></or>
```

(2) Extract a sub-corpus consisting of all murder stories with the catchline “Axe”

Start by defining a query whose scope node is <div> and whose “type” attribute is “courts” (this category defines murder stories). Then, add a word query node of “axe”. This results in 45 stories from both newspapers and the PA (this could also be filtered to allow, e.g., just newspaper stories). A new partition is defined based on this query, saved and the partition activated to limit further queries to just these texts.

(3) Which words occur most frequently in the verbatim, rewritten and new portions of the texts?

This search is more complex than the preceding because it involves creating a new XML file based on the results of a query, and then re-indexing this file to compute word frequencies. First, we index the METER corpus with <div> as the document, and <seg> as the unit to create a listing with a context of just <seg>. This listing is saved in XML and used as the basis for a new index. This index is then loaded into XARA and word frequencies can again be computed.

5. Conclusions

The METER corpus was encoded in TEI/XML format to enable the data to be shared with other members of the TEI community, and to enable generic XML and TEI parsers and search tools to be used, avoiding the need to build custom applications. The goals of the work described in this paper were (1) to verify and validate the TEI encoding scheme we have adopted for the METER corpus and (2) to verify and validate the XARA query tool. We pursued these goals by examining a variety of questions related to the annotations and lexical content of the METER corpus, questions which we hoped the corpus annotation and XARA would enable us to answer, and used XARA to try to answer them.

Our first observation is that the TEI encoding of METER is in fact valid – by indexing the corpus using XARA we have been able to verify that the texts are encoded in valid XML. Secondly, the encoding was verified to the extent that it allowed us to answer all of the questions that we set by ourselves. However, a deeper question is whether the TEI encoding was really necessary. Certainly, there can be no doubt that an XML encoding of some sort is highly useful, enabling searches over the structured annotations, as well as combined searches over lexical content as well as the structured metadata. However, it is not clear that for the METER corpus annotating the corpus in TEI provides any additional benefit beyond encoding the corpus in XML. For those who are not familiar with TEI – in this case the creators of the corpus when we set out to build it – TEI involves a steep learning curve. Using TEI offers the corpus encoder a choice of predefined tags and attributes, at the expense of sometimes difficult decisions about how exactly to map the structural units of the corpus into the available TEI elements and their associated attributes. XARA works perfectly well with any XML files, so the benefits it offers could be derived without adopting TEI. While for many applications a ready-made tag and attribute set of the sort TEI offers may be entirely adequate and promote sensible annotation, for others it may be the case that carefully designing a bespoke annotation hierarchy and attributes from the first principles is a better use of time than shoe-horning a problem into the framework that TEI provides.

One specific problem we encountered illustrates this issue and the interaction between the annotation scheme and the retrieval tool. When encoding the METER corpus in TEI, we annotated sentences and then annotated verbatim, rewritten or new word sequences using the TEI <seg> within sentences. Since XML does not permit overlapping elements, the result was that verbatim, rewritten or new word sequences that spanned sentences were split into multiple shorter sequences that fit within sentences. However, our research into detecting text reuse found that the *length* of verbatim matches was a key indicator of reuse. The TEI encoding scheme we had adopted precluded determining whether adjacent verbatim word sequences in different sentences were part of a single longer verbatim sequence. One solution to this difficulty is to use a link attribute to chain together multiple connected segments. Another solution would be to do without sentence annotations since they are not essential to the METER aims. However, this is not entirely satisfactory because not all texts in the corpus were annotated to the same degree of granularity, i.e. some are annotated to the level of word sequence and some only to the document level. Certain degree of markup more fine-grained than the whole text, e.g. the sentence, is required to enable the results of search requests in XARA to be of practical benefit.

The second aim of this paper was to verify and validate the XARA query tool as a possible search tool for release with the METER corpus. XARA has been able to support all of the initial questions we set out to answer, and has so far as we are aware, correctly answered all of them. This included queries concerned with lexical content only, those concerned with structured annotations only, and combinations of the two. The only limitations we discovered in the course of this study were that (1) partitions cannot be saved in a XML format that can be read by other applications, (2) XARA does not support pattern match searching over XML attribute value strings, which can be useful when values assigned to attributes are non-atomic. This facility has subsequently been added to the system.

A final lesson for would-be corpus creators is this: before committing to an annotation scheme, learn about the capabilities of a XML query tool such as XARA, identify key queries you would like the scheme to support and assure yourself that the query tool and annotation scheme together support the queries.

References:

Aston, G. and L. Burnard. 1998. The BNC Handbook: Exploring the British National Corpus with SARA. Edinburgh Textbooks in Empirical Linguistics, Edinburgh University Press, Cambridge, UK.

Bell, A. 1991. The Language of News Media. Blackwell, Oxford, UK.

Clough, P, R. Gaizauskas, and S. L. Piao. 2002. Building and annotating a corpus for the study of journalistic text reuse. In Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC02), pp. 1678-1691.

Gaizauskas, R., J. Foster, Y. Wilks, J. Arundel, P. Clough, and S. Piao. 2001. The METER corpus: A corpus for analysing journalistic text reuse. In Proceedings of the Corpus Linguistics 2001 Conference, pp. 214-223.

Sperberg-McQueen, C. and L. Burnard, editors. 1999. Guidelines for Electronic Text Encoding and Interchange. TEI P3 Text Encoding Initiative, Oxford, UK, revised reprint edition.