УДК 004.6

TOWARDS EASIER QUERYING OF XML-BASED LINGUISTIC CORPORA

© Gladkova G.P., Drozd A.A.

Kiev National Taras Shevchenko University
Institute of Philology
Department of English Philology
01601 Taras Shevchenko Boulevard 14 Kiev, Ukraine
e-mail: *anna.gld@gmail.com*

Moscow State University (Sevastopol Branch)
Programming Department
99000 Geroev Sevastopolya 9, Sevastopol, Ukraine
e-mail: *alexander.drozd@gmail.com*

Abstract. The paper is devoted to evaluation of general-purpose XML querying tools in respect to linguistic corpora. A specialized pattern-based query language is suggested and implemented in XCorp software.

Introduction

Corpus linguistics is one of the most actively developing trends in applied linguistics. Corpora are widely understood to be merely a "large bodies of machine-readable text containing thousands or millions of words" [6, p.48], and many popular tools for corpus analysis like Antony Lawrence's AntConc [2] presuppose the input to be simple plain text files. But current tasks in the spheres of phonology, semantics or syntax of a natural language require more complex annotation of linguistic data, not to mention issues in pragmatics and cognitive analysis of language. This leads to the problem of incorporating additional data in the text and complex querying of this information.

Corpora may be stored in a variety of formats, including the so-called vertical format and SGML. While these formats may be more advantageous for certain kinds of tasks, the most flexible solution remains to be XML, which is proved by the fact that many corpus projects have developed their own XML-based formats optimized for storage of task-specific information (well-known examples are generic TEI XML, TigerXML etc). Moreover, a great many utilities for tagging of the text on the levels of syntax and morphology can produce XML output. Yet while the XML format itself is flexible and may be tailored to meet the needs of a particular study with some basic knowledge of applied linguistics, querying the resulting data poses a more serious problem. The aim of this paper is to analyse the data model used in corpus linguistics and the applicability of the standard XML querying tools in this sphere, as well as to suggest a more convenient specialized querying tool.

The general issues of data retrieval from XML databases are discussed in variety of sources [4,5,11,12], but one can judge of the applicability of general XML queries for corpus linguistics by the fact that all major corpus projects generally develop their own tools for querying their data (e.g. Xaira for British National Corpus [3]). Therefore there exist a number of solutions developed for specialized annotation sets (e.g. TigerSearch for TigerXML). The problem of general applicability of standard tools is aggravated by the fact that they are all developed for specialists in IT and may be difficult to use for linguists.

1. General purpose tools for quiering XML

XML is claimed to be the universal format for data representation, and a great many universal solutions has been developed for querying XML data. However, as often is the case, universal solutions may be less suitable for specific tasks. First of all, XML itself, as well as the default XML querying tools, has been developed for usage in other information processing paradigm: its main purpose is storage and retrieval of structured data of database-like type, where all elements of the same level are considered equal and no importance is given to their consecutive order (employee profiles or movie collections are the typical examples in XML tutorials). In fact such XML files are merely an alternative format to a database, and typical queries for such files much resemble database requests (e.g. "find all the employees with salaries higher than 1000$").

However, linguistic corpora possess certain characteristics that make the standard XML querying tools less suitable for them. Elements of a text in a natural language are sequences of words that combine into phrases, sentences and paragraphs. The order of those elements is important for the researcher, and so is the distance between the elements. Sometimes linguists need to combine the annotation data with the patterns present in the plain text data in their search requests. For example, a study in alliteration may involve searching for sequences of words starting with the same consonant at a particular distance. A study in word-formation may require searching for roots and words derived from them occurring within one sentence or one paragraph. The register of letters may be important or not for a given task. Some of these difficulties may be solved by means of standard XML querying tools, but this might pose some difficulties even for an expert in IT sphere, while for an average linguist they turn into an unsolvable problem.

The standard means of querying XML data is XQuery language, developed by W3 Consortium as well as XML itself. However, the current version of XQuery is poorly suited for use with XML-annotated text corpora: typical tasks involving search for sequences of elements in a given order are very difficult or impossible to solve. The necessity of augmenting XQuery with text querying functionality is acknowledged by the fact that W3 Consortium itself started the work on development of XQuery tor better support of text searching (XQuery and XPath Full Text [12]). The suggested changes partly solve the problem of querying the XML data as a text in a natural language. However, the problem of complexity of XQuery for an expert in humanities is even aggravated by further sophistication of the language. Besides that, the aforementioned changes to XQuery are still in the draft stage, and it is hard to predict the time of new release of XQuery, not to mention the development of software tools to support the new search mechanisms.

Besides XQuery there exist a number of less well-known XML querying languages, but none of them meet the two aforementioned requirements at a time (simplicity and support for full-text search). For example, XML-QL [4] is simpler than XQuery, but it offers no support for regular expressions or searching for elements that occur at a given distance.

There also exist specialized software tools developed for specific corpus projects. The most famous example is Xaira [3], the successor of SGML-based SARA tool distributed with the British National Corpus. While its architecture is general, the drawbacks include complexity of corpus compilation, necessity of huge indices (sometimes five times as big as source XML files with heavy annotation), as well as instability in work. Alinea [7] is a parallel corpus tool which is somehow more difficult to use for single-language corpora. The problem with many programs of the type of UAM corpus tool [9] is that they have been implemented in script languages and are rather instable in work with large-scale corpora. Therefore this paper suggests a general tool for querying XML-based corpora that has been developed in view of the most common tasks in analysis of linguistic data that can be easily automated.

2. Sample Task in Corpus Analysis

It is worth stressing that even if particular a particular research project in linguistics has seemingly nothing to do with applied or computer linguistics, it is always based on a corpus of text data. Using electronic texts may considerably shorten the time spent on retrieving evidence of linguistic facts. The general scheme of a research project in linguistics is the following: at first a classification scheme or typology for some language phenomenon is developed. It is then applied to analysis of text data and then statistics is drawn to prove the preliminary hypothesis. Traditional approach with index cards for example is not only susceptible to mistakes, but is also difficult to follow in cases when each item to be analyzed has more then two parameters to be classified with (which is the case with all complex studies involving, e.g. analysis on the levels of semantics, syntax and pragmatics).

Let's analyze a sample task posed in a research on peculiarities of English abstract nouns ending in -ness [1]. While it is relatively easy to find such nouns in a text with regular expressions (though odd words like witness or governess have to be eliminated), the task involves analysis of semantics as well as syntactic and pragmatic behavior of such nouns in a corpus of classical British novels. Semantic analysis is brought down to defining of semantic domain of a particular noun (according to the nature of referent five general domains have been specified, four of which describe various qualities of people (physical, psychological, qualities, states of mind, and qualities denoting social behavior and attitudes) and one is reserved for other kinds of referents). Words belonging to these domains are further subdivided into a number of thematic groups. Syntactic behavior is analyzed in terms of the most common distribution models of syntactic groups including nouns ending in -ness. Pragmatics is studied in terms of who is the speaker and which character is the quality denoted by the -ness noun attributed to, as well as whether the quality denoted by the -ness noun is evaluated positively or negatively in the context of a novel. Therefore 5 units of information are to be added to each -ness noun in the corpus. Besides that, the corpus has to be tokenized, and part-of-speech information is to be added to every word in order to enable the distribution analysis. Thus a sentence from "Pride and Prejudice" by Jane Austen contained in a single line and incorporating all this information in XML format would look like this (the pos-tag information has been simplified for viewing purposes):

```
<paragraph  id="40">
  <sentence id="78">
   <w pos="noun">Mr.</w>
   <w pos="noun">Darcy</w>
   <w pos="link_verb">is</w> <w pos="adjective">all</w>
   <w pos="noun" semantics="social\_polite" evaluation="$+$"
      speaker="Elizabeth" qualified="Darcy">politeness</w>
     <w pos="verb">said</w> <w pos="noun">Elizabeth</w> <w pos="participle">smiling</w>
  </sentence>
</paragraph>
```

Performing such annotation enables the linguist to perform complex queries to check if some character is more likely to use words from a certain semantic domain, how he evaluates other characters and is characterized by them, whether words from one semantic domain are more likely appear more often in certain syntactic models and not in the others. It is possible to learn if several such nouns appear in consecutive sentences or in the same paragraph (which is of interest because -ness nouns used in groups in the same context or together with the words they are derived from produce stylistic effect).

3. XCorp Query Language

Since one of the problems of the standard XML querying languages is its excessive complexity for an average linguist, we suggest a query language based on patterns. It was developed in view of typical tasks and situations that professional linguists face when working with text corpora. The proposed tool offers a general querying functionality for XML corpora that covers and simplifies such typical tasks, that include finding segments of text matching certain criteria and gathering statistics. Suggested routines are implemented in a program called XCorp, currently released at http://sourceforge.net/projects/xcorp/. XCorp runs under Microsoft .NET framework, and can be used on any operating system supporting .NET framework.

First thing to be determined is the types of corpora to be supported. XML-based linguistic corpora generally store text as an hierarchy of structures like chapter
paragraph
sentence, and on the bottom level as a sequence of elements representing words with attributes for different linguistic categories, such as part of speech, word lemma, semantic class etc. This is the output model supported by the majority of tokenizers, lemmatizers, part-of-speech taggers and other corpus utilities. Since this is the most frequently used type of annotation, XCorp was developed primarily in its view. (More complex XML schemas with data model different from the aforementioned one are generally developed for specialized corpora like TigerCorpus that usually develop a specialized querying tool for their data). The level of nesting and names of specific nodes may be different in various corpora and thus need to be specified in the search request. Corpus texts are to

be stored in simple xml files, no indexing is required. The current version of Xcorp has command-line interface with the program file being executed on the request file, and the development of GUI with graphical query constructor is scheduled.

As search request can contain many parameters and can be rather complicated, we chose to represent it in XML format as well. The root element of the query configuration file is <config> that contains three sections. The first section of request ($< search\_scope >$) specifies the structure of corpus files and the search scope within them, i.e. how elements are nested and what elements contain target information. Target elements can be specified with XPath notation. Second section ($< search_{r}equest >$) specifies search criteria. As text is presented as a sequence of elements with words and certain attributes, XCorp software is developed to retrieve subsequences of those elements, matching certain criteria. User can specify a substring or regular expression for each element in chain as well as for each attribute. Also maximum distance between elements can be set. The last section of search request ($< search\_target >$) contains description of what kind of output is expected and how it is to be presented. Therefore searching an XML-annotated text file is reduced to filling in a template form, which should make the task considerably easier for linguists with no prior training in programming.

Currently XCorp enables the user to obtain information of four kinds. 1) basic statistics for retrieved items. XCorp computes the number of hits of target pattern for all the levels in which they are nested (e.g. those may be sentences or paragraphs containing the target item). This feature may be useful for checking the "density" of target sequence, for example, in texts of different genres, or in different sections of the same text. It simplifies searching for stylistic phenomena based on repetition, such as anaphora or epiphora. 2) KWIC (keyword-in-context) lists containing all the occurrences of the target item in the context in which they occur. The context may be specified to be a certain amount of characters to the right and to the left of the target pattern, which is the traditional way for concordancer software, or the context may be understood as the element within which the target pattern is found (e.g. sentences or paragraphs or syntactic groups within which the target pattern occurs). 3) wordlists, or rather, lists of occurrences of target pattern in every file constituting the corpus, and a general wordlist for the whole corpus. The default setting for wordlist order is the order in which they occur in the file, which may be useful for research involving linguistic analysis of fiction or newspaper discourse. The wordlist can also be sorted alphabetically. There is an option of generating a frequency list, in which all similar occurences of target pattern are merged and general statistics is given. 4) other information characterizing the target pattern and stored in xml format. This feature makes XCorp useful not only for hypothesis-driven research where one needs only to check for availability of predefined patterns, but also for discovering "clusters" of linguistic information that the user may not be aware of at the time of request. For example, if the corpus has morphological and semantic annotation, this feature may help the researcher to discover semantic patterns that correspond to the target syntactical pattern.

Let us consider a query designed for the above example from "Pride and Prejudice". To describe the way the author construes the relationship between Elizabeth and Darcy we need to know what the two characters think of each other. To learn that we can search for -ness nouns uttered by Elizabeth and concerning Darcy, together with their attributes. The request matching the above example will look like this:

```
<search_scope>
  <element name="//paragraph">
   <element name="sentence">
    <element name="w">
    </element>
   </element>
  </element>
</search_scope>
<search_request>
  <item mask="" distance="0">
   <attribute name="pos">adjective</attribute>
  </item>
  <item mask="\\wness" distance="">
   <attribute name="speaker">Elizabeth</attribute>
   <attribute name="qualified">Darcy</attribute>
  </item>
</search_request>
<search_target>
  <content sort="frequency" order="descending"/>
</search_target>
```

The adjective in the above example serves to increase the degree of Darcy's politeness so as to exaggerate it and let us feel the irony of Elizabeth, who in fact thinks him extremely rude. On the other hand, Mrs. Bennet talks of his "shocking rudeness", which is also an exaggeration, and this time it is the author who speaks ironically of her character. But as the novel progresses we witness Elizabeth starting to like Darcy and even acknowledging his "utmost politeness" in earnest.

Conclusion

The presented paper analyses the applicability of general-purpose XML querying tools in the sphere of corpus linguistics. Two main problems have been identified: the standard querying tools do not currently support full-text search functionality, and the default querying language is too difficult for experts in humanities with no programming experience. Therefore the proposed query language is pattern-based. It is implemented in software program XCorp and can be applied for querying XML corpora with various kinds of annotation. The proposed solution is universal enough to work with different kinds of linguistic data, and at the same time it is as simplified as possible. XCorp has been successfully applied for solving some practical tasks in corpus linguistics. Further work includes the development of graphical user interface and inviting the linguistic community to produce more requirements, so as to make XCorp a more universal solution and to make the suggested query language more expressive.

Список литературы

1. *Гладкова Г.П.* Особливості функціонування абстрактних іменників із суфіксом -ness у тексті роману Джейн Остін "Pride and Prejudice-/ Г.П. Гладкова // Мовні і концептуальні картини світу. – 2008. – Вип. 24. – Частина 1. – Київ: КНУ імені Т. Шевченка, 2008. – с. 180-186.
2. *Anthony, L.* AntConc: design and development of a freeware corpus analysis toolkit for the technical writing classroom / Lawrence Anthony // Professional Communication Conference, 2005. IPCC 2005. Proceedings. -pp. 729-737. - [Electronic resource]: http://www.antlab.sci.waseda.ac.jp/abstracts/ipcc05_pres_20050713/IPCC_05_Anthony_fin_handouts.pdf
3. *Aston G.* Introducing XAIRA: an XML-aware concordance program / Guy Aston, Lou Burnard. -Presentation at workshop held at TALC 2006. -[Electronic resource]: http://www.oucs.ox.ac.uk/rts/xaira/Talks/xaira-wkshop.odp.
4. A Query Language for XML / Alin Deutsch, Mary Fernandez, Daniela Florescu et al. [Electronic resource]. -http://www8.org/w8-papers/1c-xml/query/query.html.
5. *Buxton S.* Querying XML : XQuery, XPath, and SQL/XML in context / Jim Melton, Stephen Buxton. -San Francisco: Morgan Kaufmann, 2006. -845 p. -(The Morgan Kaufmann Series in Data Management Systems).
6. *Baker P. A* Glossary of Corpus Linguistics / Paul Baker, Andrew Hardie, Tony McEnery. -Edinburgh: Edinburgh University Press, 2006. -187 p.
7. Duchet, J.-L. Alinea: a language independant tool for bi-text processing / Jean-Louis Duchet, Oliever Kraif // JRC EU-Enlargement Workshop: Exploiting parallel corpora in up to 20 languages. JRC-Ispra, Italy, 26-27.09.2005. -[Electronic resource]: http://langtech.jrc.it/0509_EU-Enlargement-Workshop.html.
8. Kennedy, Graeme D. An Introduction to Corpus Linguistics / Graeme Kennedy. -London: Longman, 1998.
9. O'Donnell, M. The UAM CorpusTool: Software for corpus annotation and exploration / Michael O'Donnell // Proceedings of the XXVI Congreso de AESLA, Almeria, Spain, 3-5 April 2008. -[Electronic resource]. -http://www.wagsoft.com/Papers/AESLA08.pdf.
10. Stubbs M. Text and corpus analysis: computer-assisted studies of language and culture / Michael Stubbs. -Malden: Blackwell Publishers, 1996. -267 p. -(Volume 23 of Language in Society Series).
11. XQuery 1.0: An XML Query Language /Scott Boag, Don Chamberlin, Mary F. Fernandez et al. [Electronic resource]. -http://www.w3.org/TR/xquery/.
12. XQuery and XPath Full Text 1.0 / Sihem Amer-Yahia, Chavdar Botev, Stephen Buxton et al. [Electronic resource]. -http://www.w3.org/TR/xpath-full-text-10/.

Статья поступила в редакцию 22.09.2009