Lexical Databases in XML: A Case Study of Up-Translation of the Dictionary of Literary Czech Language.

Pavel Smrz
Faculty of Informatics, Masaryk University
Botanicka 68a
60200 Brno, Czech Republic
smrz@fi.muni.cz

Abstract

This paper deals with up-translation – a process of lexical data transformation from any source format to the XML document. Relevant aspects of the XML format and many related technologies are surveyed first. Then, information content enhancement of existing lexical resources is discussed. The last part brings information about up-translation of the Dictionary of Literary Czech Language and the way of efficient storage and retrieval of data.

Introduction

Dictionaries are the most relevant source of language vocabulary information. Their usage is not limited to human beings but they are also essential for artificial intelligence – natural language processing applications need dictionary information for almost all tasks that they solve today. Hundreds if not thousands different dictionaries are in use to support information retrieval, automated summarisation, machine translation etc.

At present, most of dictionary data integrated into applications has not been primarily for automatic processing. Many printed dictionaries have been transformed into an electronic form in last decades primarily with the aim to decrease the costs related to editorial emendations, consistency checks and modifications during preparation of new editions. Even the original purpose of the genuinely electronic WordNet database (Miller et al. 1990) was not an application usage, as it was intended as a model of mental lexicon.

Despite the history of the issuing, available electronic dictionaries, or rather electronic lexical databases, are of a great value from the point of view of their application. The acquisition of lexical information is actually very expensive and also represents a rather difficult intellectual performance. The preference of the usage of existing sources is valid in spite of inevitable investment in finding relevant information, which is to some extent hidden owing to the weak structure of entries, necessary mistakes, inconsistencies and omissions. Thus, an admittedly rational target is to keep lexical data in a versatile, widely available and reusable format. The family of standards and tools related to the XML language offers such an environment.

The following section brings a brief overview of the XML format and related standards. Then we will discuss data transformation from a source format to the final XML, i.e. the process of enhancement of information content – up-translation. The last section brings information about up-translation of the Dictionary of Literary Czech Language to the XML form that conforms to the TEI (Text Encoding Initiative) recommendations. The paper concludes with the recommendations for similar projects.

The XML Family

XML (eXtensible Markup Language) (Bray et al. 2000) is an international standard for representation and interchange of data. It presents a powerful instrument enabling general markup of all forms of a structure, mutual references and multilevel structure hierarchies.

The XML language, oriented primarily to the area of World Wide Web applications, is a simplified dialect of SGML (Standard Generalised Markup Language). Consequently, it is theoretically weaker than SGML in some aspects. However, thanks to dozens of connected technologies enabling document transformations, definitions of constraints, structure validation, and pointers within one document as well as inter-document mutual references (see below), XML is an appropriate tool allowing to keep in touch with the extreme speed of progress in the field of information technology.

Users can exploit many existing mechanisms for data access and manipulation when working with lexical databases in the XML format. We will speak about a family of XML standards. XML is a markup language in the base form and so it allows the identification of text elements, hierarchical structure and references. The structure of XML encoded documents is described by DTD (Document Type Definition) occurring already in the SGML standard. DTD defines generalised structure rules and determines what is permitted in each particular document encoding.

The merits of document validation offered by DTD are extended by the XML Schema definition language (Thompson et al. 2001, Biron, Malhotra 2001). It provides the way to restrict and document the meaning, usage and relations of particular parts of XML documents. Default values of attributes and elements can be specified, for example. From the conceptual point of view, the definition of XML Schema can be considered as an abstract data model of the described document class (Ide 2000).

Other members of the "XML family" are stylesheet languages XSL (eXtensible Stylesheet Language) (Adler et al. 2000) and XSLT (eXtensible Stylesheet Language for Transformations) (Clark 1999, Clark 2001). Stylesheet determines what action will be accomplished if the given condition is fulfilled. XSLT processors work with XML documents represented by tree structures and transform them to another arbitrary format by means of information selections, re-arrangements and additions. The XSLT language supports selection of the content of an element or its parts from one or more XML documents and the transformations of the content as well as names of elements.

Information Content Enhancement

Dictionaries contain many different types of information, encoded in many different ways. Various structural or typographic criteria for homographs, collocations, grammar information etc. are applied. A standard lexical database formalism has to define an unambiguous representation of all these entities.

Despite the ambiguity of information encoding, people are usually able just to look at a dictionary entry and they immediately know its structure. They grasp at least the relevant parts that form the entry and they understand their meaning. However, this process involves common sense and general knowledge about the function of dictionaries and the way in which they are usually used. In order to provide computers with the same information, it has to be transformed from an original implicit form to explicit data that are easily accessible by computer programs.

The value of lexical databases is radically increased if they share common markup. However, such an aim is very challenging as comprehensive sources are usually extracted from existing dictionaries with their own, specific structure. The transformation of data from a source to a target format is called "information content enhancement" or "up-translation". From the application point of view, it can be interpreted as a course to a more usable form of dictionary data.

As the previous section suggests, our aim is to transform dictionary data into the XML formats, so that up-translation represents a conversion from any source format to a valid instance of XML that corresponds a given DTD. We are seeking for (semi-)automatic methods of this transformation. The development of such methods is motivated by an effort to decrease the costs.

Taking into account the enormous variability of source formats, it is very difficult to define a universal model of the up-translation process. Nevertheless, three basic sub-processes can be identified generally (Chahuneau 1994):

1. The identification of groups of source document objects that share common formatting properties

   (typographic characteristics and typical text patterns);

2. Mapping identified classes on the types of XML elements corresponding to the target DTD.

3. Generating the final structure, possible data reorganisation and adding missing structures

   (elements and attributes) to ensure that all the entities correspond to the DTD.

All the processes can be realised in one pass. However, such an approach has some disadvantages. It is usually difficult to divide tasks for more programmers. Moreover, a monolithic form of transformation programs does not contribute to the legibility of the code and necessary manual corrections of coding errors represent also a non-trivial problem.
A solution of these challenges is a gradual, multi-pass transformation. The relevant DTDs can be defined for the outcomes of each individual phase in the form of an XML document. The entrance to "the XML arena" already in the initial phase of transformation is advantageous as the model of information content achieved by means of DTD is explicit and

allows incorporation of sophisticated tools for processing XML structures. The cases when the source format strongly defies the intended DTD can also be covered easily.

Transformation of Dictionary of Literary Czech Language

Our Natural Language Processing Laboratory at the Faculty of Informatics, Masaryk University Brno, has taken part in the transformation of the Dictionary of Literary Czech Language to the XML format. The data has been provided in the form of MS Word documents, each ten pages of source text in one file. The previous phase involved scanning of the printed dictionary, transformation using OCR and also a preliminary check in order to eliminate easily noticeable errors of recognition. The Institute of Czech Language at the Academy of Sciences of Czech Republic has completed all these tasks.

The first task involved the transformation from the MS Word format. MS Word 2000 promises data saving in HTML format that keeps all information necessary for transformation into a primitive XML form. However, our experience suggests that the transformation to XML corresponding to the standard would require an enormous amount of tedious work and even then the results would offer little support when the document structure is derived from markup. The use of OpenOffice applications, that employ the XML format for saving documents, represents another alternative (the new versions of these applications were not available in the time of our work). We chose the direct data transformation by means of special scripts implemented in the Visual Basic for Application language, that is interpreted by MS Word in the form of macros. Taking into account the fact that this process runs only once, time demand of this phase does not matter.

The next step consisted of finding abnormalities in the input format, elimination of ambiguities and correction of coding errors. We have strongly perceived that the definition of a complete grammar for recognition of text patterns and transformation of structure represents an almost endless process. It can be fulfilled only by the successive modifications of the code, which is tedious. On the other hand, this process can be straightforward, deterministic and robust, if attention is paid to debugging the transformation code.

The last and the most difficult task lies in the transformation of intermediate results into the XML format corresponding to the final DTD. In the best instances, a type of element may directly match a font. In other instances, elements can be captured in a simple, unique context (e.g. pronunciation in square brackets). Sometimes, it is advantageous to take into account the restrictions of information content when a value is included in a pre-defined list (lists of abbreviations, author names). The success of transformation depends to a big extent on the quality of source data, in our case mainly on the consistency of dictionary preparation. Most of inconveniences are connected only just with the inconsistent structure of entries. It is extremely difficult, if not impossible, to convert such entries in an automatic way.

At present, we use two variants of XML. Low-level encoding (see the first example) is suitable for corrections of errors found in the text. The form matching the final DTD (the second example) is appropriate for some queries for specific parts of entries, e.g. to restrict the query only to quotations, etymology, etc. The second form still includes certain amount of inaccurate recognised elements. These errors are gradually corrected. The errors and inconsistencies present even in the printed version of the dictionary form a special category. These corrections are recorded separately to be able to confront the original data form.

```
<entry>
<bold>terorismus</bold>
<ital>způsob vlády vymáhající terorem poslušnost; hrůzovláda,
krutovláda, despotismus:</ital>
<norm>vojenský t.; nesnesitelný t.; demagogie a t.; </norm>
<small>přen. expr.</small>
<norm>to je t., nedejte si to líbit</norm>
</entry>
```

Example 1: Low-level data encoding – typefaces and fonts markup

```
<entry>
  <hw>
    <orth>terorismus</orth>
  </hw>
  <morph>
    <paradig>socialismus</paradig>
  </morph>
  <senses>
    <sense>
      <def>způsob vlády vymáhající terorem poslušnost</def>
      <def>hrůzovláda</def>
      <def>krutovláda</def>
      <def>despotismus</def>
      <eg>vojenský terorismus</eg>
      <eg>nesnesitelný terorismus</eg>
      <eg>demagogie a terorismus</eg>
      <eg>
        <usg type=style>přen. expr.</usg>
        to je terorismus, nedejte si to líbit
      </eg>
    </sense>
  </senses>
</entry>
```

Example 2: The final encoding of one entry

Efficient storage and retrieval of dictionary data is provided by the lexical database management system MAXXL. The system originated as a practical outcome of the Masters thesis at the Faculty of Informatics, Masaryk University in Brno, Czech Republic (Karasek 2000). The basic characteristic of the system is its total independence on any particular structure of XML. Processing is based on a given DTD and indexes for an efficient retrieval are generated from additional information about individual element types. The most important information is identification of the primary key element.

The system defines its own query language, specifically tailored to reflect the needs of lexical databases. The result of a query takes the form of a sequence of XML elements or a sequence of simple words. Operators of exact match, prefix searching and general substring localisation are provided. The very efficient Karp-Rabin algorithm is employed for these tasks.

Lexical database in MAXXL is a set of XML documents. These documents can use various character encodings so that the problems associated with the usage of all different alphabets can be solved. MAXXL accepts data represented in the UNICODE format, UTF-8 coding (128 characters encoded in 1 byte – ASCII, 1920 characters in 2 bytes – all Czech characters, Greek, Hebrew, ..., 63488 characters in 3 bytes – Chinese, Japan, 4,5,6 bytes codes still unassigned). Consequently, it is possible to process XML data from different languages at the same time.

MAXXL will be massively used in the work on the Czech part of Balkanet project, in the process of up-translation of the machine-readable version of various Czech dictionaries and in several other tasks.

References

Adler S. et al. (2000) *Extensible Stylesheet Language (XSL). Version 1.0.* W3C Proposed Recommendation. http://www.w3.org/TR/xsl/.

Biron P. and Malhotra A. (2001) *XML Schema Part 2: Datatypes.* W3C Recommendation. http://www.w3.org/TR/xmlschema-2/.

Bray T. et al. (2000) *Extensible Markup Language (XML) 1.0 (Second Edition).* W3C Recommendation. http://www.w3.org/TR/1998/REC-xml.

Clark J. (1999) *XSL Transformations (XSLT). Version 1.0.* W3C Recommendation. http://www.w3.org/TR/xslt/.

Clark J. (2001) *XSL Transformations (XSLT). Version 1.1.* W3C Working Draft. http://www.w3.org/TR/xslt11/.

Karasek L. (2000) *A System for the Development and Presentation of Mono- and Multilingual Dictionaries.* Masters thesis, Faculty of Informatics, Masaryk University, Brno (in Czech).

Thompson H. S. et al. (2001) *XML Schema Part 1: Structures.* W3C Recommendation. http://www.w3.org/TR/xmlschema-1/.