Proceedings of HLT-NAACL 2003
Short Papers , pp. 1-3
Edmonton, May-June 2003

Automating XML markup of text documents

Shazia Akhtar
Department of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland
Shazia.Akhtar@ucd.ie

Ronan G. Reilly
Department of Computer Science, National University of Ireland, Maynooth, Ireland
Ronan.Reilly@may.ie

John Dunnion
Department of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland
John.Dunnion@ucd.ie

Abstract

We present a novel system for automatically marking up text documents into XML and discuss the benefits of XML markup for intelligent information retrieval. The system uses the Self-Organizing Map (SOM) algorithm to arrange XML marked-up documents on a two-dimensional map so that similar documents appear closer to each other. It then employs an inductive learning algorithm C5 to automatically extract and apply markup rules from the nearest SOM neighbours of an unmarked document. The system is designed to be adaptive, so that once a document is marked-up; its behaviour is modified to improve accuracy. The automatically marked-up documents are again categorized on the Self-Organizing Map.

1 Introduction

Vast amounts of information are now available in electronic form to which accurate and speedy access is getting more difficult. The increasing quantity of information has created a need for intelligent management and retrieval techniques. Many of the existing information retrieval systems, which deal with large volumes of documents, have poor retrieval performance because these systems can use a little knowledge in the documents. By adopting *XML* as a standard document format, content-based queries can be performed by exploiting the XML structure of the documents. In addition, specifically tagged sections of the documents can be searched rather than the entire document, thus providing fast and effective retrieval. Furthermore, using the logical structure of a document created by XML markup, different types of operations can be performed, for example, the same content can be reused in a variety of formats, specific elements can be extracted from the XML documents and full documents satisfying certain structural conditions can be retrieved from the database. These and other advantages of using XML make it a complete solution for content management and intelligent information retrieval. However, despite the advantages and the popularity of XML, we still do not have large repositories of XML because automatic XML markup is still a challenge and the process of manually marking up XML documents is complex, tedious and expensive. Most of the existing automatic markup systems are limited to certain domains and do not perform general automatic markup. In addressing the need for more general automatic markup of text documents, we present a system with a novel hybrid architecture. The system uses the techniques of *Self-Organizing Map (SOM)* algorithm (Kohonen, 1997) and an *inductive learning algorithm, C5* (Quinlan, 1993, 2000).

2 System overview

The system has two phases. The first phase of the system deals with the formation of a map of *valid* XML (a *valid* XML document is one which is well-formed and which has been validated against a DTD) marked-up documents using the SOM algorithm. The second phase deals with the automatic markup of new (unmarked) document according to the markup of existing documents. Once a document is marked-up, the system's behaviour is modified to improve accuracy. These two phases of the system are currently implemented independently but will be combined to form an integrated

hybrid system. This paper focuses on phase 2 of the system.

Phase 2 of the system is implemented as an independent automatic XML markup system, which is Figure 1. It comprises two main modules – a Rule extraction module and a Markup module. The rule extraction module deals with the extraction of rules using an inductive learning approach (Mitchell, 1997). Firstly, during a preliminary phase, *training examples* are collected from the valid XML marked-up documents. These documents should be from a specific domain and their markup should be valid and conformant to the rules of a single *Document Type Definition* (DTD). An XML document consists of a strictly nested hierarchy of *elements* with a root element. Only elements having text nodes are considered as markup elements for our system.

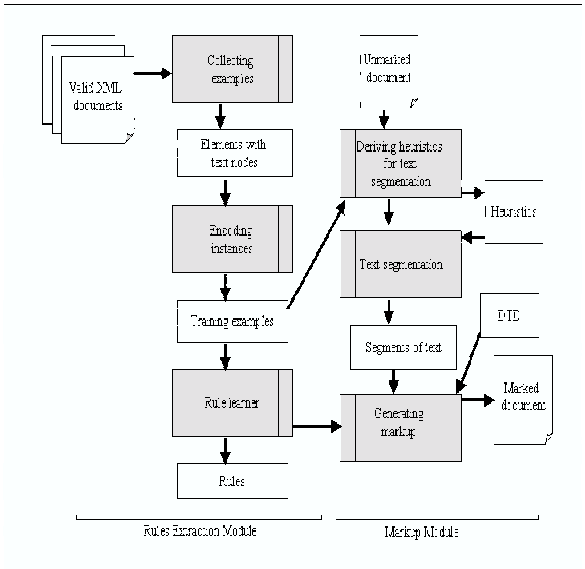

Figure 3: Automatic XML markup system

The markup of elements nested within other elements can be accomplished by using the DTD. Each *training instance* corresponds to an element containing a text node from the collection of marked-up documents. The text enclosed between the start and end tags of all occurrences of each element is encoded using a fixed-width feature vector. We have used 31 features in our experiments. The set of feature vectors is used by the system to learn classifiers. An inductive learning algorithm processes these encoded instances to develop classifiers for elements having specific tag names. These classifiers segment the text of an unmarked document into different elements of the resulting XML marked-up document. In our system, the C5 program is used to learn classifiers. These classifiers are later used to markup the *segments* of text as XML elements.

The second module deals with the creation of XML markup. The unmarked document to be used for this process should be from the same domain and should have a similar structure to the documents, which were used for learning the rules. To accomplish the markup, the unmarked document is segmented into pieces of text using a variety of heuristics. These heuristics are derived from the set of training examples. By using the DTD conformant to the document set used for learning the rules and by using the text segments stored for each element, a hierarchical structure of the document is encoded and the marked-up document is produced.

The markup produced by the system can be validated according to a DTD. However, in order to check the accuracy of the markup, we have to examine it manually and compare it with the original source (if available) as XML processors can only validate the syntax of the markup, and not its semantics.

3 Performance

We used documents from a number of different domains for our experiments, including letters from the MacGreevy archive (Schreibman, 1998, 2000), a database of employee records, Shakespearean plays (Bosak, 1998), poems from an early American encoding project, and scientific journal articles (Openly Informatics, Inc., 1999-200). Figure 2 shows a part of a scene from "A Midsummer Night's Dream" as an example of XML markup automatically produced by our system. The underlined text was not marked up by our system.

We have also evaluated our system with some of the document sets. For evaluation, we considered the elements representing the content of the document, and a human expert is required to evaluate this. We have used three performance measures in evaluating the automatic markup process. These measures are

- The percentage of markup elements determined correctly by the system
- The percentage of markup elements determined incorrectly by the system
- The percentage of markup elements not determined by the system (i.e. text nodes for these markup elements are not present in the marked-up document produced by the system)

The elements of 10 valid XML marked-up letters from the MacGreevy archive were used to learn C5 rules and text segmentation heuristics. By applying these rules and heuristics, 55 elements of five unmarked letters

from the MacGreevy archive were automatically marked up by the system with 96% accuracy (we use the term "accuracy" here to mean the number of marked-up elements correctly determined by the system). Similarly, elements of 5 valid XML marked-up Shakespeare plays were used as training examples and 13882 elements of four Shakespearean plays were automatically marked-up by the system. In this case the accuracy rate was 92%.

```
 ...
<SCENE>
     <TITLE> SCENE I. Athens.  The
     palace of THESEUS.  </TITLE>
     <STAGEDIR> Enter THESEUS,
     HIPPOLYTA, PHILOSTRATE, and
     Attendants</STAGEDIR>
     <SPEECH>
        <SPEAKER>THESEUS</SPEAKER>
        <LINE>Now, fair Hippolyta, our
        nuptial hour</LINE>
        <LINE>Draws on a pace; four
        happy days bring in</LINE>
        <LINE>Another moon: but, O, me
        thinks, how slow</LINE>
        <LINE>This old moon wanes!
        she lingers my desires,</LINE>
        <LINE>Like to a step-dame or
        a dowager</LINE>
        <LINE>Long withering out a
        young man revenue. </LINE>
     </SPEECH>
     <SPEECH>
        <SPEAKER>HIPPOLYTA</SPEAKER>
        <LINE>Four days will quickly
        steep themselves in night;
        </LINE>
        <LINE>Four nights will quickly
        dream away the time; </LINE>
        <LINE>And then the moon, like
        to a silver bow</LINE>
        <LINE>New-bent in heaven,
        shall behold the night</LINE>
        <LINE>Of our solemnities
        </LINE>
        </SPEECH>
...
```

Figure 2. A section taken from "A Midsummer Night's Dream" marked up by our system

4 Conclusion

We have described a system with a novel hybrid architecture that uses an inductive learning approach to perform automatic markup of text documents. The system automatically marks up documents by capturing markup information from the neighbouring documents on a Self-Organizing Map. Such marked-up documents can be used for management and retrieval purposes according to the structural information they contain. The results from our experiments demonstrate that our approach is practical and that our system provides a novel approach for automatically marking up text documents in XML. The functionality of our system makes it a useful tool for electronic information exchange.

Acknowledgements

The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged. The work was funded under grant PRP/00/INF/06.

References

Bosak, J. (1998). Shakespeare 2.00. [http://metalab.unc.edu/bosak/xml/eg/shaks200.zip]

Kohonen, T. (1997). *Self-Organizing Maps*. Springer Series in Information Science, Berlin, Heidelberg, New York.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Quinlan, J. R. (1993). C4.5: *Programs For Machine Learning*. Morgan Kauffman Publishers, San Mateo, Calif.

Quinlan, J. R. (2000). *Data Mining Tools See5 and C5.0*. [http://www.rulequest.com/see5-info.html]

Openly Informatics, Inc. (1999-2000). [http://www.openly.com/efirst]

Schreibman, S. (1998). *The M*