# AutoMarkup: A tool for automatically marking up text documents

Shazia Akhtar, Ronan G. Reilly, John Dunnion

Smart Media Institute, Department of Computer Science, University College Dublin,
Belfield, Dublin 4, Ireland
{Shazia.Akhtar, Ronan.Reilly, John.Dunnion}@ucd.ie

**Abstract.** In this paper we present a novel system that can automatically mark up text documents into XML. The system uses the Self-Organizing Map (SOM) algorithm to organize marked documents on a map so that similar documents are placed on nearby locations. Then by using the inductive learning algorithm C5, it automatically generates and applies the markup rules from the nearest SOM neighbours of an unmarked document. The system is adaptive in nature and learns from errors in the automatically marked-up document to improve accuracy. The automatically marked-up documents are again arranged on the SOM.

## 1 Introduction

The dramatic growth of the World Wide Web with the availability of large collections of textual resources in electronic form has created a need for intelligent text processing. Extensible Markup Language XML was developed to address the need of electronic publishing and intelligent document management. Its power goes beyond the current Hypertext Markup Language and intelligent document management. Its power goes beyond the functionality of Hypertext Markup Language (HTML) for example it makes explicit the content and structure of the documents to make them easier to identify and retrieve. XML provides key features such as extensibility, validation and structure and is considered a complete solution for content management and electronic publishing. Despite the widespread adoption and popularity of XML, it is still a significant challenge to automatically markup documents in XML. Automatic XML markup is therefore currently a major research issue and many projects are involved in such research, as manual XML markup of documents is tedious and expensive. However most systems that have been developed are limited to certain domains and require a considerable amount of human intervention. There is as yet no tool available to solve the hard problem. In addressing this need we present a novel system that automatically marks up text documents into XML by using the Self-Organizing Map (SOM) algorithm (Kohonen, 1997) and an inductive learning algorithm C5 (Quinlan, 1993, 2000).

## 2 System Overview

Our system has two phases. The first phase of the system deals with the formation of a map of valid XML documents by using the SOM algorithm. In the second phase the system automatically learns and applies rules from the nearest SOM neighbours of a new unmarked document. The system learns from markup errors of the automatically marked up document and improves the markup. These two phases of the system are independently implemented and our intention is to combine the two phases to form a hybrid system.

Phase 2 of the system dealing with the automatic markup of documents is shown in Figure 1. It comprises two main modules, a Rule Learner and a Markup module. The first module learns classifiers by using the machine-learning algorithm C5. This module processes a set of pre-tagged valid XML documents.

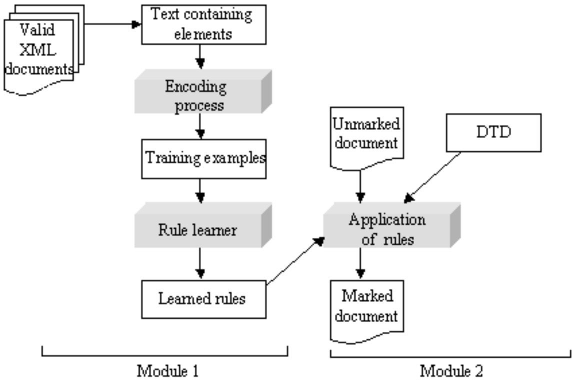
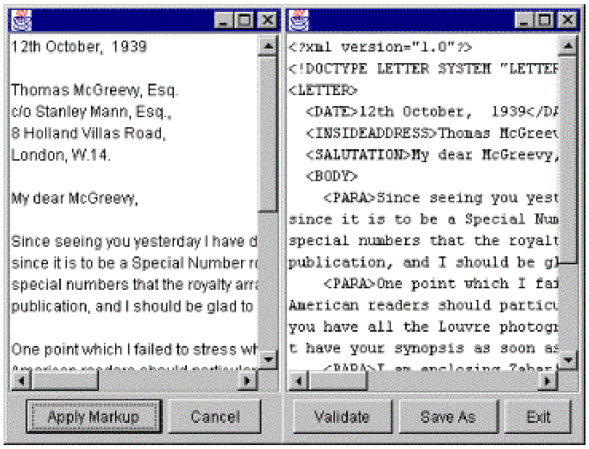


**Fig. 1.** Process of automatic markup. (a) Rule Learner (b) Markup

All documents in the set should be from a specific domain and conform to a single *Document Type Definition* (DTD). The system automatically gathers the *training examples* from the set of documents. Each *instance* corresponds to a text-containing *element* of the marked collection of documents. Instances are encoded using a fixed width feature vector. We have used twenty-two features such as word count and character count, in our experiments. All the encoded instances form a training set. Rules are learned when the training set is input to the C5 classifier. The second module deals with the markup of a new un-marked document from the same domain. The markup is obtained automatically by applying the generated rules and the rules of the DTD to an unmarked document. The DTD provides us with a set of rules using a number of operators for sequence elements (‘,’), repeated elements (‘+’), optional elements (‘?), and alternatives for recognizing the logical structure of a document. In this process, the unmarked document is chunked into pieces of text by using delimiters such as blank lines. By applying the rules of the DTD and the learned rules automatically generated by the system. The automatically marked up document is also valid XML.



12th October, 1939

Thomas McGreevy, Esq.
c/o Stanley Mann, Esq.,
8 Holland Villas Road,
London, W.14.

My dear McGreevy,

Since seeing you yesterday I have d
since it is to be a Special Number r
special numbers that the royalty arr
publication, and I should be glad to

One point which I failed to stress wh

```
<?xml version="1.0"?>
<!DOCTYPE LETTER SYSTEM "LETTEF
<LETTER>
  <DATE>12th October,  1939</DA
  <INSIDEADDRESS>Thomas McGreev
  <SALUTATION>My dear McGreevy,
  <BODY>
    <PARA>Since seeing you yest
since it is to be a Special Num
special numbers that the royalt
publication, and I should be gl
    <PARA>One point which I fai
American readers should particu
you have all the Louvre photogr
t have your synopsis as soon as
```

**Fig. 2.** Unmarked letter and valid XML markup of the same letter produced by the system

We have used documents from a few different and simple domains using simple DTDs as an initial test bed for our experiments. One example is the automatic markup of letters from the MacGreevy archive (Schreibman, 1998). We have used valid marked-up letters from this archive to learn classifiers. By using the learned classifiers, unmarked letters from the same domain are marked up (see Figure 2). In an earlier version of this system we worked with well-formed documents comprising letters from the MacGreevy archive. We tested it on the elements of about 20 letters and achieved 94% accuracy. The accuracy rate is calculated by considering the correctly marked up elements as a percentage of the total number of elements of the tested letters. We are currently working with valid documents and hope to achieve higher accuracy.

## References

1. Kohonen, T. (1997). *Self-Organizing Maps.* Springer Series in Information Science, Berlin, Heidelberg, New York.
2. Quinlan, J. R. (1993). C4.5: *Programs For Machine Learning.* Morgan Kauffman Publishers, San Mateo, Calif.
3. Quinlan, J. R. (2000). *Data Mining Tools See5 and C5.0.* [http://www.rulequest.com/see5-info.html]
4. Schreibman, S. (1998). *The MacGreevy Archive.* [http://www.ucd.ie/~cosei/archive.htm]