A TREE LEARNING APPROACH TO WEB DOCUMENT SECTIONAL HIERARCHY EXTRACTION

F. Canan Pembe
Department of Computer Engineering, Boğaziçi University, Bebek, İstanbul, Turkey
Department of Computer Engineering, Istanbul Kultur University, Ataköy, İstanbul, Turkey
canan.pembe@boun.edu.tr

Tunga Güngör
Department of Computer Engineering, Boğaziçi University, Bebek, İstanbul, Turkey
gungort@boun.edu.tr

Keywords: Machine learning, document structure, World Wide Web, Hypertext Markup Language.

Abstract: There is an increasing availability of documents in electronic form due to the widespread use of the Internet. Hypertext Markup Language (HTML) which is mostly concerned with the presentation of documents is still the most commonly used format on the Web, despite the appearance of semantically richer markup languages such as XML. Effective processing of Web documents has several uses such as the display of content on small-screen devices and summarization. In this paper, we investigate the problem of identifying the sectional hierarchy of a given HTML document together with the headings in the document. We propose and evaluate a learning approach suitable to tree representation based on Support Vector Machines.

1 INTRODUCTION

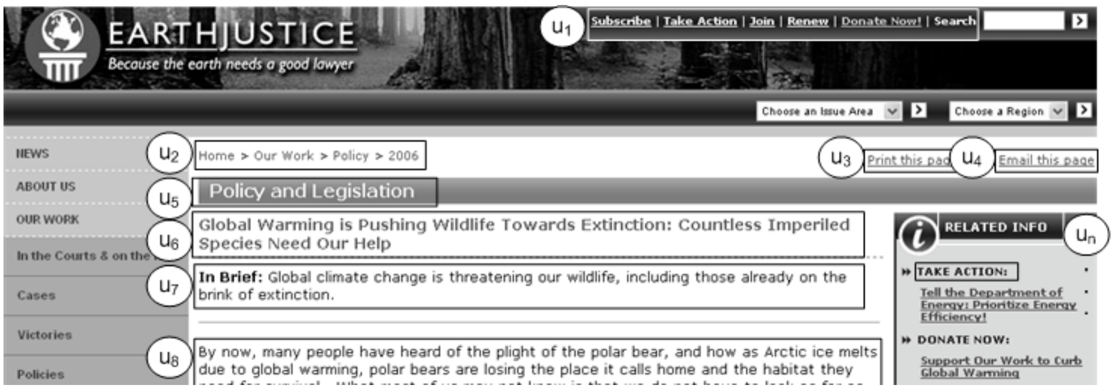
The automatic processing of Web documents is becoming increasingly important as the number of such documents grows drastically on the Internet. HTML is the most commonly used document format on the Web; however, it is mainly concerned with the presentation of contents rather than a semantic representation. Web documents usually consist of several sections and subsections with various formatting. Automatic processing of the document structure can be useful for various applications including search engines and summarization.

In this paper, we consider the problem of automatically extracting the sectional hierarchy of a Web document. We developed a machine learning approach because it can be more flexible by combining several features using a corpus rather than predefined rules. In general, a document can be represented as a tree with order and containment relations between its physical and logical components (Mao *et al.*, 2003). Therefore, a learning approach suitable to a tree representation is needed rather than the simpler case of classification. We developed an incremental algorithm making a sequence of locally optimal choices to approximate a globally optimal solution using Support Vector Machines. We evaluated the system based on the accuracy of heading and hierarchy extraction tasks.

2 RELATED WORK

In the literature, there are some studies on the extraction of the main title from documents in electronic form using machine learning (Xue *et al.*, 2007). In another study, the aim is to segment a Web document into blocks using classification (Feng *et al.*, 2005). For this purpose, a set of features is defined to represent the difference between each pair of text nodes; e.g. the difference in formatting of two nodes. The document is segmented in a flat way without considering the hierarchical structure.

In general, document structure analysis can be considered as a syntactic analysis problem (Mao *et al.*, 2003). In one of the studies, transformation-based learning is used in the conversion of HTML documents into XML format (Curran and Wong, 1999). The logical structure of a document may also be represented with a generalized n-gram model (Brugger *et al.*, 1997) or by means of a probabilistic grammar (Shilman *et al.*, 2005). There is some related work on incremental parsing using machine learning (Collins and Roark, 2004). An application



EARTHJUSTICE
Because the earth needs a good lawyer

Subscribe | Take Action | Join | Renew | Donate Now! | Search

Choose an Issue Area | Choose a Region

NEWS
ABOUT US
OUR WORK
In the Courts & on the
Cases
Victories
Policies

Home > Our Work > Policy > 2006

Print this pag | Email this page

Policy and Legislation

Global Warming is Pushing Wildlife Towards Extinction: Countless Imperiled Species Need Our Help

In Brief: Global climate change is threatening our wildlife, including those already on the brink of extinction.

By now, many people have heard of the plight of the polar bear, and how as Arctic ice melts due to global warming, polar bears are losing the place it calls home and the habitat they

RELATED INFO
TAKE ACTION:
Tell the Department of Energy: Prioritize Energy Efficiency!
DONATE NOW:
Support Our Work to Curb Global Warming

Figure 1: Part of an example HTML document.

of such an approach is to automatically generate a table-of-contents for a book (Branavan *et al*., 2007).

3 WEB DOCUMENT SECTIONAL HIERARCHY EXTRACTION

3.1 The Model

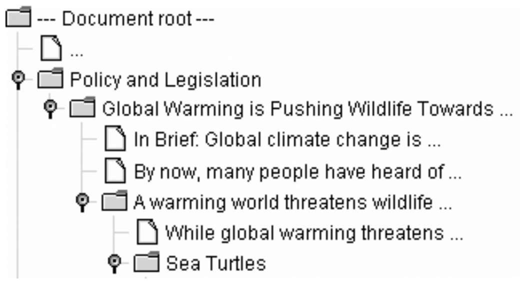
The general problem we consider may be defined as learning a mapping from inputs $x \in X$ to outputs $y \in Y$. In syntactic parsing, X is a set of sentences and Y is a set of possible parse trees (Collins and Roark, 2004). Analogously, we define X as a set of documents and Y as a set of possible sectional hierarchies using the following framework:

- Training examples (x_i, y_i) for $i = 1 \ldots n$
- A function GEN(x) which enumerates a set of possible outputs for an input x
- A representation Φ mapping each $(x_i, y_i) \in X \times Y$ to a feature vector $\Phi(x_i, y_i)$
- A parameter vector α

The learning task is to estimate the parameter vector α using the training examples such that it will give highest scores to correct outputs:

$$F(x) = \arg \max_{y \in GEN(x)} \Phi(x, y) \cdot \alpha \qquad (1)$$

In the proposed system, each document is modeled as a sequence of text units. We define a *text unit* u_i as a text fragment delimited by a new line (Figure 1). The output of the proposed system is the sectional hierarchy where headings and subheadings are at the intermediate nodes and other text units are at the leaves (Figure 2). A graph corresponding to an example hierarchy is given in Figure 3. The nodes are arranged from left to right according to their order of appearance in the document.



--- Document root ---
...
Policy and Legislation
Global Warming is Pushing Wildlife Towards ...
In Brief: Global climate change is ...
By now, many people have heard of ...
A warming world threatens wildlife ...
While global warming threatens ...
Sea Turtles

Figure 2: An example output of the proposed system.

3.2 The Learning Approach

The training set consists of (x_i, y_i) pairs where x_i is i^{th} document and y_i is the golden standard hierarchy (i.e. tree) for that document. For each document in the set, the learning algorithm (Figure 4) works on the units one by one and considers the attachment of a unit to its parent unit as a positive example (regular lines in Figure 3). The negative examples are the potential attachments not realized in the golden standard hierarchy (dashed lines in Figure 3). Here, two constraints due to the document flow are applied. First, a unit cannot be attached to a heading unit coming after it in the document order. Second, the connections cannot cross each other according to the projectivity rule (Covington, 2001).

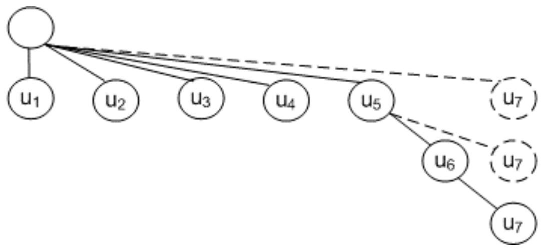


Figure 3: Part of an example document graph.

```
1: For each document $x_i$ in the training set
2:     For each unit $u_j$ in $x_i$
3:         $p$ = parent($u_j$)
4:         Set ($p$, $u_j$) as positive_example
5:         $prev = u_{j-1}$
6:         While ($prev$ != null)
7:             If ($prev$ != $p$)
8:                 Set ($prev$, $u_j$) as negative_example
9:             $prev$ = parent($prev$)
```

Figure 4: Training algorithm.

3.3 Features

Each unit is associated with the following features:

(1) Formatting features: These are related to the formatting of the unit; e.g. font size and boldness.

(2) DOM tree features: These features are related to the DOM tree of the document (W3C, 2005); e.g. the DOM address of the unit (Feng *et al*., 2005).

(3) Content features: These features specify whether a unit contains certain cue words or phrases; e.g. “back to top”, “login”, etc.

(4) Other features: These include the number of characters and the punctuation mark of the unit.

We utilize contextual information in the document hierarchy. We use u_{ij} to denote the unit with i levels above a unit u and j units to its left; e.g., u_{10} denotes the parent and u_{01} denotes the preceding sibling. We define *composite features* of two units u and u_{ij} (F_{ij}) as the difference of their features. Such information can be useful in determining the sectional hierarchy. For example, a heading unit (i.e. parent) is usually more emphasized than the underlying text unit in terms of formatting. Similarly, units under the same heading, i.e. siblings, usually have similar formatting. Finally, we define *global features* such as the depth of the tree.

3.4 Variations of the Testing Approach

In testing, we adapted an incremental approach similar to a previous work on syntactic parsing (Collins and Roark, 2004). Initially, the set of partial trees is empty. Whenever the next unit in the document is processed, its potential attachments to the partial trees are considered and the set is updated (ADVANCE). Restrictions due to the document flow are also applied. To prevent the exponential growth of the set, we maintain only the top k (i.e. beam width) highest scored trees at each step (FILTER). In the pre-processing, each heading u_j is connected with u_{j+1} because a heading is always followed by a child unit in the hierarchy. Also, a unit can only attach to a heading. We developed the following modifications of the algorithm:

M_1: The score obtained from the SVM model is converted into a probability value using a sigmoid function (Platt, 1999; Mayfield *et al*., 2003). We use *parent*(i)=j to denote that there is a parent-child dependency between the units i and j. If we assume that the probabilities of such dependencies are mutually independent, the probability of building a hierarchy with n units is:

$$\prod_{i=1}^{n} P(parent(i) = j) \qquad (2)$$

M_2: The testing algorithm is run in two levels. The first level runs on only heading units and the second level runs on non-heading units.

M_3 - M_4: The partial trees are given ranks based on the output of SVM. In M_3, the times a tree has obtained rank “1” are summed, whereas in M_4, the ranks at each step are summed to obtain the score.

4 EVALUATION

We selected 12 TREC queries (TREC, 2004) and built a corpus with the top 25 results of Google for each query. In Table 1, some statistics are given for the corpus. We implemented the system using the GATE framework (GATE, 2009), Cobra Java HTML Renderer and Parser Toolkit (The Lobo Project, 2009) and SVM-light (Joachims, 1999).

First, we evaluated the classification of units as heading or not (Model 1). We used five different feature sets (Table 2) including features of the current unit (F_n) and neighbouring units. The most accurate results were obtained when all the features were included. We evaluated the effect of different kernel types for SVM on that feature set (Table 3).

The second model is the tree learning approach for sectional hierarchy extraction (Model 2). We used different feature sets, including composite features of the current unit with candidate parent (F_{10}), siblings (F_{01}, F_{02}) and grandparent (F_{20}) as in Table 4. The accuracy is defined as the ratio of correctly identified parent-child relationships to the total number of such relationships in the hierarchy. We obtained the best accuracy when the feature set F_{10}, F_{01} was used. In Table 5, the results for the main algorithm (M_0) and the modifications (M_1 to M_4) are given for polynomial kernel and a beam width of 100. We compared the effect of using the headings identified with Model 1 and using manually marked headings. The best accuracy result was obtained (0.76) when manually marked headings were used.

Table 1: Corpus statistics.

	Training Set	Test Set
Number of documents	240	60
Avg. number of text units	107.1	97.6
Avg. hierarchy depth	4.1	4.1
Avg. number of headings	10.8	10.8

Table 2: Feature sets used in heading classification.

Feature Set	Number of Features
F_n, F_{n+1}	46
F_n, F_{n+1}, F_{n-1}	73
F_n, F_{n+1}, F_{n+2}	73
$F_n, F_{n+1}, F_{n+2}, F_{n-1}$	100
$F_n, F_{n+1}, F_{n+2}, F_{n-1}, F_{n-2}$	127

Table 3: Results for heading classification (Model 1).

	Recall	Precision	F-measure
Linear	0.71	0.80	0.75
Polynomial (d=2)	0.73	0.84	0.78
Polynomial (d=3)	0.72	0.84	0.78
Polynomial (d=4)	0.70	0.86	0.77

Table 4: Feature sets used in hierarchy extraction.

Feature Set	Number of Features
F_{10}	11
F_{10}, F_{01}	24
F_{10}, F_{01}, F_{20}	36
$F_{10}, F_{01}, F_{20}, F_{02}$	48

Table 5: Results for hierarchy extraction (Model 2).

Method	Learning Algorithm	Model 1 headings	Manual headings
M_0	Classification	0.58	0.75
M_0	Ranking	0.56	0.75
M_1	Classification	0.55	0.76
M_2	Classification	0.49	0.66
M_3	Classification	0.51	0.69
M_4	Classification	0.51	0.67

5 CONCLUSIONS

In this paper, we considered the problem of sectional hierarchy extraction from Web documents and adapted a tree learning approach as a solution. The proposed system was evaluated on a corpus of Web documents for heading and sectional hierarchy extraction. The results show that the system has acceptable results for unrestricted domain of Web documents. As further work, we will consider the division of Web documents into content blocks. This information can be used as features to improve the accuracy of the system. Also, the corpus of Web documents will be extended with additional queries.

ACKNOWLEDGMENTS

This work was supported by the Bogazici University Research Fund, Grant No. 07A106 and İstanbul Kültür University.

REFERENCES

Branavan, S. R. K., Deshpande, P., Barzilay, R., 2007. Generating a table-of-contents. In *Proc. of ACL*.

Brugger, R., Zramdini, A., Ingold, R., 1997. Modeling documents for structure recognition using generalized n-grams. In *Proc. of ICDAR*, pp. 56–60.

Collins, M., Roark, B., 2004. Incremental parsing with the perceptron algorithm. In *Proc. of ACL*.

Covington, M. A., 2001. A fundamental algorithm for dependency parsing. In *Proc. of ACM Southeast Conference*.

Curran, J. R., Wong, R. K., 1999. Transformation-based learning for automatic translation from HTML to XML. In *Proc. of ADCS99*.

Feng, J., Haffner, P., Gilbert, M., 2005. A learning approach to discovering Web page semantic structures, In *Proc. of ICDAR*, pp. 1055 – 1059.

GATE, 2009. A General Architecture for Text Engineering. Available at: http://gate.ac.uk/.

Joachims, T., 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Mao, S., Rosenfeld, A., Kanungo T., 2003. Document structure analysis algorithms: a literature survey. In *Proc. of SPIE Electronic Imaging*, pp.197—207.

Mayfield, J., McNamee, P., Piatko, C., Pearce, C., 2003. Lattice-based tagging using Support Vector Machines. In *Proc. of ICIKM*, pp. 303-308.

Platt, J. C., 1999. Probabilistic outputs for Support Vector Machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, A. Smola, P.Bartlett, B. Scholkopf, D. Schuurmans (eds.). MIT Press.

Shilman, M., Liang, P., Viola, P., 2005. Learning non-generative grammatical models for document analysis. In *Proc.of ICCV*.

The Lobo Project, 2009. Cobra: Java HTML renderer & parser. Available at: http://lobobrowser.org/cobra.jsp.

TREC, 2004. Text REtrieval Conference. Available at: http://www.trec.org.

W3C, 2005. Document Object Model (DOM). Available at: http://www.w3.org/DOM/.

Xue, Y., Hu, Y., Xin, G., Song, R., Shi, S., Cao, Y., Lin, C.-Y., Li, H., 2007. Web page title extraction and its application. In *Information Processing and Management*, Vol. 43, No. 5, pp. 1332-1347.