Structure and Content Analysis for HTML Medical Articles: A Hidden Markov Model Approach

Jie Zou, Daniel Le and George R. Thoma
Lister Hill National Center for Biomedical Communications
National Library of Medicine
8600 Rockville Pike, Bethesda, MD, 20894
{jzou, daniel, gthoma}@mail.nih.gov

ABSTRACT
We describe ongoing research on segmenting and labeling HTML medical journal articles. In contrast to existing approaches in which HTML tags usually serve as strong indicators, we seek to minimize dependence on HTML tags. Designing logical component models for general Web pages is a challenging task. However, in the narrow domain of online journal articles, we show that the HTML document, modeled with a Hidden Markov Model, can be accurately segmented into logical zones.

Categories and Subject Descriptors
H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *Indexing methods*; I.7.5 [**Document and Text Processing**]: Document Capture – *Document analysis*.

General Terms
Algorithms, Performance, Design, Experimentation

Keywords: Document Layout Analysis, Document Object Model (DOM), Web Information Retrieval, HTML Document Segmentation, HTML Document Labeling, Text Mining

1. INTRODUCTION
Maintaining MEDLINE®, the world's preeminent bibliographic database of the biomedical journal literature, containing over 14 million citations, is one of the most important tasks at the National Library of Medicine. With increasing numbers of journal articles published online in the HTML format, it is important to seek automated techniques to extract bibliographic data, including title, authors, affiliations, citations, grant numbers, databank numbers, etc., from these online journal articles.

The task of understanding the HTML document structure and content is to segment the document into blocks (*HTML segmentation*) and then assign logical labels to these blocks (*HTML labeling*). This preprocessing step significantly expedites subsequent information extraction processes and markedly increases their reliability.

Many previous Web-page-structure-understanding studies have adopted simple algorithms which depend heavily on the HTML tags: a small set of them, such as <P>, <TABLE>, <LI>/<UL> and <H1>~<H6>, serving as the segment indicators [1,3,4]. However, as shown in our previous work [7], due to the flexibility of the HTML syntax, conceptually and visually similar pages can be implemented by completely different HTML codes, and therefore the tag-dependent algorithms are not reliable.

Our HTML page content analysis algorithm identifies five logical components: Title, Authors, Affiliations, Abstract and References, from journal article Web pages. By rendering the HTML pages in a WebBrower, the extensively studied document image layout analysis algorithms can be borrowed to conduct HTML document segmentation. Besides text features, geometric features can also be extracted. We emphasize the importance of these tag-independent geometric and text features. The likelihoods of the HTML blocks to be particular logical components are calculated from feature statistics collected from historic MEDLINE data. The components of a journal article Web page are modeled with a Hidden Markov Model, followed by the Viterbi algorithm [2] to find the optimal state sequence, which concludes the structure understanding and component labeling process.

2. THE ALGORITHM
During rendering the HTML document in a WebBrowser control, a DOM (Document Object Model) [5] tree is created, at which point the logical component analysis algorithm begins.

In an HTML document, in order to implement certain features, such as changing font attributes (e.g., <FONT>, <B>), adding links (<A> tag) and so on, simple text lines often break into several DOM nodes at different levels of the DOM tree. We categorize HTML tags into two types: *Inline* tags, including <A>, <B>, <SUP>, etc., which do not introduce line breaks; *Line-Break* tags, including <TABLE>, <P>, <DIV>, etc., that do. The first step of the algorithm is to merge consecutive inline DOM nodes to avoid breaking text lines. Applying the recursive function below, starting from <BODY> node, a set of elementary component zones is collected.

```
Function CollectComponentZones (Children)
{  FOR each DOM node, _node_, of Children
   {  IF _node_ is a line-break node
      {  Save the leaf zone formed by the previous inline nodes.
         IF _node_ has no children or all children are inline nodes
            Save _node_ as a component zone
         ELSE
            CollectComponentZones (_node_'s children)
      }ELSE
      Merge _node_ with its previous adjacent inline siblings
}}
```

Copyright 2007 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
DocEng'07, August 28–31, 2007, Winnipeg, Manitoba, Canada.
Copyright 2007 ACM 978-1-59593-776-6/07/0008...$5.00.

This is the only step in our algorithm that uses the HTML tag information. Subsequent steps of the algorithm are independent of HTML tags. We are interested only in the text of the HTML document; therefore, the component zones, which contain no text or only space characters, are labeled as trivial zones. The remaining non-trivial zones are modeled and analyzed by the Hidden Markov Model described below.

The Hidden Markov Model we use to model online journal article Web pages has the following parameters:

• N, the number of states (labels) in the model, in which individual states are $S = \{S_1, S_2, \cdots, S_N\}$. Our goal is to detect *Title*, *Authors*, *Affiliations*, *Abstract* and *References*. These five states are referred to as *Major* states in the subsequent discussion. In the HTML journal articles, Abstract and References usually have headings, which are zones containing descriptive words, such as "Abstract" and "References". In order to use these distinctive landmarks, we also include the following two states in the model: *Abstract Heading* and *Reference Heading*. We refer to these as *Heading* states in the subsequent discussion. For the remaining zones, one design choice is to label them as "Other" zones. However, this will introduce loops. For example, the state transition instance, Title → Other → Author → Other → Title, is legitimate according to the model, but will not appear in real HTML articles. For most articles, the Major and Heading states are usually in the following order: starting from Title, then Authors, Affiliations, Abstract Heading, Abstract, Reference Heading, and finally References. In order to explicitly incorporate this constraint into the model, we introduce a further 9 states. These 9 states, as listed in Figure 1, are referred to as *Minor* states in the subsequent discussion. In our HMM, N therefore equals 16. The complete HMM structure, illustrated in Figure 1, is a Barkis-like (left-right) HMM. The model is able to accommodate certain missing states, but requires the states in the defined order.

• $A = \{a_{ij}\}$, the state transition probability matrix, where $a_{ij} = P[S_j \mid S_i]$, $1 \le i, j \le N$. In our model, this is a 16 by 16 matrix. If there is no link between two states in Figure 1, the corresponding element of A is 0. The remaining elements of A are estimated from a small set of manually-labeled training samples.

• $\pi = \{\pi_i\}$, the initial state distribution, where $\pi_i = P[S_i]$, $1 \le i \le N$. In our algorithm, the initial state is always "PreTitle".

• $B = \{b_j(O_t)\}$, the likelihoods of an observation O_t to be state S_j. In our case, the total number of observations, T, is the number of component zones collected by the recursive function. The observation, O_t, includes the text and geometric features. The likelihood, $b_j(O_t)$, is calculated with a Naive Bayesian classifier.

We collect the word frequencies from 10 years of MEDLINE historic data of Title, Authors, Affiliations, and Abstract. MEDLINE citations do not contain information for References and Minor zones. We merge the Title and Author word collections to build the word frequency for the References zones. For the Minor zones, we collect the word frequencies from a set of 25 training documents.

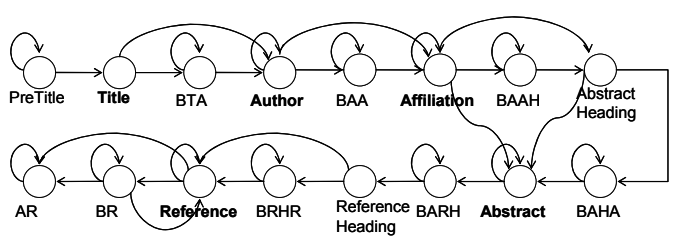
We assume independence among the words, and the likelihood of a zone having label j given an observation of k words, $W = \{w_1, \cdots w_k\}$, in the zone, can be calculated as: $b_j(W) = \prod_k b_j(w_k)$.

The other features (F) used for the classification include: the number of words (nw) in the zone, the left position (*left*), the top position (*top*) and the height (*height*) of the zone. These features and the words are assumed to be independent, and therefore the likelihood of a zone having label j is:

$$b_j(O) = b_j(F) b_j(W) = b_j(nw) b_j(left) b_j(top) b_j(height) \prod_k b_j(w_k).$$

After the likelihood of every component zone is calculated, the Viterbi algorithm [2] is applied to find an optimal state sequence, which assigns a logical label to each zone and concludes the HTML segmentation and labeling process.

An important property of our structure and content analysis algorithm is that it is minimally dependent on the HTML tags, and is therefore tolerant to different HTML implementation styles. Provided that the states (labels) do not switch their order in the document, the Hidden Markov Model and the labeling algorithm are applicable.



PreTitle: Before Title
BTA: Between Title and Authors
BAA: Between Authors and Affiliations
BAAH: Between Affiliations and Abstract Heading
BAHA: Between Abstract Heading and Abstract
BARH: Between Abstract and Reference Heading
BRHR: Between Reference Heading and References
BR: Between References
AR: After References

Fig. 1. The HMM model for HTML journal articles. It is nearly a Barkis (left-right) model, except that Reference and BR (Between References) may iterate by themselves.

3. EXPERIMENTAL RESULTS

We have applied the algorithm to a set of medical journal article HTML pages. Figure 2 shows the labeling result for one of these articles. In the top portion of the article, note that the Title, Author, Affiliation and Abstract zones are correctly identified (indicated by solid arrows and the bounding box). In the bottom portion of the article, even though the references are in a concise format, they are also correctly identified.

Our preliminary evaluation is on 129 articles from 22 medical journals. They follow very different HTML implementation styles. The ground truth labeling is manually created. Only the Major labels are considered in the evaluation. The F-measures are 99.6% for Title, 100.0% for Author, 88.6% for Affiliation, 91.7% for Abstract, and 99.5% for References.

The errors in Affiliations are mostly due to over-labeling, as indicated by a dotted arrow in Figure 2, and are not considered serious. The Abstract contains heterogeneous text and poses difficulties in identification, especially when the paragraphs are

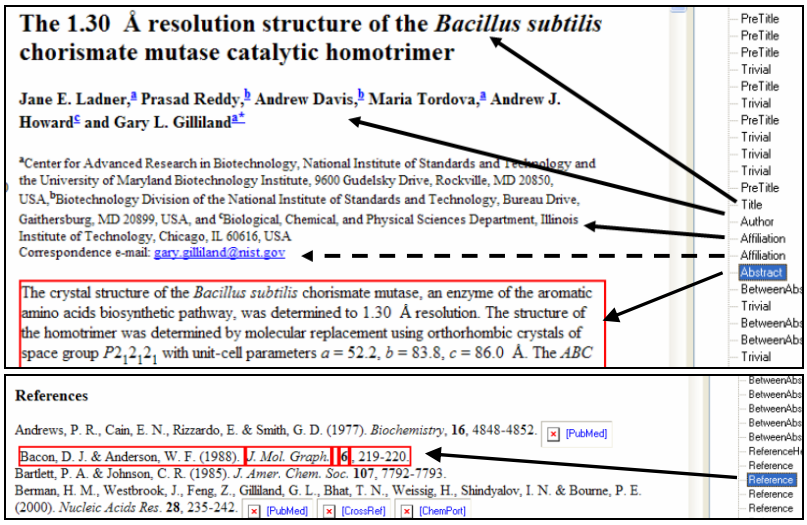


Fig. 2. An example of logical labeling. The top and bottom portions of an article are shown.

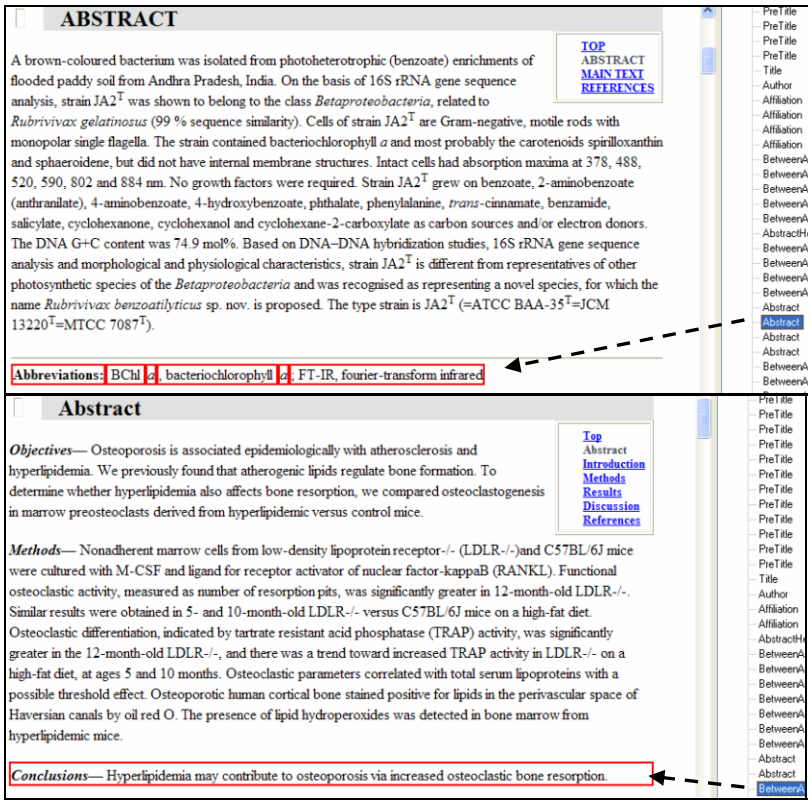


Fig. 3. Typical errors of abstract zones. Top: over-labeling; Bottom: under-labeling.

short, as shown in Figure 3. Further research is under way to solve these problems.

4. DISCUSSION AND CONCLUSIONS

We have described an HMM-based structure and content analysis algorithm for online medical journal articles. Although larger scale evaluation remains to be conducted, a preliminary test shows promising results. The performance can be further improved by using more advanced techniques in the algorithm and acquiring more training samples. For example, the Naïve Bayesian classifier can be replaced by Support Vector Machine and AdaBoost, which have proven to be successful in Text Classification [6].

The HMM allows for missing components, but requires components to be in a predefined order. However, these components sometimes follow a different order, such as in commentaries and editorial articles. Further research is also required to address this problem.

We emphasize that the main advantage of our labeling algorithm is that it relies more on text and geometric features than on HTML tags. HTML tags, due to the flexible HTML syntax, are not reliable features (indicators). Designing a logical model for Web pages in general remains a challenging and interesting research topic, but our method may also be applicable to other narrow domains, such as Blogs and News pages.

5. ACKNOWLEDGMENTS

We thank Dr. Jong Woo Kim for collecting the MEDLINE historic data and Dr. Song Mao for several valuable discussions. This research was supported by the Intramural Research Program of the National Institutes of Health, National Library of Medicine, and Lister Hill National Center for Biomedical Communications.

6. REFERENCES

[1] Diao, Y., Lu, H., Chen, S., and Tian, Z., Toward Learning Based Web Query Processing, *Proc. of International Conference on Very Large Databases*, 2000, 317-328.

[2] Forney, G.D. Jr., The Viterbi Algorithm, *Proceedings of the IEEE*, 61, 3, 1973, 268-278.

[3] Kaasinen, E., Aaltonen, M., Kolari, J., Melakoski, S., and Laakko, T., Two Approaches to Bringing Internet Services to WAP Devices, *Proc. 9th WWW Conference*, 2000, 231-246.

[4] Lin, S.-H., and Ho, J.-M., Discovering Informative Content Blocks from Web Documents, *Proc. ACM SIGKDD*, 2002.

[5] Marini, J., *The Document Object Model, Processing Structured Documents*, McGraw-Hill/Osborne, 2002.

[6] Sebastiani, F., Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, 34, 1, 2002, 1-47.

[7] Zou, J., Le, D., Thoma, G.R. , Combining DOM tree and Geometric Layout Analysis for Online Medical Journal Article Segmentation, *Proc. JCDL*, 2006, 119-128.