Automated Article Links Identification for Web-based Online Medical Journals

Daniel X. Le and George R. Thoma
Lister Hill National Center for Biomedical Communications
National Library of Medicine
Bethesda, MD 20894

ABSTRACT

As part of research into Web-based document analysis including Web page downloading and classification, an algorithm has been developed to automatically identify article links in Web-based online journals. This algorithm is based on feature vectors calculated from attributes and contents of links extracted from HTML files, and an instance-based learning algorithm using a nearest neighbor methodology to identify article links. The performance of the algorithm has been evaluated using a sample size of several thousand HTML links of Web-based medical journals. Evaluation shows that the algorithm is capable of identifying article links at an accuracy greater than 99 %.

Keywords: Instance-based learning algorithm, a nearest neighbor methodology

1. INTRODUCTION AND BACKGROUND

In recent years, the Internet and the World Wide Web have become the most popular and efficient medium to exchange information worldwide, and an increasing number of biomedical journal publishers provide their subscribers with access to online journals. As a result, the extraction of bibliographic records from these online journals is important for automatic document searching, document delivery and automated data entry. The Lister Hill National Center for Biomedical Communications, a research and development division of the National Library of Medicine (NLM), is developing an automated system, the Web-based Medical Article Records System (WebMARS)[1], to download, analyze and extract bibliographic information from Web-based journal articles to produce citation records for its MEDLINE® database. This system (1) classifies and downloads Web document articles, (2) converts PDF files to HTML files, if necessary, (3) parses HTML files to create and label text zones, (4) extracts, modifies and reformats the citation information in labeled text zones for validation by an operator, and finally (5) uploads the citation records to another NLM database for indexing by content experts. This paper describes one component of the downloading process of this system: the identification of article links in Web-based online journals. In this paper, we propose an automated technique to identify article links for downloading using feature vectors calculated from attributes and contents of links extracted from HTML files and an instance-based learning algorithm using a nearest neighbor methodology.

The rest of this paper is divided into five sections. Section 2 provides a discussion on HTML links and identification features. Section 3 presents an instance-based learning algorithm and a nearest neighbor methodology. Section 4 describes the Web-based article links identification process. Experimental results and summary are in Sections 5 and 6.

2. HTML LINKS AND IDENTIFICATION FEATURES

As defined in the HTML 4.01 specification published by the World Wide Web Consortium (W3C), a link is a connection from one Web resource to another. It has two ends -- called *anchors* -- and a direction. The link starts at the "source" anchor and points to the "destination" anchor, which may be any Web resource (e.g., an image, a video clip, a sound bite, a program, an HTML document, an element within an HTML document, etc.) [3]. Links can be defined by the “A” element or by the “LINK” element. The former is used to retrieve another Web resource such as an HTML or PDF article, while the second is used to show document relationships such as the position of a document within a series of documents. Since our system is required to identify article links and then download their associated HTML or PDF articles, we are only interested in analyzing the “A” element links.

The format of the “A” element link is defined as “<A attribute>content</A>”. The link content has 4 features which are defined as “tag”, “tag-attribute”, “tag-attribute-value”, and “caption”. The link attribute consists of 30 features: “id”, “class”, “style”, “title”, “lang”, “dir”, “onclick”, “ondblclick”, “onmousedown”, “onmouseup”, “onmouseover”, “onmousemove”, “onmouseout”, “onkeypress”, “onkeydown”, “onkeyup”, “charset”, “type”, “name”, “href”, “hreflang”, “target”, “rel”, “rev”, “accesskey”, “shape”, “coords”, “tabindex”, “onfocus”, and “onblur” [4].

Furthermore, several link attribute features such as “lang”, “charset”, “type”, “href”, and “hreflang” have their own sub-features; however, in this paper we considered the sub-features of “href” features only. The general format of HREF is “<scheme>://<authority><path>?<query>#fragment”[2]. The “scheme” component defines the namespace of the Uniform Resource Identifiers (URI) and some common schemes are “ftp”, “gopher”, “http”, “mailto”, “news”, “telnet”, “rlogin”, “tn3270”, “wais”, and “file”. The “authority” component is usually defined by a Web server and often it represents a specified server on the Internet. The “path” component identifies the specific resource under its scheme and authority. The “query” component represents a string of information passed to a Web resource for specific information processing. The “#fragment” component represents a fragment of an object such as a Web-based text document and so it can be used to refer to parts of a document.

In the example that follows, the “A” element link is analyzed to provide its attribute features, its content features, and its detailed HREF scheme sub-features.

```
<A href="http://nih.nlm.gov/help/" target=_top>
<FONT face=arial color=# ffffff size=2>HELP</FONT></A>
```

content:
    <FONT face=arial color=#ffffff size=-2>HELP</FONT>

    tag:
        <FONT>
    tag-attribute:
        <[face][color][size]>
    tag-attribute-value:
        <[arial][#ffffff][-2]>
    caption:
        HELP

attribute:
    “href="http://nih.nlm.gov/help/" target=_top”

    “href”:
        http://nih.nlm.gov/help/
        scheme:
            http
        authority:
            nih.nlm.gov
        path:
            help
    “target”:
        _top

Since the attributes and the contents of the “A” element links consist of useful information such as “class”, “href”, “tag”, “tag-attribute”, “caption”, etc., they can be used to cluster data links and to identify article links. In this paper, we create a 9-component identification feature vector in which 5 components are from the link attributes (3 of them derived from the “href” link attributes), and 4 components from the link contents. The components of a feature vector are: (1) attribute-name, (2) attribute-value, (3) href-scheme, (4) href-fragment, (5) href-query-name, (6) content-tag, (7) content-tag-attribute, (8) content-tag-attribute-value, and (9) content-caption.

As an example, the 9-component identification feature vector of the above “A” element link is as follows:

(1) <href><target>
(2) <http://nih.nlm.gov/help/><_top>
(3) <http>
(4) <>
(5) <>
(6) <FONT>
(7) <[face][color][size]>
(8) <[arial][#ffffff][-2]>
(9) <HELP>

3. AN INSTANCE-BASED LEARNING ALGORITHM AND A NEAREST NEIGHBOR METHODOLOGY

The article links identification algorithm proposed in this paper is based on an instance-based learning algorithm using a nearest neighbor methodology. The instance-based learning algorithm starts by storing training examples of each Web-based journal name and later uses them to classify links in the new journal issue instance using a nearest neighbor methodology. The selection of the instance-based learning algorithm for this system was based on its ability to handle the enormous variation in the document structure of Web documents from different publisher Web sites. The stored training examples for each journal are considered as journal specific information, so the accuracy of the system depend on how well the stored information represents links for a particular journal. Unlike most commonly used learning algorithms that construct an explicit global representation of the target function, the instance-based learning algorithm forms a local approximation of the target function when a new instance must be classified.

A nearest neighbor methodology classifies a new instance into a class of a training example in which its distance to the new instance is minimum. Given a query feature vector X and a set of K training examples where each example consists of a pair of a feature vector and its class, the method calculates distances from the query feature vector to all training feature vectors. It then locates the closest training example Ki for which the distance to the query feature vector is the smallest. Finally, it assigns the class of the training example Ki to the query vector X.

4. WEB-BASED ARTICLE LINKS IDENTIFICATION PROCESS

The article links identification process consists of three steps: (1) parse an HTML file to extract all “A” element links, (2) create an identification feature vector for each link based on its attribute and content, and (3) apply an instance-based learning algorithm and a nearest neighbor methodology using a set of training examples to identify article links. Each step is discussed below.

4.1 Extract HTML “A” element links

In this step, the system connects to a publisher Web site, selects the initial page of a particular journal issue, and extracts all “A” element links in this page. Each “A” element link begins with “<A ” and ends with “</A>”. Note that in a production system this step must be permitted by copyright and subscription agreements.

4.2 Create identification feature vector

Based on the format of the “A” element link as “<A attribute>content</A>”, the system first parses the link to get the link attribute and the link content. Second, the system extracts the “href” string from the link attribute to build three “href” components: href-scheme, href-fragment, and href-query-name. The system then parses the remained link attribute to create two “attribute” components: attribute-name and attribute-value. Finally, the system analyzes the link content to get four “content” components: content-tag, content-tag-attribute, content-tag-attribute-value, and content-caption.

In order to build the three “href” components from the “href” string, the system reverses the procedure of building the URI string algorithm described in RFC 2396 [2] which can be summarized as follows:

The URI string building algorithm:

```
result = ""
a. if scheme is defined, then (1) append scheme to
     result and (2) append ":" to result
b. if authority is defined, then (1) append "//" to
     result and (2) append authority to result
c. append path to result
d. if query is defined, then (1) append "?" to
     result and (2) append query to result
e. if fragment is defined, then (1) append "#" to
     result and (2) append fragment to result
f. return result
```

The following is the algorithm to build the “href” components:

```
source = href-string
a. search the source string backward for “#”
     if found then extract the last part of the string
         (after “#”) and assign it to href-fragment
     else href-fragment = <>
b. search the remaining source string backward for “?”
     if found then extract and parse the last part of
         the string (after “?”) for query names and
         assign it to href-query-name
     else href-query-name = <>
c. search the remaining source string backward for “://”
     if found then extract the first part of the string
         (before “://”) and assign it to href-scheme
     else href-scheme = <>
```

4.3 Apply an instance-based learning algorithm using a nearest neighbor methodology

Finally, the article links identification algorithm compares the feature vector X of an “A” element link generated in the above step against all stored training examples for the same Web-based journal.

```
For each training example Ki
    Calculate the distance between X and Ki
End For
Calculate the minimum distance and locate the
    corresponding training example Kn
Assign the class of Kn to X
```

5. EXPERIMENTAL RESULTS

The article links identification algorithm described above has been implemented, and experiments have been conducted with Web-based document articles selected from 7 different medical journals. The system was trained with one journal issue in a training set, and tested with the remaining issues at each publisher Web site. The training set consisted of 7 Web pages containing 893 links (510 article links and 383 non-article links) and the testing set had 23 Web pages with 2,550 links (1282 article links, 1268 non-article links). Examples of a non-article link include an image, an advertisement, and a video clip. The experimental results show that the article links identification achieved an accuracy on the test data set of over 99.0 %. Errors were due to the incomplete representation of training examples in one of the journals. Figure 1 shows the results of the experiments and note that in the training and testing set columns, there are three numbers displayed - the first number is the total “A” element links in a journal issue, the second and third numbers located inside parentheses are the total article links and the total non-article links, respectively.

6. SUMMARY

The identification of article links from Web-based online medical journals using an instance-based learning algorithm, a nearest neighbor methodology, and HTML links features has been presented. The experimental results on a test set of 23 Web pages consisting of 2,550 links (1282 article links, 1268 non-article links) drawn from 7 different publisher Web sites are very encouraging and they showed that the system is able to successfully handle the variation of document structure in Web documents from different publisher Web sites.

7. REFERENCES

[1] D. X. Le, L. Q. Tran, J. Chow, J. Kim, S. E. Hauser, C. W. Moon, and G. Thoma, “Automated Medical Citation Records Creation for Web-Based Online Journals,” *The Fourteenth IEEE Symposium on Computer-Based Medical Systems*, Bethesda, MD, pp. 315-320, 2001.

[2] T. Berners-Lee, R. Fielding, and Masinter L., “Uniform Resource Identifiers (URI): Generic Syntax,” IETF RFC 2396, August 1998.

[3] http://www.w3.org/TR/html401/struct/links.html

[4] http://www.w3.org/TR/html401/struct/links.html#h-12.2

Journal Name	Training Set	Testing Set Actual	Testing Set Classified	Accuracy %
AAPS PharmSci	71 (7, 64)	72 (8, 64)	72 (8, 64)	100
(1 issue for training)		66 (0, 66)	66 (0, 66)	100
(5 issues for testing)		67 (3, 64)	67 (3, 64)	100
		78 (14, 64)	78 (14, 64)	100
		76 (12, 64)	76 (12, 64)	100
Abdom_Imaging	44 (25, 19)	41 (24, 17)	41 (24, 17)	100
(1 issue for training)		44 (25, 19)	44 (25, 19)	100
(3 issues for testing)		41 (24, 17)	41 (24, 17)	100
Academic Emergency Medicine	106 (46, 60)	103 (41, 62)	103 (41, 62)	100
(1 issue for training)		97 (43, 54)	97 (43, 54)	100
(3 issues for testing)		111 (44, 67)	111 (44, 67)	100
Heart	144 (69, 75)	146 (72, 74)	146 (72, 74)	100
(1 issue for training)		144 (67, 77)	144 (67, 77)	100
(3 issues for testing)		145 (70, 75)	145 (70, 75)	100
Journal of Agromedicine	66 (14, 52)	69 (16, 53)	69 (16, 53)	100
(1 issue for training)		61 (8, 53)	61 (8, 53)	100
(3 issues for testing)		67 (13, 54)	67 (13, 54)	100
Pediatrics	326 (276, 50)	214 (170, 44)	214 (169, 45)	99.53
(1 issue for training)		245 (198, 47)	245 (195, 50)	98.78
(3 issues for testing)		247 (205, 42)	247 (204, 43)	99.60
The New England Journal of Medicine	136 (73, 63)	137 (73, 64)	137 (73, 64)	100
(1 issue for training)		134 (73, 61)	134 (73, 61)	100
(3 issues for testing)		145 (79, 66)	145 (79, 66)	100

Figure 1: Experimental results on 7 publisher journal Web sites.