Flexible Web Document Analysis for Delivery to Narrow-Bandwidth Devices

Gerald Penn[1,2] gpenn@cs.toronto.edu
Jianying Hu[1,3] jianhu@avaya.com
Hengbin Luo[1] hengbinluo@hotmail.com
Ryan McDonald[2] r.mcdonald@toronto.edu

[1]Lucent Bell Labs
Language Modeling Research
600 Mountain Ave.
Murray Hill, NJ 07974, USA

[2]University of Toronto
Dept. of Computer Science
10 King's College Rd.
Toronto M5S 3G4, Canada

[3]Avaya Laboratories
Multimedia Technologies Research
211 Mt. Airy Road
Basking Ridge, NJ 07920, USA

Abstract

We propose a set of baseline heuristics for identifying genuinely tabular information and news links in HTML documents. A prototype implementation of these heuristics is described for delivering content from news providers' home pages to a narrow-bandwidth device such as a portable digital assistant or cellular phone display. Its evaluation on 75 web-sites is provided, along with a discussion of topics for future research.

1. Motivation

Delivery of content from the World Wide Web to devices other than computers with standard-sized displays is typically achieved through the use of a portal. The portal's function is to collect and prepare the content for delivery to the particular device with a great deal of attention to one of its particular modes of presentation. Ultimately, the range and potential sources of content are restricted by the portal to make this provision feasible.

The present research programme was motivated by speculation that recent advances in computational linguistics and document analysis could be used to replace portals, ultimately a filter to enforce syntactic restrictions on content, with more functional restrictions gleaned from the topic or domain to which the content pertains. Given that semantic restriction, linguistic and document-based analysis of semi-structured text can, in principle, be used to automatically provide the most relevant content at the right level of detail to a user and in the right format for the mode of delivery.

This is very much in keeping with the original aims of markup languages such as SGML and HTML, whose containers and attributes were meant to facilitate the identification of function, salience and other semantic relations as an abstraction from how they should be used to guide layout or presentation through a particular modality. The problem is that nearly all professionally designed web-sites (ab)use HTML as a means of customising the presentation of their content through a standard browser such as Netscape or Internet Explorer on a desktop or laptop computer. As a result, to the extent that their content is semantically or functionally marked at all, it is not structured well enough to be of much assistance when attempting to render it elsewhere.[1] Recent innovations in the W3C HTML standard for separating content from presentation, such as cascading style sheets, are widely used, but not so much as to replace the various proprietary extensions of HTML that actively discourage this distinction.

To test this speculation, we have chosen to focus on delivering news content through narrow-bandwidth devices (NBDs), e.g., portable digital assistants and cellular phone displays, where the means of transmission and the small dimensions of the display require a drastic reduction in the size of the content, and a reformatting of the resulting summary. Although we are concerned here with providing textual content, the requirements of this task are obviously related to those of delivering HTML news content by voice using a text-to-speech synthesiser. A number of both voice-based and text-based portals already exist for this domain, of course, but as methods improve, it will be possible to refine this domain to very specific topics of news, or expand it to include content such as personal home pages, for which there is unlikely to be sufficient commercial demand to support a portal.

Front pages of news providers functionally contain headlines plus possibly short summaries of news stories with links to fuller versions of those stories, as well as tabular information such as sports scores and stock quotes. Intermixed with this can be photos, links to streaming au-

[1]Of course, this also requires constant readjustment to parsers that strobe other web-sites for particular kinds of content (price quotes from competing corporations' web-sites, for example).

dio/video feeds as well as *text buttons* — small phrases such as 'Subscribe,' 'FAQ,' 'Shop,' etc., that facilitate navigation through or use of the site's services without necessarily providing news content in the strict sense. Syntactically, the grouping, wording and organisation of this information in HTML source can differ greatly among sites. Notably, the preferred means of precisely placing content in roughly columnar form on a web-browser's canvas using HTML is with tables, even if the intended mode of presentation is in no real sense tabular. This complicates the identification of genuinely tabular information, which is often a convenient enough and terse enough presentation of content to be highly preferable for delivery to an NBD. Recently, the HTML standard has allowed for free-text attributes to be added that could in principle describe function, but at present these seem never to be used on professionally designed sites.

In this paper, we consider two kinds of simple heuristics to assist in extracting relevant information for presentation to NBDs. The first seeks to identify genuinely tabular content (document entities whose layout is semantically significant), amidst the heavy use of tabular mark-up for non-tabular-content layout on a standard browser. The second seeks to identify reasonably self-explanatory hypertext links to news stories. These links and genuine-table captions can be delivered together as a "front page" to an NBD, with each news link linked to a summary of its story obtained using automatic text summarisation techniques, and each caption linked to its table. There is certainly a great deal of room for improvement in both categories — crucially, neither benefits from an automated training phase on real data, nor from the use of a language model of any kind; they are proposed here as a reasonable baseline for future work in this area that relies exclusively on very noisy structural cues from HTML source. These are described in the next two sections, followed by a description of a prototype implementation we have constructed based on them and its evaluation on 75 news, television/radio and corporate websites.

2. Tables

The TABLE container in HTML allows its content to be laid out in a two dimensional grid. We define *genuine tables* to be document entities where such a two dimensional grid is semantically significant. In particular, the correct interpretation of the content of a genuine table cell requires references to its row index, column index, or both. HTML pages use TABLE containers both as genuine tables, i.e., as a mechanism to convey certain categorized information in a concise manner, and for grouping contents visually into clusters for easy viewing.

This distinction in functionalities leads to the following characteristics of genuine tables:

1. they are truly two dimensional;
2. each cell is simple, i.e., short in length and containing no complex structures in itself; and
3. there is a high level of coherence both syntactically and semantically within rows and columns.

This third characteristic is somewhat vague and hence the most difficult to measure. It has been explored before with limited success in the context of detecting tables from plain ASCII text and scanned document images [9, 8, 7]. Defining coherence across table cells in the web domain is even more difficult because of the many formatting containers (affecting font, size, color etc.) often applied within table cells. For our baseline system we decided to start with a heuristic exploring the first two characteristics only, since they are particularly relevant and well defined in the web domain.

Before describing the heuristic, some terminology should be introduced. A TABLE container is said to be a *leaf table* iff there are no TABLE containers inside it. A TABLE container is *multi-row* iff it contains more than one TR container, the container used for denoting table rows, headers and footers. The actual entries in a table are identified in HTML by TD (table data) and TH (table header) containers. A TABLE container is *multi-column* iff one of its TR containers contains more than one TD or TH container that contains text other than punctuation and whitespace. There is also a class of HTML tags called *text-level formatting tags* [10]. These are the tags that control the font, font size, boldfacing etc. of displayed text. In other words, they format the text itself, not its layout.

Heuristic for identifying a TABLE container as a genuine table:

1. it is a leaf table;
2. it is multi-row and multi-column,
3. its table cells have:
    (a) at most one non-text-level-formatting tag, and
    (b) no lists, frames, forms or image tags; and
4. the length (in whitespace-delimited words) of the content of each cell is less than a threshold, δ.

The threshold is necessary because many TABLE containers consisting of multiple columns of news links satisfy the remaining conditions, and, as explained below, were therefore being excluded from consideration as news links. By adding a bound on the length of each table cell to the table heuristic, these were excluded. This bound also improves table precision slightly.

Once a table has been identified as genuine, a caption can be found for it:

1. in the table's CAPTION container, if present, or

2. from the first row provided it has:

   (a) a single cell/column, and

   (b) no tags except text-level formatting tags.

Captions are typically short enough to be suitable for presentation on NBDs as well.

3. News Links

Front-page *news items*, summaries of news that contain links to full news stories, cannot be found inside a single kind of container with the same reliability as tables — sometimes they occur as paragraphs, sometimes as one or more items in a list, sometimes as one or more cells of a (non-genuine) table, etc. Instead, we must be prepared to amalgamate strings of text together that may occur in different containers but still conceptually belong to the same news item. We can define *text regions* as the largest nodes in an HTML parse tree that:

1. contain text,

2. do not contain any tags except formatting, list, hyperlink, and image tags [10], and

3. are not contained in a genuine table.

Another challenge is to distinguish these from contiguous strings of text buttons, descriptions of news categories, e.g. "today's sports highlights," and from other potentially hyperlinked text on the page such as disclaimers and legal information. This last category can be excluded quite effectively on the basis of their position within the file (normally near the end) and the occurrence of certain key phrases such as `all rights reserved.' Text buttons and categories tend to be quite short. We can apply the following *continuity measure* to the string of words in a text region, w:

$$c(w) = \frac{\sum_{i=1}^{j} \min^k\{|w_i|, \mu\}}{\sum_{i=1}^{j} \min\{|w_i|, \mu\}}$$

where w can be split into j substrings, $w_1, \ldots, w_j$, each of which is either a single hyperlink container, or entirely free of hyperlink containers. When $k > 1$, this measure penalises strings that have a large number of very short hypertext links and rewards those that have long substrings of unlinked or continuously linked text. The threshold μ is used to prevent very long strings (such as paragraphs of news stories) from overwhelming the value.

Heuristic for identifying a string of text, w, as a news item:

1. w is exactly the text of a text region;

2. w is not contained inside a genuine table (as identified above); and

3. $c(w) > \theta$, for some threshold value, θ.

We then **heuristically identify news links** as those substrings of a news item that are:

1. the text contents of hyperlinked containers, and

2. whose lengths (in whitespace-delimited words) are greater than a threshold, ν.

4. Implementation

Our prototype implementation is written in Java 1.2 and uses the Swing XML parser with the W3C HTML 3.2 DTD,[2] and includes a graphical tool for inspecting a transformed HTML tree after each stage of operation. Given the URL of a news provider, it first identifies all leaf tables, then applies the other constraints of the heuristic to locate the genuine tables, using a value of $\delta = 4$ determined from experiments on a smaller data set. For each genuine table, it attempts to locate a caption using the two methods described above. It then locates all text regions (ignoring genuine tables), discarding those that have three or more key phrases from a pre-defined list of terms likely to appear in disclaimers and legal information. It then calculates the continuity measure on the remaining regions and applies the threshold value, θ. Experimentation with several parameters on three test sites yielded $k = 3$, $\theta = 16$, $\nu = 2$, and $\mu = 5$ as reasonable choices. Once the news links have been identified, the system follows each one to its full news story and creates a summary of it by using the first paragraph with the largest number of occurrences of non-stop-words in the news link itself.

The tables, captions, news links and summaries are then wrapped in a set of WML decks that can be transmitted to a PDA or cellular phone.

5. Evaluation

We tested our heuristics on 75 web-site front-pages, consisting of 49 news providers (such as `latimes.com` and `usatoday.com`), 16 television and radio sites (such as

[2] At the time this was implemented, the W3C HTML 4.0 DTD was incompatible with the Swing parser.

Site	Tables Precision	Tables Recall	News Links Precision	News Links Recall
News	85.2%	86.8%	75.1%	91.6%
TV/Radio	89.4	95.4	80.2	93.4
Corporate	0[1]	0[1]	81.7	100
Overall	86.3	89.8	76.1	92.2

Table 1. Evaluation Results.

pbs.org and wnbc.org), and ten corporate sites (such as lucent.com and ge.com). The content of their front pages was cached along with all pages accessible by one link and ground truth was determined by hand. For the purposes of evaluation, text displayed as image files (common among certain news providers who wish to preserve the "look" of their print media) were not counted. Extending this implementation with a means of recognising text in images [11, 1] remains an area of further research.

The results are shown in Table 1.

The primary source of errors is the lack of isomorphy between "units" of content — either genuine tables or news items — and nodes of the HTML parse tree, which our implementation still assumes. News link precision, however, was actually more adversely affected by the presence of content-free links such as "click here for full story," and "enter here to subscribe," and of some advertisements that were placed within news regions. By our definition, these do not count as news links because the links themselves provide no information on the content of a news story. One could, of course, make reference to the text of the pages that the links refer to in order to make a better decision. A small number of errors are also due to text being inserted by Javascript attachments, which our implementation cannot recognise yet.

6. Comparison with Other Work

This work bears a similarity to several other recent approaches to improving the state-of-the-art in web-based content delivery. Some, e.g., the Power Browser project [5, 4], have focussed on a range of problems connected to web-surfing with PDAs such as bandwidth, power consumption, pen-based query input, etc., and use a combination of techniques such as incremental indexing and keyword completion to address these. Others, e.g., Brin [3] and other work at Google, have used a bootstrapping technique to extract particular kinds of relations from on-line text, such as author-title relations for books. Ashish and Knoblock [2] describe a semi-automated method for "wrapper" generation that uses regular expressions for locating keywords combined with information about font size and indentation to identify structural units of interest within HTML documents. Cohen and Fan [6] use the automated rule-learning system, RIPPER, to detect simple lists of strings and lists of associations of strings and URLs in noisy HTML source based on a collection of nineteen features pertaining to the structure of the HTML parse tree.

[1] Among the ten corporate sites, one TABLE container was wrongly identified as genuine, and there was one genuine table, which was not identified.

7. Conclusion

A set of baseline heuristics have been presented for identifying genuine table content and news links in HTML documents, along with their evaluation in a general news domain on 75 news, television/radio and corporate web-sites. With the exception of news link precision, we find that these heuristics performed quite respectably in our evaluation and set a fairly high watermark for more sophisticated statistical approaches to achieve.

The heuristics provided in this paper operate without any training, either on HTML source (as in [6]) or on a language model; finding a statistical model with which to enhance these heuristics remains the most important future extension of this work. Incorporating a means of recognising text from images (as in [11, 1]) as well as better text summarisation methods for news summaries are also natural areas for further development. Developing a statistical model of coherence among table columns/rows that performs well in this domain is also an important area of enquiry.

References

[1] A. Antonacopoulos and D. Karatzas. Accessing textual information embedded in internet images. In *Proceedings of Internet Imaging II (IS&T/SPIE Electronic Imaging), San Jose*, volume 4311, 2001.

[2] N. Ashish and C. Knoblock. Wrapper generation for semi-structured internet sources. In *ACM SIGMOD Record, Special Issue on Management of Semi-structured Data*, December 1997. Invited paper.

[3] S. Brin. Extracting patterns and relations from the world wide web. In P. Atzeni, A. O. Mendelzon, and G. Mecca, editors, *The World Wide Web and Databases: International Workshop WebDB'98, Valencia, Spain, March 27-28, 1998, Selected Papers*, volume 1590 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 1999.

[4] O. Buyukkokten, H. Garcia Molina, and A. Paepcke. Focused web searching with pdas. In *The 9th International WWW Conference (WWW9), Amsterdam, Netherlands - May 15-19, 2000*, 2000.

[5] O. Buyukkokten, H. Garcia Molina, A. Paepcke, and T. Winograd. Power browser: Efficient web browsing for pdas. In *Proceedings of the Conference on Human Factors in Computing Systems*, 2000.

[6] W. Cohen and W. Fan. Learning page-independent heuristics for extracting data from web pages, 1999.

[7] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. A system for understanding and reformulating tables. In *Proceedings of the Fourth IAPR International Workshop on Document Analysis Systems*, pages 361–372, Rio de Janeiro, Brazil, December 2000.

[8] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Table structure recognition and its evaluation. In P. B. Kantor, D. P. Lopresti, and J. Zhou, editors, *Proceedings of Document Recognition and Retrieval VIII (IS&T/SPIE Electronic Imaging)*, volume 4307, pages 44–55, San Jose, CA, January 2001.

[9] M. Hurst and S. Douglas. Layout and language: preliminary investigations in recognizing the structure of tables. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 1043–1047, 1997.

[10] E. Ladd and J. O'Donnell. *Platinum Edition Using HTML 4, XML, and Java 1.2*. Que Books, 1999.

[11] D. Lopresti and J. Zhou. Locating and recognizing text in www images. *Information Retrieval*, (2):177–206, 2000.