Webpage Understanding: Beyond Page-Level Search

Zaiqing Nie Ji-Rong Wen Wei-Ying Ma

Web Search & Mining Group
Microsoft Research Asia
Beijing, P. R. China
{znie, jrwen, wyma}@microsoft.com

Abstract

In this paper we introduce the webpage understanding problem which consists of three subtasks: webpage segmentation, webpage structure labeling, and webpage text segmentation and labeling. The problem is motivated by the search applications we have been working on including Microsoft Academic Search, Windows Live Product Search and Renlifang Entity Relationship Search. We believe that integrated webpage understanding will be an important direction for future research in Web mining.

1. Introduction

The World Wide Web is a vast and rapidly growing repository of information, and various kinds of valuable semantic information are embedded in webpages. Some basic understanding of the structure and the semantics of webpages could significantly improve people's browsing and searching experience.

We have been working on novel Web applications beyond page-level search with different levels of webpage understanding granularity. Specifically,

Block-based Search: We segment a webpage into semantic blocks and label the importance values of the blocks using a block importance model [11]. Then the semantic blocks, along with their importance values, are used to build block-based Web search engines [1][3].

Object-Level Vertical Search: We extract and integrate all the Web information about a real world object/entity and generate a pseudo page for this object. These object pseudo pages are indexed to answer user queries, and users can get integrated information about a real-world object in one stop, instead of browsing through a long list of pages. Our object-level vertical search technologies have been used to build Microsoft Academic Search (http://libra.msra.cn) and Windows Live Product Search (http://products.live.com).

Entity Relationship Search: We have deployed an Entity Relationship Search Engine in the China search market called *Renlifang* (http://renlifang.msra.cn). In Renlifang, users can query the system about people, locations, and organizations and explore their relationships. These entities and their relationships are automatically mined from the text content on the Web (more than 1 billion Chinese webpages).

As we can clearly see, large-scale Web mining (especially webpage understanding) plays a critical role in the above search technologies. In this paper, we first introduce these search applications in detail to motivate the webpage understanding tasks: webpage segmentation, webpage structure labeling, webpage text segmentation and labeling. Then we formally define the webpage understanding problem. Finally we present integrated statistical models for these webpage understanding tasks.

2. Beyond Page-Level Search

Nowadays, major commercial search engines take a webpage as the basic information unit and return a list of pages as search results to users. *Is a webpage the only or best atomic unit for information search on the Web?* We have developed several novel search technologies and systems, in ways going beyond the current page-level search paradigm.

2.1 Block-based Search

The content of a webpage is usually much more diverse compared with a traditional plain text document and encompasses multiple regions with unrelated topics. Moreover, for the purpose of browsing and publication, non-content materials, such as navigation bars, decoration fragments, interaction forms, copyright notices, and contact information, are usually embedded in webpages. Instead of treating a whole webpage as a unit of retrieval, we believe that the characteristics of webpages make passage a more effective mechanism for information retrieval.

In [2], we propose a VIPS (**VI**sion-based **P**age **S**egmentation) algorithm to segment a webpage into multiple semantic blocks. VIPS makes use of page layout features such as font, color, and size to segment a page. It first extracts all suitable nodes from the tag-tree of the page, and then finds the separators between these nodes. Here,

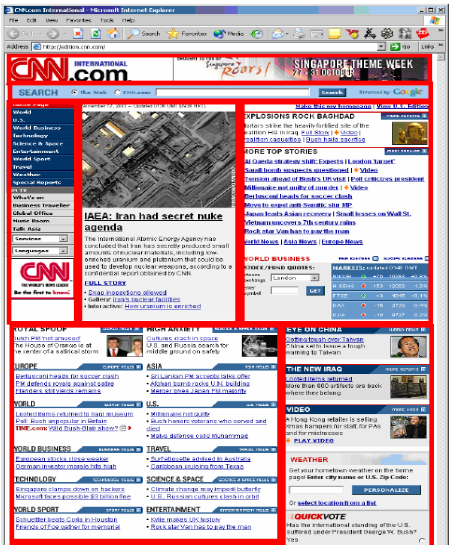


Figure 1. VIPS segmentation of a sample webpage

separators denote the horizontal or vertical lines in a page that visually do not cross any node. Figure 1 shows the result of using VIPS to segment a sample CNN webpage.

After segmenting a webpage into semantic blocks, we compute the importance values of the blocks using a block importance model [11]. Then the semantic blocks, along with their importance values, are used to build block-based Web search engines with block-level link analysis [1] and block-based ranking [3] algorithms, and finally to improve the relevance of search results.

2.2 Object-Level Vertical Search

Much structured information about real-world objects is embedded in webpages and online databases. We explored a new paradigm to enable web search at the object level. We developed a set of technologies to automatically extract, integrate and rank structured Web objects [6][7][8][14], and then build powerful object-level vertical search engines for specific domains such as product search, academic search, and local search.

Information (e.g., attributes) about a web object is usually distributed in many web sources and within small segments of webpages. The task of an object extractor is to extract meta-data about a given type of objects from every webpage containing this type of objects. For example, for each crawled product webpage, we extract the *name*, *image*, *price* and *description* of each product from it using machine learning algorithms. If all of these product pages or just a portion of them are correctly extracted, we will have a huge collection of meta-data about real-world products that could be used for further knowledge discovery and query answering. Our statistical study on 51,000 randomly crawled webpages shows that about 12.6 percent are product pages. That is, there are about 1 billion product pages within a search index of 9 billion webpages.

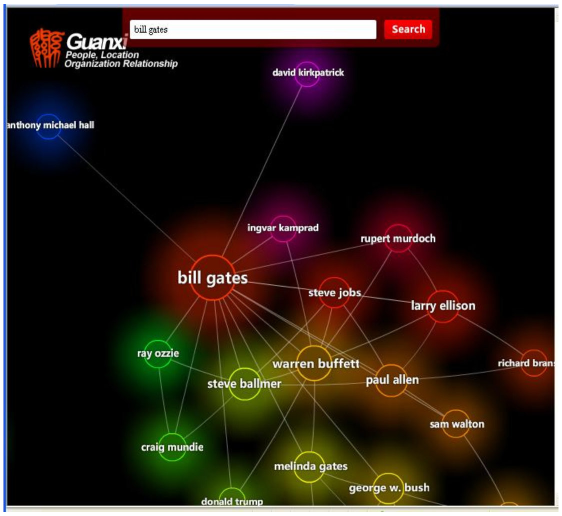


Figure 2. Automatically generated entity relationship graph for the query "Bill Gates" by our entity relationship search engine

However, how to extract product information from webpages generated by many (maybe tens of thousands of) different templates is non-trivial. One possible solution is that we first distinguish webpages generated by different templates, and then build an extractor for each template. We say that this type of solution is *template-dependent* (i.e., wrappers). However, accurately identifying webpages for each template is not a trivial task because even webpages from the same website may be generated by dozens of templates. Even if we can distinguish webpages, template-dependent methods are still impractical because the learning and maintenance of so many different extractors for different templates will require substantial human effort. By empirically studying webpages across websites about the same type of objects, we found strong template-independent features. The information about an object in a webpage is usually grouped together as an *object block*, as shown in Figure 4.

Using our Vision-based Page Segmentation (i.e., VIPS) algorithm and data record extraction technologies, we can automatically detect these object blocks, which are further segmented into atomic extraction units (i.e., HTML elements) called *object elements*. Each object element provides (partial) information about a single attribute of the web object. The web object extraction problem can be solved as a webpage structure labeling problem assuming we don't need to further segment the text content within the HTML elements [14].

With the Web object extraction and ranking technology, people can get integrated information about a real-world object in one stop, instead of browsing through a long list of pages. Our object-level vertical search technologies have been used to build Microsoft Academic Search and Windows Live Product Search. For more information about our object-level vertical search work, please refer to [6].

2.3 Entity Relationship Mining and Search

We have deployed an Entity Relationship Search Engine in the China search market called Renlifang. Currently Renlifang only serves in the Chinese language domain, and the knowledge is automatically mined from more than 1 billion crawled Chinese webpages.

Renlifang is a different kind of search engine, one that explores relationships between entities. In Renlifang, users can query the system about people, locations, and organizations and explore their relationships. These entities and their relationships are automatically mined from the text content on the Web. For each crawled webpage in Renlifang, the system extracts entity information and detects relationships, covering a spectrum of everyday individuals and well-known people, locations, or organizations.

Below we list the key features of Renlifang:

- **Entity Relationship Mining and Navigation**. Renlifang enables users to explore highly relevant information during searches to discover interesting relationships about entities associated with their query.
- **Expertise Finding**. For example, Renlifang could return a ranked list of people known for dancing or any other topic.
- **Web-Prominence Ranking**. Renlifang detects the popularity of an entity and enables users to browse entities in different categories ranked by their prominence on the Web during a given time period.
- **People Bio Ranking**. Renlifang ranks text blocks from webpages by the likelihood of being biography/description blocks.

Renlifang has been well received by Chinese Internet users and media with positive comments and millions of daily page-views in peak days. The English version of Renlifang is under development. In Figure 2, we show an automatically generated entity relationship graph using our English Renlifang prototype.

In entity relationship search, we need to extract the entity names and the related entities from both free text content and structured HTML elements within every webpage we crawled. So we need to use both page structure labeling and text segmentation and labeling technologies.

2.4 Categorization of Search Engines

Figure 3 shows a matrix about the categorization of search engines. Traditional web search engines can be classified as "page-level general search" engines, which simply crawl as many webpages as possible and treat each page as the basic retrieval unit. We see that there are two important trends towards next generation search engines. One trend is from general search to vertical search. We can restrict the search to a special domain and build page-level vertical search. For example, Google Scholar is an academic vertical search engine designed to help users to identify academic information. Another trend is from page-level to object-level. In some domains, data are more structured and uniform and it's relatively easy to define object schemas and conduct object extraction. Libra academic search is a typical application of object-level vertical search technologies. If we can automatically mine all types of entities and their relationships, we can build an object-level general search engine. This is actually the so-called Web Database dream, which aims to treat the whole Web as a huge database. The key here is to provide generic entity relationship mining and search technologies. Renlifang is a preliminary attempt towards this direction. For example, to extend Renlifang to support product relationship search, we only need to re-train the Named Entity Recognition model to optimize for product name extraction.

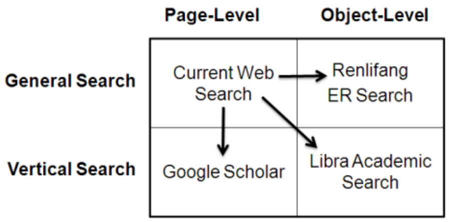


Figure 3. From Page-Level Search to Object-Level Search

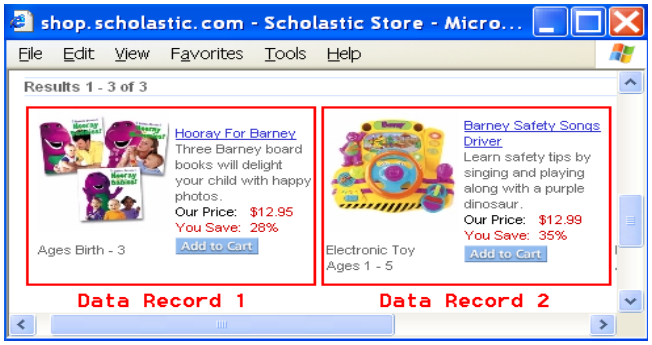
As we can see from the above search applications including block-based search, object-level vertical search and entity relationship search, some shallow understanding of the webpages will significantly improve users' browsing and searching experiences.

3. Problem Definition

In this section we define the webpage understanding problem which consists of three sub-tasks: webpage segmentation, webpage structure labeling, and webpage text segmentation and labeling.

3.1 Webpage Segmentation

To segment a webpage into semantically coherent units, the visual presentation of the page contains a lot of useful cues. Generally, a webpage designer would organize the content of a webpage to make it easy for reading. Thus, semantically coherent content is usually grouped together and the entire page is divided into regions for different content using explicit or implicit visual separators such as



Results 1 - 3 of 3

Hooray For Barney
Three Barney board books will delight your child with happy photos.
Our Price: $12.95
You Save: 28%
Ages Birth - 3

Barney Safety Songs Driver
Learn safety tips by singing and playing along with a purple dinosaur.
Our Price: $12.99
You Save: 35%
Electronic Toy
Ages 1 - 5

Figure 4. A sample webpage with two similar data records

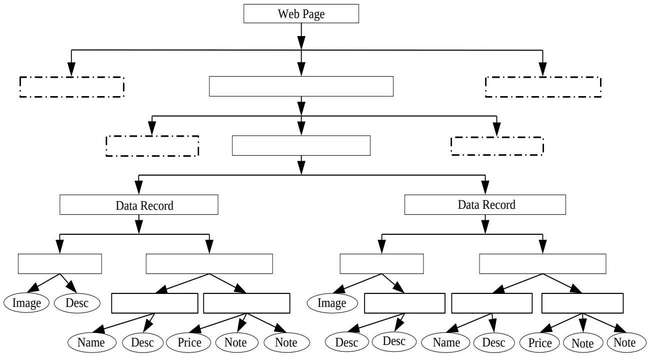


Figure 5. The vision-tree of the page in Figure 4.

lines, blank areas, images, font sizes, and colors [2]. Our goal is to derive this content structure from the visual presentation of a webpage.

We formally define the webpage segmentation problem below.

Definition 3.1 (Webpage Segmentation)*: Given a webpage, webpage segmentation is the task of partitioning the page at the semantic level and constructing a vision-tree for the page. Each node in the vision-tree will correspond to a block of coherent content in the original page (See Figure 1 for an example).*

Based on the definition, the output of webpage segmentation is the vision-tree of a webpage. Each node on this vision-tree represents a data region in the webpage, which is called a *block*. The root block represents the whole page. Each inner block is the aggregation of all its child blocks. All leaf blocks are atomic units (*i.e.*, elements) and form a flat segmentation of the webpage. Since vision-tree can effectively keep related content together while separating semantically different blocks from one another, we use it as the data representation format of the webpage segmentation results. Figure 5 is a vision-tree for the page in Figure 4, where we use rectangles to denote the inner blocks and use ellipses to denote the leaf blocks (or elements). Due to space limitations, the blocks denoted by dotted rectangles are not fully expanded.

3.2 Webpage Structure Labeling

After webpage segmentation, we will have a vision-tree representation of a webpage keeping semantically coherent content together as web blocks. The webpage structure labeling task is to assign semantic labels to the blocks on a webpage (i.e., nodes on vision-tree). For different applications, the semantic label space could be different. For example,

- For Web object extraction, the label space consists of a label called Object Block and several labels corresponding to the individual attribute names of the object (for example, the *name*, *image*, *price* and *description* of a product for sale). The web object extraction problem can be solved as a webpage structure labeling problem assuming we don't need to further segment the HTML elements which are the leaf nodes of the vision-tree [14].
- For the webpage main block detection application, the label space could consist of the following: Main Block, Navigation Bar, Copyright, Advertisement, etc.

Below we define the webpage structure labeling problem.

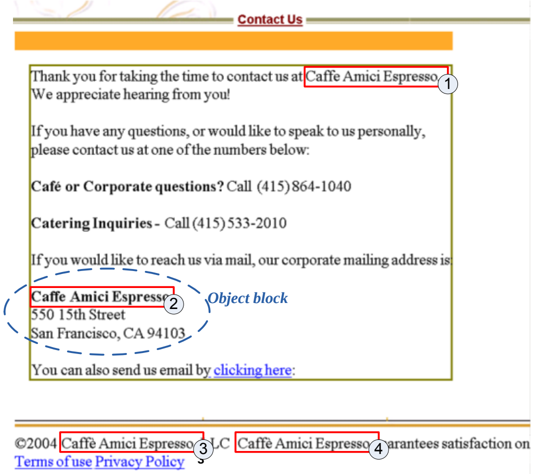
Definition 3.2 (Webpage Structure Labeling)*: Give a vision-tree of a page, let* $x = \{x_0, x_1, \cdots, x_N\}$ *be the features of all the blocks and each component* x_i *is a feature vector of one block, and let* $y = \{y_0, y_1, \cdots, y_N\}$ *be one possible label assignment of the corresponding blocks. The goal of webpage structure labeling is to compute maximum a posteriori (MAP) probability of* y *:*

$$y^* = \arg\max p(y \mid x)$$

3.3 Webpage Text Segmentation and Labeling

The page segmentation task segments a webpage into blocks and constructs a vision-tree for the webpage, and then the webpage structure labeling task detects the object blocks and labels HTML elements on the vision-tree using attribute names. However, how to effectively segment and label the text content inside HTML elements is still an open problem. As we pointed out in section 2, text segmentation and labeling are critical for entity relationship search engines which need to extract entity information from both the free text content and structured blocks of billions of crawled webpages. Since much of the text content on a webpage is often text fragments and not strictly grammatical, traditional natural language processing techniques that typically expect grammatical sentences, are no longer directly applicable.

In Figure 6, we show an example webpage containing local entity information. As we can see, the address information



Contact Us

Thank you for taking the time to contact us at Caffe Amici Espresso
We appreciate hearing from you!

If you have any questions, or would like to speak to us personally, please contact us at one of the numbers below:

Café or Corporate questions? Call (415) 864-1040

Catering Inquiries - Call (415) 533-2010

If you would like to reach us via mail, our corporate mailing address is:

Caffe Amici Espress
550 15th Street
San Francisco, CA 94103

You can also send us email by clicking here:

©2004 Caffè Amici Espresso LC Caffè Amici Espresso arantees satisfaction on
Terms of use Privacy Policy

Figure 6. An example webpage containing local objects

of the local business on the webpage is regularly formatted in a visually structured block: the first line of the block contains the business name with bold font; the second line contains the street information; the third line contains the city, state and zip code. As we can see, the attributes in an object block are always short strings. It is quite difficult to identify business names correctly only with the structure (i.e., visual layout) information and text features of these short strings (e.g., regular expression and word emission probability). For example, there is not much evidence for “Caffe Amici Espresso” to be labeled as the business name in the object block shown in Figure 6. Fortunately, the business name is mentioned not only in the object block but also in the natural language sentences outside the object block, such as “Thank you for taking the time to contact us at Caffe Amici Espresso” and “Caffe Amici Espresso guarantees satisfaction on all products we sell”.

We believe that if we could collect more information about an object, we can make better decisions on it. For example, it would be much more accurate and easier if we could label all the mentions of the business name “Caffe Amici Espresso” together, no matter where it appeared in the webpage: object blocks or natural language sentences.

Below we define the webpage text segmentation and labeling problem.

Definition 3.3 (Webpage Text Segmentation and Labeling): *Given a vision-tree of a page, let* $\mathrm{x} = \{\mathrm{x}_0, \mathrm{x}_1, \cdots, \mathrm{x}_N\}$ *be the features of all the word occurrences on the tree and each component* x_i *is a feature vector of one word occurrence. The goal of webpage text segmentation and labeling is to find the optimization segmentation and labeling* ***S*****:*

$$\mathbf{S}^* = \arg\max_{\mathrm{S}} p(\mathbf{S} \mid \mathbf{X})$$

4. Models for Webpage Understanding

In this section, we introduce models/algorithms for webpage understanding subtasks: webpage segmentation, structure labeling, and web text segmentation and labeling. In particular, in Section 4.4, we argue that joint optimization of the subtasks significantly improves the performance of the individual subtasks.

4.1 Webpage Segmentation

An intuitive way to segment a page is based on the layout of a webpage. This way, a webpage is generally separated into 5 regions: top, down, left, right and center [5]. The drawback of this method is that such a layout template cannot fit into all webpages. Furthermore, the segmentation is too rough to exhibit semantic coherence.

Compared with the above segmentation, Vision-based Page Segmentation (VIPS) excels in both an appropriate partition granularity and a coherent semantic aggregation. By detecting useful visual cues based on DOM structure, a tree-like vision-based content structure of a webpage is obtained. The granularity is controlled by *a degree of coherence* (DOC) which indicates how coherent each block is. VIPS can efficiently keep related content together while separating semantically different blocks from each other. Each block in VIPS is represented as a node in a tree. The root is the whole page; inner nodes are the top level coarser blocks; child nodes are obtained by partitioning the parent node into finer blocks; all leaf nodes consist of a flat segmentation of a webpage with an appropriate coherent degree. The stopping of the VIPS algorithm is controlled by a predefined DOC (PDOC), which plays a role as a threshold to indicate the finest granularity that we are satisfied with [2]. The segmentation only stops when the DOCs of all blocks are no smaller than the PDOC.

However, if we combine webpage segmentation and structure labeling together, we don’t need to guess the predefined DOC to segment a webpage into blocks with a satisfied degree of granularity. The leaf nodes of the vision-tree could be the HTML elements, and we just need to assign labels to the nodes of the tree to detect the blocks we are interested in.

4.2 Webpage Structure Labeling

After the webpage segmentation task, a webpage is represented as a vision-tree, and the webpage structure labeling task becomes the task of assigning labels to the nodes on a vision-tree. We introduce a probabilistic model called Hierarchical Conditional Random Field (HCRF) model for webpage structure labeling.

For the page in Figure 4, the HCRF model is shown in Figure 7, where we also use rectangles to denote inner nodes and use ovals to denote leaf nodes. The dotted rectangles are for the blocks that are not fully expanded. Each node on the graph is associated with a random

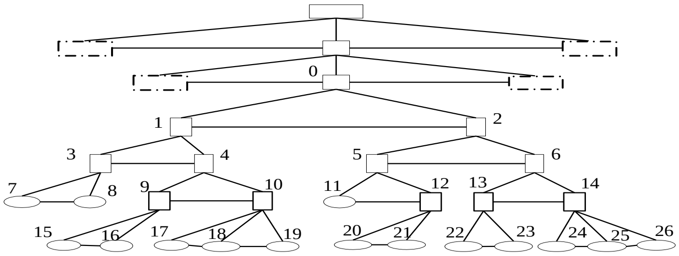


Figure 7. The HCRF model for the page in Figure 4.

variable Y_i. We currently model the interactions of sibling variables via a linear-chain, although more complex structure such as two-dimensional grid can also be used.

As a conditional model, HCRF can efficiently incorporate any useful features for webpage structure labeling. By incorporating hierarchical interactions, HCRF could incorporate long distance dependencies and achieve promising results [14].

4.3 Webpage Text Segmentation and Labeling

The existing work on text processing cannot be directly applied to web text understanding. This is because the text content on webpages is often not as regular as those in natural language documents and many of them are less grammatical text fragments. One possible method of using NLP techniques for web text understanding is to first manually or automatically identify logically coherent data blocks, and then concatenate the text fragments within each block into one string via some pre-defined ordering method. The concatenated strings are finally put into a text processing method, such as CRYSTAL [10] or Semi-CRF [9], to identify target information. [4][10] are two attempts in this direction.

It is natural to leverage the webpage structure labeling results to first concatenate the text fragments within the blocks generated by VIPS, and then use Semi-CRF to process the concatenated strings with the help of structure labeling results. However it would be more effective if we could jointly optimize the structure labeling task and the text segmentation and labeling task together.

4.4 Integrated Webpage Understanding

Now we have introduced the three subtasks of webpage understanding: webpage segmentation, webpage structure labeling, and webpage text segmentation and labeling. We argue that we need a unified model to jointly optimize these webpage understanding tasks. This is because with more semantic understanding of the text tokens we could perform better structure labeling, and with better structure labeling we can perform better page segmentation, and vice versa.

We don’t have a unified model to integrate all the three subtasks yet, but we have done some initial work to jointly optimize two subtasks. Our recent work [15] shows that the joint optimization of webpage segmentation and structure labeling tasks improves the performance of both tasks, and we will introduce another recent work [12] on integrated webpage structure labeling and text segmentation and labeling below.

Given the data representation of the page structure and text strings, we can define the joint optimization problem formally as follows.

Definition 4.1 (Joint Optimization of Structure Understanding and Text Segmentation and Labeling): *Given a vision tree X, the goal of joint optimization of structure understanding and text segmentation and labeling is to find the optimal assignment of the node labels and text segmentations (**H, S**)*:*

$$(\mathbf{H},\mathbf{S})^{*} = \arg\max_{(\mathbf{S},\mathbf{H})} p(\mathbf{H},\mathbf{S} \mid \mathbf{X}) \cdot$$

Here, all the segmentation and labels of the leaf nodes on the vision tree are denoted as $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2 \ldots \mathbf{s}_i \ldots \mathbf{s}_{|\mathbf{S}|}\}$, and $\mathbf{H} = \{h_1, h_2 \ldots h_i \ldots h_{|\mathbf{X}|}\}$ represents the labels of the nodes on the vision-tree **X.**

This problem is too hard because the searching space is the Cartesian product of both label spaces. Fortunately, the negative logarithm of the posterior will be a convex function, if we use the exponential function as the potential function. Then we can use the coordinate-wise optimization to optimize **H** and **S** iteratively. In this manner, we can solve two simpler conditional optimization problems instead, i.e., we perform the structure understanding and text understanding separately and iteratively. As we introduced before, the state-of-the-art models for structure understanding and text understating are HCRF and Semi-CRF respectively. However, we need to make them interact with each other. Therefore, we extend them by introducing additional parameters. We extend the HCRF model by introducing text segment feature functions with the segmentation of the text strings as their input. The Semi-CRF model is extended by introducing both the label of the corresponding node on the vision-tree and the segmentation results over all the nodes on the vision-tree in the last iteration.

We evaluated the performance of the joint optimization using a local entity extraction task. The extraction results show that the accuracy of all the attributes of the joint optimization model are significantly better than optimizing webpage structure labeling and text segmentation and labeling separately.

5. CONCLUSION

Internet search engines process billions of webpages on a weekly basis, and the text content of these webpages are indexed to answer user queries. We believe that some shallow understanding of the webpages will significantly improve users' browsing and searching experiences. In this paper, we formally define the webpage understanding problem, which consists of three subtasks: webpage segmentation, webpage structure labeling, and webpage text segmentation and labeling. The solutions to the problem have to be template-independent because of its web-scale nature. In this paper, we introduce partially integrated statistical models for these webpage understanding tasks. However we believe that fully integrated webpage understanding models will be an important direction for future research in Web mining for search applications.

6. REFERENCES

[1] Deng Cai, Xiaofei He, Ji-Rong Wen, and Wei-Ying Ma. Block-Level Link Analysis. In Proceedings of SIGIR, 2004.

[2] Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Ying Ma. VIPS: a Vision-based Page Segmentation Algorithm. Microsoft Technical Report, MSR-TR-2003-79, 2003.

[3] Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Ying Ma. Block-based Web Search. In Proceedings of SIGIR, 2004.

[4] D. DiPasquo. Using HTML Formatting to Aid in Natural Language Processing on the World Wide Web. Senior Honors Thesis, Carnegie Mellon University, 1998.

[5] M. Kovacevic, M. Diligenti, M. Gori and V. Milutinovic. Recognition of Common Areas in a Webpage Using Visual Information: a possible application in a page classification. ICDM, 2002.

[6] Zaiqing Nie, Ji-Rong Wen and Wei-Ying Ma. Object-Level Vertical Search. Proc. of CIDR, 2007.

[7] Zaiqing Nie, Yunxiao Ma, Shuming Shi, Ji-Rong Wen, Wei-Ying Ma. Web Object Retrieval. In Proc. of WWW, 2007.

[8] Zaiqing Nie, Yuanzhi Zhang, Ji-Rong Wen and Wei-Ying Ma. Object-Level Ranking: Bringing Order to Web Objects. In Proc. of WWW, 2005.

[9] S. Sarawagi and W. W. Cohen. Semi-Markov. Conditional Random Fields for Information Extraction. Proc. of NIPS, 2004.

[10] S. Soderland. Learning to Extract Text-based Information from the World Wide Web. Proc. of SIGKDD, 1997.

[11] Ruihua Song, Haifeng Liu, Ji-Rong Wen, and Wei-Ying Ma. Learning Block Importance Models for Webpages. In *Proc. of WWW*, 2004.

[12] Chunyu Yang, Yong Cao, Zaiqing Nie, Jie Zhou, Ji-Rong Wen. Closing the Loop in Webpage Understanding. Proc. of CIKM, 2008.

[13] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang and Hsiao-Wuen Hon. Webpage Understanding: An Integrated Approach. Proc. of SIGKDD, 2007.

[14] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang and Wei-Ying Ma. Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. Proc. of SIGKDD, 2006.

[15] Jun Zhu, Zaiqing Nie, Bo Zhang and Ji-Rong Wen. Dynamic Hierarchical Markov Random Fields for Integrated Web Data Extraction. Journal of Machine Learning Research. 9(Jul): 1583--1614, 2008.