Closing the Loop in Webpage Understanding

Chunyu Yang, *Student Member, IEEE,* Yong Cao, Zaiqing Nie,
Jie Zhou, *Senior Member, IEEE,* and Ji-Rong Wen

Abstract—The two most important tasks in information extraction from the Web are webpage structure understanding and natural language sentences processing. However, little work has been done towards an integrated statistical model for understanding webpage structures and processing natural language sentences within the HTML elements. Our recent work on webpage understanding introduces a joint model of Hierarchical Conditional Random Fields (i.e. HCRF) and extended Semi-Markov Conditional Random Fields (i.e. Semi-CRF) to leverage the page structure understanding results in free text segmentation and labeling. In this top-down integration model, the decision of the HCRF model could guide the decision-making of the Semi-CRF model. However, the drawback of the top-down integration strategy is also apparent, i.e., the decision of the Semi-CRF model could not be used by the HCRF model to guide its decision-making. This paper proposed a novel framework called WebNLP, which enables bidirectional integration of page structure understanding and text understanding in an iterative manner. We have applied the proposed framework to local business entity extraction and Chinese person and organization name extraction. Experiments show that the WebNLP framework achieved significantly better performance than existing methods.

Index Terms—Natural Language Processing, Webpage Understanding, Conditional Random Fields.

1 INTRODUCTION

The World Wide Web contains huge amounts of data. However, we cannot benefit very much from the large amount of raw webpages unless the information within them is extracted accurately and organized well. Therefore, information extraction (IE) [1], [2], [3] plays an important role in web knowledge discovery and management. Among various information extraction tasks, extracting structured Web information about real-world entities (such as people, organizations, locations, publications, products) has received much attention of late [4], [5], [6], [7], [8]. However, little work has been done towards an integrated statistical model for understanding webpage structures and processing natural language sentences within the HTML elements of the webpage. Our recent work on Web object extraction has introduced a template-independent approach to understand the visual layout structure of a webpage and to effectively label the HTML elements with attribute names of an entity [9], [10].

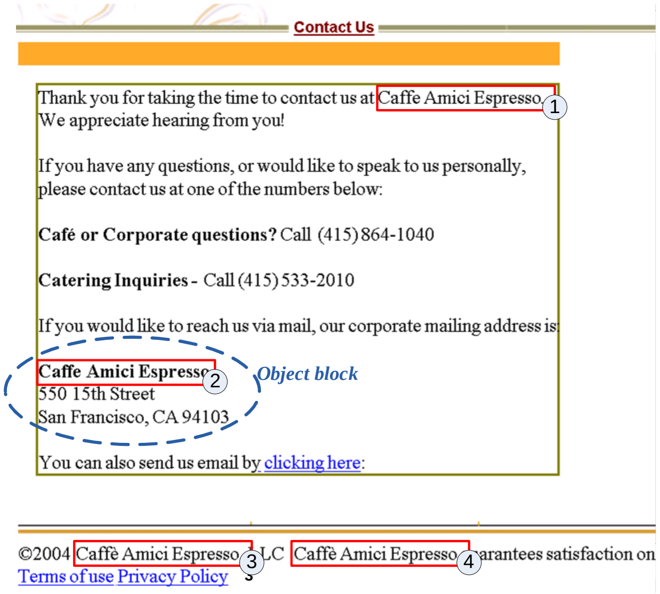
Our latest work on webpage understanding [11] introduces a joint model of the Hierarchical Conditional Random Fields (i.e. HCRF) model and the extended Semi-Markov Conditional Random Fields (i.e. Semi-CRF) model to leverage the page structure understanding results in free text segmentation and labeling. The HCRF model can reflect the structure and the Semi-CRF model can make use of the gazetteers. In this top-down integration model, the decision of the HCRF model could guide the decision of the Semi-CRF model. However, the drawback of the top-down strategy is that the decision of the Semi-CRF model could not be used by the HCRF model to refine its decision-making. In this paper, we introduce a novel framework called WebNLP that enables bidirectional integration of page structure understanding and text understanding in an iterative manner. In this manner, the results of page structure understanding and text understanding can be used to guide the decision-making of each other, and the performance of the two understanding procedures are boosted iteratively.

1.1 Motivating Example

We have been working on local entity extraction to increase the data coverage of the Windows Live Local search service by automatically extracting structured information about local businesses from the crawled webpages. In Fig. 1, we show an example webpage containing local entity information. As we can see, the address information 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 in bold font; the second line contains the street information; the third line contains the city, state and zip code. Such a structured block containing multiple attribute values of an object is called an object block. We can use the HCRF algorithm [9] together with the Semi-CRF algorithm

- *The work was done when Chunyu Yang was visiting Microsoft Research Asia.*
- *Chunyu Yang and Jie Zhou are with the State Key Laboratory on Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Automation, Tsinghua University, Beijing 100084, China.*
  E-mail: yangchunyu@mails.thu.edu.cn; jzhou@tsinghua.edu.cn
- *Yong Cao, Zaiqing Nie and Ji-Rong Wen are with Microsoft Research Asia, Beijing 100080, China.*
  E-mail: yongc@microsoft.com; znie@microsoft.com, jrwen@microsoft.com

[12] to detect the object block first and then label the attributes within the block [11].



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 Espresso
550 15th Street
San Francisco, CA 94103

You can also send us email by clicking here:

©2004 Caffè Amici Espresso, LLC Caffè Amici Espresso guarantees satisfaction on
Terms of use Privacy Policy

Fig. 1. An Example webpage Containing Local Objects.

Though such a method works well on extracting simple attributes like address information, it performs rather badly on the business name attribute, which is the most important attribute for a local search service. We found that the attributes in an object block are always short isolated strings. The features we can extract to identify the label of theses isolated strings are rather limited. 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). In other words, the evidence supporting the decision to label an isolated string as the business name is usually not strong enough. For example, there is rather little evidence for "Caffe Amici Espresso" marked with "2" to be labeled as the business name in the object block shown in Fig. 1. Fortunately, in the whole webpage, the business name is mentioned multiple times, 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". These mentions could also provide some information for the business name extraction.

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: within object blocks or in natural language sentences. That is because there is more evidence supporting "Caffe Amici Espresso" to be labeled as business name from the multiple mentions, compared to the evidence from any single instance. There are many ways to share the evidence among these mentions.

A naive way of leveraging the evidence of multiple mentions is to share the features of all the mentions without changing the statistical model [13], [14], [15]. Combining multiple occurrences at feature level will definitely improve the accuracy of each mention. However, in the classical CRF model, many feature functions are constructed to reflect the sequential dependencies of the labels on adjacent segments. Therefore, it is impossible to share such kinds of features among the multiple occurrences directly because of the local dependencies.

We need a well-defined joint statistical model that can integrate structure information and natural language information together, so that the labeling results of the HTML elements can give a prior for the natural language understanding, while the decision of the natural language processing can also give semantic suggestions to improve the HTML element labeling.

Our latest work on webpage understanding [11] makes the first attempt toward such an integrated solution. It first uses the HCRF model to label the HTML elements and then uses the Semi-CRF model to segment the text fragment within the HTML elements considering their labels assigned by the HCRF model. In this top-down integration model, the decision of the HCRF model could guide the decision-making of the Semi-CRF model. The advantage of this model is that it reduces the possible search space of the Semi-CRF model, thus making the decision more efficiently. For example, if an HTML element is labeled by the HCRF model as CITY_STATE_ZIP, the Semi-CRF model will only need to search the label space {CITY, STATE, ZIP, NOTE}, where NOTE means anything other than CITY, STATE and ZIP.

However, the drawback of this top-down model is also apparent: the HCRF model could not use the decision of the Semi-CRF model in its decision-making. Without a mechanism to leverage the text segmentation and labeling results of Semi-CRF, not only could the page structure understanding model not be improved further using these semantic labeling results, but also the text features with sequential label dependencies could not be shared among the multiple mentions in a webpage.

It is natural to close the loop in webpage understanding by introducing a bidirectional integration model, where the bottom-up model using text understanding to guide structure understanding is integrated with the top-down model mentioned above. For example, the instances of "Caffe Amici Espresso" in the natural language sentences "Thank you for taking the time to contact us at Caffe Amici Espresso" and "Caffe Amici Espresso guarantees satisfaction on all products we sell" are easy to be recognized as a business name using statistical language features. Using a bidirectional integration model with the capability to share features among all mentions of the same entity in the webpage, the other text fragment "Caffe Amici Espresso" in the same webpage will be more likely recognized as a business name, and the block containing "Caffe Amici Espresso" and some address-like text will be more confidently marked as an object block with business name and address attributes.

However, the natural language understanding component in the loop needs to be accurate enough to provide positive feedback to the structure understanding component. In [11], the Semi-CRF model is designed to handle simple text fragment segmentation, such as the segmentation between city, state and zip code. Therefore, the model only contains some regular expression features. For these regular expression features, the model can be trained to achieve nearly optimal parameters with only hundreds of labeled webpages. However, these features are not comprehensive enough to segment and label the natural language sentences in the webpage for tasks like business name extraction. We need to introduce more statistical natural language features to make the text understanding component accurate enough. The number of sentences required to train a satisfactory natural language understanding model with tens of thousands of statistical natural language features is much larger. Therefore, we need to introduce an auxiliary corpus to learn the parameters of these features.

In this paper, we propose a novel framework called WebNLP which enables bidirectional integration of page structure understanding and shallow natural language processing in an iterative manner. The WebNLP framework closes the loop of information flow in webpage understanding. Specifically, we extend both the HCRF model and the Semi-CRF model to enable the iterative labeling process, so that the labeling decision made by HCRF on page structure understanding and the decision made by semi-CRF on free text understanding could be treated as features in both models iteratively. In the WebNLP framework, the weights of the natural language features could be trained on existing large NLP corpus to guarantee accurate text segmentation and labeling.

Although the WebNLP framework is motivated by multiple mentions of object attributes (named entities) in a webpage, it will also improve entity extraction from webpages without multiple mentions because of the joint optimization nature of the framework.

The main contributions of this work are as follows:

1) We introduce a novel framework for webpage understanding called WebNLP to boost the performance of page structure understanding and shallow natural language processing iteratively, compared to Zhu's top-down strategy.
2) We introduce the multiple occurrence features to the WebNLP framework. It improved both the precision and recall of the named entity extraction and structured web object extraction on a webpage.
3) Shallow natural language processing features are applied to the WebNLP framework, which allows training of the natural language features on existing large corpus different from the limited labeled webpages.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the related work on webpage understanding. Section 3 gives a formal definition of the problem to be solved. Section 4 describes the necessary background of the CRF model, the HCRF model, and the Semi-CRF model. We introduce our WebNLP framework in detail in Section 5. The experimental results and analysis are shown in Section 6. Section 7 concludes this paper.

2 RELATED WORK

Webpage understanding plays an important role in information retrieval from the Web [16], [10]. There are two main branches of work for webpage understanding: template-dependent approaches and template-independent approaches.

Template-dependent approaches (i.e., wrapper-based approaches) can generate wrappers either with supervision [6], [8] or without supervision [17], [18], [4], [7], [19]. The supervised approaches take in some manually labeled webpages and learn some extraction rules (i.e. wrappers) based on the labeling results. Unsupervised approaches do not need labeled training samples. They first automatically discover clusters of the webpages, and then produce wrappers from the clustered webpages. No matter how the wrappers are generated, they can only work on the webpages generated by the same template. Therefore, they are not suitable for general-purpose webpage understanding.

In contrast, template-independent approaches can process various pages from different templates. However, most of the methods in literature can only handle some special kinds of pages or specific tasks such as object block (i.e. data record) detection. For example, [20], [21], [22] can only segment list pages and [23] can only detect the main block in the page. [20] segments data on list pages using the information contained in their detail pages. The need of detail pages is a limitation because automatically identifying links that point to detail pages is nontrivial and there are also many pages that do not have detail pages behind them. [21], [22] proposed an algorithm to extract structured data from list pages. The method consists of two steps. It first identifies individual records based on visual information and a tree matching method. Then a partial tree alignment technique is used to align and extract data items from the identified records. [23] defines the block importance estimation as a learning problem. First, they use the Vision-based Page Segmentation (i.e. VIPS) [16] algorithm to partition a webpage into semantic blocks with a hierarchy structure. Then spatial features (such as position, size) and content features (such as the number of images and links) are extracted to construct a feature vector for each block. Based on these features, learning algorithms, such as SVM and neural network, are applied to train various block importance models.

The task of the template-independent webpage understanding is defined in this paper as the task of page structure understanding and text content segmentation and labeling. The state-of-the-art webpage structure understanding algorithm is the Hierarchical Conditional Random Fields (HCRF) algorithm proposed in [9]. The idea of the HCRF algorithm is based on the vision-based webpage segmentation (VIPS) approach [16], [23]. HCRF organizes the segmentation of the page hierarchically to form a tree structure, and conducts inference on the vision tree to tag each vision node (vision block) with a label. The HCRF model has been proven effective for product information extraction. Since the attribute values of product objects (such as the product price, image, and description) are usually the entire text content of an HTML element, text segmentation of the content within HTML elements is done as a post processing step. Since text segmentation is not included, the HCRF model is less effective for the applications where multiple attributes of an object lie in the text in one vision node.

The requirement of text understanding in information retrieval is simpler than classical natural language understanding. Deep parsing of the sentences is unnecessary in most of the cases. Shallow parsing that can extract some important named entities is usually enough. The most popular technique used for named entity recognition is Conditional Random Fields (CRF) [24], which is language independent. For the Chinese language, the algorithm proposed in [25] works well using CRF. The named entity recognition task was modeled as a character labeling problem in [25], which is quite suitable for the characteristics of Chinese. For the English language, Semi-CRF [12] proves to be more effective than the linear chain CRF model in named entity recognition. That is because Semi-CRF introduced the non-Markov property between tokens inside entity name segments.

The combination of structure understanding and text understanding is natural [26], [27], [11]. All this work holds the belief that the structure understanding can help the text understanding. For example, [11] described a joint model that was able to segment and label the text within the vision node. It integrates the HCRF model and the Semi-CRF model. It also extends the Semi-CRF model to take the vision node label assignment as an input of the feature functions. The label of the vision node is actually a switch. It eliminates unnecessary searching paths in the optimization procedure in the Semi-CRF model. This joint model is in fact only a top-down integration, where only the label of the vision node can guide the segmentation and labeling of its inner text. The labeling of the text strings cannot be used to refine the labeling of the vision nodes.

Observing the drawback of existing models, we propose our WebNLP framework. The differences between the model in [11] and the WebNLP framework are obvious. First, the WebNLP framework is a bidirectional integration strategy where the page structure understanding and the text understanding are reinforced by each other in an iterative way. It closes the loop in the webpage understanding. Second, we introduce multiple-mention features in this new framework. Our model treats the segmentation and labeling decision at all mentions of one same entity as its observation. Such treatment greatly expands the valid features of the entity to make more accurate decisions. Third, we introduce an auxiliary corpus to train the weights of the statistical language features of the extended Semi-CRF model. It makes our model perform much better than the extended Semi-CRF model in [11] with only regular expression matching features and sequential structure features. A similar work is the bootstrapping relation extraction method proposed by Etzioni et al. [30], [31]. They focused more on the relationship between named entities, whereas we mainly focus on the attributes of an object.

3 PROBLEM DEFINITION

This paper aims at introducing a joint framework that can segment and label both the structure layout and text in the webpage. In this section, we first introduce the data representation of the structure layout of the webpage and the text content within the webpage. Then we formally define the webpage understanding problem.

3.1 Data Representation

We use the VIPS approach to segment a webpage into visually coherent blocks [16]. VIPS makes use of page layout features, such as client region, font, color, and size to construct a vision tree representation of the webpage. Different from the HTML DOM tree, each node in the vision tree represents a region on the webpage. The region of the parent node is the aggregation of those of all its child nodes. The root node represents the whole webpage. All the leaf nodes form the most detailed flat segmentation of the webpage. Only leaf nodes have inner text content. The text content inside leaf nodes may contain information like business name. The text content could be structured text like address lines or grammatical paragraphs, which contain the attribute values of an entity. Details about the usage of VIPS to segment the structural webpages can be referred to [16].

In this study, we use the vision tree as the data representation for the structure understanding. We use $\mathbf{X} = \{x_1, x_2, \cdots, x_i, \cdots, x_{|\mathbf{X}|}\}$ to denote the entire vision tree of a webpage. x_i is the observation on the i-th vision node, which can be either inner node or leaf node. The observation contains both the visual information, e.g., the position of the node, and the semantic information, e.g., the text string within the node. Each vision node is associated with a label h to represent the role of the node on the whole tree, e.g., whether the node contains all or some of the attributes of an object. So $\mathbf{H} = \{h_1, h_2, \cdots, h_i, \cdots, h_{|\mathbf{X}|}\}$ represents the label of the vision tree $\mathbf{X}$. We denote the label space of h as $\mathbf{Q}$.

The text string within the leaf node is represented by a character sequence. Understanding the text means to segment the text into non-overlapping pieces and tag each piece with a semantic label. In this paper, text understanding is equal to text segmentation and labeling. We use $\mathrm{s} = \{s_1, s_2, \cdots, s_m, \cdots, s_{|\mathrm{s}|}\}$ to represent the segmentation and tagging over the text string within a leaf node x. Each segment in s is a triple $s_m = \{\alpha_m, \beta_m, y_m\}$, in which α_m is the starting position; β_m is the end position; y_m is the segment label which is assigned to all the characters within the segment. We use $|x|$ to denote the length of the text string within the vision node x. Then segment s_m satisfies $0 \leq \alpha_m < \beta_m \leq |x|$ and $\alpha_{m+1} = \beta_m + 1$. Named entities are some special segments differentiated from other segments by their labels. We denote the label space of y as $\mathbf{Y}$. All the segmentation and tagging of the leaf nodes in the vision tree are denoted as $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_i, \cdots, \mathbf{s}_{|\mathbf{S}|}\}$.

Unless otherwise specified, these symbols defined above have the same meaning throughout this paper.

3.2 Problem Definition

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

Definition 1: (**Joint optimization of structure understanding and text understanding**): Given a vision tree $\mathbf{X}$, the goal of joint optimization of structure understanding and text understanding is to find both the optimal assignment of the node labels and text segmentations $(\mathbf{H}, \mathbf{S})^*$:

$$(\mathbf{H}, \mathbf{S})^* = \arg\max_{(\mathbf{S},\mathbf{H})} p\left(\mathbf{H}, \mathbf{S}|\mathbf{X}\right). \tag{1}$$

This definition is the ultimate goal of webpage understanding, i.e., the page structure and the text content should be understood together. However, such a definition of the problem is too hard because the search space is the Cartesian product of $\mathbf{Q}$ and $\mathbf{Y}$. Fortunately, the negative logarithm of the posterior in (1) will be a convex function, if we use the exponential function as the potential function [24]. Then we can use the coordinate-wise optimization to optimize $\mathbf{H}$ and $\mathbf{S}$ iteratively. In this manner, we can solve two simpler conditional optimization problems instead of solving the joint optimization problem in (1) directly, i.e., we do structure understanding and text understanding separately and iteratively. The formal definitions of the two conditional optimization problems are as follows.

Definition 2: (**Structure understanding**): Given a vision tree $\mathbf{X}$ and the text segmentation and labeling results $\mathbf{S}$ on the leaf nodes of the tree, structure understanding is to find the optimal label assignment of all the nodes in the vision tree $\mathbf{H}^*$:

$$\mathbf{H}^* = \arg\max_{\mathbf{H}} p\left(\mathbf{H}|\mathbf{X}, \mathbf{S}\right). \tag{2}$$

The objective of the structure understanding is to identify the labels of all the vision nodes in the vision tree. Both the raw observations of the nodes in the vision tree and the understanding results about the text within each leaf node are used to find the optimal label assignment of all the nodes on the tree.

Definition 3: (**Text understanding**): Given a vision tree $\mathbf{X}$ and the label assignment $\mathbf{H}$ on all vision nodes, text understanding is to find the optimal segmentation and labeling $\mathbf{S}^*$ on the leaf nodes:

$$\mathbf{S}^* = \arg\max_{\mathbf{S}} p\left(\mathbf{S}|\mathbf{X}, \mathbf{H}\right). \tag{3}$$

The task of the text understanding problem in entity extraction is to identify all the named entities in the webpage. The labeling results of the vision nodes will constrain the text understanding component to search only part of the label space of the named entities. The labels of the named entities within a vision node are forced to be compatible with the label of the node assigned by the structure understanding.

The problem described in Definition 1 can be solved by solving the two sub problems in Definition 2 and Definition 3 iteratively, starting from any reasonable initial solution. In Definition 2, the $\mathbf{S}$ in the condition is the optimum of the text understanding in the last iteration, and in Definition 3, the $\mathbf{H}$ in the condition is the optima of the structure understanding in the last iteration. The iteration can begin with either the structure understanding or text understanding. In this work, we will begin with the text understanding. The features related to the label given by structure understanding are set as zero in the first run of text understanding. The loop stops when the optima in two adjacent iterations are close enough. The details about the solution to above problems will be introduced in Section 5.

4 BACKGROUND

To make this paper as self-contained as possible, we introduce some necessary background about related models. We will only introduce the definition of the potential function in each model in this section. Details about other parts of these algorithms, e.g. parameters inference and optimal label assignment, should be referenced to the original papers.

4.1 CRF

The linear chain CRF tags elements in a sequence $\mathbf{x}$ with transition features [24]. Let the label of the elements in the sequence be $\mathbf{y}$, and then the conditional probability of $\mathbf{y}$ is defined as follows:

$$p\left(\mathbf{y}|\mathbf{x}\right) = \frac{1}{Z(\mathbf{x})}\phi\left(\mathbf{y}|\mathbf{x}\right), \tag{4}$$

where $Z\left(\mathbf{x}\right) = \sum_{\mathbf{y}} \phi\left(\mathbf{y}, \mathbf{x}\right)$ is the normalization factor to make it a distribution function. The potential function $\phi\left(\mathbf{y}, \mathbf{x}\right)$ is defined as:

$$\phi\left(\mathbf{y}, \mathbf{x}\right) = \exp\left[\sum_{\nu,k} \mu_k g_k(\mathbf{y}|_v, \mathbf{x}) + \sum_{e,k} \lambda_k f_k(\mathbf{y}|_e, \mathbf{x})\right]. \tag{5}$$

v is vertex corresponding to a single element and e is edge corresponding to a pair of neighboring elements. $\mathbf{y}|_v$ are the components of $\mathbf{y}$ associated with the vertex v and $\mathbf{y}|_e$ are the components of $\mathbf{y}$ associated with the edge e. $g_k(\cdot)$ is the k-th state function and $f_k(\cdot)$ is the k-th transition feature function; μ_k and λ_k are the corresponding weights of the feature functions $g_k(\cdot)$ and $f_k(\cdot)$ respectively.

4.2 HCRF

HCRF [9] is an extension of the CRF model on graphs. The main idea of the HCRF model is to convert the vision tree representation of a webpage to a graph by introducing edges between adjacent siblings. Then the junction tree algorithm [28] is used to infer the label of the vertices on the graph. These vertices correspond to the vision nodes in the vision tree. Similar to the CRF model introduced in Section 4.1, the conditional distribution of the labels given the observations is defined as follows:

$$p\left(\mathbf{H}|\mathbf{X}\right) = \frac{1}{Z(\mathbf{X})}\phi\left(\mathbf{H},\mathbf{X}\right), \tag{6}$$

where $Z\left(\mathbf{X}\right) = \sum_{\mathbf{H}} \phi\left(\mathbf{H},\mathbf{X}\right)$ is the normalization factor to make it a distribution function, and $\phi\left(\mathbf{H},\mathbf{X}\right)$ is the potential function of the label assignment $\mathbf{H}$ on the vision tree. It has the following form:

$$\phi\left(\mathbf{H},\mathbf{X}\right) = \exp\left[\begin{array}{l}\sum\limits_{\nu,k} \mu_k g_k(\mathbf{H}|_v,\mathbf{X}) + \sum\limits_{e,k} \lambda_k f_k(\mathbf{H}|_e,\mathbf{X}) \\ + \sum\limits_{t,k} \gamma_k b_k(\mathbf{H}|_t,\mathbf{X})\end{array}\right]. \tag{7}$$

v and e still represent vertex and edge respectively. t is the triangle consisting of three vertices and three edges connecting each pair of the three vertices. Please see Fig. 2 for an intuitive example. The solid squares represent inner nodes and the hollow squares represent leaf nodes. They all correspond to v. The pairs of squares connected by a solid line correspond to e. The triangles with three squares and three solid lines connecting each pair of the squares correspond to t.

In the potential function $\phi\left(\mathbf{H},\mathbf{X}\right)$ in (7), $\mathbf{H}|_v$ are the components of $\mathbf{H}$ associated with the vertex v; $\mathbf{H}|_e$ are the components of $\mathbf{H}$ associated with the edge e; $\mathbf{H}|_t$ are the components of $\mathbf{H}$ associated with the triangle t. $g_k(\cdot)$, $f_k(\cdot)$ and $b_k(\cdot)$ are feature functions on the vertices, edges and triangles respectively; μ_k, λ_k and γ_k are the corresponding weights of the feature functions $g_k(\cdot)$, $f_k(\cdot)$ and $b_k(\cdot)$ respectively. We can see that the main differences between the HCRF and CRF models are the feature functions $b_k(\cdot)$ on the triangles.

4.3 Semi-CRF

Semi-CRF [12] is an extension of the linear chain CRF. As is defined in Section 3.1, the segmentation of a text string x, is $\mathbf{s} = \{s_1, s_2, \cdots, s_m, \cdots, s_{|\mathbf{s}|}\}$. Let $q_k(s_m, s_{m-1}, x)$ be the k-th feature function at segment m. The value of $q_k(s_m, s_{m-1}, x)$ depends on the current segment s_m, the previous segment s_{m-1} and the whole observation of the string x. Let ξ_k be the weight of $q_k(\cdot)$. The conditional probability is then defined as follows:

$$p\left(\mathbf{s}|x\right) = \frac{1}{Z(x)}\phi\left(\mathbf{s},x\right), \tag{8}$$

where $Z\left(x\right) = \sum_{\mathbf{s}} \phi\left(\mathbf{s},x\right)$ is the normalization factor to make $p\left(\mathbf{s}|x\right)$ a distribution function and the potential function $\phi\left(\mathbf{s},x\right)$ is as:

$$\phi\left(\mathbf{s},x\right) = \exp\left[\sum_m \sum_k \xi_k q_k(s_m, s_{m-1}, x)\right]. \tag{9}$$

5 WEBNLP FRAMEWORK

In this section we introduce the WebNLP framework to solve the webpage understanding problem defined in Section 3.2. We first introduce the framework intuitively and describe the individual models within the framework formally. Then we describe how we integrate the page structure understanding model and the text understanding model together in the framework. The parameter learning method and the label assignment procedure will be explained last.

5.1 Overview

The WebNLP framework consists of two components, i.e., a structure understanding component and a text understanding component. The observations of these two components are both from the webpage. The understanding results of one component can be used by the other component to make a decision. The information flows between the two components form a closed loop. The beginning of the loop is not very important. However, we will show that starting from the text understanding component is a good choice.

The structure understanding component assigns labels to the vision blocks in a webpage, considering visual layout features directly from the webpage and the segments returned by the text understanding component together. If the segments of the inner text are not available, it will work without such information. The text understanding component segments the text string within the vision block according to the statistical language features and the label of the vision block assigned by the structure understanding component. If the label of the vision block is not available, it can also work without such information. The two components run iteratively until some stop criteria is met. Such iterative optimization can boost both the performance of the structure understanding component and text understanding component. Details about the models used in structure understanding and text understanding will be introduced in the rest of this section.

5.2 The Extended Models

As we introduced previously, the state-of-the-art models for webpage structure understanding and text understanding are the HCRF model and the Semi-CRF model respectively. However, there is no way to make them interact with each other in their original forms. Therefore, we extend them by introducing additional input parameters to the feature functions. The original forms of the HCRF model and the Semi-CRF model have been introduced in Section 4. Therefore, we will only introduce the forms of the extended HCRF model and the extended Semi-CRF model in this section.

We first extend the HCRF model by introducing other kinds of feature functions. These feature functions take the segmentation of the text strings as their input. Analogizing to the feature functions defined in Section 4.2, we use $e_k(\mathbf{H}|_t, \mathbf{X}, \mathbf{S})$ to represent the feature functions having text strings segmentation input. To simplify the expression, we use the functions defined on the triangle to represent all functions defined on the vertex, edge or triangle. As the WebNLP framework is an iterative one, we further use the superscript j to indicate the decision in the j-th iteration. Then the potential function of the extended HCRF algorithm in the j-th iteration can be defined as follows:

$$\phi\left(\mathbf{H}^j, \mathbf{X}, \mathbf{S}^{j-1}\right) = \exp\left[\begin{array}{l} \sum\limits_{\nu,k} \mu_k g_k\left(\mathbf{H}^j|_\nu, \mathbf{X}\right) + \sum\limits_{e,k} \lambda_k f_k\left(\mathbf{H}^j|_e, \mathbf{X}\right) \\ + \sum\limits_{t,k} \gamma_k h_k\left(\mathbf{H}^j|_t, \mathbf{X}\right) \\ + \sum\limits_{t,k} \chi_k e_k\left(\mathbf{H}^j|_t, \mathbf{X}, \mathbf{S}^{j-1}\right) \end{array}\right]. \quad (10)$$

The newly introduced feature function $e_k(\cdot)$ uses the decision of the text understanding component in the $(j-1)$-th iteration $\mathbf{S}_{j-1}$ as its additional input. χ_k is the weight of the feature function $e_k(\cdot)$. Other symbols keep the same meanings as in the original HCRF model described in section 4.2.

Then we can get the conditional distribution function of the extended HCRF model in the j-th iteration as follows:

$$p\left(\mathbf{H}^j|\mathbf{X}, \mathbf{S}^{j-1}\right) = \frac{1}{Z\left(\mathbf{X}, \mathbf{S}^{j-1}\right)} \phi\left(\mathbf{H}^j, \mathbf{X}, \mathbf{S}^{j-1}\right), \quad (11)$$

where $Z\left(\mathbf{X}, \mathbf{S}^{j-1}\right) = \sum_{\mathbf{H}^j} \phi\left(\mathbf{H}^j, \mathbf{X}, \mathbf{S}^{j-1}\right)$ is the normalization factor to make $p\left(\mathbf{H}^j|\mathbf{X}, \mathbf{S}^{j-1}\right)$ a distribution function.

The Semi-CRF model is extended by introducing both the label of the vision node and the segmentation results of the text strings within all the vision nodes in the last iteration. Therefore, the potential function of the extended Semi-CRF model is:

$$\phi\left(\mathbf{X}, \mathbf{H}^j, \mathbf{S}^{j-1}, \mathbf{S}^j\right) = \exp\left[\begin{array}{l} \sum\limits_i \sum\limits_m \sum\limits_k \xi_k q_k\left(s^j_{i,m-1}, s^j_{i,m}, x_i\right) \\ + \sum\limits_i \sum\limits_m \sum\limits_k \theta_k r_k\left(s^j_{i,m-1}, s^j_{i,m}, h^j_i, x_i\right) \\ + \sum\limits_i \sum\limits_m \sum\limits_k \eta_k u_k\left(s^j_{i,m-1}, s^j_{i,m}, \mathbf{X}, \mathbf{S}^{j-1}\right) \end{array}\right]. \quad (12)$$

In (12), $q_k(\cdot)$ is the statistical language feature function that has been mentioned in Section 4.3 in the original Semi-CRF model. $r_k(\cdot)$ is the feature function considering the label of the vision node containing the text string x_i. $u_k(\cdot)$ is the global feature function, which can include the observation on the whole webpage and the text segmentation results in the last iteration. ξ_k, θ_k and η_k are the corresponding feature weights of feature functions $q_k(\cdot)$, $r_k(\cdot)$ and $u_k(\cdot)$ respectively.

The conditional distribution function of the extended Semi-CRF model can be expressed as follows:

$$p\left(\mathbf{S}^j|\mathbf{X}, \mathbf{H}^j, \mathbf{S}^{j-1}\right) = \frac{\phi\left(\mathbf{S}^j, \mathbf{X}, \mathbf{H}^j, \mathbf{S}^{j-1}\right)}{Z\left(\mathbf{X}, \mathbf{H}^j, \mathbf{S}^{j-1}\right)}, \quad (13)$$

where $Z\left(\mathbf{X}, \mathbf{H}^j, \mathbf{S}^{j-1}\right) = \sum_{\mathbf{S}^j} \phi\left(\mathbf{S}^j, \mathbf{X}, \mathbf{H}^j, \mathbf{S}^{j-1}\right)$ is the normalization factor to make it a distribution function.

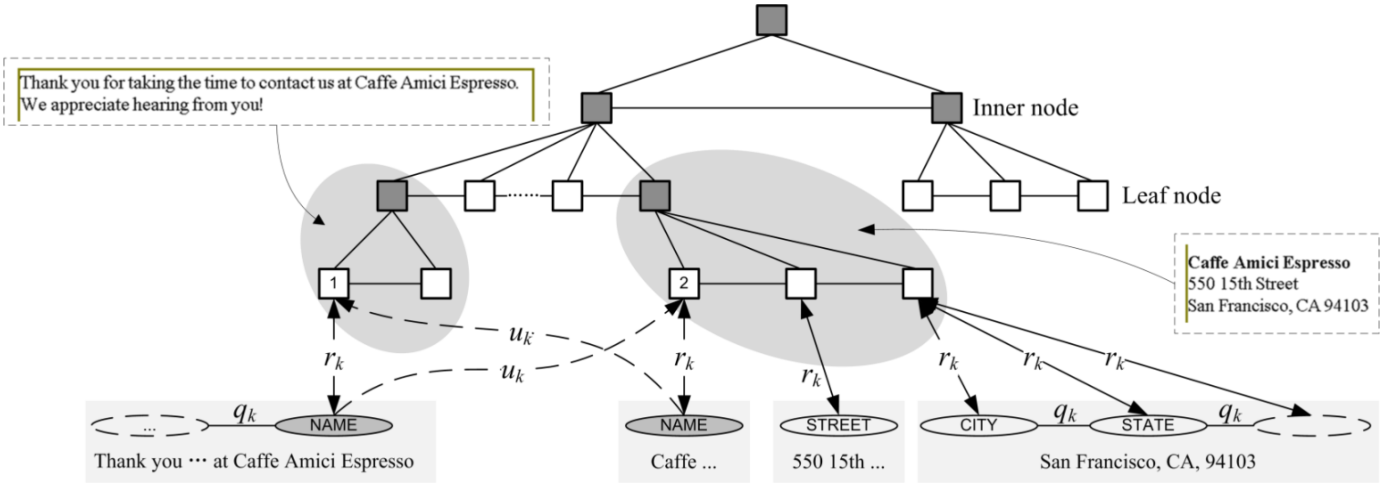
5.3 Model Integration

In Section 5.2, we introduced the extended HCRF and Semi-CRF models. We will analyze how these two models are integrated together in this section. Fig. 2 gives an illustrative example of the connection between the extended HCRF model and the extended Semi-CRF model in a webpage. It is generated based on the example webpage shown in Fig. 1. There are two types of connections in the integrated model. One is the connection between the vision node label and the segmentation of the inner text. The other is the connection between multiple mentions of a same named entity.

5.3.1 *Vision tree node and its inner text*

The natural connection between the extended HCRF model and the extended Semi-CRF model is via the vision tree node and its inner text. The feature functions that connect the two models are $r_k(\cdot)$ in the extended Semi-CRF model and $e_k(\cdot)$ in the extended HCRF model.

Feature function $r_k(\cdot)$ in the extended Semi-CRF model takes the labeling results of the leaf node given by the extended HCRF model as its input. For example, if a leaf node x is labeled as ADDRESS (which indicates that x contains and only contains address information), then $r_k(\cdot)$ will return a positive value only when the tagging of the text only contain labels such as CITY, STREET and ZIP. Therefore, evidence from the vision tree node is then delivered downward to the extended Semi-CRF model via function $r_k(\cdot)$.



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

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

Fig. 2. Connection of the two modules. The rectangle represents the vision tree node; the ellipse represents named entity; the two gray ellipses represent two mentions of one named entity with “NAME” label.

Feature function $e_k(\cdot)$ in the extended HCRF model uses the segmentation and labeling results of the extended Semi-CRF model as its input. For example, if the text within a node is segmented to CITY, STATE and ZIP, then $e_k(\cdot)$ will return a positive value only when the potential label of the node is ADDRESS. Thus, the evidence from the underlying text is delivered upward to HCRF via function $e_k(\cdot)$.

Such connections are illustrated in Fig. 2 as solid bidirectional arrow lines marked with r_k. Actually, each bidirectional arrow line represents both $r_k(\cdot)$ (downward) and $e_k(\cdot)$ (upward) between the two models.

5.3.2 Multiple mentions

In many cases, a named entity has more than one mention within a webpage. Therefore, it is natural to collect evidence from all the different mentions of one same named entity to make a decision on all these occurrences together. The evidence from all the other mentions of a named entity are delivered to the vision tree node where one of the mentions of the named entity lies via feature function $u_k(\cdot)$, when the extended Semi-CRF model is working.

$u_k(\cdot)$ can introduce the segmentation and labeling evidence from other occurrences of the text fragment all over the current webpage. By referencing the decision S_{j-1} all over the text strings in last iteration, $u_k(\cdot)$ can determine whether the same text fragment has been labeled as an ORGANIZATION elsewhere, or whether it has been given a label other than STREET. By this means, the evidence for a same named entity is shared among all its occurrences within the webpage.

The single arrow dashed lines in Fig. 2 illustrate such connections. The vision node 1 and vision node 2 both contain a mention to one named entity “Caffe Amici Espresso” with label “NAME”.

5.4 Learning the Parameters

Given the labeling results, the extended HCRF model and the extended Semi-CRF model are independent. Therefore, the parameters of the two models can be learnt separately. The two models will not interact during the parameters inference stage. We will introduce the parameter inference methods one by one.

5.4.1 The Extended HCRF Model

The parameter learning for the extended HCRF model is relatively straightforward. In this model, the feature function set is relatively small. Therefore, it does not need a large number of labeled samples to train the model. Usually, hundreds of samples are enough. Though the potential of the extended HCRF model has an additional parameter $\mathbf{S}_{j-1}$ compared with the original HCRF model, it still can be trained using exactly the same method as the original HCRF model by simply treating $\mathbf{S}_{j-1}$ as a part of the observation of the vision tree. Actually, for the labeled training webpages, $\mathbf{S}_{j-1}$ is provided as the labeling result on all the text strings in the page. Then the parameter learning method for the original HCRF model [11] is taken on the extended observation.

5.4.2 The Extended Semi-CRF Model

The parameter learning for the extended Semi-CRF model is not as straightforward as the extended HCRF model described above. The statistical language feature functions $q_k(\cdot)$ in the extended Semi-CRF model in (12) are mainly the statistics of the language elements (unigrams, bigrams etc.), whose number is usually several million. In order to get reasonable weights for these features, the model should be trained on a language corpus large enough to avoid bias. Usually, tens of thousands of sentences are required. Unfortunately, the labeled webpages for training the extended HCRF model are usually too limited compared to such requirements, e.g. there are usually only a few hundred manually labeled webpages. The number of sentences that can be used to train $q_k(\cdot)$ are usually only a few thousand. It is impossible to learn satisfactory weights for $q_k(\cdot)$ on such a limited training set. Meanwhile, the number of features other than q_k in the extended Semi-CRF model is relatively small. Their weights can be trained rather accurately with only a few hundred webpages. The unbalanced training sample requirement can be solved in two ways. One solution is to label thousands of webpages, and then we can get tens of thousands of sentences for all the feature functions. It costs too much in labor resources to label so many webpages. The other solution is to introduce an auxiliary language corpus and train the weights of $q_k(\cdot)$ on it while training the weights of other features on the hundreds of webpages. Because there are many existing labeled corpora that are large enough, we do not need to perform costly labeling on thousands of webpages. The second solution uses existing resources elegantly, so we choose this solution to learn the parameters of the extended Semi-CRF model.

We first train the weights ξ_k of $q_k(\cdot)$ on the auxiliary corpus using the original Semi-CRF model. Details about the estimation of ξ_k can be found in [12] and are omitted in this paper. These weights ξ_k are then fixed in the extended Semi-CRF model. Then the weights of other feature functions are learnt from the labeled webpages. We only present the parameter estimation for these features. The logarithmic likelihood function was defined on the training data set D as follows:

$$\begin{aligned} L &= \sum_{\mathbf{X}} \log\left[p\left(\mathbf{S}^j|\mathbf{X},\mathbf{H}^j,\mathbf{S}^{j-1}\right)\right] \\ &= \sum_{\mathbf{X}} \left\{ \begin{array}{l} \log\left[\phi\left(\mathbf{S}^j,\mathbf{X},\mathbf{H}^j,\mathbf{S}^{j-1}\right)\right] \\ -\log\left[Z\left(\mathbf{X},\mathbf{H}^j,\mathbf{S}^{j-1}\right)\right] \end{array} \right\}. \end{aligned} \tag{14}$$

To simplify the expression, let $c_k\left(s^j_{i,m-1}, s^j_{i,m}, h^j_i, \mathbf{X}, \mathbf{S}^{j-1}\right)$ be the general form of the feature functions and δ_k be the general representation of the feature weights. Since ξ_k is fixed after training on the auxiliary corpus, it is excluded from the concept of δ_k. Then the gradient of the logarithmic likelihood over parameter δ_k is:

$$\begin{aligned} \frac{\partial L}{\partial \delta_k} &= \sum_{\mathbf{X}} \sum_i \left\{ \begin{array}{l} \sum\limits_m c_k\left(s^j_{i,m-1}, s^j_{i,m}, h^j_i, \mathbf{X}, \mathbf{S}^{j-1}\right) \\ -\sum\limits_m c_k\left(s^j_{i,m-1}, s^j_{i,m}, h^j_i, \mathbf{X}, \mathbf{S}^{j-1}\right) \\ \dfrac{\phi\left(\mathbf{X},\mathbf{H}^j,\mathbf{S}^{j-1},\mathbf{S}^j\right)}{Z\left(\mathbf{X},\mathbf{H}^j,\mathbf{S}^{j-1}\right)} \end{array} \right\}. \\ &= \sum_{\mathbf{X}} \sum_i \left\{ \begin{array}{l} \sum\limits_m c_k\left(s^j_{i,m-1}, s^j_{i,m}, h^j_i, \mathbf{X}, \mathbf{S}^{j-1}\right) \\ -E\left(c_k, \mathbf{x}\right) \end{array} \right\} \end{aligned} \tag{15}$$

The superscripts j and $j-1$ should be removed because the node and text are labeled on the training webpages. The second item $E(\cdot)$ is the expectation of the feature function c_k under the current model parameters. Then the L-BFGS [29] gradient search algorithm can be used to find the optima.

5.5 Finding the Optimal Assignment

After obtaining the parameters of the models, we can process fresh webpages under the WebNLP framework. The assignments of the vision nodes and the segmentations of the text should be optimized iteratively, according to the WebNLP framework described in Section 5.1. Concretely, the following steps should be repeated until the convergence of the assignments is reached.

Step 1. The extended Semi-CRF model generates the segmentation candidates within the text leaf nodes of the vision tree using only the available natural language features.

Step 2. The extended HCRF model infers the optimal label of the vision nodes based on the text segmentation and labeling results given by the extended Semi-CRF model and other visual features.

Step 3. The extended Semi-CRF model generates the segmentation candidates within the text leaf node of the vision tree using the full feature set: the natural language features, the labeling results from the extended HCRF model and the multiple mention features of the same entities. Go to Step 2, until the segmentation and labeling results are similar enough in two adjacent iterations.

Note that we first run the extended Semi-CRF model with only partial features to understand the text strings before running the extended HCRF model. That is because the language features in the extended Semi-CRF model are powerful enough to make a reasonable decision, while the visual features in the extended HCRF model alone cannot provide accurate assignment.

6 EXPERIMENTS

We have carried out two sets of experiments to illustrate the effectiveness of the WebNLP framework. The first set of experiments is based on the Windows Live Local search service. It focuses on the local business object extraction from English webpages. The second set of experiments is to extract named entities from Chinese webpages for a social network research project. We will describe the experiments in detail in the following parts of this section.

6.1 Experiments on Local Search Service

The task of this application is to extract local entities within a webpage for Windows Live Local search. The attributes of a business entity in this experiment include the business NAME, STREET, CITY and STATE. These attributes are essential to identify a specific business in Windows Live Local search.

6.1.1 Data Set

The webpages used in this experiment are crawled according to a business name list. Because it is required that every webpage should contain all the four attributes to identify a business entity, we select 456 pages from all crawled pages satisfying this requirement. An example of the page used in this experiment has been shown in Fig. 1. The attributes in these webpages are all manually labeled. We randomly select 200 pages for training and the remaining 256 pages are left for testing.

The auxiliary training corpus for the statistical language features in the extended Semi-CRF model come from two sources. The first one is the text in the labeled webpages. There were 3,030 sentences from this source. The second one was automatically generated from Microsoft Live search. We sent queries with quoted company names, which are randomly selected from yellow pages data, to Live search engine. Then, we filtered the returned snippets to select the sentences containing the company name. In this way, we could conveniently get a large amount of accurate labeled data quickly. We picked 30,000 sentences from this source.

6.1.2 Methods and Evaluation Metrics

We compared four different algorithms in this experiment. They are BHS, NHS, MHS and WebNLP.

The first algorithm is the original HCRF and extended Semi-CRF framework. We name it the BHS (Basic HCRF and extended Semi-CRF) algorithm. It is the algorithm described in [11].

The second algorithm is similar to the BHS algorithm. The only difference from BHS is that it adds the natural language features directly into the extended Semi-CRF model. We name it the NHS (Natural language HCRF and extended Semi-CRF) algorithm. The extended Semi-CRF model in the NHS algorithm is trained using both the text nodes from the labeled webpages and the corpus data. The super labels of all sentences from the Live search are set as NAME, because we only queried the NAME. The rest of this algorithm is identical to the BHS algorithm.

The third algorithm is based on the NHS algorithm. It further adds global multiple mentions feature functions to the HCRF model. We name this algorithm MHS (Multiple mentions HCRF and extended Semi-CRF). These feature functions are all for the business NAME attribute. The summaries of the features at other mentions of the same business NAME candidate are used as feature functions for the current mention. It is some kind of feature sharing. An example of the multiple mention features is as follows:

$$u_k\left(s_{m-1}, s_m, \mathbf{X}\right) = \begin{cases} 1, & \begin{array}{l} \text{if } s_m.y = \text{NAME} \\ \text{and } s_m \text{ satisfies more than two functions} \\ \text{as NAME in other sentences in } \mathbf{X} \end{array} \\ 0, & \text{otherwise} \end{cases} . \tag{16}$$

The constraint of the minimum number of functions limitation in this example is used to avoid noise. The rest of the MHS algorithm is identical to the NHS algorithm.

The last one is the WebNLP framework. The feature functions used in the extended HCRF model are similar to BHS. Examples of the three types of feature functions on the i-th sentence x_i in the extended Semi-CRF model are as follows.

Because the functions $q_k(\cdot)$ are statistical language features, we can only give a simple illustrative example as:

$$q_k\left(s_{i,m-1}^j, s_{i,m}^j, x_i\right) = \begin{cases} 1, & \begin{array}{l} \text{if } s_{i,m-1}^j.y = \text{NOTE} \\ \text{and } s_{i,m}^j.y = \text{NAME} \\ \text{and } x_i\left[s_{i,m-1}^j.\alpha, s_{i,m-1}^j.\beta\right] = \text{“at”} \\ \text{and Capital\_Init}\left(x_i\left[s_{i,m}^j.\alpha, s_{i,m}^j.\beta\right]\right) \end{array} \\ 0, & \text{otherwise} \end{cases} . \tag{17}$$

It means that if all the words within a sub string begin with capital letters and the word exactly before it is “at”, then the sub string is possible to be marked as NAME and the word “at” is possible to be marked as NOTE.

The function $r_k(\cdot)$ is used to guide the decision of the segmentation by the label of the leaf node, i.e. the decision of the HCRF on the vision tree:

$$r_k\left(s_{i,m-1}^j, s_{i,m}^j, h_i^j, x_i\right) = \begin{cases} 1, & \begin{array}{l} \text{if } h_i^j = \text{ADDRESS} \\ \text{and } s_{i,m-1}^j.y = \text{CITY} \\ \text{and } s_{i,m}^j.y = \text{STATE} \\ \text{and } x_i\left[s_{i,m-1}^j.\alpha, s_{i,m-1}^j.\beta\right] \text{ is a valid city} \\ \text{and } x_i\left[s_{i,m}^j.\alpha, s_{i,m}^j.\beta\right] \text{ is a valid state} \end{array} \\ 0, & \text{otherwise} \end{cases} . \tag{18}$$

It means that if the node is labeled as ADDRESS block and a segment is a valid state in the gazetteer and its preceding segment is a valid city, then the segment is possible to be marked as STATE and the preceding segment is possible to be marked as CITY.

The function $u_k(\cdot)$ is used to collect evidence from the other occurrences of the current segmentation fragment:

$$u_k\left(s_{i,m-1}^j, s_{i,m}^j, \mathbf{X}, \mathbf{S}^{j-1}\right) = \begin{cases} 1, & \begin{array}{l} \text{if } \exists\ s_1 \in \mathbf{S}^{j-1} \text{ and } s_2 \in \mathbf{S}^{j-1} \\ \text{and } s_1.y = s_2.y = s_{i,m}^j.y = \text{NAME} \\ \text{and strings in } s_1,\ s_2 \text{ and } s_{i,m}^j \text{ are identical} \end{array} \\ 0, & \text{otherwise} \end{cases} . \tag{19}$$

The evaluation of the algorithms is based the performance on all attributes. For each attribute, the standard Precision, Recall and F1 measure are evaluated. We also evaluate the P/R/F1 of the local business as a whole, i.e., all the four attributes of the local business should be correctly extracted.

6.1.3 Results and Discussions

The object extraction results, as well as the attributes extraction results, of different algorithms are reported in Table 1. We can see that the P/R/F1 of all the attributes and the object of the proposed WebNLP framework is the highest among the four algorithms. An interesting result in Table 1 is that the precision of the attributes CITY and STATE of all these algorithms are 100%. We checked the value of the features related to CITY and STATE, and found out that the 100% precision is because the validation features from the gazetteer are quite strong, i.e., all erroneous extractions of the city and state are filtered by the gazetteer-based features. We also analyzed the contribution of different components of the framework. It shows that all components of the WebMLP framework contribute to its good performance.

The contribution of the statistical language features can be seen from the comparison of the NHS algorithm and the BHS algorithm. The statistical language features in the NHS algorithm help to improve the precision of the business NAME and STREET, because the two attributes may not be easy to segment precisely without some statistical language evidence. The NLP features provide accurate segmentation suggestions for the extended Semi-CRF model. Though the recall of some attributes becomes low, it can be amended by future components added to the framework. We have to admit that the comparison between NHS and BHS is a bit unfair, because NHS used an additional corpus that was not seen by BHS. However, it is a practical strategy to incorporate as many resources as possible, as long as these resources are easy to obtain and the algorithm can handle them. For the HNS algorithm, the additional corpus is easy to obtain and it can handle it without too much effort.

The contribution of the multiple mentions features is reflected by the difference between the MHS algorithm and the NHS algorithm. The multiple mentions features helped the MHS algorithm to increase both the precision and recall of the business NAME compared with the NHS algorithm. However, we can see that the improvement is limited. It proves that the simple feature sharing mechanism could not fully utilize the information.

The WebNLP framework gets the best numbers on all attributes and the object as a whole. It amends the decrease of the recall of CITY and STATE by reusing part of the Semi-CRF model in BHS. The iterative labeling procedure greatly improved the recall of the business NAME. In our experiment, we found out that two iterations were enough to make the labeling procedure converged. Therefore, the process of the WebNLP algorithm in this experiment was Semi-CRF → HCRF → Semi-CRF → HCRF → Semi-CRF.

We can also conclude from Table 1 that the object extraction benefits from the improvement of the attribute extraction, i.e., the extended Semi-CRF model helps the extended HCRF model to make a better decision on the object block extraction. Essentially, the object is described by its associated attributes. The more accurate the attribute extraction is, the more accurate the object extraction is.

6.2 Experiments on Chinese Named Entity Extraction

Now we introduce our experiments on the Chinese named entity extraction, which is motivated by our application on building a social network among persons and organizations by extracting their relationship from crawled webpages. Therefore, in this experiment, we focus on the two most important named entities, i.e., the PERSON name and ORGINIZATION name.

6.2.1 Data Set

The webpages used in the experiments are crawled automatically from news sites and organization websites. In order to better show the effectiveness of the proposed framework, we only selected some webpages containing multiple mentions of the same entity. These pages include news pages, biography and personal homepages. We randomly sampled 33 pages for training and 100 pages for testing.

Similar to the experiment on business object extraction, we also train the statistical language features of WebNLP on a large auxiliary corpus. We use the MSRA Chinese named entity corpus containing 23,182 Chinese sentences for training.

6.2.2 Methods and Evaluation Metrics

In this experiment, we only compared the results from the traditional named entity recognition algorithm and WebNLP, because the gazetteer features in Chinese language are usually too long, e.g., many organization names have more than 10 characters. If we use the Semi-CRF model, the search space will be too large. Therefore, we use the CRF model instead. We use the name NLP to refer to the CRF model used for the Chinese named entity extraction proposed in [25]. The extension for the CRF model in the WebNLP framework is analogous to the extension for the Semi-CRF model introduced in Section 4.3. The CRF model is also trained on the MSRA Chinese named entity corpus.

Chinese is a character-based language. A named entity in Chinese is a segment of characters with no delimiter at either end. Therefore, NER in Chinese is often modeled as a character labeling problem. We used seven different labels in the extended CRF model. For both PERSON and ORGANIZATION, three labels were used to represent the first character, the last character and the remaining character of the entity respectively. One label was used to represent the non-entity character.

TABLE 1
Comparison of the extraction results on the NAME (N), STREET (A), CITY (C), STATE (S) and OBJECT (O) of different algorithms.

Attributes		**N**	**A**	**C**	**S**	**O**
BHS	P	61.8%	90.2%	**100.0%**	**100.0%**	54.1%
	R	58.8%	77.7%	**90.7%**	95.6%	39.1%
	F1	60.3%	83.5%	**95.1%**	97.7%	45.4%
NHS	P	65.1%	93.0%	**100.0%**	**100.0%**	60.1%
	R	47.4%	**78.3%**	86.7%	94.8%	33.6%
	F1	54.9%	85.0%	92.9%	97.3%	43.1%
MHS	P	65.8%	93.3%	**100.0%**	**100.0%**	61.2%
	R	51.1%	77.7%	86.7%	94.8%	33.2%
	F1	57.5%	84.8%	92.9%	97.3%	43.0%
WebNLP	P	**67.1%**	**96.5%**	**100.0%**	**100.0%**	**64.3%**
	R	**66.2%**	**78.3%**	**90.7%**	**96.0%**	**46.5%**
	F1	**66.6%**	**86.5%**	**95.1%**	**97.9%**	**54.0%**

The statistical language features we used included boolean value feature functions and real value feature functions. Boolean value feature functions include unigram features, bigram features and type features. Real value features include character statistics numbers from an external entity list.

Similar to the experiment on English local entity extraction, the evaluation criteria in this experiment are also the standard Precision, Recall and F1 measure.

6.2.3 Results and Discussions

The extraction results of different algorithms are reported in Table 2. We can see that the proposed WebNLP framework improved both the precision and the recall of the Chinese NER task. Especially, it increased the recall significantly. For example, for PERSON, the recall of the NLP was only 54.4%, but the recall of the WebNLP was 79.0%, which was increased by nearly 50% compared to NLP. In total, the WebNLP framework increased the recall of the two types of Chinese Named Entities from 55.8% to 76.8%, and increased the F1 measure from 62.1% to 74.5%.

Since the grammar of the Chinese language is very flexible, the NER model trained on the Chinese language corpus always suffers from the flexibility of the language when working on the text from webpages, where many text strings are irregular. Fortunately, the decision on irregular text can be reinforced by those on regular text, e.g., the content of the news story and the formal introduction to an organization. The WebNLP framework enables such reinforcement by connecting the multiple mentions of the same entity in the extended CRF model. The experimental results proved its efficiency.

7 CONCLUSIONS

Webpage understanding plays an important role in web search and mining. It contains two main tasks, i.e., page structure understanding and natural language understanding. However, little work has been done towards an integrated statistical model for understanding webpage structures and processing natural language sentences within the HTML elements.

In this paper, we introduced the WebNLP framework for webpage understanding. It enables bidirectional integration of page structure understanding and natural language understanding. Specifically, the WebNLP framework is composed of two models, i.e., the extended HCRF model for structure understanding and the extended Semi-CRF model for text understanding. The performance of both models can be boosted in the iterative optimization procedure. The auxiliary corpus is introduced to train the statistical language features in the extended Semi-CRF model for text understanding, and the multiple occurrence features are also used in the extended Semi-CRF model by adding the decision of the model in last iteration. Therefore, the extended Semi-CRF model is improved by using both the label of the vision nodes assigned by the HCRF model and the text segmentation and labeling results, given by the extended Semi-CRF model itself in last iteration as additional input parameters in some feature functions; the extended HCRF model benefits from the extended Semi-CRF model via using the segmentation and labeling results of the text strings explicitly in the feature functions. The WebNLP framework closes the loop in webpage understanding for the first time. The experimental results show that the WebNLP framework performs significantly better than the state-of-the-art algorithms on English local entity extraction and Chinese named entity extraction on webpages.

REFERENCES

[1] J. Cowie and W. Lehnert, "Information extraction," *Commun. ACM*, vol. 39, no. 1, pp. 80–91, 1996.

[2] C. Cardie, "Empirical methods in information extraction," *AI Magazine*, vol. 18, no. 4, pp. 65–80, 1997.

[3] R. Baumgartner, S. Flesca, and G. Gottlob, "Visual web information extraction with lixto," in *Proceeding of VLDB*, 2001, pp. 119–128.

[4] A. Arasu and H. Garcia-Molina, "Extracting structured data from web pages," in *Proceeding of SIGMOD Conference*, 2003, pp. 337–348.

[5] D. W. Embley, Y. S. Jiang, and Y.-K. Ng, "Record-boundary discovery in web documents," in *Proceeding of SIGMOD Conference*, 1999, pp. 467–478.

[6] N. Kushmerick, "Wrapper induction: Efficiency and expressiveness," *Artif. Intell.*, vol. 118, no. 1-2, pp. 15–68, 2000.

TABLE 2
Extraction evaluation of NLP and WebNLP.

Attributes		PERSON	ORGANIZATION	TOTAL
NLP	P	87.0%	58.5%	69.9%
	R	54.4%	57.4%	55.8%
	F1	66.9%	57.9%	62.1%
WebNLP	P	**87.4%**	**60.6%**	**72.3%**
	R	**79.0%**	**74.5%**	**76.8%**
	F1	83.0%	66.8%	74.5%

[7] K. Lerman, S. Minton, and C. A. Knoblock, "Wrapper maintenance: A machine learning approach," *J. Artif. Intell. Res. (JAIR)*, vol. 18, pp. 149–181, 2003.

[8] I. Muslea, S. Minton, and C. A. Knoblock, "Hierarchical wrapper induction for semistructured information sources," *Autonomous Agents and Multi-Agent Systems*, vol. 4, no. 1/2, pp. 93–114, 2001.

[9] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma, "Simultaneous record detection and attribute labeling in web data extraction," in *Proceeding of KDD*, 2006, pp. 494–503.

[10] Z. Nie, Y. Ma, S. Shi, J.-R. Wen, and W.-Y. Ma, "Web object retrieval," in *Proceeding of WWW*, 2007, pp. 81–90.

[11] J. Zhu, B. Zhang, Z. Nie, J.-R. Wen, and H.-W. Hon, "Webpage understanding: an integrated approach," in *Proceeding of KDD*, 2007, pp. 903–912.

[12] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," in *Proceeding of NIPS*, 2004.

[13] R. C. Bunescu and R. J. Mooney, "Collective information extraction with relational markov networks," in *Proceeding of ACL*, 2004, pp. 438–445.

[14] H. L. Chieu and H. T. Ng, "Named entity recognition: A maximum entropy approach using global information," in *Proceeding of COLING*, 2002.

[15] C. Sutton and A. McCallum, "Collective segmentation and labeling of distant entities in information extraction," in *ICML Workshop on Statistical Relational Learning and Its connections to Other Fields*, 2004.

[16] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "Block-based web search," in *Proceeding of SIGIR*, 2004, pp. 456–463.

[17] C.-H. Chang and S.-C. Lui, "Iepad: information extraction based on pattern discovery," in *Proceeding of WWW*, 2001, pp. 681–688.

[18] V. Crescenzi, G. Mecca, and P. Merialdo, "Roadrunner: Towards automatic data extraction from large web sites," in *Proceeding of VLDB*, 2001, pp. 109–118.

[19] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. T. Yu, "Fully automatic wrapper generation for search engines," in *Proceeding of WWW*, 2005, pp. 66–75.

[20] K. Lerman, L. Getoor, S. Minton, and C. A. Knoblock, "Using the structure of web sites for automatic segmentation of tables," in *Proceeding of SIGMOD Conference*, 2004, pp. 119–130.

[21] Y. Zhai and B. Liu, "Web data extraction based on partial tree alignment," in *Proceeding of WWW*, 2005, pp. 76–85.

[22] Y. Zhai and B. Liu, "Structured data extraction from the web based on partial tree alignment," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 12, pp. 1614–1628, Dec. 2006.

[23] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma, "Learning block importance models for web pages," in *Proceeding of WWW*, 2004, pp. 203–211.

[24] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceeding of ICML*, 2001, pp. 282–289.

[25] A. Chen, F. Peng, R. Shan, and G. Sun, "Chinese named entity recognition with conditional probabilistic models," in *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 2006, pp. 173–176.

[26] D. DiPasquo, "Using html formatting to aid in natural language processing on the world wide web," 1998.

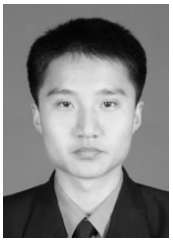
[27] C. Jacquemin and C. Bush, "Combining lexical and formatting cues for named entity acquisition from the web," in *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, 2000, pp. 181–189.

[28] R. G. Cowell, P. A. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems*, ser. Statistics for Engineering and Information Science. New York, NY: Springer, 1999.

[29] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Math. Program.*, vol. 45, no. 3, pp. 503–528, 1989.

[30] O. Etzioni, M. J. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," *Artif. Intell.*, vol. 165, no. 1, pp. 91–134, 2005.

[31] D. Downey, M. Broadhead, and O. Etzioni, "Locating complex named entities in web text," in *IJCAI*, 2007, pp. 2733–2739.

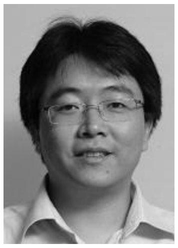


Chunyu Yang received the BE degree in control science and engineering from Department of Automation, Tsinghua University in 2004, where he is currently pursuing the PhD degree. His research interests are in pattern recognition, data mining, computer vision, and intelligent information processing. He is a student member of the IEEE.

Yong Cao joined Microsoft Research Asia in July 2007. He graduated in June 2007 with a Ph.D in signal processing & diagnose from University of Science and Technology of China (USTC). He received his bachelor degree in Electronic Engineering in USTC in 2002. Now, his research interests include data mining, machine learning and web search.

Zaiqing Nie is a lead researcher in the Web Data Management Group at Microsoft Research Asia. He graduated in May 2004 with a Ph.D. in Computer Science from Arizona State University. He received both his Master and Bachelor of Engineering degree in Computer Science from Tsinghua University. His research interests include entity search, data mining, machine learning, Web information integration and retrieval. Nie has many publications in prestigious conferences and journals including SIGKDD, ICML, WWW, CIDR, ICDE, TKDE, and JMLR. His recent academic activities include Program Committee co-chair of IIWeb 2007, proceedings chair of WWW 2008, sponsor co-chair of CIKM 2009, and program committee member of KDD 2008, SIGIR 2008, WWW 2008, ICML 2008, ACL 2009, etc. Some technologies he developed have been transferred to Microsoft products/services including Windows Live Product Search and Windows Live Search in China, Libra Academic Search (http://libra.msra.cn), and Renlifang Guanxi Search (http://renlifang.msra.cn)

Jie Zhou received the BS and MS degrees from Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the PhD degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995. From 1995 to 1997, he was a postdoctoral fellow in the Department of Automation, Tsinghua University, Beijing, where he is currently a full professor. His research area includes pattern recognition, image processing, computer vision, and information fusion. In recent years, he has authored more than 20 papers in international journals and more than 50 papers in international conferences. He received the Best Doctoral Thesis Award from HUST in 1995, the First Class Science and Technology Progress Award from the Ministry of Education (MOE) in 1998, the Excellent Teaching Award from Tsinghua University in 2003, and the Best Advisor Awards from Tsinghua University in 2004 and 2005, respectively. He was selected into the outstanding scholar plan of MOE in 2005. He is an associate editor for the International Journal of Robotics and Automation and Acta Automatica Sinica. He is a senior member of the IEEE.



Ji-Rong Wen received the BS and MS degrees from Renmin University of China and the PhD degree in 1999 from the Institute of Computing Technology, Chinese Academy of Science. He is currently a senior researcher with Microsoft Research Asia, Beijing. His main research interests are Web data management, information retrieval (especially Web IR), data mining and machine learning.