Statistical Entity Extraction from Web

Zaiqing Nie, Ji-Rong Wen, and Wei-Ying Ma, *Fellow, IEEE*

Abstract— **There are various kinds of valuable semantic information about real-world entities embedded in web pages and databases. Extracting and integrating these entity information from the Web is of great significance. Comparing to traditional information extraction problems, web entity extraction needs to solve several new challenges to fully take advantage of the unique characteristic of the Web. In this paper, we introduce our recent work on statistical extraction of structured entities, named entities, entity facts and relations from Web. We also briefly introduce iKnoweb, an interactive knowledge mining framework for entity information integration. We will use two novel web applications, Microsoft Academic Search (aka Libra) and EntityCube, as working examples.**

***Index Terms*— Entity Extraction, Named Entity Extraction, Entity Search, Entity Relationship Mining, Natural Language Processing, Web Page Segmentation, Interactive Knowledge Mining, Crowdsourcing**

I. Introduction

The need for collecting and understanding Web information about a real-world entity (such as a person or a product) is currently fulfilled manually through search engines. However, information about a single entity might appear in thousands of Web pages. Even if a search engine could find all the relevant Web pages about an entity, the user would need to sift through all these pages to get a complete view of the entity. Some basic understanding of the structure and the semantics of the web pages could significantly improve people's browsing and searching experience. In this paper, we will discuss the recent results and trends in web entity extraction, in the context of two novel web applications.

A. Motivating Example

Based our entity extraction and search technologies, we have been developing entity search engines to generate summaries of web entities from billions of public web pages and to allow for exploration of their relationships. Specifically, we deployed:

Manuscript received June 15, 2011.

Zaiqing Nie, Ji-Rong Wen and Wei-Ying Ma are all with the Microsoft Research Asia, Building 2, No. 5 Dan Ling Street, Haidian District, Beijing, P. R. China, 100080. (phone: (86) 10-59173309; e-mail: {znie, jrwen, wyma} @ Microsoft.com).

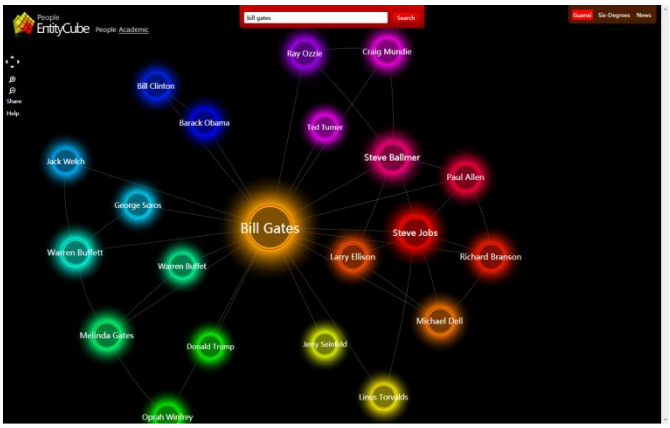

Figure 1. EntityCube**:** An automatically generated entity relationship graph based on knowledge extracted from billions of web pages.

- **EntityCube** (*http://www.entitycube.com*, see Figure 1) for users to search and browse summaries of entities including people, organizations, and locations. The Chinese version of EntityCube is called *Renlifang* (*http://renlifang.msra.cn*);
- **Microsoft Academic Search** (aka Libra Academic, *http://academic.research.microsoft.com*) for users to search and browse information about academic entities including papers, authors, organizations, conferences, and journals.

The entities and their relationships in EntityCube and Libra are automatically mined from billions of crawled web pages and integrated with existing structured knowledge from content providers. For each crawled web page, we extract entity information and detect relationships, covering a spectrum of everyday individuals and well-known people, locations, conferences, journals, and organizations. Below we list the key features of entity search engines:

- **Entity Retrieval:** entity search engines can return a ranked list of entities most relevant for a user query.
- **Entity Relationship/Fact Mining and Navigation:** entity search engines enable users to explore highly relevant information during searches to discover interesting relationships/facts about the entities associated with their queries.
- **Prominence Ranking:** entity search engines detect the popularity of an entity and enable users to browse entities in different categories ranked by their prominence during a given time period.
- **Entity Description Retrieval:** entity search engines rank text blocks from web pages by the likelihood of their being the entity description blocks.

The main objective of this paper is to introduce the web entity extraction problem and to summarize the solutions for this problem.

Web entity extraction is different from traditional information extraction in the following ways:

- **Visual Layout**: In a web page, there is much visual structure which could be very useful in segmenting the web pages into a set of appropriate atomic elements instead of a set of words and in tagging the atomic elements using the attribute names;
- **Information Redundancy**: The same knowledge/fact about an entity may redundantly exist in multiple heterogeneous web pages with different text or layout patterns, and this redundancy could be very useful in statistical pattern discovery;
- **Information Fragmentation**: Information about a single entity is distributed in diverse web sources, each source may only have a small piece of its information, and the format of web pages across heterogeneous data sources is very different;
- **Knowledge Base**: The existing structured information about an entity in the knowledge databases could be very useful in extracting knowledge from other sources about this entity.

Our recent work on web entity extraction ([18] [23] [33] [35] [36] [38] [39]) proposes to take advantage of these unique characteristics of the Web in extracting and integrating entity information. Specifically,

- **Vision-based Web Entity Extraction:** Given a web page, we partition the page at the semantic level and construct a vision-tree for the page according to its visual layout [7]. Each node in the vision-tree will correspond to a block of coherent content in the original page, and the leaf nodes are the HTML elements of the web page. The page structure understanding task can be treated as assigning semantic labels to the nodes on vision-tree (i.e., blocks on a web page) [38]. After the page structure understanding task, we further segment and label the text content inside HTML elements to extract the attribute values of an entity. Since much of the text content on a web page is often text fragments and not strictly grammatical, traditional natural language processing techniques that typically expect grammatical sentences, are no longer directly applicable. We propose a vision-based web entity extraction approach to jointly optimize both page structure understanding and web text labeling [33].

- **Statistical Snowball for Pattern Discovery:** Because of the information redundancy nature of the Web, the same entity facts may be repeatedly written in different web pages with different text patterns (or layout patterns). If we could find all possible patterns in describing entity facts and relationships, we could greatly improve the web entity extraction accuracy. In the literature, how to exploit information redundancy to improve information extraction has been considered as an interesting research problem ([1][11][14][19][34]). We introduce a Statistical Snowball (StatSnowball) approach to iteratively discover extraction patterns in a bootstrapping manner ([18] [35]). Starting with a handful set of initial seeds, it iteratively generates new extraction patterns and extracts new entity facts. The discovered extraction patterns can be used as the text features for web entity extraction in general.
- **Interactive Entity Information Integration:** Because the information about a single entity may be distributed in diverse web sources, entity information integration is required. The most challenging problem in entity information integration is name disambiguation. This is because we simply don't have enough signals on the Web to make automated disambiguation decisions with high confidence. In many cases, we need knowledge in users' minds to help connect knowledge pieces automatically mined by algorithms. We propose a novel knowledge mining framework (called iKnoweb) to add people into the knowledge mining loop and to interactively solve the name disambiguation problem with users.
- **Using Structured Knowledge in Entity Extraction**: We can imagine the significant growth of the knowledge base after we extract and integrate entity information from even a small portion of the Web. When we extract the entity information from a newly crawled web page, it's very likely we already have some information in the knowledge base about the entities to be extracted from the page. Our empirical results show that the extraction accuracy could be significantly improved if we use the knowledge about these entities during extraction [23].

The rest of the paper is organized as follows. In the next section, we formally define the web entity extraction problem and introduce the background of our research on web entity extraction and search. Section 3 summarizes our work on vision-based web entity extraction and shows that using structured knowledge in entity extraction could significantly improve the extraction accuracy. Section 4 summarizes our work on using statistical snowball to discovery new extraction patterns and entity facts and descriptions. Section 5 introduces our most recent idea on interactive entity information integration, and Section 6 concludes the paper.

II. Background & Problem Formulation

In this section, we introduce the background information and define the web entity extraction problem.

A. Web Entities

We define the concept of *Web Entity* as the principal data units about which Web information is to be collected, indexed and ranked. Web entities are usually recognizable concepts, such as people, organization, locations, products, papers, conferences, or journals, which have relevance to the application domain. Different types of entities are used to represent the information for different concepts. We assume the same type of entities follows a common relational schema:

$$R(a_1, a_2, \ldots, a_m)$$

Attributes, $A = \{a_1, a_2, \ldots, a_m\}$, are properties which describe the entities. The key attributes of an entity are properties which can uniquely identify an entity.

The designer of an entity search engine needs to determine the types of entities which are relevant to the application, and the key attributes of these entities.

B. *Entity Search Engine*

Figure 2 shows the brief architecture of an entity search engine. First, a crawler fetches web data related to the targeted entities, and the crawled data is classified into different entity types, such as papers, authors, products, and locations. For each type, a specific entity extractor is built to extract structured entity information from the web data. At the same time, information about the same entity is aggregated from different data sources including both unstructured web pages and the structured data feeds from content providers. Once the entity information is extracted and integrated, it is put into the web entity store, and entity search engines can be constructed based on the structured information in the entity store. Moreover, advanced entity ranking and mining techniques can be applied to make search more accurate and intelligent ([20] [22] [24]).

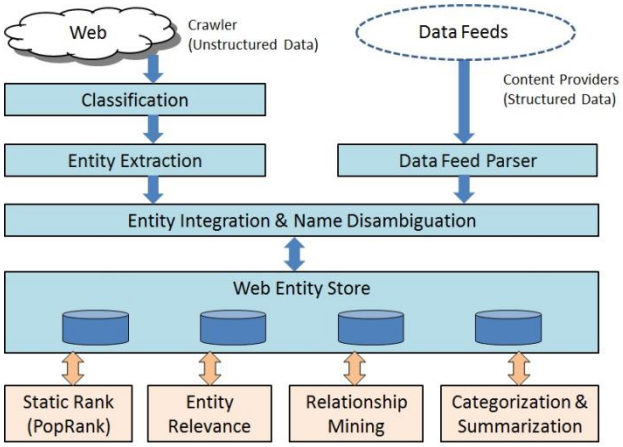

Figure 2. System Architecture of Entity Search Engines

C. *Blocks & Vision-Trees*

For web entity extraction, a good representation format for web pages can make the extraction task easier and improve the extraction accuracy.

In most previous work, tag-tree, which is a natural representation of the tag structure, is commonly used to represent a web page. However, as [7] pointed out, tag-trees tend to reveal presentation structure rather than content structure, and are often not accurate enough to discriminate different semantic portions in a web page. Moreover, since authors have different styles to compose web pages, tag-trees are often complex and diverse.

A vision-based page segmentation (VIPS) approach is proposed to overcome these difficulties [7]. VIPS makes use of page layout features such as font, color, and size to construct a *vision-tree* for a page. It first extracts all suitable nodes from the tag-tree, and then finds the separators between these nodes. Here, separators denote the horizontal or vertical lines in a web page that visually do not cross any node. Based on these separators, the vision-tree of the web page is constructed. Each node on this tree represents a data region in the web page, which is called a *block*. In Figure 3, we show two example blocks (marked by two red rectangles) of the web page. The root block represents the whole page. Each inner block is the

Figure 3. A sample web page with two similar data records

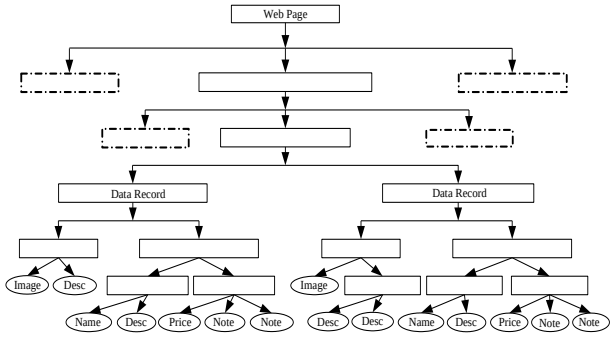

Figure 4. The vision-tree of the page in Figure 3

aggregation of all its child blocks. All leaf blocks are atomic units (*i.e.* elements) and form a flat segmentation of the web page.

Since vision-tree can effectively keep related content together while separating semantically different blocks from one another, we use it as our data representation format. Figure 4 is a vision-tree for the page in Figure 3, where we use rectangles to denote inner blocks and use ellipses to denote leaf blocks (or elements). Notice that the blocks denoted by dotted rectangles are not fully expanded.

D. *Web Entity Extraction*

Given a web corpus, web entity extraction is the task of extracting knowledge pieces of an entity from each individual web page within the web corpus and integrating all the pieces of the entity together. Below we formally define the web entity extraction problem using the terms we have defined in this section. See Figure 5 for a real example of web entity extraction.

Definition 2.1 (Web Entity Extraction): *Given a vision tree X, a knowledge base K, and an entity schema* $R(a_1, a_2, \ldots, a_m)$*, the goal of web entity extraction is:*

- *To find the optimal segmentation of the text on the vision tree and the optimal assignment of the attribute names of the entity schema to the corresponding text segments* $\mathbf{S}^*$*:*

$$\mathbf{S}^* = \arg\max\ p(\mathbf{S}|\mathbf{X}, \mathbf{K})$$

- *To integrate the attribute values in X with the existing information about the entity in the knowledge base K.*

Here, the text segmentation and labeling results of the vision tree **X** are denoted as $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2 \ldots \mathbf{s}_i \ldots \mathbf{s}_{|S|}\}$.

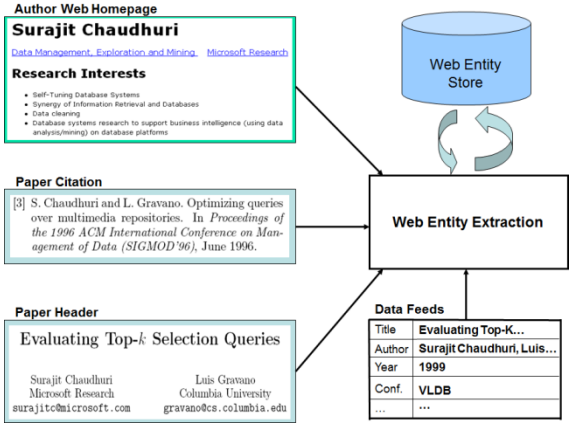

Figure 5. Web Entity Extraction Examples

III. Vision-Based Web Entity Extraction

In this section, we summarize our work on web entity extraction. Specifically, we first introduce three types of features we use in web entity extraction: visual layout features, text patterns, and knowledge base features. Then we present a statistical model to jointly optimize both page layout understanding and text understanding for web entity extraction leveraging these three types of features.

A. Features for Vision-Based Web Entity Extraction

As we mentioned above, there exist three types of information that could be utilized for web entity extraction: visual layout features, text patterns, and knowledge base features. In the following, we will discuss them respectively.

Visual Layout Features

Web pages usually contain many explicit or implicit visual separators such as lines, blank area, image, font size and color, element size and position. They are very valuable for the extraction process. Specifically, it affects two aspects in our framework: block segmentation and feature function construction.

Using visual information together with delimiters is easy to segment a web page into semantically coherent blocks, and to segment each block of the page into appropriate sequence of elements for web entity extraction.

Visual information itself can also produce powerful features to assist the extraction. For example, if an element has the maximal font-size and centered at the top of a paper header, it will be the *title* with high probability. If two sub-blocks have similar patterns in appearance (for example, two authors' address information in the paper header in Figure 5), the corresponding items in them should have the same labels. Though tag information is unstable across multiple heterogeneous website, the visual information is much more robust, because people are always trying to display information on the web orderly and clearly, and this desirability makes the visual appearances of the same kind of entities vary much less than tags.

In [36], we show that page layout understanding can improve Web entity extraction compared to pure text understanding methods. Specifically, to test the effectiveness of our 2D CRF model incorporating 2D layout understanding for Web IE, we choose linear-chain CRFs as the baseline models for their outstanding performance over other sequential models. We carry out our experiments in the domain of product entity extraction. In the experiments, four attributes ("name", "image", "price", and "description") are evaluated. 400 product blocks with two-dimensional neighborhood dependencies are randomly selected as training samples. Another 1000 such blocks are used as testing sets. We show the experimental results in Figure 6. As we can see that the 2D CRF model leveraging page layout information can significantly improve both the F1 of each attribute extraction results and the average block instance accuracy (i.e. the percentage of blocks of which the key attributes [name, image, and price] are all correctly labeled).

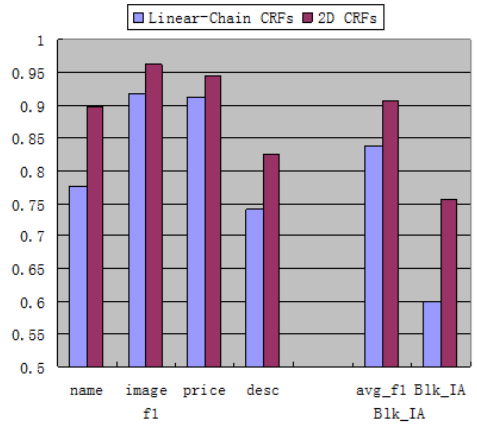

Figure 6. 2D Page Layout Helps Web Entity Extraction

Text Features

Text content is the most natural feature to use for entity extraction. Traditionally, the text information is treated as a sequence of words to be labeled. Statistics about word emission probabilities and state transition probabilities are computed on the training dataset, and then these statistics are used to assist labeling the words one by one.

In web pages, there are a lot of HTML elements which only contain very short text fragments (which are not natural sentences). We do not further segment these short text fragments into individual words. Instead, we consider them as the atomic labeling units for web entity extraction. For long text sentences/paragraphs within web pages, however, we further segment them into text fragments using algorithms like Semi-CRF [26] (see detailed discussions on how we segment the text content of a web page in sub-section B).

We prefer to use the natural text segments of a web page as atomic labeling units because of the following reasons.

- First of all, these short text fragments themselves are not natural language sentences and it is difficult to

guess the semantic meanings based on single words. For example, given "A. J. Black", we could say with high confidence that it is an *author* name. But little could be told based on individual word separately: "A.", "J.", and "Black". Given "Data Mining", we have no idea whether the labels should be *title* or *conference*, because they have similar emission probabilities for these two attributes. But if we treat "International Conference on Data Mining" as a whole, we could almost definitely say that labels of the five words are all *conference*.

- Secondly, because only one word's label is determined in one round, the labeling efficiency is impaired.
- Thirdly, usually it is straightforward to convert the information of a block on the Web to an appropriate sequence of elements, using visual features like font and position and delimiters like punctuation.

The text features are very effective in web entity extraction and they are different for different entity types. For example, for product entity extraction, below are two example text features:

- The text fragment only contains "$" and digits
- Percentage of digits in the text fragment

The HTML tags of the web pages are another type of text information which is widely utilized in traditional wrappers. But they are not so useful here because of their website-dependent nature. Due to different designing styles among individual website creators, information implied by tags is not stable. We will only use the tag information to estimate the visual layout during the page layout understanding task.

Another type of text patterns we use in web entity extraction is the patterns we automatically discovered in a bootstrapping manner. These patterns are used to describe entity facts and their relationships in nature language sentences. Because of the information redundancy nature of the Web, the same entity facts may be repeatedly written in different web pages with different text patterns [1]. In Section IV, we introduce a Statistical Snowball approach to iteratively discover extraction patterns in a bootstrapping manner ([18] [35]). Starting with a handful set of initial seeds, it iteratively generates new extraction patterns and extracts new entity facts.

Knowledge Base Features

For some web entities, there may be some structured information in the knowledge base about them already. This structured information can be used to remarkably improve the extraction accuracy in three ways.

- First of all, we can treat the information in the knowledge base as additional training examples to compute the *element (i.e. text fragment) emission probability*, which is computed using a linear combination of the emission probability of each word within the element. In this way we can build more robust feature functions based on the element emission probabilities than those on the word emission probabilities.
- Secondly, the knowledge base can be used to see if there are some matches between the current text fragment and stored attributes. We can apply the set of domain-independent string transformations to compute the matching degrees between them [31]. These matching degrees, which are normalized to the range of $[0,1]$, can be used as a knowledge base feature to determine the label. For example, when extracting from the paper citation in Figure 5, its first element is "S. Chaudhuri and L. Gravano". It has a good match with the *author* attribute of the second record in the knowledge base. Then we can say with certain confidence that the label of the first element is *author*.
- Thirdly, if we found a good match between the entity information in the web page and the key attributes of an entity in the knowledge base, we can say with high confidence that the information on the web page refers to the same entity in the knowledge base. Then we can use other attributes of this entity in the knowledge base to label the rest elements of the web page or rectify wrong labels. Take paper header in Figure 5 for an example. *Title* is a key attribute of a paper. For the first element, "Evaluating Top-k Selection Queries", we will find a good match with the *title* attribute of the paper entity in the knowledge base (which is collected from the structured data feed). It is of high probability that the header and the paper in the knowledge base are about the same paper. We then use all the matching results to direct further extraction or rectification of structured entity information from the paper header.

We have done some initial experiments to show that utilizing knowledge base features achieves an obvious improvement on extraction accuracy. To test the effectiveness of utilizing knowledge base information, we vary the size of knowledge base during the extraction of paper entities from PDF files (crawled from the Web) in Libra. Specifically, we randomly selected 0, 5000, 30000, and 150000 paper entities from ACM DL to derive different knowledge bases, and conducted individual experiment on each of them. The accuracy results are shown in Figure 7. Here the accuracy is defined as the percentage of instances in which all words are correctly labeled. As we can see, when we increase the size of the knowledge base, we obtain a gradual improvement on accuracy.

Although we can clearly see the improvement by leveraging a knowledge base, we do need to guarantee the quality of the knowledge. Otherwise the errors in the knowledge base will be further amplified through the knowledge base features used in

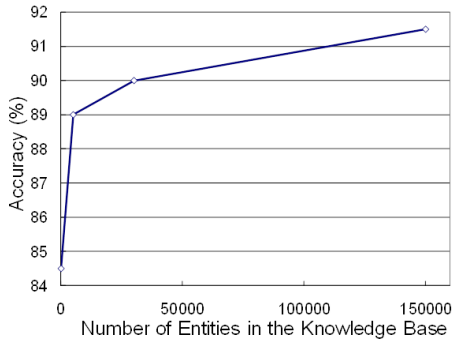

Figure 7. Extraction Accuracy V.S. Knowledge Base Size

web entity extraction. In Section V, we discuss how to build an accurate knowledge base which integrates all structured information from the Web through an interactive knowledge mining approach.

B. *Models for Vision-Based Web Entity Extraction*

We need a well-defined joint statistical model that can integrate both the visual layout understanding and the web text understanding (considering visual layout features, text patterns, and knowledge base features) together, so that the labeling results of the HTML elements and page layout can give a priori for further understanding the texts within the HTML elements, while the understanding of the text fragments with the HTML elements can also give semantic suggestions to improve page layout understanding.

Vision-based Page Layout Understanding

As a web page is represented as a vision-tree, and the page layout understanding task becomes the task of assigning labels to the nodes on a vision-tree. In [38], we introduce a probabilistic model called Hierarchical Conditional Random Field (HCRF) model for page layout understanding.

For the page in Figure 3, the HCRF model is shown in Figure 8, 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 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 [36].

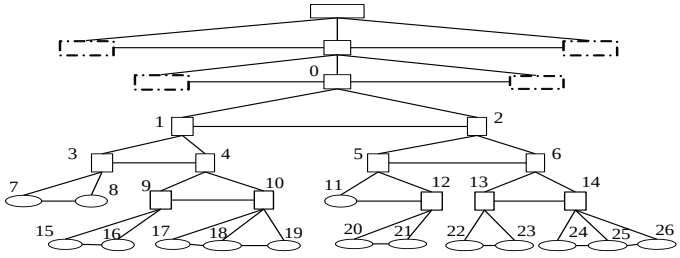

Figure 8. The HCRF model for the page in Figure 3.

As a conditional model, HCRF can efficiently incorporate any useful features for page layout understanding. By incorporating hierarchical interactions, HCRF could incorporate long distance dependencies and achieve promising results [38].

Web Page 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 web pages 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 [28] or Semi-CRF [26], to identify target information. [10] [28] are two attempts in this direction.

It is natural to leverage the page layout understanding 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 page layout understanding task and the text segmentation and labeling task together.

Joint Optimization of Layout and Text Understanding

In [36], we make the first attempt toward such solution. It first use HCRF to label the html elements and nodes on the vision-tree, and then use the Semi-CRF to segment the text content within the html element according to the assigned label. It is a top-down integration model. The decision of the HCRF model could guide the decision of the Semi-CRF model, i.e., it reduces the possible searching space of the Semi-CRF model to make the decision more efficient.

The drawback of such top-down strategy is apparent. The HCRF model could not use the decision of the Semi-CRF model. That means the entity block detection cannot benefit from the understanding of the attributes contained in the text. Without knowing the decision of Semi-CRF, i.e., the attribute extraction result, the entity block detection cannot be improved further because no extra evidence is provided. Furthermore, the text features with sequential label dependencies still could be shared among the multiple mentions of the same text fragment. We need to find a way to make use of such information better.

Therefore, the extension to bidirectional integration is natural. By introducing the feedback from the text segmentation to HTML element labeling in [33], we close the loop in web page understanding, from page layout understanding to text understanding. Specifically, in [33], we introduce a novel framework called WebNLP (see Figure 9), which enables bidirectional integration of page layout understanding and shallow natural language processing in an iterative manner. In WebNLP framework, the labeling decision made by HCRF on page layout understanding and the decision made by semi-CRF on free text understanding could be treated as features in both models iteratively.

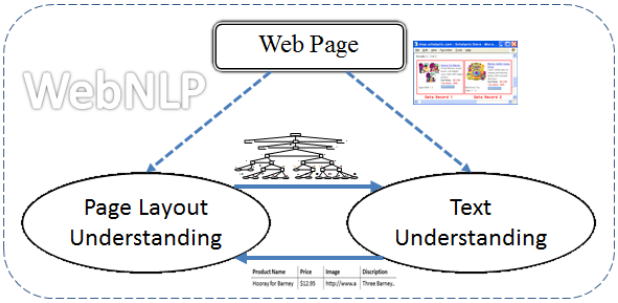

Figure 9. The WebNLP Framework

IV. Statistical Snowball for Pattern Discovery

Based on the overwhelming response from Chinese Internet users of our entity search engine Renlifang, we found that automatically extracting a large number of highly accurate entity relations and facts of different types from unstructured web texts is important to improve the user experience and to fulfill users' information needs.

The task of entity extraction from free web texts can be solved as two sub-problems: named entity recognition to extract the name of the entity and fact/relation extraction to extract other attributes/facts of the entity. For example, in the text paragraph shown in Figure 10 below, we can extract the following entity information below (**Name**: *William Henry "Bill" Gates III*, **Birthday**: *October 28, 1955*, **Affiliation**: *Microsoft*, **Title**: *Chairman*) for the people entities with schema ***Person*** *(Name, Birthday, Affiliation, Title).*

William Henry "Bill" Gates III (born October 28, 1955) is an American business magnate, philanthropist, author, and chairman of Microsoft, the software company he founded with Paul Allen. He is ranked consistently one of the world's wealthiest people and the wealthiest overall as of 2009. During his career at Microsoft, Gates held the positions of CEO and chief software architect, and remains the largest individual shareholder with more than 8 percent of the common stock. He has also authored or co-authored several books.

Figure 10. An Example Page with Biography Information

To solve these two sub-problems (i.e. NER and Relation/Fact Extraction), we need to write a lot text patterns as features in supervised statistical extraction models (including our vision-based web entity extraction models). It is prohibitory expensive to manually write all the possible text patterns. In this section, we introduce our work on automatically discovering text patterns for web entity extraction leveraging the information redundancy property of the Web. Because the same knowledge may be represented using different text patterns in different web pages, this motivates us to use bootstrapping methods to interactively discover new patterns through some popular seed knowledge.

Existing work on entity and relation extraction in the literature could not meet the requirements of automated text pattern discovery for web-scale entity search engines ([1][12][13]). Snowball [1] was the state of the art work on automated text pattern discovery and fact/relation extraction, which serves as the basis of our proposed Statistical Snowball. Snowball takes a small set of seed tuples as inputs, and employs the pattern-entity duality [5] to iteratively generate extraction patterns and identify new relation tuples. From the generated patterns and identified tuples, some confidence measures are carefully crafted to select good ones and add them to Snowball as new knowledge. Evaluating patterns and tuples is one key component, since it is crucial to select good patterns and good new seed tuples to make sure the system will not be drifted by errors. Another bootstrapping system—KnowItAll ([12][13]) requires large numbers of search engine queries and webpage downloads. Although the bootstrapping architecture is promising, Snowball has at least two obvious limitations, which make it unsuitable for web-scale text pattern discovery and relation extraction as motivated by EntityCube (and its Chinese version Renlifang). First, since the target of Snowball is to extract a specific type of relation (e.g., companies and their headquarters) the extraction patterns in Snowball are mainly based on strict keyword-matching. Although these patterns can identify highly accurate results, the recall will be limited. Second, Snowball does not have an appropriate evaluation measure, such as the probability/likelihood of a probabilistic model, to evaluate generated patterns. The carefully crafted measures and pattern selection criteria are not directly adaptable to general patterns (e.g., POS tag sequences), which can significantly improve the recall as shown in our empirical studies. This is because many tuples extracted by a general pattern are more likely not to be the target relations of Snowball, although they can be other types of relations. In this case, the confidence scores will be very small, and it is inappropriate to use the criteria as used in Snowball to select these patterns.

In [35], we address these issues as suffered by Snowball to improve the recall while keeping a high precision. We present a system called Statistical Snowball (StatSnowball). StatSnowball adopts the bootstrapping architecture and applies the recently developed feature selection method using $\ell 1$-norm [15] [32] to select extraction patterns—both keyword matching and general patterns. Starting with a handful set of initial seeds, it iteratively generates new extraction patterns; performs a $\ell 1$-norm regularized maximum likelihood estimation (MLE) to select good patterns; and extracts new relation tuples. StatSnowball is a general framework and the statistical model can be any probabilistic model. StatSnowball uses the general discriminative Markov logic networks (MLN) [25], which subsume logistic regression (LR) and conditional random fields (CRF) [17]. Discriminative models can incorporate arbitrary useful features without strong independence assumptions as made in generative models, like naïve Bayes (NB) and Hidden Markov Models (HMM).

By incorporating general patterns, StatSnowball can perform both traditional relation extractions like Snowball to extract pre-specified relations and open information extraction (Open IE) [3] to identify general types of relations. Open IE is a novel domain-independent extraction paradigm, which has been studied in both the natural language document corpus [27] and the Web environment [3]. Although the existing Open IE systems are self-supervised, they require a set of human-selected features in order to learn a good extractor.

In contrast, StatSnowball automatically generates and selects the extraction patterns. Moreover, the Open IE systems require expensive deep linguistic parsing techniques to correctly label training samples, while StatSnowball only uses cheaper and more robust shallow parsing techniques to generate its patterns. Finally, by using the MLN model, StatSnowball can perform joint inference, while the O-CRFs [3] treat sentences independently.

To the best of our knowledge, StatSnowball is the first working system that takes a bootstrapping architecture and applies the

well-developed $\ell 1$-norm regularized MLE to incrementally identify entity relations and discover text patterns.

The task of StatSnowball is to iteratively discover new text patterns and to identify relation/fact tuples. We have a strict mathematical formulation for StatSnowball. Formally, StatSnowball iteratively solves a $\ell 1$-norm regularized optimization problem:

$$\mathrm{P}: \quad \mathbf{w}^{\star} = \arg\min_{\mathbf{w}} LL(\mathcal{D}, \mathcal{R}, \mathbf{w}) + \lambda \|\mathbf{w}\|_1.$$

Where $LL(\mathcal{D}, \mathcal{R}, \mathrm{w})$ is the loss defined on the corpus $\mathcal{D}$ given a set of patterns (which are represented as formulae in the probabilistic model) $\mathcal{R}$ and the model weights w; and $\|.\|_1$ is the $\ell 1$-norm. The data corpus $\mathcal{D}$ and the pattern set $\mathcal{R}$ are updated at each iteration. For $\mathcal{D}$, by changing, we mean that new relation tuples are identified. For $\mathcal{R}$, the change is in the sense that new patterns are added. In the problem P, the loss can be the log-loss as used in probabilistic models or the hinge loss as used in support vector machines [9]. In [35], we focus on the logloss. This $\ell 1$-norm regularized MLE problem yields a sparse estimate by setting some components of w to exact zeros [15] [30] and has efficient solvers, such as the Orthant-Wise Limited-memory Quasi-Newton (OWL-QN) method [2].

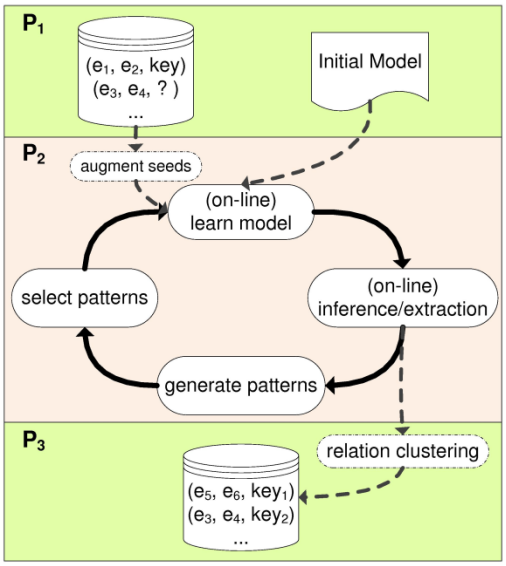

Figure 11. The StatSnowball Framework, with three parts: P1 (input), P2 (statistical extraction model), and P3 (output).

Figure 11 shows the architecture of StatSnowball. Generally, StatSnowball has three parts. The first part P1 is the input, which contains a set of seeds and an initial model.

The seeds are not required to contain relation keywords that indicate the relationship. Thus, we have two types of seeds, i.e., seeds with relation keywords like (e1, e2, key) or seeds without relation keywords like (e3, e4, ?). If the initial model is empty, we will first use the seeds to generate extraction patterns in order to start the process.

The second part P2 is the statistical extraction model. To start the iterative extraction process, StatSnowball takes the input seeds and the initial model (can be empty) in P1 to learn an extractor. We apply the $\ell 2$-norm regularized maximum likelihood estimation (MLE) at this step. Online learning is an alternative if batch learning is expensive.

Then, StatSnowball uses the learned model to extract new relation tuples on the data corpus. The third step in P2 is to generate extraction patterns with the newly identified relation tuples. These patterns are used to compose formulae of MLN. Finally, it selects good formulae to add to the probabilistic model and re-train the model. In this step, we first do $\ell 1$-norm regularized MLE, which will set some formulae's weights to zeros. Then, we remove these zero weighted formulae and send the resultant model to the next step for re-training. StatSnowball iteratively performs these four steps until no new extraction tuples are identified or no new patterns are generated. In this part, an optional component is the augmenting seeds, which can be used to find more seeds to start the process. In order to get high quality training seeds, this component applies strict keyword matching rules. We do not use it in the current system.

The third part P3 is the output, which is necessary only when StatSnowball is configured to do Open IE [3]. When StatSnowball performs Open IE, the extraction results in P2 are general relation tuples. To make the results more readable, we can apply clustering methods to group the relation tuples and assign relation keywords to them. The missing keywords of the seeds can be filled in this part.

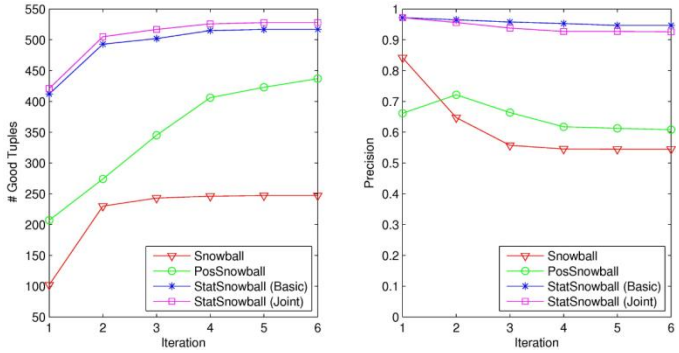

Figure 12. StatSnowbal V.S. Snowball

In [35], we compared StatSnowball with Snowball ([1]). To start the iteration process, StatSnowball uses 30 seeds (15 wife seeds and 15 husband seeds) to a dataset of 1 million Web blocks with text content. All the other systems perform the extraction of "Wife" and "Husband" separately with the corresponding seeds. All the extracted tuples are sent to human readers to judge whether they are correct extractions. Figure 12 shows the number of correct tuples and the precision of the identified tuples with respect to the number of iterations. From the results, we can see that StatSnowball systems identify much more correct relation tuples with a significantly higher precision on all the identified tuples than the Snowball systems, especially the Snowball using only keyword-matching patterns.

In summary, StatSnowball iteratively discovers both new facts/relations of an entity and more importantly new text patterns, which are useful for improving web entity extraction in general. In addition to the entity relation/fact extraction task, the discovered text patterns can also be used as text features

both in named entity extraction for web pages with long text paragraphs and in our vision-based web entity extraction for web pages with short text fragments but rich visual layout information.

V. Interactive Entity Information Integration

As we discussed before, the web information about a single entity may be distributed in diverse web sources, the web entity extraction task should integrate all the knowledge pieces extracted from different web pages (and data feeds). The most challenging problem in entity information integration is name disambiguation. Name disambiguation problem is a ubiquitous and challenging task in improving the quality of web search. This is because we simply don't have enough signals on the Web to make automated disambiguation decisions with high confidence. In many cases, we need knowledge in users' minds to help connect knowledge pieces automatically mined by algorithms. In this section, we propose a novel entity disambiguation framework (called iKnoweb) to add people into the knowledge mining loop and to interactively solve the name disambiguation problem with users. Similar to interactive models for other domains, our goal is to minimize the human effort in getting a nearly perfect solution.

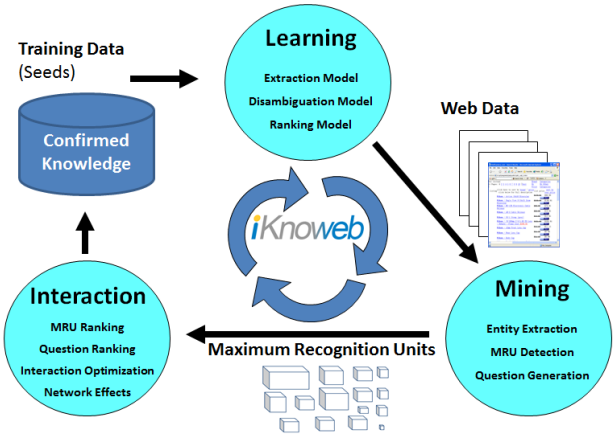

Figure 13. The iKnoweb Framework

To our best knowledge, iKnoweb is the first serious effort to interactively involve human intelligence for entity knowledge mining problems. iKnoweb is a crowdsourcing approach which combine both the power of knowledge mining algorithms and user contributions. More specifically, we expect that a user just needs to spend little effort to help us achieve the goal of accurately integrating all extracted knowledge pieces about an entity. The knowledge pieces could be facts extracted from general web pages about ambiguous names like "Michael Jordan", or scientific papers of different researchers named "Lei Zhang".

A. iKnoweb Overview

One important concept we propose in iKnoweb is *Maximum Recognition Units (MRU)*, which serves as atomic units in the interactive name disambiguation process.

Definition 5.1 (Maximum Recognition Unit): *A Maximum Recognition Unit is a group of knowledge pieces (such as web appearances, scientific papers, entity facts, or data records), which are fully automatically assigned to the same entity identifier with 100% confidence that they refer to the same entity (or at lease with accuracy equal to or higher than that of human performance), and each Maximum Recognition Unit contains the maximal number of knowledge pieces which could be automatically assigned to the entity given the available technology and information.*

Basically, MRU represents the best performance that the current technology can do to automatically connect the knowledge pieces about the same entity.

In Figure 13, we show the iKnoweb framework for interactive knowledge mining. The overall process is like follows. We first train machine learning models to automatically extract entity information from web pages using the available training data. The extracted knowledge pieces are then merged into MRUs. When a user wants to find the information about a particular entity, he/she will interact with the iKnoweb system by selecting some MRUs or answer some questions whose answers can help the system rank the relevant MRUs on the top for users to confirm. The confirmed knowledge will be stored into entity store, and the confirmed knowledge (and the original web pages containing the knowledge) can be used as training data to further improve our entity extraction models.

Specifically, the iKnoweb framework contains the following components.

- **Detecting Maximum Recognition Units**: We need to automatically detect highly accurate knowledge units, and the key here is to ensure that the precision is higher than or equal to that of human performance.
- **Question Generation**: By asking easy questions, iKnoweb can gain broad knowledge about the targeted entity. An example question could be: "Is the person a researcher? (Yes or No)", the answer can help the system find the topic of the web appearances of the entity.
- **MRU and Question Re-Ranking:** iKnoweb learns from user interactions, and the users will see more and more relevant MRUs and questions after several user interactions.
- **Network Effects**: A new User will directly benefit from the knowledge contributed by others, and our learning algorithm will be improved through users' participation.
- **Interaction Optimization**: This component is used to determine when to ask questions, and when to invite users to initiate the interaction and to provide more signals.

B. iKnoweb Applications

We are applying the iKnoweb framework to solving the name disambiguation problems together with users in both Microsoft Academic Search and EntityCube/Renlifang.

In Microsoft Academic Search, the iKnoweb framework is used to disambiguate scientific papers of authors with popular names. For some popular names, we have thousands of papers in our system. Our goal here is to help a researcher with a popular name disambiguate all his publications within 5 minutes. The academic papers are a special kind of Web documents with the following properties since they are more structured than general Web documents: most publications have some informative attributes, including a list of authors, their emails and/or homepages, references, citations, conference, title, abstract and download URLs. We need to first merge the papers into MRUs, and then a user just needs to select these MRUs. After each user selection, we will re-rank the rest MRUs (based on users previous actions) to move the relevant ones to the top for users to confirm.

In EntityCube/Renlifang, the problem of name disambiguation on general web pages is more complicated, mainly because the web pages are more diversified (including home pages, news, etc) and less structured. However, we can extract structured knowledge from the context of the entity and use them to generate MRUs. For example, if two web pages all mentioned the same friends (more than two) of a person name, these two pages can be merged into a MRU of the person name (note that, in real implementations, we need to take care of some out layer situations).

We recently deployed Renlifang 2.0 (http://renlifang.msra.cn) with several interactive mining and crowdsourcing features. In particular, we developed a novel interactive mining feature called Guanxi Wiki, which provides an easy and fun way of disambiguating people's web appearances and building wiki entries for anyone with a modest web presence. We also developed a 20 question game to encourage user participation and collect knowledge from the crowd.

In summary, iKnoweb is an interactive knowledge mining framework that enables users to interact with and contribute to our automated entity-extraction and disambiguation systems, such as EntityCube/Renlifang, Microsoft Academic Search, and Bing. iKnoweb can learn from both underlying, web-scale data and user interactions. With the underlying learned model, iKnoweb then can extract and disambiguate knowledge. iKnoweb also can interact with users to retrieve the knowledge in their minds and keep learning through interacting with people. As more users interact with iKnoweb, more knowledge will be accumulated. At the same time, relationships within this knowledge also will be established. This builds a huge knowledge web.

VI. CONCLUSION

How to accurately extract structured information about real-world entities from the Web has led to significant interest recently. This paper summarizes our recent research work on statistical web entity extraction, which targets to extract and integrate all the related web information about the same entity together as an information unit. In web entity extraction, it is important to take advantage of the following unique characteristics of the Web: visual layout, information redundancy, information fragmentation, and the availability of a knowledge base. Specifically, we first introduced our vision-based web entity extraction work, which considers visual layout information and knowledge base features in understanding the page structure and the text content of a web page. We then introduced our statistical snowball work to automatically discover text patterns from billions of web pages leveraging the information redundancy property of the Web. We also introduced iKnoweb, an interactive knowledge mining framework, which collaborates with the end users to connect the extracted knowledge pieces mined from Web and builds an accurate entity knowledge web.

VII. ACKNOWLEDGMENTS

We wish to thank Jun Zhu, Xiaojiang Liu, Yong Cao, Gang Luo, Yunxiao Ma, Zhengdong Lu, Chunyu Yang, Yuanzhi Zhang, Fei Wu, and Deng Cai for research collaboration and system implementation of Libra, Renlifang and EntityCube. This paper is mostly a summarization of the research work we have collaborated with them during the past 8 years. Please read the cited papers for their individual contributions.

REFERENCES

[1] Eugene Agichtein, Luis Gravano: *Snowball*: extracting relations from large plain-text collections. In Proceedings of the fifth ACM conference on Digital libraries, pp. 85-94, June 02-07, 2000, San Antonio, Texas, United States. [DOI：10.1145/336597.336644]

[2] G. Andrew and J. Gao. Scalable training of l1-regularized log-linear models. In Proceedings of International Conference on Machine Learning (ICML), Corvallis, OR, June 2007. [DOI：10.1145/1273496.1273501]

[3] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 2670–2676.

[4] M. Banko and O. Etzioni. The tradeoffs between open and traditional relation extraction. In Proceedings of the Association for Computational Linguistics (ACL), 2008, pp. 28-36.

[5] S. Brin. Extraction patterns and relations from the World Wide Web. In International Workshop on the Web and Databases (WebDB), 1998, pp. 172—183.

[6] Deng Cai, Xiaofei He, Ji-Rong Wen, and Wei-Ying Ma. Block-Level Link Analysis. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK (SIGIR), 2004, pp. 440-447.

[7] 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.

[8] Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Ying Ma. Block-based Web Search. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,

Sheffield, UK (SIGIR), pp. 456-463, 2004. [DOI: 10.1145/1008992.1009070]

[9] C. Cortes and V. Vapnik. Support-vector networks. Machine Learing, Vol. 20, Nr. 3 (1995) , p. 273-297 , 1995.

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

[11] D. Downey, O. Etzioni, S. Soderland. A Probabilistic Model of Redundancy in Information Extraction. In Proceedings of the 19th International Joint Conference on Artificial Intelligence,2005, pp. 1034-1041.

[12] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in knowitall. InProceedings of the 13th international conference on World Wide Web, 2004, pp. 100-110.

[13] O. Etzioni, M. 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. Artificial Intelligence, 165(1):91–134, 2005.

[14] H. Ji, R. Grishman. Refining Event Extraction through Cross-document Inference. In Proceedings of the Association for Computational Linguistics (ACL), 2008, pp. 254-262.

[15] A. Kaban. On Bayesian classification with laplace priors. Pattern Recognition Letters, 28(10):1271–1282, 2007.

[16]

[17] 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. In Proceedings of the Sixth International Conference on Data Mining, Maebashi City, Japan 2002, IEEE Press, pp. 250-258. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th International Conference on Machine Learning (ICML), 2001, pp. 282—289.

[18] Xiaojiang Liu, Zaiqing Nie, Nenghai Yu, Ji-Rong Wen: BioSnowball. automated population of Wikis. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), 2010, pp. 969-978. [DOI: 10.1145/1835804.1835926]

[19] G. Mann. 2007. Multi-document Relationship Fusion via Constraints on Probabilistic Databases. In Proceedings of the NAACL Conference on Human Language Technology (HLT/NAACL 2007), *pp. 332-339*. Rochester, NY, US.

[20] Zaiqing Nie, Ji-Rong Wen and Wei-Ying Ma. Object-Level Vertical Search.In Proceedings of the 3rd Biennial Conference on Innovative Data Systems Research (CIDR), 2007, pp. 235--246.

[21] Zaiqing Nie, Ji-Rong Wen, Wei-Ying Ma. Webpage understanding: beyond page-level search. SIGMOD Record 37 (4): 48-54 (2008).

[22] Zaiqing Nie, Yunxiao Ma, Shuming Shi, Ji-Rong Wen, Wei-Ying Ma. Web Object Retrieval. In Proceedings of the 16th international conference on World Wide Web (WWW), 2007, pp. 81-90. [DOI: 10.1145/1242572.1242584]

[23] Zaiqing Nie, Fei Wu, Ji-Rong Wen, Wei-Ying Ma. Extracting Objects from the Web. In Proceedings of the 22nd International Conference on Data Engineering (ICDE), p. 123, 2006. [DOI: 10.1109/ICDE.2006.69]

[24] Zaiqing Nie, Yuanzhi Zhang, Ji-Rong Wen and Wei-Ying Ma. Object-Level Ranking: Bringing Order to Web Objects. In Proceedings of the 14th international conference on World Wide Web (WWW), pp. 567--574, 2005. [DOI: 10.1145/1060745.1060828]

[25] M. Richardson and P. Domingos. Markov logic networks. Machine Learing, 62(1-2):107–136, 2006.

[26] S. Sarawagi and W. W. Cohen. Semi-Markov. Conditional Random Fields for Information Extraction. In Proceedings of Advanced in Neural Information Processing Systems (NIPS), pp. 1185–1192, 2004.

[27] Y. Shinyama and S. Sekine. Preemptive information extraction using unrestricted relation discovery. In Proceedings of the NAACL Conference on Human Language Technology (HLT/NAACL), pp. 304–311, 2006.

[28] S. Soderland. Learning to Extract Text-based Information from the World Wide Web. In Proceedings of Third International Conference on Knowledge Discovery and Data Mining (SIGKDD), 1997, pp. 251-254.

[29] Ruihua Song, Haifeng Liu, Ji-Rong Wen, and Wei-Ying Ma. Learning Block Importance Models for Webpages. In Proceedings of the 13th international conference on World Wide Web (WWW), 2004, pp. 203-211.

[30] C. H. Teo, Q. Le, A. Smola, and S. Vishwanathan. A scalable modular convex solver for regularized risk minimization. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD), pp. 727-736, 2007. [DOI: 10.1145/1281192.1281270]

[31] Sheila Tejada, Craig A. Knoblock, and Steven Minton. Learning Domain-Independent String Transformation Weights for High Accuracy Object Identification. *In* Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD), *pp. 350—359, 2002. [DOI: 10.1145/775047.775099]*

[32] R. Tibshirani. Regression shrinkage and selection via the LASSO. J. Royal. Statist. Soc., B(58):267–288, 1996.

[33] Chunyu Yang, Yong Cao, Zaiqing Nie, Jie Zhou, Ji-Rong Wen. Closing the Loop in Webpage Understanding. In Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM), pp. 1397-1398, 2008. [DOI: 10.1145/1458082.1458298]

[34] Roman Yangarber. Verification of Facts across Document Boundaries. *Proc. International Workshop on Intelligent*

Information Access. July 6-8, 2006, Marina Congress Center, Helsinki, Finland.

[35] Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, Ji-Rong Wen. StatSnowball: A Statistical Approach to Extracting Entity Relationships. In Proceedings of the 18th international conference on World Wide Web (WWW), 2009, pp. 101—110. [DOI: 10.1145/1526709.1526724]

[36] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang and W.-Y. Ma.2D Conditional Random Fields for Web Information Extraction. In Proceedings of International Conference on Machine Learning (ICML), 2005, pp. 1044-1051. [DOI: 10.1145/1102351.1102483]

[37] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang and Hsiao-Wuen Hon. Webpage Understanding: An Integrated Approach. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD), pp. 903-912, 2007. [DOI: 10.1145/1281192.1281288]

[38] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang and Wei-Ying Ma. Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD), pp. 494-503, 2006. [10.1145/1150402.1150457]

[39] 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.

Zaiqing Nie is a Lead Researcher in the Web Search & Mining 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 in 1998 and 1996, respectively. His research interests include data mining, machine learning, Web information integration and retrieval. Nie has many publications in high quality conferences and journals including SIGKDD, WWW, ICML, CIDR, ICDE, JMLR, and TKDE. His recent academic activities include PC co-chair of IIWeb (2007 and 2012), vice PC chair of ICDM 2010, senior PC of AAAI 2010 (AI and Web track) and KDD 2012, and PC member of conferences including WWW, KDD, ACL, WSDM, ICML etc. Some technologies he developed have been transferred to Microsoft products/services including Bing, Microsoft Academic Search, Renlifang and EntityCube.

Ji-Rong Wen is currently a senior researcher and group manager of the Web Search and Mining Group at Microsoft Research Asia (MSRA). Dr. Wen received B.S. and M.S. degrees from Renmin University of China, Beijing, in 1994 and 1996, respectively.. He received his Ph.D. degree in 1999 from the Institute of Computing Technology, the Chinese Academy of Science. Since then, he joined Microsoft Research Asia and conducted research on Web data management, information retrieval (especially Web search), data mining and machine learning. In the past 13 years at MSRA, he has filed 50+ U.S. patents in Web search and related areas. Many of his research results have been or are being integrated into important Microsoft products (e.g. Bing). He has published extensively on prestigious international conferences and journals, such as WWW, SIGIR, SIGKDD, VLDB, ICDE, ICML, ACM TOIS, IEEE TKDE, etc. He is also very active in related academic communities and served as program committee members or chairs in many international conferences and workshops. He is the co-chair of the "WWW in China" Track in WWW2008 held in Beijing.

Wei-Ying Ma is an Assistant Managing Director at Microsoft Research Asia where he oversees multiple research groups in the area of Web Search, Data Mining, and Natural Language Computing. He and his team of researchers have developed many key technologies that have been transferred to Microsoft's Bing Search Engine. He has published more than 250 papers at international conferences and journals. He is a Fellow of the IEEE and a Distinguished Scientist of the ACM. He currently serves on the editorial boards of ACM Transactions on Information System (TOIS) and ACM/Springer Multimedia Systems Journal. In recent years, he served as program co-chair of WWW 2008, program co-chair of Pacific Rim Conference on Multimedia (PCM) 2007, and general co-chair of Asia Information Retrieval Symposium (AIRS) 2008. He is the general co-chair of ACM SIGIR 2011. Before joining Microsoft in 2001, Wei-Ying was with Hewlett-Packard Labs in Palo Alto, California where he worked in the fields of multimedia content analysis and adaptation. From 1994 to 1997, he was engaged in the Alexandria Digital Library project at the University of California, Santa Barbara. He received a bachelor of science in electrical engineering from the National Tsing Hua University in Taiwan in 1990. He earned a Master of Science degree and doctorate in electrical and computer engineering from the University of California at Santa Barbara in 1994 and 1997, respectively.