A Web of Concepts

Nilesh Dalvi Ravi Kumar Bo Pang Raghu Ramakrishnan Andrew Tomkins
Philip Bohannon Sathiya Keerthi Srujana Merugu
Yahoo! Research
701 First Ave.
Sunnyvale, CA 94089, USA.
{ndalvi,ravikumar,bopang,ramakris,atomkins,plb,selvarak,srujana}@yahoo-inc.com

ABSTRACT

We make the case for developing a *web of concepts* by starting with the current view of web (comprised of hyperlinked pages, or documents, each seen as a bag of words), extracting concept-centric metadata, and stitching it together to create a semantically rich aggregate view of all the information available on the web for each concept instance. The goal of building and maintaining such a web of concepts presents many challenges, but also offers the promise of enabling many powerful applications, including novel search and information discovery paradigms. We present the goal, motivate it with example usage scenarios and some analysis of Yahoo! logs, and discuss the challenges in building and leveraging such a web of concepts. We place this ambitious research agenda in the context of the state of the art in the literature, and describe various ongoing efforts at Yahoo! Research that are related.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Miscellaneous*

General Terms

Experimentation, Measurement, Theory

Keywords

Concepts, Extraction, Ranking, Selection

1. INTRODUCTION

The way we gather, represent, and index the web is changing fundamentally to allow a more semantic view of content. The state of the art in indexing and searching the web is essentially based on viewing it as a collection of hyper-linked pages (each containing a bag of words). The value of the web, however, lies in the wealth of information provided by these pages on a broad range of entities, events, and topics (which we collectively refer to as *concepts*). We believe that extracting and organizing metadata in a concept-centric way that allows us to retrieve and present the most relevant information for the concepts related to a user's information need is the next transformative step in the evolution of the web. This will not only improve how current search engines identify and rank relevant content but also allow us to support richer attribute-oriented search criteria and to produce results that are meaningful concept-centric syntheses of information scattered across multiple web pages.

In this paper we outline a research agenda towards this ambitious goal, drawing upon many research threads from the database, information retrieval, machine learning, and web search communities. We begin with a discussion of what we mean by concepts and some of the challenges in representing and organizing concepts and instances of concepts in Section 2. We then motivate our proposal for developing a web of concepts using a combination of example scenarios and data from Yahoo! logs in Section 3. We discuss the challenges in extracting and reconciling concept-centric information from the web in Section 4. A central point of this paper is that we need to develop domain-centric extraction and maintenance systems in order to enable a web of concepts, going beyond the state of the art in site-centric wrappers, domain-independent extraction, and the development of techniques that address a part (e.g., CRF-based attribute extraction algorithms) of the problem but not the whole end-to-end challenge. In Section 5 we discuss several applications, including novel web search paradigms, that are enabled by a web of concepts. We discuss some related work in Section 6 and outline several research challenges in Section 7.

2. CONCEPTS

A discussion of entities, concepts, identities, and objects quickly enters deep epistemological and ontological waters. In this paper our focus is on a pragmatic approach to organizing information gathered from the web in a concept-centric fashion; thus, we want to examine how to describe and represent concepts (whatever they are!) in a way that is amenable to indexing and retrieval. To this end, we will avoid rigorous definitions of central terms such as *concepts* and *objects*, and offer informal discussion and examples instead. We will, however, present one specific proposal for representing and storing concept-centric metadata, and the challenges associated with it, in order to make our discussion more concrete.

PODS'09, June 29–July 2, 2009, Providence, Rhode Island, USA.

2.1 What are Concepts and Objects?

We use the term *concept* to refer to things of interest to users of the web who are either searching for information (e.g., is there a good Indo-Chinese restaurant in San Jose?) or trying to accomplish some task (e.g., find a good restaurant near downtown and make dinner reservations). Common types of concepts include entities, events, and topics.

Examples of concepts, in our sense of the term, include *restaurants, events,* and *academia*. Consider the concept of restaurants. Instances, of course, are specific restaurants, and each is described by *attributes* such as location and cuisine type. It is natural to think of a restaurant as an *object*, with a unique identifier (perhaps its address), state, etc. However, not all concepts have this property. Consider *events*, such as such as concerts and sports; instances include specific events such as the heavyweight title match on Jan 20, 2010 or the 2010 NFL Superbowl between the Packers and the Steelers. Sites such as `upcoming.yahoo.com` provide extensive event-centric data. One could associate an id with an event, but are events concepts? To take the discussion one step further, consider the concept of *academia*. We all understand what this denotes, and appreciate that there are many facets to it. We may want to get more information on some aspect of it (e.g., models of tenure in the US versus Europe), but there is no underlying object that corresponds to the topic *academia*. Nonetheless, when we focus on a specific facet, say research publications, we may identify related concepts such as *publications* and *research institutions*, each with instances that are concepts, described by one or more attributes.

2.2 Representing a Concept

One of our goals is to enable an evolutionary shift from current search technology, which is based primarily on massively scalable inverted index implementations. We therefore propose to describe an instance of a concept as a *loosely-structured record* (or just *lrec*) that consists of *(attribute-key, value)* pairs. We make two important stipulations:

1. There is a distinguished key called *id*, with the property that it uniquely identifies the record in the stored corpus. However, we may discover that two distinct records in fact describe the same real-world concept, or that a record conflates information about two distinct real-world concepts. This is part of the challenge in determining what information on the web pertains to a given concept and to a given instance of a concept, and is closely related to the *entity matching* problem in the literature; we will discuss this in Section 6.

2. For each concept that is represented in our corpus, we have metadata, including such things as a listing of attributes for which we might have values for one or more instances of that concept. Given a record, we can determine the corresponding concept.

Thus, a concept can be loosely thought of as a *type*, in that we typically store data about several instances, all of which have something in common based on the nature of the underlying concept. In concrete terms, several instances of a concept are likely to have values defined for each of a set of attributes. However, we do not assume that every concept instance defines values for all attributes. Indeed, the set of attributes for which an instance has defined values may evolve, and the set of attributes associated with a concept may also evolve.

This is a minimalist representational model. In practice, we may need to augment it with additional details at both the record and the concept levels. This model, however, gives us considerable flexibility in dealing with issues characteristic of the web, including missing data, uncertainty about many aspects of the data, and constant evolution (see Section 7.3).

We close this section by introducing one additional notion. A *domain* is a set of related concepts. Thus, people, publications and conferences are examples of concepts in the academic community domain.

2.3 Concept-Centric Data Organization

Undertaking to organize all the information on the web at a semantically useful level is no mean task, and that is essentially what is involved here. We make no claim to having a comprehensive solution, but outline some important issues to be addressed. There is a considerable literature on knowledge representation [13], to guide any detailed discussion of representational issues. For concreteness in our discussion, however, we have deliberately separated what we believe to be a minimal core for representing semantic, concept-centric descriptions in a manner that is amenable to leveraging existing search engine infrastructure; this is the basic lrec representation we presented above. Almost certainly, it will be worth considering extensions to this core, and the question is which extensions make sense and what they additionally support. In this context, some nuances worth considering include the following:

- Should the basic model allow for nested structure, e.g., XML-style paths and nested references to lrec ids, rather than just a flat collection of attribute-value pairs in an lrec? We have suggested a simpler alternative in part because retrieval is more readily mapped to existing inverted indexes, but also because populating more sophisticated representational structures using extraction is likely to be harder.

- Should there be support for provenance, versions, and uncertainty? It is by now traditional for large search engines to include information directly on the search results page providing metadata and links for an object of interest. If the query references a restaurant, for instance, there will be a box with a map, contact information, ratings, and so forth. Connecting users to key pieces of text (professional reviews, user reviews, blog mentions, etc.) is a critical part of building concept-oriented search engines. Furthermore, a piece of text may reference multiple concepts, the association to concepts is inferred and therefore uncertain, and may change over time, so simply associating the text with a single object is not sufficient: we will need to identify and maintain references between concepts of different types, and perhaps maintain versions of important concept instances over windows of time. These are computationally challenging requirements. How best can we approximate what is essential with an acceptable cost?

- In the real world, concepts are interrelated by many natural taxonomies and containment or inheritance re-

lationships. How far should we extend support for organizing lrecs into corresponding hierarchical relationships? As examples of the nuances we may want to capture, consider the following:

- We may want to be able to talk about the Nikon D40 model of digital cameras; how it is a particular kind of digital camera, which in turn is a particular kind of camera; and how it is a kind of Nikon cameras.
- We may want to be able to talk about the D40 camera as a concept, with each physical unit of this type as an instance (e.g., in the Amazon inventory); we may also want to talk about the D40 camera as an instance of the concept of camera models (e.g., in the `dpreview.com` review forum).
- We may want to talk about how the D40 camera *is part of* a special camera package.
- We may want to talk about abstract concepts such as academia or war that have many facets, and are hard to categorize unambiguously into given taxonomies. Yet, a collection of such concepts may lend itself to hierarchical categorization techniques that yield a data-driven taxonomy. What is the relative role of the two approaches—categorizing concepts and instances into curator-developed taxonomies versus data-driven taxonomy construction?

The distinctions we have drawn are only illustrative; clearly, there are more representational issues to be considered. Beyond issues of expressiveness and computational cost, however, in our setting there are two new considerations. First, is it feasible to extract or otherwise obtain (e.g., via contractual feeds) information about concepts and instances at this level of detail, reflecting such taxonomic niceties? Second, can our ability to interpret what users are looking for and to match it with the concept database take such nuances into account reliably? What is the right granularity for a concept hierarchy, given our ability to automatically categorize and match to user intent?

3. USAGE STUDIES

In this section we consider the implicit role of concepts in online user activities such as web search and browsing. We envision the following scenarios:

- A user may have a specific instance of a concept in mind and wants to search or browse for various attributes associated with that instance, e.g., find the menu, or reviews, for a given restaurant.
- A user may want to search for a set of concept instances whose values satisfy certain conditions, e.g., the best bakeries in Cupertino, or the closest restaurant to a given theater.

By examining users' search/browsing behavior, we hope to gain some insights into their activities with respect to concepts. Notice that in the search setting, users may also benefit from aggregated information, e.g., information on a specific concept hosted by different websites.

We present a set of empirical observations based on analyzing Yahoo! Search and Yahoo! Toolbar logs, focusing on how the notion of concepts may be relevant to user online experience and how they may benefit from a concept-oriented interpretation of the web content on the part of search engines.

Concepts vs. Search. The question we seek to answer is if users indeed search for both a specific instance of a concept and a set of concept instances with values for some keys satisfying certain constraints. To do this, we looked at queries (over a month) resulting in a click on a URL from `yelp.com`, a site hosting reviews on various local businesses. We identified three main sub-categories of these URLs:

- 59% are `biz` URLs, where each page is about an individual business,
- 19% are `search` URLs, where each page is a search result page in `yelp.com`, which can be searching for either a specific business (e.g., a restaurant named `churro factory` in Chicago) or a group of businesses (e.g., `wedding cakes Los Angeles`).
- 11% are `c` URLs, or category URLs, where each page is about a group of businesses in a pre-defined category, e.g., `San Jose Italian Restaurants`.

Thus, at least for searches that result in clicks on this website, we estimate that people search for a specific instance roughly 60%–70% of the time and search for a set of instances roughly 10%–20% of the time. Even if these specific numbers might vary for other websites, our study provides some evidence that users do conduct significant amounts of both types of search.

Searching for Attributes of a Concept. We now proceed to ask if users explicitly search for different attributes of a concept. To study this, we performed experiments on the search logs. First we obtained a list of restaurant homepage URLs from `yelp.com`. We then examined queries that led to a click on one of these restaurant homepage URLs. After removing the restaurant names and location information from the queries, we tallied the remaining tokens. Top words that co-occur in conjunction with restaurant names are the following: `menu` (3%), `coupons` (1.8%), `online`, `weekly specials`, `locations` (1.5%), etc., where the numbers in the parenthesis show the fraction of occurrence. Note that these are queries that led to a click on the restaurant's homepage, even when the user was actually looking for a specific attribute. The fraction of co-occurring words remain largely the same even when the query merely contained the word `restaurant`, regardless of what the user clicked, or when the query contained more specific terms, say, `chinese restaurant`. Other interesting but less frequently occurring attributes that surface from our study include `nutrition`, `to go`, `delivery`, `careers`, and `cod`.

Value in Aggregation. Would users benefit from aggregated information? If we look at users who clicked on a `biz` type URL from Yelp, more than 59% of the time they also clicked on at least one other URL for the same query, and 35% of the time they clicked on at least two other URLs. Manual examination of a small sample of the queries that

led to these URLs finds most of them to be indeed looking for a specific business and other pages clicked can include the homepage of the business, profile pages from other aggregation sites such as `citysearch.com`, `local.yahoo.com`, `yellowpages.com`, etc., as well as blogs and reviews written about the business. Clearly, even when users are searching for a specific instance of a concept, they seek diverse information beyond one source. An aggregation page can facilitate this process (Section 5.2).

As a concrete example of a more sophisticated search, here is a query from the log: `mexican food chicago best salsa` and the user clicked on a Yelp category page, a Citysearch search result page, as well as a page of expert reviews hosted by a restaurant. Most likely the user was consulting multiple sources to reach a conclusion regarding the best salsa. An aggregated page with locations of different mexican food places in chicago, accompanied by reviews that commented on salsa from different sources, with meta information on the trust-worthiness of these sources could probably add more value to this user's search for the best salsa.

Concepts vs. Browsing. How do users get to concepts on the web? To understand this, we analyzed the toolbar logs. As before, we obtained a list of restaurant homepages from `yelp.com`; these homepages are meant to represent the instances of the restaurant concept. We then examined the user trails that pass through one of these restaurant homepages. First, we observed that about 42% of the homepage visits are immediately preceded by a query to a search engine. Next, we examined the URL that is surfed immediately after the homepage is visited and observed that about 11.5% of these URLS are the location/address of the restaurant, 9% of them are the menu pages, and about 1% of them are coupons. These studies demonstrate that users are already looking for natural attributes of the restaurant concept, even in a browsing mode. Furthermore, about 10.5% of the user trails contain more than one distinct instance of the restaurant concept, suggesting that an aggregation of instances might be beneficial to the user.

4. CONSTRUCTING A WEB OF CONCEPTS

We can view today's web as a simplified web of concepts, where each record is of type "Document." We want to start from here and extract records of richer types. We use the term *extraction* broadly to refer to any of the operations that either create new records belonging to the concept or enrich existing records. The following list of extraction operations is typical.

- *Information extraction.* This extracts structured data from documents, e.g., an address from a restaurant website or a list of publications from a personal homepage.
- *Linking.* This creates links between existing records. Examples include linking a restaurant review to its corresponding restaurant or matching two different mentions of the same author.
- *Analysis.* This attaches metadata to records, e.g., identify and tag the cuisine type for a restaurant website or assign sentiments to a review.

We propose constructing a web of concepts using *domain-centric* extraction. A domain specifies a set of concepts of interest that may not be specific to any individual source. For example, a restaurant domain might specify the concepts `menu`, `location`, `review`, an academic domain might specify `author`, `publication`, and a shopping domain might specify `product`, `seller`, `review`. Most of the current extraction methods are *site-centric*, i.e., they can only be deployed to extract from a specific website or data source. In contrast, to construct a web of concepts, we propose to look at extraction holistically on a domain level. To do this, we need to bridge the gap between the capabilities of existing methods and the requirements for domain-centric extraction.

We start by describing the current extraction methods and then describe the domain-centric extraction methods that we are currently pursuing.

4.1 Site-Centric Extraction

The existing site-centric (or source-centric) methods for extraction can be classified along two dimensions: structural and semantic.

The *structural* approaches typically rely on the rich HTML structure employed by the author for presenting the content. For example, tables and lists are frequently used in webpages to present structured content. Restaurant aggregators (e.g., `yelp.com`, `citysearch.com`), product catalogs (e.g., `shopping.yahoo.com`, `amazon.com`), music and movie websites (e.g., `imdb.com`), and virtually any website that serves pages off a database often uses scripts to generate highly structured and regular HTML, and thus impart structure across multiple pages within the website. In this direction, *wrapper induction* [22, 44, 49, 40, 21, 39, 58, 7, 50, 4] has been used as a powerful and robust way for structural site-centric extraction. With relatively few labeled examples, extraction rules, called *wrappers*, can be learnt to extract information from a specific website. The main drawback with wrappers is that they rely on the existence of a structure.

In contrast, *semantic* approaches view web pages as text documents and employ natural language processing [16] and machine-learned probabilistic models. For instance, Conditional Random Fields [33, 36, 46] have been used effectively to parse postal addresses and lists of publications. Semantic techniques often require large supervised training data, and are sensitive to the construction of this training data; e.g., a probabilistic model learnt to extract US restaurant addresses may completely fail on European restaurant addresses, and a model learnt to extract Computer Science publications may perform poorly on Physics publications.

4.2 Domain-Centric Extraction

The main challenge in constructing a web of concepts is to extrapolate site-level extraction techniques to work at the domain level. A domain has an unbounded number of websites, which renders pure wrapper-style extraction infeasible, and a large amount of diversity, which cannot be easily captured by probabilistic models. While a complete solution towards constructing a web of concepts requires several challenges to be solved (Section 7.2), we describe some of the ongoing work at Yahoo! Research, and briefly discuss work that is especially relevant from other research groups.

Domain-Centric List Extraction. The goal of this work [41] at Yahoo! Research is a domain-centric extraction of lists. The motivation is that much information on the web is present in the form of lists or tables, e.g., a list of restau-

rants, a list of menu items for a restaurant, a list of publications, etc. This work tries to extract content in the form of lists by combining structure and semantics. A list can often be identified on a webpage by a repeating pattern of HTML structure. However, webpages often contain several lists, and we need to identify the lists that we are interested in; this typically requires us to combine domain knowledge with structural cues. For example, to extract a list of restaurants, we might have two kinds of domain knowledge: first, the fields of interest such as the address, city, zip code, phone number, and hours of operation, along with rules to identify zips/phones, and second, certain statistical properties (e.g., each restaurant is associated with a single zip code and has one or two phone numbers). This domain knowledge, along with an analysis of the repeated structure, is used to identify lists of addresses in a completely unsupervised, site-independent fashion.

Relational Classification. Most large web sites have pages covering a variety of content. One might be interested in filtering out only those pages that belong to a certain category and then doing further extraction on them. For example a city site such as `sanjose.com` has pages on diverse categories such as hotels, attractions, night-life, restaurants, events, etc. Let us say we are interested in selecting only events pages and extracting individual event information from them. If we are also interested in repeating the same process on thousands of city sites, then it is not scalable to develop an events classifier for each site since it requires labeling several pages on each site as events/non-events. Developing a global events classifier is easier, but it tends to be noisy given the vastly different content in the large collection of sites. Fortunately, the link and directory relationships in a site contain valuable signals for solving such a classification problem. For instance, all the events pages in `sanjose.com` are placed in a directory called `calendar`. Note that this relational structure will be different for different web sites. After bootstrapping the pages of a site with the classification labels given by an inaccurate classifier (such as the global classifier mentioned above), the relational structure present in that site can be used to revise them and get highly accurate classification. Graph-based methods for doing the above are discussed in [60].

Aggregator Mining. Independently maintained concept-centric web corpora will emerge and grow in importance. Many examples already exist: Wikipedia and numerous aggregator sites such as Yelp; sites such as LinkedIn that gather and selectively expose structured information about individuals; and structured integrated corpora such as *Freebase* and *KnowItAll* [30], and community driven *mass collaboration* sites such as *DBLife* [25, 48]. Automatically discovering such concept-centric web corpora, extracting data from them, and integrating with data from the rest of the web can significantly increase the quality of the unified web of concepts that we seek to build.

The aggregator mining work currently in progress at Yahoo! Research aims to address this problem by applying *bootstrapping* to make extraction scalable. The main idea is to use already extracted records to automatically generate labeled data and use it to extract more records. For instance, suppose we have already extracted a small set of Italian restaurant menu items and stored them in a database. Consider a restaurant website that lists a menu. As discussed before, we can identify lists by a structural analysis of the HTML content; this will yield us the menu items. Now, if we can map a few of the menu items to our database (that reflects our current knowledge of Italian restaurant menu items), then we can infer that the list represents an Italian restaurant menu and can extract additional menu items from the list to add to the database. Thus, we can start from a small set of seed records and bootstrap to extract more records from sources that overlap with the current set or records.

Matching. The matching work done at Yahoo! Research [23] aims to bridge the structured and unstructured views of a record by establishing that a piece of text is "about" a record. For example, matching a piece of text that is known to be a restaurant review to its corresponding restaurant will fall in this category. The main idea is to develop a domain-centric generative model of text that takes into account the domain knowledge (e.g., address, city, cuisine, etc. for restaurants) and use this model in order to choose the most likely restaurant a given review is about. This work is related to the widely studied entity matching problem, which we discuss in Section 6.

Other Work. The *DBLife* project at Wisconsin [25] shares the goal of building concept-centric integrated web repositories, and has led to the creation of `dblife.cs.wisc.edu` and influenced our work at Yahoo!. In contrast to our strong emphasis on domain-centric training of machine learned models, however, the DBlife project has taken a more rule-oriented approach and emphasized issues such as mass collaboration, managing site evolution, and optimizing extraction. The *Avatar* project at IBM Almaden [42] aims to apply extraction to a diverse range of datasets to improve search over enterprise datasets. It also takes a rule-oriented approach, and emphasizes rapid rule development and optimization. The *WebTables* work of Cafarella et al. [15] aims to extract all tables from the web and store them as relational data. The techniques are domain-centric. The main challenge here is to attach semantics to the extracted data for it to be useful in populating a web of concepts. An orthogonal approach is *DeepWeb* [45], which aims to surface the deep web by automated form-filling techniques. The *KnowItAll* [30] and the *TextRunner* work at the University of Washington also aim to extract information from the web in a site-independent manner using simple yet effective natural language techniques. The focus however is on extracting "common knowledge", i.e., the facts described textually in natural language and that occur often at several places on the web.

5. APPLICATIONS

In this section we discuss a variety of applications that are enabled by a web of concepts.

5.1 Augmenting Web Search

We begin with applications in the domain of web search. First, it is by now traditional for large search engines to include information directly on the search results page providing metadata and links for an object of interest. If the query references a restaurant, for instance, there will be a box with a map, contact information, ratings, and so forth. If

the query asks for a restaurant, e.g., `gochi cupertino`, then there will be a box containing a map showing the location of Gochi along with directions, reviews, and a pointer to the official homepage of the restaurant (Figure 1). To provide this capability, search engines must first build a database of concepts (often through licensing arrangements with data providers), and must then deploy technology to trigger the special box when appropriate for the query.

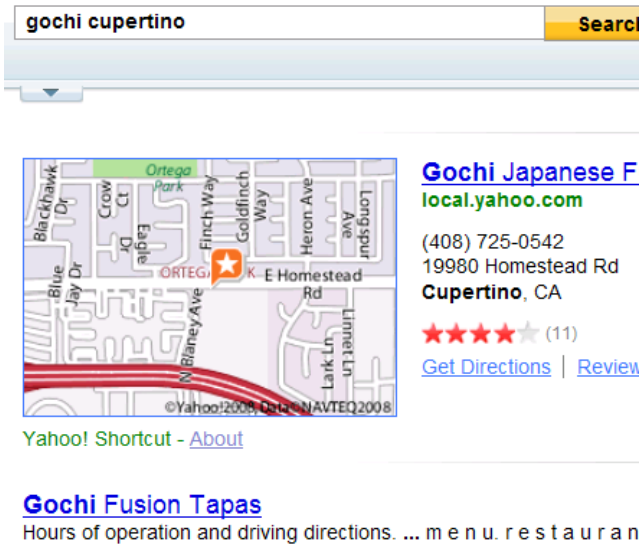

Figure 1: Search Results for `gochi cupertino`

A web of concepts influences each of these stages. First, gathering of the appropriate database becomes trivial, as it can be imported directly from a web of concepts; in the example above, we require all records of the concept `restaurant`. Second, the triggering algorithms tend to be data-hungry machine-learned recognizers that will readily incorporate the extensive metadata available through a web of concepts.

In addition to direct inclusion of concept records in the search results, a web of concepts has applications in the ranking of documents. Consider an example query, `Osteria Palo Alto`, which can be matched to a record in a web of concepts. This record has a key called `homepage`, whose value is a URL. This URL should be given preferential treatment by the ranker, as the official homepage of the requested entity. Similarly, other URLs that are linked to the record should be augmented with appropriate features. In practice, there are several ways to implement such a scheme, but where possible, it is efficient to pre-compute associations between documents and record identifiers, then store these associations with the document in the web search index. At query time, a separate procedure extracts relevant records and passes this information with the query to the search engine backend, which matches incoming record identifiers with the record identifiers associated with candidate documents, and produces features indicating that the document mentions the entity, is a homepage of the entity, includes a review of the entity, and so forth.

5.2 Concept Search

Traditional web search returns a list of documents. As described above, these documents can be annotated with information about one or more matching records such as a restaurant, or a product, or an indicator of the local weather. However, this type of result page is fundamentally different from the result page encountered in real estate listing search or personals search, in which the core results are of a concept other than document.

Search over documents has amassed a vast body of research literature. Search over other concepts is less well developed. This is true of both the algorithms and the user interaction paradigm. Today, the accepted paradigm is that users navigate to a vertical website and search within the concepts understood by that website, which are typically quite narrow: `yelp.com` hosts local entities, while `mlslistings.com` provides real estate listings, and so forth. Developing the science of ranking such disparate sources could improve user experience across all these verticals. However, as records of these different concepts become increasingly co-located and even interconnected, it is possible that concept search should adopt a different paradigm, in which users search a highly heterogeneous collection of records through a uniform interface. Simple navigational intents, in which a user intends to retrieve a single record using a few keywords, might be well served in such a model. This corresponds to the simple form of heterogeneous concept search available in web search today, as described above. However, particular vertical search providers offer users more sophisticated domain-specific approaches to searching. These can include refinement using specialized features (e.g., show only Chinese restaurants), special query parsing (e.g., geographic locations), custom query processing (e.g., combining locational proximity and genre proximity in a query for pizza in San Jose), and so forth. It is not clear how to provide these richer search and result set navigation capabilities. The Correlator work at Yahoo! Research [5] is a step in this direction.

5.3 Session Optimization

Web search and advertising today perform sophisticated pixel-by-pixel measurement and optimization of page real-estate. We anticipate that as these techniques diffuse into the internet industry, they will become increasingly common for other page types.

Consider the following example. A user visits `yahoo.com` and encounters a customized list of interesting articles. The user selects an article about TV series that may not be continued. A reference to the possible demise of Kings mentions actor Ian McShane's appearance in Deadwood, which was also terminated early in its lifecycle. The user performs a search for further information on Deadwood stars and follows links to a page about Timothy Olyphant, discovering to her surprise that the same actor played the spooky "Mr. Joshua" in Die Hard 4.

This type of browsing experience will increasingly be optimized based on the nature and context of the user, as well as the increasingly rich information available about the backend content. Thus, one may characterize an important trend in the evolution of online experience as optimization based on deeper understanding of both users and content. We will now discuss how a web of concepts impacts each of these areas, and then will cover how the user experience might change as a result.

Understanding Users. User modeling draws on techniques from areas ranging from psychology to temporal mod-

eling. Two keys areas of focus are *historical modeling*, which captures the long-standing predilections of the user (e.g., a preference for sophisticated text-rich academic documents, or an interest in jai alai), and *session modeling*, which models the current (short-term) interest of the user (e.g., booking a ticket to Madrid on a particular date, or researching family-friendly parks that allow charcoal grilling). In both cases, an understanding of the user's past interactions with records from a web of concepts are a key data source. Historical modeling benefits from information of the form: "this user consumes information referencing the concept jai alai with an average weekly inter-arrival time." Likewise, session modeling benefits from information of the form "this user consumed reviews for three steak restaurants in zipcode 95054 during the past hour." Concept information of this form is clearly just one part of a user model, but it is an important part.

Understanding Content. Given the above, it is natural to consider matching content to a particular user in a particular context based on the concepts represented in the content. The extent to which a piece of content will be interesting to a user depends on both the historical and session model for that user. A user who searches for `Birks` without context will probably be shown luxury jewelers Birks & Mayors. However, given that the user has been searching recently for restaurants in zipcode 95054, the best result page should instead include information drawn from the record of Birk's Steakhouse, along with pointers to web pages about that restaurant. This match is not possible without a model of the user's task, as well as the corresponding metadata allowing the geographic location of the restaurant to be employed in the ranking. Likewise, a user arriving at `yahoo.com` will encounter content that does not respond to a particular query, but is intended to be interesting and informative. An article about penetration of jai alai into the western US where the user is employed might be highly relevant to this user, but deeply uninteresting to other users.

5.4 Browse Optimization

Having sketched how a web of concepts might contribute to an understanding of users and content, we now offer a more futuristic view of how these capabilities might lead to the optimization of browse experiences online. Revisiting the prototypical interaction around TV series described above, we can now step back to discuss how an improved understanding of users and content might lead to a different experience. Notice that our user above encounters three types of pages:

1. Result pages, showing multiple records. E.g., an initial homepage listing several links that may be of interest to the user, or search results for a query on Deadwood.
2. Concept pages, showing information about some instance of a concept. E.g., page about Ian McShane; page about Deadwood.
3. Article pages, showcasing a piece of authored text. E.g., article about a TV series or an actor.

Websites ranging from verticals like `shopping.yahoo.com` to social sites like `myspace.com` to broad-ranging sites like `yahoo.com` offer these same three types of pages, and seek to provide users with compelling experiences by allowing users to traverse sequences of these page types in either targeted or serendipitous exploration. Table 1 shows the technologies employed to connect from one type of page (called **p**) to another type of page (called **q**).

Each cell of the table might be filled with a range of more speculative transition types; we have included only a few important entries. The first row shows connections from result pages to other types of pages, and hence represents the different types of links that might appear in a result list. Links from one result page to another represent the various forms of assistance, including suggestions for new queries, or opportunities to filter or reshape the result set. Links from a result page to a concept page represent the output of a concept search algorithm, as described in Section 5.2. And of course, a result page will show links to an article according to traditional search mechanisms for ranking text, but notice that different types of article pages might employ different ranking signals: web pages make use of hyperlinks, reviews make use of usefulness indications, blog posts employ blog-specific authority measures, and so forth.

The second row of the table characterizes the types of connections that can be made from concept pages. The first entry captures ways to generate results listings from a concept; there are many of these, but the most natural is probably the notion of searching "within" the concept. As an example, consider the page corresponding to jai alai, within the Sports concept. There are on the order of one million pages on the web that reference this sport, and the aggregate information included in those pages cannot be distilled into a single browsable summary. If the user wishes to search specifically within the "Jai Alai web" for information about, for instance, the MGM Grand casino, the results will center around the brief period during which the MGM Grand offered a facility to support gambling on the outcome of jai alai matches.

Links from a concept page to another concept page represent different forms of concept recommendation. Consider two key instances of this technology.

- *Alternatives.* A user visiting a concept page for Birks Steakhouse may be interested in other restaurants in a similar location, perhaps offering a similar level of quality or a similar cuisine type. The user is interested in discovering options that might displace Birks as a place to dine, so the goal of the system is to suppress recommendations that the user finds less preferable overall then Birks.

- *Augmentations.* A user viewing a page about the Canon G10 camera may also be interested in the NB-7L battery. In this instance, the battery augments rather than displacing the camera. There is no analog to the desire to suppress less preferable alternatives; rather, the goal is to rank augmentations by the degree of interest conditioned on engagement with the primary record.

As these examples show, concept recommendation should not be viewed as a single problem with a single optimization criterion. Instead, the problem is akin to collaborative filtering over a rich domain in which the user possesses one of a variety of poorly understood preference criteria.

Finally, links from a concept page to an article page result may be produced in many ways, but it is natural to consider

p ⇓ q ⇒	Result	Concept	Article
Result	Assistance	Concept search	Vanilla search
Concept	Search w/in concept	Concept recommendation	Semantic linking
Article	–	Semantic linking	Related pages

Table 1: Technologies for Interconnecting Different Page Types

mining articles to understand references to records in a web of concepts. We refer to this class of analyses as *semantic linking*. Techniques such as named entity recognition play a key role in this setting. One should imagine that this capability produces a bipartite graph linking concept records to articles, and allowing users to pivot back and forth between the two. For instance, a user might pivot from a concept page about an actor to an article mentioning that actor in a particular series, and then might pivot again from the article to the concept page for the series. Concept recommendation, on the other hand, should be viewed as enabling edges on the left side of this bipartite graph, connecting concept pages to concept pages, as in the direction connection followed by the user above in linking from the concept page on Deadwood to the concept page on Timothy Olyphant.

The last row of the table covers links from article pages. We will not focus in this context on links from article pages to search results. Links from article pages to concept pages have been discussed above under semantic linking. Links between article pages represent the family of techniques to find related pages, typically based on document similarity functions, perhaps employing concept references as part of the feature vector characterizing an article.

5.5 Advertising

No discussion of a web of concepts would be complete without touching on the applications to advertising. These fall into two categories: matching and marketplace.

Matching. Given a characterization of a user through the lens of the web of concepts, an advertising system may select ads targeted to the concepts of interest to the user. A user involved in booking a vacation to Europe may be offered appropriate hotels, travel gear, clothing, online services, and so forth. A user with an interest in a particular sport might be offered tickets, paraphernalia, and so forth.

Marketplace. In addition to employing a web of concepts as a source of features to match ads appropriately, it is possible also to explore modifying the advertising marketplace based on concepts. For example, in the context of web search advertising, advertisers bid on keywords, but there might be a targeting advantage in bidding on concepts. If a pageview can be associated with a concept using the concept search capabilities described above, the proprietor of Birks Steakhouse, for instance, might place a bid on any query that hits on a restaurant in zipcode 95054. An advertiser of travel deals might be willing to pay to have a graphical ad shown on any page as long as the user in question is planning a trip to Europe. Users might then choose whether to make such information available to advertisers in order to receive more relevant advertising.

6. RELATED WORK

Our goal of creating structured metadata about information originally culled from web pages has an important consequence—*data integration now becomes a central concern for web information management*. Hitherto, the web was simply a collection of linked pages, each seen as a bag of words, and there was little or no semantic interpretation; in turn, there were no semantic mappings to be established and no discrepancies to be reconciled. Now, we might interpret a page on eBay as containing a list of items for sale, with each item described by attributes such as price, model, and category. Some of these items might also be described in an Amazon page, but with attribute names that differ, prices quoted in different currencies, and in general, all the differences arising in traditional structured data integration scenarios, e.g., [54, 10]. The problem of integration in our setting, however, is somewhat stylized and embedded in the extraction challenge. Our proposed approach is essentially to build machine-learned models and is closer to the approach taken in [26, 55]. However, we seek to train a model per concept of interest, or possibly for a set of related concepts in a given domain, rather than per site. We always seek to extract structured data for a given concept, and anticipate that most records will have values for one or more of a set of target attributes associated with the concept. Thus, we have to learn mappings from several source schemas, but the target schema is fixed (at least, by the time our models are trained—we might learn about attributes of relevance to our target concepts by exploring data from several websites!). Our extraction models are trained on the basis of common structural and linguistic cues found to be correlated with instances of these attributes on relevant pages (and of course, learning which pages are relevant is part of the training for extraction models). We also believe that it is essential to exploit markup and other contextual cues on web pages, over and above the key-value pairs we seek to extract [41].

The problems of identifying which pieces of information pertain to the same concept is a variant of the well-studied *entity matching (EM)* problem, also known as record linkage, deduplication, object reconciliation, and the merge/purge problem. Initial approaches to EM focused on pairwise attribute similarities between entities. Newcombe [52] and Fellegi and Sunter [31], gave the problem a probabilistic foundation by posing EM as a classification problem (i.e., deciding a pair to be a match or a non-match) based on attribute-similarity scores. The bulk of follow up work on EM then focused on constructing good attribute-similarity measures (e.g., using approximate string-matching techniques) [51, 20, 17]. Entity matching can be significantly improved by using relational information in addition to attribute similarities [2, 43], e.g., co-authorship and citation graph analysis can significantly improve the resolution of author references. Collective approaches take the relational approaches

a step further, and collectively make all the matching decisions. Collective approaches are either iterative [12, 29], where matching decisions trigger new matches, or use various advanced probabilistic models, e.g. Conditional Random Fields (CRFs) [47, 28], relational Bayesian networks [53], latent Dirichlet models [11, 38], and Markov Logic Networks [59].

Halevy et al. [37] observe that in many data integration scenarios today, all the data cannot be fit into any one data model or repository, and it is necessary to support unified retrieval from collections of loosely connected and independently managed sources. They propose that we should develop a new approach to providing integrated access to such collections. The approach, which they call *dataspace management systems*, aims to allow a baseline of access to all sources in the collection (e.g., keyword search), with additional capabilities enabled as sources are more completely integrated. They observe that the challenges in building such a system include dealing with uncertainty in how data descriptions (e.g., schemas) in one source map to other sources, providing best-effort query processing in the face of these uncertainties, combining structured and unstructured data, etc. The goal of realizing a web of concepts shares many of these challenges, and some of these philosophies (in particular, best-effort integration with collaborative, incremental refinement, and usability of partially integrated data), but also differs in several respects. First, we have concentrated on how to create a (logically) centralized and unified store that serves as the basis of query processing. As we observed in Section 4, however, independently created structured web corpora must be integrated into this unified store; we seek to do this through extraction. Over time, we expect independent, community-driven mass-collaboration to play a significant role by creating and maintaining high-quality integrated concept-centric repositories in many domains [57]. Leveraging these leads to corresponding improvements in the unified web of concepts. Thus, we look for the same "pay as you go" characteristic sought in the Dataspaces approach. Second, our focus is primarily on web pages, and less on heterogeneous formats such as spreadsheets or on integration of tables in relational databases. We are thinking of how to build the next generation of search engines, e.g., as in [42, 6, 34]. This is not to say that spreadsheets cannot be found or the web, or that relational databases are not exposed through form interfaces and dynamically constructed pages in various ways. Rather, our goal is to consider such data only insofar as it is exposed through markup structure and other cues that give us a way to interpret it with minimal human intervention.

Finally, our goals are closely related to the semantic web [9], and we see the two approaches as synergistic. Indeed, Yahoo! has a program called SearchMonkey aimed at enabling site publishers to provide more structured content, and they provide guidelines for creating many concept schemas (e.g., businesses, job postings, media items, products) using semantic web terminology: `http://developer.yahoo.com/searchmonkey/smguide/profile_vocab.html`. Our emphasis is on taking what exists on the web today and interpreting it and enabling richer applications (in particular, search), whereas the semantic web approach is to empower authors to publish content in a more interpretable form.

7. RESEARCH CHALLENGES

The goal of this paper is to highlight an emerging research area at the intersection of database systems, machine learning, and web search, and thus far, we have (we hope) provided a general overview along with some motivation and context. In this section, we try to outline several key technical challenges in constructing, maintaining and using a web of concepts.

7.1 Representation, Organization, and Maintenance

In Section 2.3, we discussed several open issues in how to extract, organize, and retrieve information in a concept-centric manner. There are several challenges with concept representation, organization, and maintenance. These issues obviously need to be resolved. We suggest that interested researchers tackle this set of issues by undertaking to build a concept-centric repository in a domain of their choice, similar to DBLife in the domain of database research, but with an eye to generalizability to other domains. In this arena, it is hard to come to grips without getting one's hands dirty with real data in the context of a specific goal. Fortunately, there is no difficulty in accessing data from a wide range of websites, and the problems should be readily accessible to researchers everywhere, not just those at the major web companies. We also suggest creating an open source initiative to pool the development of common software building blocks, and creating shared datasets and benchmarks.

7.2 Extraction Challenges

In Section 4.2, we discussed a number of issues in domain-centric extraction and described some of the research being pursued at Yahoo! In this section, we aim complement that discussion without repeating the same points.

A fundamental observation that we want to emphasize is that domain-centric extraction is central to realizing a web of concepts, because generating labeled data for every website or data source in order to learn wrappers or probabilistic models is clearly not feasible. Therefore, we need techniques that require minimal supervision yet can work at a domain level. In Section 4.2, we described two approaches in this direction: (i) combining wrapper-based methods with domain knowledge to achieve unsupervised extraction, and (ii) bootstrapping, i.e., using already extracted records to automatically label and extract more records. Ideally, we want to leverage/reuse extraction efforts as much as possible across sources. For instance, suppose we produce sufficient labeled data to develop a good extractor for restaurant location from `yelp.com`. Then, even if the extractor cannot be directly applied to restaurants in `citysearch.com`, we should not require the full efforts to develop a new extractor. An area of research that has a huge potential in this direction is transfer learning [3, 56].

The second challenge in this category is how to link records better. One of the key features of a web of concepts is the possibility to aggregate information about records from multiple sources. This requires the ability to link co-referent records. The problem of matching two structured entities is well studied in the literature [12, 59, 53]; the techniques, however, tend to be source-centric and not domain-centric. Furthermore, the current notion of linking two structured entities needs to be extended to include the following: matching a record against a text fragment (e.g., matching reviews

to restaurants) and matching a text fragment to a record (e.g., identifying that a name mention in a blog posting refers to a given person record). This requires combining traditional entity matching techniques with those from information retrieval and named-entity recognition.

7.3 Managing Information Extraction: Uncertainty, Noise, and Change

Building a web of concepts is not a one-time affair; we will need to constantly crawl the web and redo the extraction and organization of extracted information in order to maintain the derived web of concepts. Managing this process is a fundamental and novel challenge at the intersection of traditional database management and information extraction [27]. Building a web of concepts will be an inherently noisy process since several operators such as classifiers, segmenters, information extractors, and entity matchers produce probabilistic/uncertain output. Thus, for quality assessment, we need to track the uncertainty in the extracted records as data flows through various operators. In addition, web content varies a lot in its reliability, and often contains outdated and even contradictory information. Thus, the extracted information will often be inconsistent and will need to be reconciled to meet integrity constraints. Dealing with uncertain and inconsistent information is an active area of current research [24, 61, 32], and applying it to a web of concepts will be a challenging research problem.

Managing lineage, i.e., keeping track of the documents and the sequence of operators that result in a given extracted record, is an important problem in managing a web of concepts. Lineage is important for two reasons. The first is to improve the quality of extraction and track errors. For instance, if one of the records has an error, it might be because the extractor failed or it might be because the page classifier prior to the extractor misclassified. Keeping track of lineage helps us pinpoint the locations of errors and effectively use feedback to retrain operators. The second important use of lineage is to provide explanations to user queries. Presented with a piece of extracted information, the user might want to look at the documents or fragments of documents used to construct the information. Lineage for traditional databases is another active area of research [14, 8, 19], but managing lineage for various extractors and linking algorithms in a web of concepts is a research challenge.

The web, as we know it, is highly dynamic. New content gets added every moment and existing content keeps getting updated—restaurants close down, move to a new location, or change phone numbers, researchers publish new papers or change their affiliations, and so on. There is an obvious efficiency challenge in processing the same web pages repeatedly without re-incurring the full cost of extraction when the page is not modified in a material way [18]. A less obvious but equally important issue is that we must develop extraction techniques that work robustly in the face of such change [50, 22]. We also need to maintain a web of concepts whose content tracks the changing web of documents. There are several research issues that arise in this context, some of which overlap with the issues of linking and managing inconsistencies. When we process new or updated documents, we need to link them to the existing records to correctly update existing records rather than create new ones. Also, as new content gets disseminated on the web, inconsistencies crop up with websites containing outdated information. Some concepts, like stock tickers and city temperatures, are so dynamic that they always need to be tied to their underlying source documents. A good overview of topic detection and tracking can be found in [1]. Several ideas from the line of research on view maintenance in databases [35] might also be useful in tackling these issues.

7.4 Application Challenges

On the application front, concepts bring in a fresh set of challenges in analyzing and interpreting user behavior. First of all, online user behavior constantly changes and evolves. Therefore, a static set of concepts and concept attributes might become obsolete quite rapidly. This bring about the question: how can user behavior in search and browsing be studied in order to extract the concepts and attributes that might be valuable to improving the user experience? This will involve formulating appropriate models for user behavior, together with techniques from information retrieval. It will be convenient to envision a general framework to do such analyses.

The second challenge is how to design and infer meaningful metrics for concept-driven search applications such as concept search (Section 5.2). There are two issues here. The traditional relevance notions developed in information retrieval may not be appropriate for concept search. The challenge is to take a holistic view of the result set, with concepts in mind. The second issue is how to measure the satisfaction of the users with concept-based results. Once again, traditional notions such as click-through rates may be inadequate and an aggregate notion of user satisfaction with respect to the concepts will be needed.

8. CONCLUSIONS

We live in exciting times. The nature of the web is changing, reflecting how people have come to view it and use it, and the value of organizing and delivering information on the web has never been greater. Technological developments in data management and machine learning, and sociological developments such as the demonstrated willingness of people to work together in online communities to create shared repositories of common value (e.g., Wikipedia) encourage us to dream big. The realization of an ambitious vision such as a web of concepts would transform how we obtain information from the web, and have huge social and financial impact. It is an opportunity for the research community, and in particular for the database research community, to make a major contribution in what we think might be the next big evolution in web information management. In this paper we have tried to share our thoughts on the technical problems involved in the hope that this will stimulate interest in these emerging and foundational problems.

Acknowledgments

We would like to thank colleagues at Yahoo! who have influenced our thinking on the issues discussed in this paper and are involved in many of the research challenges that we have mentioned: Mani Abrol, Vipul Agarwal, Arup Choudhury, David Ciemiewicz, Arun Iyer, Ankur Jain, Vinay Kakade, Alok Kirpal, Ashwin Machanavajjhala, Mridul Muralidharan, Rajeev Rastogi, and Cong Yu.

9. REFERENCES

[1] J. Allan. *Topic Detection and Tracking*. Kluwer Academic, 2002.

[2] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *VLDB*, pages 586–596, 2002.

[3] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853, 2005.

[4] T. Anton. Xpath-wrapper induction by generating tree traversal patterns. In *LWA*, pages 126–133, 2005.

[5] J. Atserias, H. Zaragoza, M. Ciaramita, and G. Attardi. Semantically annotated snapshot of the English Wikipedia. In *LREC*, 2008.

[6] H. Bast, A. Chitea, F. Suchanek, and I. Weber. Ester: Efficient search on text, entities and relations. In *SIGIR*, pages 671–678, 2007.

[7] R. Baumgartner, S. Flesca, and G. Gottlob. Visual web information extraction with Lixto. In *VLDB*, pages 119–128, 2001.

[8] O. Benjelloun, A. D. Sarma, A. Halevy, and J. Widom. ULDBs: Databases with uncertainty and lineage. In *VLDB*, pages 953–964, 2006.

[9] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[10] P. A. Bernstein and L. Haas. Information integration in the enterprise. *CACM*, 51(9):72–79, 2008.

[11] I. Bhattacharya and L. Getoor. A latent Dirichlet model for unsupervised entity resolution. In *SDM*, 2006.

[12] I. Bhattacharya and L. Getoor. Collective entity resolution in relational data. *TKDD*, 1(1), 2007.

[13] R. Brachman and H. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann, 2004.

[14] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. In *ICDT*, pages 316–330, 2001.

[15] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Webtables: exploring the power of tables on the web. *VLDB*, 1(1):538–549, 2008.

[16] C. Cardie. Empirical methods in information extraction. *AI Magazine*, 18(4):65–79, 1997.

[17] S. Chaudhuri, V. Ganti, and R. Motwani. Robust identification of fuzzy duplicates. In *ICDE*, pages 865–876, 2005.

[18] F. Chen, A. Doan, J. Yang, and R. Ramakrishnan. Efficient information extraction over evolving text data. In *ICDE*, pages 943–952, 2008.

[19] J. Cheney, P. Buneman, and B. Ludäscher. Report on the principles of provenance workshop. *SIGMOD Record*, 37(1):62–65, 2008.

[20] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IJCAI Workshop on Information Integration on the Web*, pages 73–78, 2003.

[21] V. Crescenzi, G. Mecca, and P. Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *VLDB*, pages 109–118, 2001.

[22] N. Dalvi, P. Bohannon, and F. Sha. Robust web extraction : An approach based on a probabilistic tree-edit model. In *SIGMOD*, 2009.

[23] N. Dalvi, R. Kumar, B. Pang, and A. Tomkins. Matching reviews with objects using a language model. In *Manuscript*, 2008.

[24] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB*, 16(4):523–544, 2004.

[25] P. DeRose, W. Shen, F. Chen, A. Doan, and R. Ramakrishnan. Building structured web community portals: A top-down, compositional, and incremental approach. In *VLDB*, pages 399–410, 2007.

[26] A. Doan, J. Madhavan, P. Domingos, and A. Y. Halevy. Ontology matching: A machine learning approach. In *Handbook on Ontologies*, pages 385–404, 2004.

[27] A. Doan, R. Ramakrishnan, and S. Vaithyanathan. Managing information extraction: State of the art and research directions. In *SIGMOD*, pages 799–800, 2006.

[28] P. Domingos. Multi-relational record linkage. In *KDD Workshop on Multi-Relational Data Mining*, pages 31–48, 2004.

[29] X. Dong, A. Halevy, and J. Madhavan. Reference reconciliation in complex information spaces. In *SIGMOD*, pages 85–96, 2005.

[30] 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: (preliminary results). In *WWW*, pages 100–110, 2004.

[31] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *JASA*, 64:1183–1210, 1969.

[32] A. D. Fuxman and R. J. Miller. First-order query rewriting for inconsistent databases. In *ICDT*, pages 337–351, 2005.

[33] R. Gilleron, F. Jousse, I. Tellier, and M. Tommasi. XML document transformation with conditional random fields. In *INEX*, 2006.

[34] M. N. Gubanov and P. A. Bernstein. Structural text search and comparison using automatically extracted schema. In *WebDB*, 2006.

[35] A. Gupta and I. S. Mumick. *Materialized Views: Techniques, Implementations, and Applications*. MIT Press, 1999.

[36] R. Gupta and S. Sarawagi. Creating probabilistic databases from information extraction models. In *VLDB*, pages 965–976, 2006.

[37] A. Y. Halevy, M. J. Franklin, and D. Maier. Principles of dataspace systems. In *PODS*, pages 1–9, 2006.

[38] R. Hall, C. Sutton, and A. McCallum. Unsupervised deduplication using cross-field dependencies. In *KDD*, pages 310–317, 2008.

[39] W. Han, D. Buttler, and C. Pu. Wrapping web data into XML. *SIGMOD Record*, 30(3):33–38, 2001.

[40] C.-N. Hsu and M.-T. Dung. Generating finite-state transducers for semi-structured data extraction from the web. *Information Systems*, 23(8):521–538, 1998.

[41] A. Jain, D. Kifer, A. Kirpal, S. Merugu, S. Keerthi, P. Bohannon, and R. Ramakrishnan. Concept-centric extraction: using domain knowledge and local learning. In *Manuscript*, 2008.

[42] T. S. Jayram, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. Zhu. Avatar information extraction system. *IEEE Data Engineering Bulletin*, 29(1):40–48, 2006.

[43] D. V. Kalashnikov, S. Mehrotra, and Z. Chen. Exploiting relationships for domain-independent data cleaning. In *SDM*, 2005.

[44] N. Kushmerick, D. S. Weld, and R. B. Doorenbos. Wrapper induction for information extraction. In *IJCAI*, pages 729–737, 1997.

[45] J. Madhavan, L. Afanasiev, L. Antova, and A. Y. Halevy. Harnessing the deep web: Present and future. In *CIDR*, 2009.

[46] A. McCallum. Information extraction: Distilling structured data from unstructured text. *Queue*, 3(9):48–57, 2005.

[47] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *NIPS*, 2004.

[48] R. McCann, A. Kramnik, W. Shen, V. Varadarajan, O. Sobulo, and A. Doan. Integrating data from disparate sources: A mass collaboration approach. In *ICDE*, pages 487–488, 2005.

[49] I. Muslea, S. Minton, and C. Knoblock. STALKER: Learning extraction rules for semistructured. In *AAAI: Workshop on AI and Information Integration*, 1998.

[50] J. Myllymaki and J. Jackson. Robust web data extraction with XML path expressions. Technical Report RJ 10245, IBM, 2002.

[51] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[52] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. andJames. Automatic linkage of vital records. *Science*, 130:954–959, 1959.

[53] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *NIPS*, 2002.

[54] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

[55] E. Rahm, A. Thor, D. Aumueller, H. H. Do, N. Golovin, and T. Kirsten. iFuice: Information fusion utilizing instance correspondences and peer mappings. In *WebDB*, pages 7–12, 2005.

[56] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML*, pages 759–766, 2007.

[57] R. Ramakrishnan and A. Tomkins. Toward a peopleweb. *IEEE Computer*, 40(8):63–72, 2007.

[58] A. Sahuguet and F. Azavant. Building light-weight wrappers for legacy web data-sources using W4F. In *VLDB*, pages 738–741, 1999.

[59] P. Singla and P. Domingos. Entity resolution with Markov logic. In *ICDM*, pages 572–582, 2006.

[60] S. Sundararajan and S. Keerthi. Graph based classification methods using inaccurate external classifier information. In *Manuscript*, 2008.

[61] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *CIDR*, pages 262–276, 2005.