Mining Semantics for Large Scale Integration on the Web: Evidences, Insights, and Challenges

Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang

Computer Science Department
University of Illinois at Urbana-Champaign
{kcchang, binhe, zhang2}@uiuc.edu

ABSTRACT

The Web has been rapidly "deepened"– with myriad searchable databases online, where data are hidden behind query interfaces. Toward large scale integration over this "deep Web," we are facing a new challenge– With its dynamic and ad-hoc nature, such large scale integration mandates *dynamic semantics discovery*. That is, we must *on-the-fly* cope with "semantics" of dynamically discovered sources without pre-configured source-specific knowledge. To tackle this challenge, our initial works hinge on the insight that the large scale is itself also a unique opportunity: We observe that the desired "semantics" often connects to surface presentation characteristics, through some hidden regularities over many sources. Such regularities can be essentially leveraged in enabling semantics discovery. In particular, we report our evidences in three initial tasks for integrating the deep Web: *interface extraction*, *schema matching*, and *query translation*. Generalizing these specific evidences, we thus propose our "unified insight" of "mining" semantics for large scale integration by exploiting hidden regularities across holistic sources. Further, to fulfill the promise of such holistic mining, we discuss challenges toward its realization for dynamic semantics discovery. As our initial works as well as several related efforts have witnessed, we believe our unified insight, holistic mining for semantics discovery, is a promising methodology toward enabling large scale integration.

1. INTRODUCTION

Recently, the Web has been rapidly deepened with the prevalence of databases on the Internet. As Figure 1 conceptually illustrates, on this so-called "deep Web," numerous online databases provide dynamic query-based data access through their query interfaces, instead of static URL links. A July 2000 study [1] estimated 43,000-96,000 such search sites (and 550 billion content pages) on the Web. Our recent survey [2] in April 2004 estimated 450,000 online databases. As current crawlers cannot effectively query databases, such data are invisible to search engines, and thus remain largely hidden from users.

However, while there are myriad useful databases online, users often have difficulties in first *finding* the right sources and then *querying* over them. Consider user Amy, who is moving to a new town. To start with, different queries need different sources to answer: Where can she look for real estate listings? (*e.g., realtor.com.*) Studying for a new car? (*cars.com.*) Looking for a job? (*monster.com.*) Further, different sources support different query capabilities: After source hunting, Amy must then learn the grueling

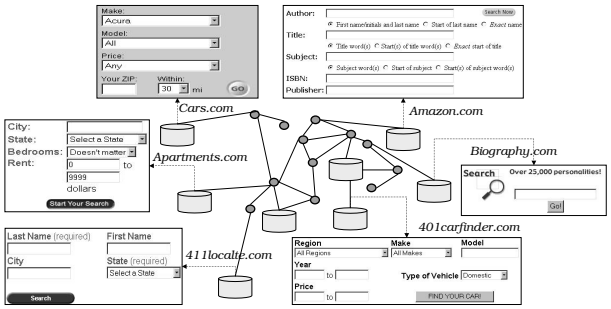


Figure 1: Databases on the Web.

details of querying each source.

To enable effective access to databases on the Web, it is critical to integrate these large scale deep Web sources. Such deep Web integration brings a new challenge of *on-the-fly* discovering integration-related semantics (*e.g.*, source query capabilities, semantic correspondences of attributes) without pre-configured source-specific knowledge. To tackle this challenge, our "thesis" of solutions builds upon the observation that databases on the Web are not arbitrarily complex– There seem to be some "convergence" or "regularity" naturally emerging across many sources. This "concerted complexity" sheds light on pursuing a "holistic mining" paradigm for discovering semantics dynamically. This paper presents evidences, insights and challenges of such semantics mining for large scale integration– as learned from our initial experience in building a "MetaQuerier"[1] for exploring and integrating the deep Web.

In particular, toward large-scale integration, we are facing new challenges. For coping with the *large scale*: The deep Web is a large collection of queryable databases (well on the order of 10^5, as mentioned earlier). As the large scale mandates, first, such integration is *dynamic*: Since sources are proliferating and evolving on the Web, they cannot be statically configured for integration and consequently must be dynamically discovered for integration. Second, it is *ad-hoc*: Since queries are submitted by users for different needs, they will each interact with different sources– *e.g.*, in Amy's case: those of real estates, automobiles, and jobs. As queries are ad-hoc, we must mediate them on-the-fly for relevant sources, with no pre-configured source-specific knowledge.

While the need is tantalizing– for effectively accessing the deep Web– the order is also tall. The challenge arises from the mandate of on-the-fly *semantics discovery*: Given the dynamically-discovered sources, to achieve on-the-fly query mediation, we must cope with various "semantics." To name a few: *What are the query capabilities of a source*? (So as to characterize a source and query it.) *How to match between query interfaces*? (So as to mediate

[1] *metaquerier.cs.uiuc.edu*

queries.) While the challenge of semantics is not new to any information integration effort, for smaller and static scenarios, automatic semantics discovery is often an *option* to reduce human labor, as an aid to manually configured semantics (*e.g.*, source descriptions and translation rules). In contrast, for large scale scenarios, semantics discovery is simply a *mandate*, since sources are collected dynamically and queried on-the-fly.

As our critical insight, while the large scale presents new challenges, we believe it also reveals itself as novel opportunities. In particular, we conducted a survey of the deep Web [2] by exploring about 500 sources in eight domains, *e.g.*, Books, Airfares. The survey revealed some inspiring observations: Databases on the Web are *not* arbitrarily complex; there seem to be some "convergence" and "regularity" *naturally* emerging across many sources. This "concerted complexity" sheds light on the challenge of dynamic semantics discovery. (So, it is perhaps hopeful to achieve large scale metaquerying.) In hindsight, such behavior is indeed natural at a large scale: As sources proliferate, they tend to be influenced by peers– which we intuitively understand as the *Amazon effect*.[2]

To begin with, as motivating evidences, our initial works for integrating the deep Web have essentially built upon this very insight (Section 2). First, *interface extraction*: For solving the problem of automatically extracting attributes from a query interface in HTML format, we introduce a *parsing* paradigm by hypothesizing the existence of *hidden syntax*, which describes the layout and semantics across query interfaces [20]. Second, *schema matching*: To discover semantic correspondences among attributes, we propose a holistic matching approach by matching all the schemas at the same time with the hypothesis of a *hidden schema model*, which guides the generation of schemas[7; 8]. Third, *query translation*: To translate queries between two query interfaces, we develop a *type-based search-driven* translation framework by observing the existence of *hidden localities* among query patterns [19].

To generalize, as a unified insight, and as the main thesis of this paper, we propose a new "philosophy" as a generic approach for integration at a large scale: *Holistic mining for semantics discovery*, as Figure 2 conceptually shows. Consider an integration task that requires discovery of semantics. To begin with, our philosophy builds upon two hypotheses: First, *shallow observable clues*: The desired "underlying" semantics often connects (Figure 2, top "$\mathcal{S}$") to the "observable" presentations, or shallow clues. Second, *holistic hidden regularities*: Such connections often follow some implicit properties, or hidden regularities (Figure 2, middle "$\mathcal{H}$"), which will reveal holistically across many sources.

Therefore, our integration task, or the discovery of the desired semantics, is naturally the *inverse* of this semantics-to-presentations connection: We thus propose to tackle with large scale integration by developing such "reverse analysis" (Figure 2, bottom) which holistically "mines" the shallow clues, as guided by the hidden regularity, to discover the desired semantics. As Section 3 will discuss, our three initial evidences in Section 2 can all be viewed as materializations of this holistic mining framework.

In our development, we also observe some challenges in pursuing such a mining approach for dynamic semantics discovery. In particular, what are the new "meta-mining" and "mining" issues arisen in holistic integration? How to deal with noises in input data, *i.e.*, "presentation clues," to be mined? What if the (often-hypothetical) hidden regularity cannot precisely capture the characteristics of observations? Beyond exploring hidden regularities with a mining approach, are there other ways to exploit "large scale" in holistic integration? Section 4 discusses these further challenges toward realizing holistic mining for large scale integration.

[2] Online bookstores seem to follow *Amazon.com* as a de facto standard.

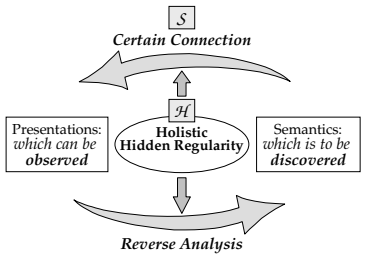


Figure 2: Unified insight: Holistic mining for semantics discovery.

We start with Section 2 for our evidences of exploiting "hidden regularity" for coping with semantics. We next, in Section 3, present our unified insight of the "holistic mining" paradigm for semantics discovery toward large scale integration. Section 4 raises some challenges and issues in pursuing such a mining approach. Section 5 reviews related work and Section 6 concludes.

2. EVIDENCES: SAMPLE TASKS

We now report three initial tasks, *interface extraction*, *schema matching* and *query translation*, as the key components for realizing large scale integration of the deep Web. For each task, we briefly summarize its functionality, motivate the essential insights, and present the specific approach. As we will see, these tasks, while different in their specific problems and solutions, are themselves "evidences" of exploiting certain hidden regularities for semantics discovery.

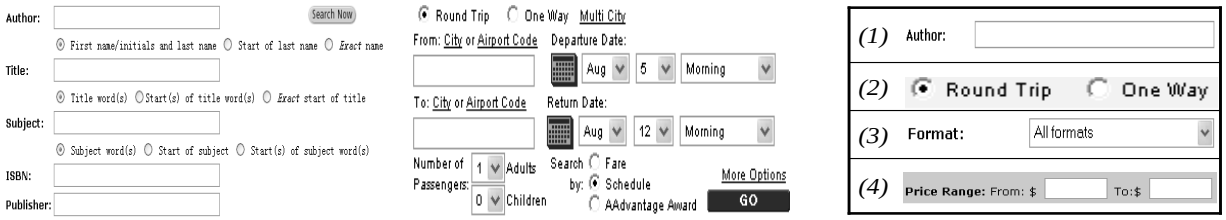
2.1 Task 1: Interface Extraction

For integrating Web databases, as the very first step, we studied the problem of *interface extraction*[20] - to "recognize" the basic condition templates presented in query interfaces.

A query interface essentially represent *query capabilities* that a source supports through its interface, as templates of specifiable query conditions. For instance, *amazon.com* (Figure 3(a)) supports a set of five condition templates (on author, title, . . ., publisher). Such query condition templates establish the target *semantics* underlying a Web query interface that our task seeks to discover.

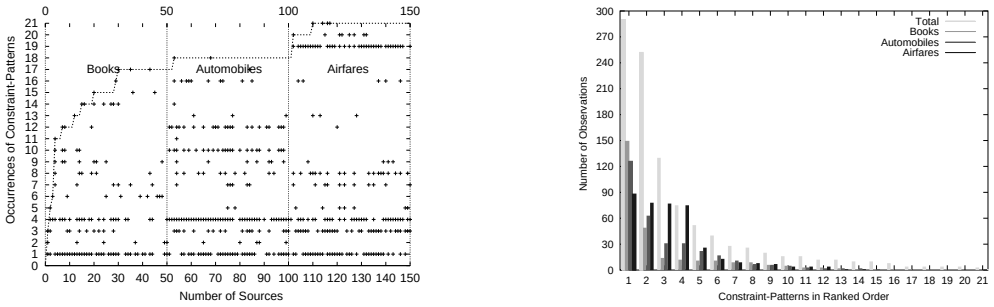
Such form extraction essentially requires both *grouping* elements hierarchically (*e.g.*, the condition template about author in *amazon.com* is a group of 8 elements: a *text* `"author"`, a *textbox*, three *radio buttons* and their associated *text*'s) and *tagging* their semantic roles (*e.g.*, `"author"` has the role of an *attribute* and the textbox an *input domain*.) The tasks are challenging – it seems to be rather "heuristic" in nature with no clear criteria but only a few fuzzy *heuristics*, as well as *exceptions*. *First*, grouping is hard, because a condition is generally n-ary, with various numbers of elements nested in different ways. ([*heuristics*]: Pair closest elements by spatial proximity. [*exception*]: Grouping is often not pairwise.) *Second*, tagging is also hard– There is no semantic labelling in HTML forms. ([*heuristics*]: A text element closest to a textbox field is its attribute. [*exception*]: Such an element can instead be an operator of this or next field.) *Finally*, with various form designs, their extraction can be inherently confusing.

Insight: We observe that query interfaces, although presented differently, often share similar or common query patterns. For instance, as Figure 3 shows, *amazon.com* has five condition templates (on author, title, *etc.*) and *aa.com* nine (on from, to, *etc.*). These condition templates seem to share some common "patterns": Those *template patterns* present condition templates in certain visual ar-



(a) Query interface Q_{am}: *amazon.com.* **(b)** Query interface Q_{aa}: *aa.com.* **(c)** Examples of condition template patterns.

Figure 3: Query interfaces examples.



(a) Growth of condition template patterns over sources. **(b)** Frequencies over ranks for condition template patterns.

Figure 4: Regularity of query condition templates across sources.

rangement (or layout)– Figure 3(c) shows several examples. For instance, pattern 1 represents a common format for condition templates of the form [attribute; *contain*; text], by arranging attribute to be followed by a textbox. Such condition templates represent keyword search (by an implicit *contain* operator) on a textual attribute (*e.g.*, author).

To understand the "complexity" of such template patterns, we conducted a survey (with more details available at [2]). In particular, we explore 150 deep Web sources in three domains - Books, Automobiles, and Airfares - from the TEL-8 dataset at The UIUC Web Integration Repository[3]. (TEL-8 dataset contains about 500 deep Web sources on eight domains.) Our survey finds that the template patterns in those query interfaces reveal some concerted structure. We find only 25 template patterns overall– which is surprisingly small as a vocabulary for online queries. As just mentioned, Figure 3(c) shows several frequently-used patterns. The distribution is extremely non-uniform: Figure 4(b) ranks the patterns according to their frequencies (and omits 4 rare attributes in the tail, which occur only once in 150 sources), for each domain and overall. We observe a characteristic Zipf-distribution, which confirms that a small set of top-ranked patterns will dominate.

We also observe the convergence behavior, both within and across domains. Figure 4(a) summarizes the occurrences of patterns. (To simplify, it similarly omits the rare "only-once" patterns.): The figure marks (x, y) with a "+" if pattern y occurs in source x. As more sources are seen (along the x-axis), the growth (along y) slows down and thus the curve flattens rapidly. Further, we observe that the convergence generally spans across different domains, which indicates that most template patterns are quite generic and not domain specific.

Such observation motivates us to hypothesize the existence of a *hidden syntax* across holistic sources. That is, we rationalize the concerted structure by asserting the creation of query interfaces as guided by some hypothetical syntax: The hypothetical syntax guides a syntactic composition process from condition templates to their visual patterns. This hypothesis effectively transforms the problem into a new paradigm: We can view query interfaces as a *visual language* [13], whose composition conforms to a hidden, *i.e.*, *non-prescribed*, grammar. The extraction of their semantics, as the reverse, is thus a *parsing* problem.

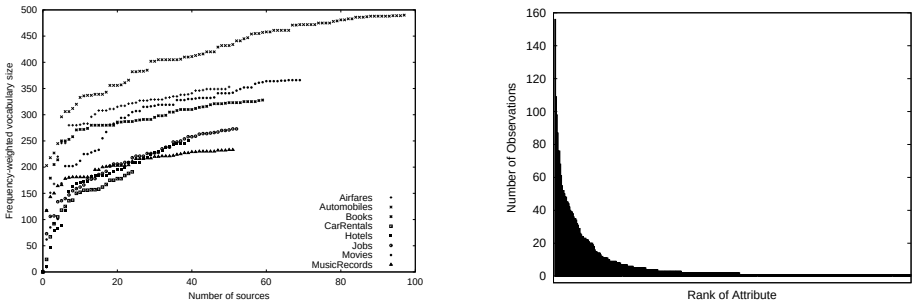
Approach: We thus introduce a *parsing* paradigm by hypothesizing that there exists *hidden syntax* to describe the layout and semantic of query interfaces [20]. Specifically, we develop the interface extractor as a visual language parser, as Figure 5 shows. Given a query interface in HTML format, the interface extractor tokenizes the page, parses the tokens, and then merges potentially multiple parse trees, to finally generate the query capability. At its heart, we develop a *2P grammar* and a *best-effort parser*.

First, by examining many interfaces, a human expert summarizes and encodes two complementary types of presentation conventions as the 2P grammar. On one hand, we need to write *productions* to capture conventionally deployed hidden patterns. On the other hand, however, by capturing many patterns, some will conflict, and thus we also need to capture their conventional precedence (or "priorities") as *preferences*.

Second, to work with a hypothetical syntax, we develop our parser to perform "best-effort." As a non-prescribed grammar is inherently ambiguous and incomplete, we need a "soft parsing" semantic– The parser will assemble parse trees that may be multiple (because of ambiguities) and partial (because of incompleteness), instead of insisting on a single perfect parse. On one hand, it will prune ambiguities, as much as possible, by employing preferences (as in the 2P grammar). On the other hand, it will recognize the structure (by applying productions) of the input form, as much as possible, by maximizing partial results.

When there are multiple parse trees for the same query interface, we need an error handling mechanism to generate the final output. While the parser framework is rather generic, error handling is often application specific. As our "base" implementation, our "Merger" (Figure 5) simply merges all query condition templates covered in all parse trees, to enhance the "recall" (or coverage) of extraction.

2.2 Task 2: Schema Matching



(a) Growth of attribute vocabulary in each domain. **(b)** Frequencies over ranks for all the attributes.

Figure 6: Regularity of attribute vocabularies across sources.

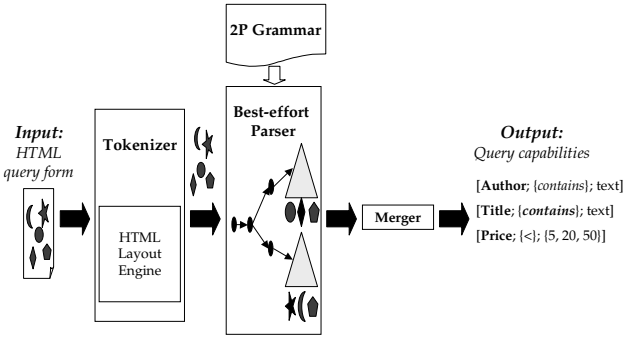


Figure 5: Form extractor for Web query interfaces.

To enable query translation across different sources, we studied the *schema matching* problem [7; 8] - to discover semantic correspondences of attributes across Web interfaces.

Schema matching is critical for mediating queries among deep Web sources. For instance, in Books domain, we may find subject is the synonym of category, *i.e.*, subject = category. In particular, we generally consider to discover complex matchings. In contrast to simple 1:1 matching, complex matching matches a set of m attributes to another set of n attributes, which is thus also called m:n *matching*. For instance, in Books domain, {author} = {first name, last name}; in Airfares domain, {passengers} = {adults, seniors, children, infants}.

While schema matching has been a central issue in data integration, the large scale sets new requirements on the matching task. Traditional schema matching works (*e.g.*, [16; 12; 5]) are developed for small scale and static integration scenarios, in which automatic matching technique is often an option to reduce human labor, as an aid to manually configured semantics. Schema matching under such scenarios is abstracted as finding pairwise attribute correspondences between two sources and thus cannot scale well. In contrast, in large scale data integration scenarios, the matching process needs to be as automatic as possible and scalable to large quantities of sources, as the large scale mandates.

Insight: We observe that the aggregate vocabulary of attributes in the same domain are not arbitrarily large - they tends to converge at relatively small size. To understand the complexity of such "schema vocabulary," we once again conduct a survey using the TEL-8 dataset as we used for *interface extraction*. (More details about the survey can be found at [2].) In particular, we surveyed all 400+ sources in eight domains. Figure 6(a) analyzes the growth of *frequency-weighted* vocabulary size for each domain. In particular, we weight the vocabulary growth by the "importance" of a new attribute– For the purpose of integration, an attribute that occurs in many sources will be more important. To quantify, let the *frequency* of an attribute be the number of sources in which it occurs. When counting the vocabulary size, each attribute is now weighted by its frequency in the corresponding domain. We see a very rapid convergence– In other words, as sources proliferate, their vocabularies will tend to stabilize. Note that the sources are sorted in the same order as they were collected without any bias.

In fact, the vocabularies will converge more rapidly, if we exclude "rare" attributes. To quantify, let the frequency of an attribute be the number of sources in which it occurs. Figure 6(b) orders these frequencies for all the attributes over their ranks. It is interesting but perhaps not surprising to observe that the distribution obeys the Zipf's law: The frequencies are inversely proportional to their ranks. Many low-ranked attributes thus rarely occur; in fact, 48% (203/422) attributes occur in only one source. Further, frequent attributes dominate: we observe that the top-20 attributes, or 4.7% (20/422) attributes, constitute 43% (1291/2992) of all the occurrences. What are the most "popular" attributes across all these sources? The top 5 frequent attributes are, in this order, title, keyword, price, make, and artist.

Approach: To tackle the challenge of large scale matching, as well as to take advantage of its new opportunity, we propose a new approach, *holistic schema matching*, to match many schemas at the same time and find all the matchings at once. Such a holistic view enables us to explore the *context* information across all schemas, which are not available when they are matched only in pairs.

In particular, we started by developing the MGS matching approach with the assumption of the existence of a hidden generative schema model, which generates query interfaces from a finite vocabulary of attributes [7]. Specifically, the observations of converging attribute vocabularies lead us to hypothesize the existence of a hidden generative model, which probabilistically generates, from a finite vocabulary, the schemas we observed. Intuitively, such a model gives the statistical properties that constrains how synonym attributes may co-occur across interfaces. The hidden generative model guides a statistic generation process from attribute correspondences (among the vocabulary) to their occurrences in interfaces. Given a set of query schemas as statistical "observations," schema matching is thus the discovery of such a hidden statistical model, which embeds attributes correspondence relationships.

To realize such hidden model discovery, we have proposed a general abstract framework, MGS, with three steps: (1) *Hypothesis modeling*: We first specify a parameterized structure of the hypothetical hidden models. Such models should capture the specific "synonym" semantics we want to discover. (2) *Hypothesis generation*: We then generate all "consistent" models that instantiate the observed schemas with non-zero probabilities. (3) *Hypothesis selection*: Finally, we select hypotheses that are consistent with the observed schemas with sufficient statistical significance.

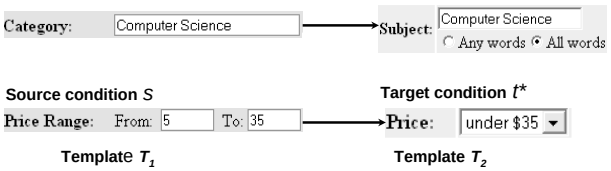


Figure 7: Examples of condition mapping.

In our further study, and in our current implementation, we explore the co-occurrence patterns of attributes to discover complex matchings [8]. For instance, we may observe that last name and first name have a high probability to co-occur in schemas, while they together rarely co-occur with author. More generally, we observe that *grouping attributes* (*i.e.*, attributes in one group of a matching *e.g.*, {last name, first name}) tend to be co-present and thus positively correlated across sources. In contrast, *synonym attributes* (*i.e.*, attribute groups in a matching) are negatively correlated because they rarely co-occur in schemas.

This observation motivates us to abstract the schema matching problem as correlation mining [8]. Specifically, we develop the DCM approach for mining complex matchings, consisting of automatic data preparation and correlation mining. As preprocessing, the data preparation step cleans the extracted query capabilities to prepare "schema transactions" for mining. Then the correlation mining step discovers complex matchings with dual correlation mining of positive and negative correlations.

2.3 Task 3: Query Translation

At the core of on-the-fly information integration, we studied the query translation problem [19] - to map a source query (which may be issued from a dynamically constructed unified query interface) to a target query form on-the-fly, *i.e.*, without manually crafted per-source knowledge pre-configured for individual sources.

As sources present different query capabilities, query translation, in essence, is to match and express queries in terms of such capabilities. As discussed in *interface extraction*, in general, query capability of a source presents templates of queries acceptable to the back-end database. To translate a query is thus to instantiate the target query template - by populating the parameters in the template with concrete values - into a target query which is semantically close to the source query.

As complex queries are built upon atomic conditions, translation eventually resorts to mapping between semantically related conditions, as discovered by *schema matching*. As Figure 7 indicates, given a specific *source condition* (*e.g.*, s = [price range; *between*; 5,35]), with respect to a matching *target template* (*e.g.*, T = [price; $\leq$; $val]), what is the closest mapping? In this case, condition mapping is to instantiate T into t^*=[price; $\leq$; 35] (assuming we want the target condition to minimally subsume the source condition), which best matches s, *i.e.*, $s \rightarrow t^*$ with respect to T . Figure 7 shows some example mappings.

With *interface extraction* syntactically recognizing the conditions, to enable query translation, the essential challenge is to really understand what a condition "means," *i.e.*, the subset of values, or *result range*, restricted by the condition. For instance, for condition [price; *between*; 5,35], its semantic meaning is to constrain the value of price in a range of $(5, 35)$. Similarly, [price; $\leq$; 35] constrained the value to $(0, 35)$. If we are able to understand such semantic meaning of each condition, we can compare the closeness of a target condition with the source condition in terms of the result ranges constrained by these two conditions. Therefore, query translation naturally becomes a search problem, *i.e.*, among all possible instantiations of the target condition template, query translation is to find the one which is semantically closest to the source condition.

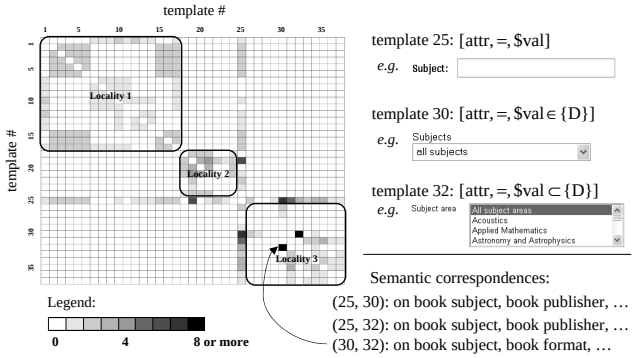


Figure 8: The correspondence matrix.

Insight: We observe that while conditions are presented differently in query interfaces, for expressing a particular semantic meaning on a concept, *e.g.*, a range of price $(0, 35)$, there are concerted ways across different interfaces. For instance, to express range $(0, 35)$, an interface may use template T_1 to form a condition [price; *between*; 0,35], while another interface may use T_2 to form [price; $\leq$; 35]. However, when looking at many interfaces collectively, we find that the "alternative" templates for expressing such semantics are rather limited.

To understand possible alternatives of expressing certain semantics, we conduct a survey on condition templates using the same dataset as we used for *interface extraction*, *i.e.*, query interfaces from Books, Airfares and Automobiles domains in the TEL-8 dataset. We notice that a condition template can be an alternative of another template only if there exist a concept (which may correspond to multiple semantically corresponding, *i.e.*, matching, attributes) that is expressed alternatively using the two templates in two different interfaces. For example, the two templates T_1 and T_2 are used to express the conditions on the same concept of book price (with possibly different attribute names such as price range and price), and thus they can be alternatives. We study the *alternative correspondences* between any two condition templates we collected in the survey. We use a *correspondence matrix* CM to report for any two condition templates (i, j), whether they have such alternative correspondence. In particular, $CM(i, j)$ denotes the number of concepts that are expressed using both templates i and j in different sources. Figure 8 shows our survey result (*i.e.*, the correspondence matrix), where the value of $CM(i, j)$ is illustrated as the degree of grayness at each cell (i, j).

¿From Figure 8, we observe that alternative templates form certain clusters or "localities"– That is, templates in a locality are often alternatives to each other, while templates across different localities cannot be used as alternatives to express the same semantic meaning. Further, we find that such correspondence localities are consistent with the notion of "data types." That is, while the condition templates in a locality are used by various concepts, those concepts often share the same data type. In particular, the first locality in Figure 8 corresponds to templates usually used by concepts of `datetime` type (*e.g.*, concept departure date, drop-off time), the second one by concepts of `numeric` type (*e.g.*, concept price, mileage) and the third one by concepts of `text` type (*e.g.*, concept author, title).

The observations of localities and their consistency with data types indicate that: For expressing a constrained subset of values, *e.g.*, price in range $(0, 35)$, there are only limited ways of presentation in query interfaces. The possible variations are restricted by the localities of alternatives or data types. For instance, to express price range $(0, 35)$, since it is of `numeric` type, we thus will use tem-

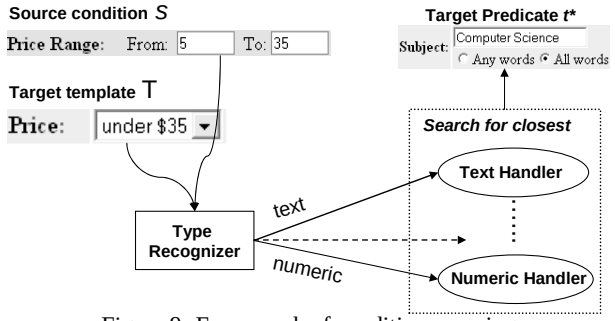


Figure 9: Framework of condition mapping.

plates *e.g.*, T_1 or T_2, in the corresponding "numeric locality."

As we further observe, data type provides a platform (which is not specific to any source) for simulating the "effects" of conditions by encoding the semantics, as constrained subset, of different operators with respect to the platform. For example, for numeric type condition, a line of real numbers is such a platform for simulating the semantics. Upon this numeric line, we can encode the meaning of operators, *e.g.*, *between*, $\leq$ or $\geq$ as partitioned ranges on the numeric line. For example, operator *between*(v_1, v_2) is encoded as a range from v_1 to v_2 on the numeric line. To understand a condition, *e.g.*, [price; *between*; 0,35], is simply to apply the encoded knowledge to simulate the result against the platform. Therefore, data type gives us a source-generic platform for encoding the semantic knowledge needed for query translation. Based on such knowledge, we are able to translate queries for those unseen sources, as long as it reuses the templates in the type-based template localities.

Approach: To translate a condition, we develop a search-driven approach. That is, among all possible instantiations of the target template, we search for the one, which is closest to the source condition. Figure 9 gives an overview of the condition mapping machinery. It takes a source condition s and a matching target condition template T as input, and outputs the closest target translation t^* for s. Specifically, starting from source condition s and target condition template T, the *type recognizer* first recognizes the type of the two conditions by exploiting their syntactic features (*e.g.*, the *from-to* pattern for numeric type, the $\leq$ operator, the values in the target template domain, *etc.*). It then further dispatches the conditions accordingly to the *type handler*. The *type handler* encodes our type-specific knowledge for understanding the semantics of query conditions. Based on such understanding, it performs search in all possible instantiations of T to find the closest mapping t^* as the output of the translation.

3. UNIFIED INSIGHT: *HOLISTIC MINING FOR SEMANTICS DISCOVERY*

Toward building the MetaQuerier system, we are inspired to observe that there seem to emerge common insight across several integration tasks. While we have developed these tasks (Section 2) separately, each with its specific techniques, as we put them together, they seem to share the same methodology, which reveals a common insight that conceptually unifies the seemingly different approaches. This section discusses this *methodology*, "holistic integration," and its underlying unified *insight*, "mining for semantic discovery."

To begin with, as Section 1 motivated, we note that any integration task is, to a large extent, about *semantics discovery*– to discover certain target *semantics*: *e.g.*, for task *interface extraction*: "extracting" query conditions; for *schema matching*: "matching" these query conditions across different interfaces; for *query translation*: "understanding" each condition. The major barrier for large scale integration, with its dynamic and on-the-fly nature, is exactly such semantics discovery, for the lack of pre-configured per-source knowledge.

As a shared "methodology," for such large scale integration, to tackle the very challenge of semantics discovery, our solutions have essentially assumed the same holistic integration framework. By *holistic integration*, we take a holistic view to account for *many* sources together in integration, by globally exploiting "clues" across all sources for resolving the "semantics" of interest– To our surprise, although not obvious by their own, when put together, many of our integration tasks implicitly share the same holistic-integration framework– which thus conceptually "unifies" our various techniques.

As a hindsight, we thus "propose" holistic integration as a conceptually unified methodology for large scale integration. As evident from our experience (albeit limited), we believe such holistic-integration is promising: It is intriguing to observe, as we are inspired, that the "challenge" of large scale can lend itself as a unique "opportunity" to solve integration tasks– The essence of holistic integration hinges on seeing not only the "tree" of each source individually but also the "forest" of many sources as a whole. That is, it will explore holistic clues (as we will see) across many sources as a "community" to take advantage of the large scale (with sufficient "samples"). Such "holistic" approaches will likely be essential for large scale integration tasks.

In particular, such holistic integration, as a common methodology, reveals several interesting "underlying principles," for enabling semantics discovery. In particular, in its materializations for various tasks, we have observed that holistic integration can resort to, among others, *hidden regularity*– as an enabling principle of the solutions. (As a related note, while not a focus of this paper, we have also observed the use of "peer majority," as we report in [4].) Intuitively, holistic integration can leverage "hidden regularity" across many sources, to discover the desired semantics– For our various integration tasks, *interface extraction* exploits hidden "syntax," *schema matching* hidden "schema model," and *query translation* hidden condition "localities," as we will explain.

Thus, as the main thesis of this paper, by holistic integration, we propose to explore "hidden regularity" existing across sources– which leads to a unified insight of "mining" as a main resort for semantics discovery. As just discussed, any integration task is essentially the discovery of certain target semantics– but, we can only observe some "surface" *presentations*. As a unified principle, several of our tasks– in particular, *interface extraction*, *schema matching*, and *query translation* as Section 2 reported– have exploited hidden regularities of surface presentations for semantics discovery. In retrospect, as Figure 2 conceptually illustrates, we observe that, under the same holistic-integration spirit, these tasks have built upon two common hypotheses, which relate underlying semantics to observable presentations, across many sources.

($\mathcal{S}$) *Shallow observable clues:* The "underlying" semantics often relates to the "observable" presentations, or shallow clues, in some way of *connection*. Thus, we can often identify certain observable clues, which reflect the underlying semantics.

($\mathcal{H}$) *Holistic hidden regularity:* Such connections often follow some implicit properties, which will reveal holistically across sources. Thus, by observing many sources, we can often identify certain hidden regularity that guides how the semantics connects to the presentations.

These hypotheses shed light for dynamic semantics discovery– To tackle this main challenge in large scale integration, we take the

approach of *holistic mining* of shallow presentations across many sources to uncover underlying semantics: By identifying the holistic regularity, our integration task, to discover the desired semantics, is thus the "inverse" of this semantics-to-presentations connection. Our "holistic integration" framework can then develop some *reverse analysis*, which holistically analyzes the shallow clues, as guided by the *hidden regularity*, to discover the desired semantics– Overall, holistic integration is thus translated to holistic "mining" of observable presentations for underlying semantics.

While a unified insight, as we will see, such holistic mining can materialize into different concrete solutions, depending on the specific integration task at hands. That is, to specifically realize such holistic mining, as the dual hypotheses dictate, our holistic mining framework must address two "enabling" questions for making the framework possible:

$\mathcal{M}1$: As a "meta-mining" issue, for the semantics-to-presentations connection, *what is the hidden regularity?* (Does it even exist?) Such regularity, if identified, will guide the reverse analysis for mining the semantics.

$\mathcal{M}2$: As a more traditional "mining" issue, with the regularity identified, *what is the reverse analysis?* That is, what reverse analysis shall serve as our "mining" technique?

With this conceptual framework (Figure 2), Section 3.1 will next present how it "unifies" our example tasks with their specific techniques, and Section 3.2 then discusses if this insight can generalize beyond our initial task studies.

3.1 Unification: Task Studies

This general "holistic mining" framework, as Figure 2 sketches, conceptually unifies our approaches for several tasks as its specific realizations. We now demonstrate with *interface extraction*, *schema matching*, and *query translation*– As Figure 10 contrasts, while addressing different problems with different techniques, these tasks are consistently unified under the same conceptual framework of holistic mining for semantics discovery.

Task 1: Interface Extraction

First, consider *interface extraction*: As Section 2 introduced, the observation of condition "patterns" motivated us to hypothesize the existence of *hidden syntax*– In our term now, this converging syntactical structure is the hidden regularity, which we identify across holistic sources. We thus rationalize the common condition layout patterns by asserting query-form creation, although at various autonomous sources, as guided by some hypothetical syntax: As Figure 10(a) shows, the hypothetical syntax (as *hidden regularity*) guides an interface composition process (as *connection*) from query conditions (as *semantics*) to their visual patterns (as *presentations*).

That is, in terms of our holistic integration framework, there exists a compositional connection (Hypothesis $\mathcal{S}$), and such connections at various sources share the same syntactical grammar as the regularity (Hypothesis $\mathcal{H}$). This syntactical interface composition behavior constrains how conditions may be arranged visually in interfaces– *e.g.*, attributes may be aligned with their input fields in certain ways. This hidden syntax effectively transforms the problem: The *reverse analysis* to find query conditions is thus the "mining" of syntactically connected components and thus, since we view query interfaces as a "visual language," a *visual-language parsing* approach.

Task 2: Schema Matching

Second, consider *schema matching*. As Section 2 introduced, we hypothesize a hidden generative behavior, which probabilistically generates, from a finite vocabulary, the schemas we observed– In our term now, this consistent generative behavior is the hidden regularity, which we identify across holistic sources. We thus rationalize the common attribute occurrence patterns by asserting query-schema creation, although at various autonomous sources, as guided by some hypothetical generative behavior. As Figure 10(b) shows, the hypothetical generative behavior (as *hidden regularity*) guides a schema generation process (as *connection*) from the matchings of attributes (as *semantics*) to their occurrences in schemas (as *presentations*).

That is, in terms of our holistic integration framework, there exists a generative connection (Hypothesis $\mathcal{S}$), and such connections at various sources share the same statistical behavior as the regularity (Hypothesis $\mathcal{H}$). This statistical schema generative behavior constrains how attributes may occur in schemas– *e.g.*, grouping attributes tend to positively co-occur while synonym attributes negatively. This hidden generative model effectively transforms the problem: The *reverse analysis* to find attribute matchings is thus the "mining" of statistical correlated attributes, and thus a *correlation mining* approach.

Task 3: Query Translation

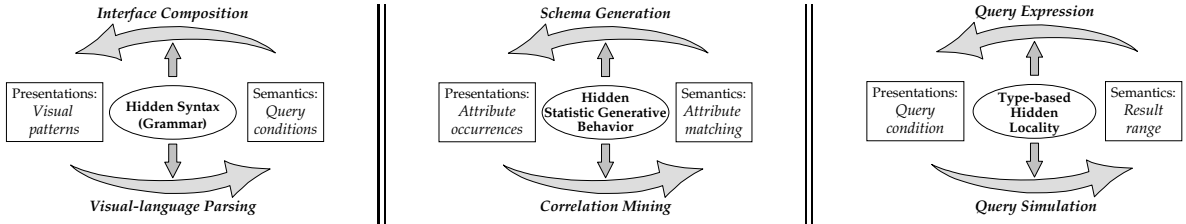
Third, consider *query translation*: As Section 2 introduced, the main challenge in on-the-fly query translation lies in *condition understanding*– the automatic "understanding" of a condition (as an instantiation of a condition template at the target query interface) for what it "means"– The "semantics" to discover is thus, for a given *query condition*, what is the subset of values, or the *result range*, that the condition constrains.

The observation of condition-correspondence "clusters" motivated us to hypothesize the existence of *hidden localities*, each of which consists of a small set of alternative condition templates for certain data type– In our term now, this type-based condition clustering is the hidden regularity, which we identify across holistic sources. We thus rationalize the common condition-correspondence patterns by asserting the expression of queries on a certain concept, although at various autonomous sources, as guided by some hypothetical condition locality: As Figure 10(c) shows, the hypothetical condition locality (as *hidden regularity*) guides a query expression process (as *connection*) from a result range (as *semantics*) to its query condition (as *presentations*).

That is, in terms of our holistic integration framework, there exists an expressional connection (Hypothesis $\mathcal{S}$), and such connections at various sources share the same localities of templates as the regularity (Hypothesis $\mathcal{H}$). This type-based query expression behavior constrains what templates may be used to construct query conditions– *e.g.*, numeric attributes may be constructed with $\leq$, $\geq$, or range selections. This hidden locality effectively transforms the problem: The *reverse analysis* to find the semantics of a query condition is thus the "mining" of its result range. Since we view a condition as constructed within a generic locality (which is not source specific), this understanding becomes a *type-based simulation* of the condition's effect on the intended type of data– *i.e.*, recognizing the right type and simulating within its understood scope of the locality.

Putting Together: Contrast and Unification

Putting together, we have seen that these semantics discovery tasks, while with rather different techniques, are conceptually unified in the same holistic mining framework (Figure 2). However, these concrete approaches have demonstrated different realizations of the two "enabling" questions:



(a) Interface extraction. **(b)** Schema matching. **(c)** Query translation (condition understanding).

Figure 10: Holistic mining for semantics discovery: Materializations in different integration tasks.

First, for $\mathcal{M}1$, we observed that the hidden regularities between semantics and presentations, although of the same nature, can take rather different forms. Even in our limited examples, in clear contrast, the spectrum ranges from *syntactical* regularities (of hidden grammar for interface composition and locality for query expression) on one hand to *statistical* ones (of hidden generative behavior for schema generation) on the other hand. However, they are of essentially the same nature– as a "creation" process connecting semantics to presentations– for our three tasks, that of creating interfaces, schemas, and queries respectively.

Second, for $\mathcal{M}2$, we observed that the reverse analysis or the "mining techniques," although naturally implied by the forward connection, can resort to rather different "mining" techniques: In our examples, in clear contrast, the spectrum ranges from more "standard" *correlation mining* techniques to the less traditional approaches of *visual parsing* and *simulation*. Nevertheless, they are all naturally implied as the appropriate "inverse" of their respective forward connections.

3.2 Generalization: Hypotheses Valid?

While we have "unified" from our three initial tasks, as a hindsight, the conceptual "holistic mining" framework for semantics discovery, will the same insight "generalize" to other tasks? As the heart of our unification, the dual hypotheses underpin the feasibility of the conceptual framework. In this section, we attempt to qualitatively argue that these hypotheses (and thus the desired generalization) are likely to hold, as the nature of Web presentations dictates. While our conjecture is informal at best, we hope to highlight intuitively the promise of holistic mining for large scale integration.

Thus, we ask– As the basis of our approach, *are the dual hypotheses valid?* First, for Hypothesis $\mathcal{S}$, are there certain presentation patterns as observable "clues" that connect to and reflect the desired underlying semantics? If such patterns emerge, this hypothesis will stand to guide our semantics discovery, by analyzing such clues. Further, for Hypothesis $\mathcal{H}$, even if such patterns exist, can we observe them consistently as a "regularity" across a large scale? If such patterns are indeed "holistic," this hypothesis can guide us to exercise the large scale of the Web to identify the regularity, which will then guide the reverse analysis.

Patterns are likely to emerge: First, we observe that such presentation patterns will often naturally emerge, for presenting certain underlying semantics.

As the nature of the Web dictates, to convey information to human users, its presentations will naturally show the following characteristics: One one hand, the presentations will be "meaningful"– to naturally bear the implications of the underlying semantics. In our examples, *e.g.*, for *schema matching*: As Section 2 discussed, attribute occurrences indeed connect to the "semantics"– The matchings (or the lack of) between attributes (*e.g.*, author and name) imply if they are mutually redundant, leading to their occurrences in a schema as observable clues. On the other hand, to be friendly to users, the presentations will be "intuitive"– to naturally appeal to users consumption of information. For instance, for *interface extraction*: To make query interfaces easy to use, query conditions (the underlying semantics) are often arranged in a table-like appearance– Thus the visual layout serves as such an observable clue. Overall, to bridge underlying semantics to users, as both ends demand, Web presentations are likely to exhibit meaningful and intuitive patterns.

Patterns are likely to converge: Further, we believe such patterns can often be consistently observed across many sources and thus surface as holistic patterns.

As the nature of the Web dictates, there are many good reasons for this conjecture: To begin with, first, the same "meaningfulness" upon which a pattern may emerge (as just mentioned) at a single source will hold true at every source (thus, for *schema matching*, author and name will not co-occur in general). Second, for "intuitiveness"– Since the Web has become our "ultimate information source," the need of *Web usability*[3] naturally advocates common design patterns that many sources will likely follow. For instance, for *interface extraction*: The analogy to ill-designed "forms" (*e.g.*, the infamous Florida "butterfly" ballots in US Election 2000) has generated discussions[4] on Web-form designs. Moreover, as the Web is a heavily interlinked community, as in any social network, "peer influence" will naturally forge the convergence of conventions. For instance, most personal homepages follow similar format or adopt uniform templates. In fact, on the Web, autonomous sources may even converge to de facto standards– *e.g.*, online bookstores seem to follow *Amazon.com* as a standard interface.

Overall, with the nature of the Web, we believe that presentation patterns will not only emerge (thus Hypothesis $\mathcal{S}$) but also converge (thus Hypothesis $\mathcal{H}$) across many sources holistically. For our example tasks, we have observed the hidden syntax, the generative behavior, and the condition localities– We stress that, although almost taken for granted now, these regularities were not obvious (as they were "hidden") to begin with; they were only revealed as we surveyed real Web sources, as Section 2 reported. We believe similar observations will surface in other tasks. We note that, as further evidences, several research efforts have also emerged recently to leverage such "holistic mining" insight for integration, as Section 5 will discuss.

4. CHALLENGES: ISSUES & AGENDA

[3] *e.g.*, www.useit.com/alertbox/20040913.html.

[4] *e.g.*, www.larrysworld.com/articles/ups_ballot.htm.

In our development of the holistic mining insight for semantics discovery, we also observed some open issues that warrant further research. These issues can be categorized into three aspects: the realization of the mining framework, the robustness of mining techniques, and the exploration of holistic insight.

The Realization of the Holistic Mining Framework:

As Section 3 discussed, to realize the holistic mining for semantics discovery (Figure 2), we need to address two enabling questions (*i.e.*, $\mathcal{M}1$ and $\mathcal{M}2$ as Section 3 discussed): *Meta-mining* for discovering the hidden regularity and *mining* for the semantics. Our evidences show that, for various integration tasks, the meta-mining phase may cover a wide range of different hidden regularities and consequently the mining phase may exploit different mining techniques, including non-traditional ones.

First, as key to the framework, meta-mining is difficult to automate– It is unclear how we can automatically discover the hidden regularity with respect to a specific integration task. In our evidences (Section 2), we, as human experts, observe the regularities of hidden syntax (for *interface extraction*), hidden generative model (for *schema matching*) and hidden locality (for *query translation*). It seems that, with current techniques, it is very difficult to automate such an "observation" stage.

While hard to automate, as we learned from our experience, such "meta-mining" suggests a more profound lesson– To see "hidden opportunities," it is imperative that we get to know the real data, by surveys and experiments, to gather a good grasp of the characteristics of our problems. In particular, as we decided to "get our hands dirty" early on, our survey [2] of the deep Web "frontier" essentially guided us in shaping our "meta-mining" insight.

Second, as our evidences show, the mining techniques for semantics discovery may exploit non-traditional approaches. To begin with, such integration tasks, as novel applications for data mining, may raise new issues even for existing techniques. For instance, in *schema matching*, our DCM approach translates finding complex matchings into correlation mining, a well-studied problem in data mining. However, in our scenario, we are more interested in negative correlations, which reflect the "synonym" attributes. As existing correlation mining works mostly focus on positive correlations, we need to address the new issue of developing a robust measure for negative correlations. Further, for many tasks, their mining techniques can go beyond existing mining abstractions– and are not even always "statistical." We have witnessed several such non-standard techniques: For *schema matching*: our MGS approach exploits hidden model discovery with hypothesis testing; for *interface extraction*, we resort to visual parsing as the mining technique– a syntactical rather than statistical approach.

The Robustness of Mining Techniques:

An essential difference between our mining for semantics and traditional data mining is the interpretation of the mining result. Traditional data mining does not require a strong connection between semantics and presentations, and consequently, the interpretation of mining result is rather subjective since there is no clear ground truth to measure the "accuracy" of the result (*e.g.*, when do we consider an association rule accurate?) In contrast, for integration scenarios, the mining result has clear semantic meaning and thus the accuracy matters (*e.g.*, whether an extracted query capability or discovered matching is correct). Thus, this demand of accuracy sets a "semantic" metric for measuring mining techniques. In particular, as our hidden regularity is often hypothetical in nature (*e.g.*, the hidden grammar for *interface extraction* is at best imaginary), the mining techniques must be robust against potential inconsistency between the actual observations and our hypotheses.

First, such inconsistency can result from noises in the observations. Such noises may be collected from source containing erroneous information or generated by a preceding integration task. For instance, the input of *schema matching* is in fact a set of extracted interfaces outputted by *interface extraction*. As an automatic process, *interface extraction* inevitably incurs errors, which make the input of *schema matching* noisy. It is thus critical to develop robust mining approaches that can sustain noisy observations. In [4], we propose an *ensemble* framework as a sample technique to cope with this noisy data problem.

Second, such inconsistency can also result from over-simplified modeling of a hidden regularity. Hypothetical in nature, the hidden regularity may not capture full data characteristics and thus the mining approach may be working under an "inconsistent" assumption. We can remove such an inconsistency by more complete modeling of the hidden regularity– For instance, for *interface extraction*, the visual grammar (as the "regularity") consists of not only production rules but also *preferences*– without which the parser cannot arbitrate between conflicting patterns, which will naturally arise as the syntax is only hypothetical.

Further Exploration of Holistic Insight:

In this paper, we propose a mining view to discover semantics by exploring holistic sources across the deep Web. We believe there are other ways to exploit this holistic insight. In particular, in [4], we propose *peer majority* as a new approach for "error correction," based on the same insight– In short, we can exploit the *majority* of peers for correcting errors made by relative *few*. Thus, while this hidden regularity-based mining approach is promising, as our experience extrapolates, we believe there are likely more novel opportunities for leveraging holistic sources as a principled solution for enabling integration.

5. RELATED WORK

Our study of large scale information integration has a distinct focus in terms of both the problem and the solutions. On one hand, as the problem, motivated by enabling large scale integration, our goal is to dynamically integrate numerous sources on the deep Web. On the other hand, as the solutions, our observations of hidden regularities lead to a "mining" paradigm for semantics discovery.

To begin with, information integration has traditionally focused on relatively small-scaled pre-configured systems [6; 17] (*e.g.*, Information Manifold [10], TSIMMIS [15], Clio [14]). In contrast, we are facing a "dynamic" and "ad-hoc" scenario (Section 1) of integrating at a large scale, for databases on the Web. Such a large scale integration imposes different requirements and thus faces its own challenges. In particular, to deal with this large scale, many tasks have to be automated as much as possible, unlike integration in a small scale where sources can be manually prepared. These challenges essentially boil down to the requirement of on-the-fly semantics discovery.

Further, to enable such semantics discovery, as our main thesis, we propose to exploit the "hidden regularities" revealed by holistic sources, which leads to a mining paradigm for semantics discovery. While the idea of leveraging the regularities for semantics discovery has also been observed in other research works, especially wrapper induction [11], their regularities are explicit "assertions" rather than implicit "hypotheses." The focus of those works is thus to "induce" such regularities from repetitive examples: *e.g.*, dynamically generated Web pages from an underlying "template." In a sense, we generalize the inductive spirit to large scale scenarios, where myriad sources, albeit *heterogeneous* and *autonomous*, exhibit certain hidden regularities. We thus propose the dual hypothe-

ses ($\mathcal{S}$ and $\mathcal{H}$; Section 3), upon which we develop holistic mining as a conceptually unified framework for large scale integration.

Toward such semantics discovery with hidden regularities, several research efforts, which also emerged recently, essentially share similar insights– but specifically for the *schema matching* task, which we have also studied in our context [7; 8] (as Section 2 reported). In particular, references [18; 9] exploit clustering for holistically matching many schemas. Reference [11] proposes a "corpus-based" idea, which uses a separately-built schema corpus as a holistic "knowledge base" for assisting matching of unseen sources.

While sharing similar holistic frameworks, in contrast to these efforts, we have developed our holistic mining insight to generally tackle with "semantics discovery" common in many large scale integration tasks, which we believe will generalize well beyond the specific task of *schema matching* (*e.g.*, *interface extraction* and *query translation*), as Section 3 discussed. Further, we believe, besides "statistical" analysis (which most other works have based upon), there are a wide range of applicable techniques (*e.g.*, syntactical parsing [20] for *interface extraction*) to generally explore holistic hidden regularities for semantics discovery.

6. CONCLUSION

Toward large scale integration on the Web, where the discovery of integration-related semantics is dynamic and thus necessarily on-the-fly, this paper proposes our "philosophy" of holistic mining for semantics discovery as a general approach. Motivated by the concerted-complexity observations of Web sources, our insights hinge on the hypotheses that the target semantics to be discovered often connects to certain shallow observable clues, in a way guided by some holistic hidden regularities across many sources. Integration is thus a reverse analysis, which holistically mines the shallow clues to discover the underlying semantics.

As concrete evidences, we have studied three different deep Web integration tasks, *interface extraction* [20], *schema matching* [7; 8], and *query translation* [19], each of which materializes holistic mining with syntactical or statistical approaches. Our experience indicates the promise of such techniques: For *interface extraction*, our experiment shows that the parsing approach achieves above 85% accuracy for extracting query conditions across randomly selected deep Web sources. For *schema matching*, our experiment in eight popular domains shows that both the MGS and DCM frameworks can achieve about 80-100% accuracy. For *query translation*, our experiment in three domains shows that the type-base search-driven translation framework can achieve about 90% accuracy.

Thus, as our experience (although limited) clearly suggests, such holistic approaches are well suited for the new frontier of large-scale networked databases in general and our focus of the deep Web in particular. As our key insight, in these settings, as sources proliferate, their aggregate complexity does not grow indefinitely– Instead, holistic hidden regularities often naturally emerge across sources. We thus propose "holistic mining for semantics discovery" as a general approach for leveraging the large scale challenge as an opportunity for new integration techniques.

7. REFERENCES

[1] M. K. Bergman. The deep web: Surfacing hidden value. Technical report, BrightPlanet LLC, Dec. 2000.

[2] K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang. Structured databases on the web: Observations and implications. *SIGMOD Record*, 33(3):61–70, Sept. 2004.

[3] K. C.-C. Chang, B. He, C. Li, and Z. Zhang. The UIUC web integration repository. Computer Science Department, University of Illinois at Urbana-Champaign. http://metaquerier.cs.uiuc.edu/repository, 2003.

[4] K. C.-C. Chang, B. He, and Z. Zhang. Toward large scale integration: Building a metaquerier over databases on the web. In *Proceedings of the Second Biennial Conference on Innovative Data Systems Research (CIDR 2005)*, Asilomar, CA, Jan. 2005.

[5] A. Doan, P. Domingos, and A. Y. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *SIGMOD Conference*, 2001.

[6] D. Florescu, A. Y. Levy, and A. O. Mendelzon. Database techniques for the world-wide web: A survey. *SIGMOD Record*, 27(3):59–74, 1998.

[7] B. He and K. C.-C. Chang. Statistical schema matching across web query interfaces. In *SIGMOD Conference*, 2003.

[8] B. He, K. C.-C. Chang, and J. Han. Discovering complex matchings across web query interfaces: A correlation mining approach. In *SIGKDD Conference*, 2004.

[9] H. He, W. Meng, C. T. Yu, and Z. Wu. Wise-integrator: An automatic integrator of web search interfaces for e-commerce. In *VLDB Conference*, pages 357–368, 2003.

[10] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogeneous information sources using source descriptions. In *VLDB Conference*, 1996.

[11] J. Madhavan, P. A. Bernstein, A. Doan, and A. Halevy. Corpus-based schema matching. In *ICDE Conference*, 2005.

[12] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *VLDB Conference*, 2001.

[13] K. Marriott. Constraint multiset grammars. In *Proceedings of IEEE Symposium on Visual Languages*, pages 118–125, 1994.

[14] R. J. Miller, M. A. Hernández, L. M. Haas, L. Yan, C. T. Howard Ho, R. Fagin, and L. Popa. The Clio project: managing heterogeneity. *SIGMOD Rec.*, 30(1):78–83, 2001.

[15] Y. Papakonstantinou, H. García-Molina, and J. Ullman. Medmaker: A mediation system based on declarative specifications. In *ICDE Conference*, 1996.

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

[17] J. D. Ullman. Information integration using logical views. In *ICDT Conference*, Jan. 1997.

[18] W. Wu, C. T. Yu, A. Doan, and W. Meng. An interactive clustering-based approach to integrating source query interfaces on the deep web. In *SIGMOD Conference*, 2004.

[19] Z. Zhang, B. He, and K. C.-C. Chang. On-the-fly constraint mapping across web query interfaces. In *Proceedings of the VLDB Workshop on Information Integration on the Web (VLDB-IIWeb'04)*, 2004.

[20] Z. Zhang, B. He, and K. C.-C. Chang. Understanding web query interfaces: Best effort parsing with hidden syntax. In *SIGMOD Conference*, 2004.