PROJECT DESCRIPTION

1 Introduction

People want to know! And so do government agencies, information providers, search-and-retrieval companies, electronic publishers, corporate enterprises, and business-intelligence professionals. But they're swamped with volumes of data spewed forth from search engines, corporate intranets, scientific instrumentation, news feeds, and the mountain of historical data found in books, microfilms, and government archival documents. People want critical information extracted expertly, organized automatically, and summarized smartly in a usable personalized view.

Critical information is difficult to locate. Once located, its incompatible formats and chaotic presentation styles make it difficult to use effectively. Large volumes of text—some of it not even digitized—must be digested and integrated into an easy-to-use, organized, uniform format to support querying, focused searching, personalized information products, and on-line transaction systems. The challenges are many: automatic search, automatic filtering, automatic extraction, automatic organization, automatic integration, automatic analysis, and automatic summarization.

We propose a particular approach to address these challenges—an approach that spans the spectrum of these challenges and provides a framework for resolution, but leaves open the possibility of adding many technical details. The approach we propose is called TIDIE (Target-based, Independent-of-Document Information Extraction, pronounced "Tidy"). It is "Target-based" because it requires a target description of what a user wants. It is "Document Independent" because it relies on linguistic, geometric, ontological, and metatextual clues that are specified independently of any particular document.

Because TIDIE is target-based and document-independent, it applies broadly across the spectrum of challenges. The ontological part of a target description serves as a conceptual model and establishes a basic scheme for the data to be extracted. We thus immediately have a framework for information organization, a basis for information integration, and a database that can be queried for analysis and summarization. With the addition of naturally occurring linguistic clues that can identify self-describing data or data in context, geometric clues that can identify layout patterns for groups of related data, and metatextual clues that can signal the location of information boundaries, we can also turn the target description into a mechanism for automatic search, automatic filtering, and automatic information extraction.

Our thesis is this: finding, extracting, structuring, synthesizing, and rendering information is easier given a detailed, target-based, document-independent description of what is wanted. Since TIDIE requires a detailed target description, it should be clear that we are *not* proposing it as an alternative search-engine technology to aid in ad-hoc, one-time requests to find a Web page of interest. Instead, we are proposing TIDIE for inquiries that gather and organize information of long-lasting interest, for inquiries that can be delegated to autonomous agents, or for common inquiries that can be amortized over large user groups.

We proceed with our proposal for TIDIE as follows. In Section 2 we discuss two initial experiments we have conducted—one on extracting and structuring data from multiple-record Web documents and the other on extracting and structuring data from tables found in

non-electronic archival documents. In Section 3 we discuss our current efforts—integration, extraction of data from business reports, Web document filtering, information culling, microfilm extraction, sentence boundary recognition, and extraction of data behind forms. We also discuss the vision of future work we see for TIDIE. Throughout Sections 2 and 3 we also review the relevant literature and show how the TIDIE approach is similar to and different from the work of other researchers. In Section 4 we summarize and state the significance of the proposed work for TIDIE. Finally, in Section 5 we lay out a research plan to achieve the objectives we have set for for TIDIE.

2 Background

Over the past two years, we have experimented successfully with extracting data from multiple-record Web documents such as car ads and job ads ([ECLS98]), obituaries ([ECJ+98]), and other applications ranging from real estate to stocks to personals to musical instruments ([Dem]). Also, we have experimented successfully with converting scanned images of tables into electronic data ([Haa98]). This section briefly explains these initial experiments and compares them with the research work of others.

Before discussing the details of these two initial experiments, we mention four other projects we have completed whose results directly apply to TIDIE. (1) In [Emb80] we introduced the idea of a "data frame" to capture the appearance of a particular data item—such as a license-plate number, a social-security number, or a bank-account number—along with constraints that apply to the data item, canonical written forms for the data item, and procedures and functions that apply to the data item. This kind of knowledge about a data item permits document-independent extraction of atomic data values. (2) In [EK85] we described a way to extract attributes, data values, and comparison operators from natural-language queries and showed how to use them to formulate SQL queries. We formulated queries using a given ER model instance as a guide, and thus we began to explore the combined use of text extraction and ontological declarations. (3) We continued using this combination as we turned our attention to ordinary business forms and developed NFQL (the Natural Forms Query Language) [Emb89]. (4) Along the way, we developed OSM [EKW92], a conceptual model with a greater ontological orientation than the earlier ER models. These four projects contributed ideas and techniques that directly apply both to our current and future work on TIDIE.

2.1 Initial Experimentation—Data Extraction for Multiple-Record Web Documents

Our approach to Web data extraction consists of the following steps.

1. We begin with a raw HTML document that contains unstructured chunks of text for an application of interest. (See Figure 1 as an example that shows a partial document for some raw-text car ads.)

2. For the application of interest, we develop a conceptual-model instance that describes the application's objects, the relationships among objects, the application's constant

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>NAC/The Salt Lake Tribune Transportation Classifieds</title>
</head>
...
<h4><font face="Arial" size="2">
'98 BUICK Century, like brand new, Only $14,995. 461-8509
<br><img src="/logo/mecham.gif" align="bottom" WIDTH="125"><br>
</font></h4>
<hr>
...
</html>
```

Figure 1: Raw Sample HTML Document (Partial).

values, and the application's keywords (keywords help identify which values belong to which object sets). (Figure 2 shows a conceptual-model snippit from the car-ad application.)

3. We parse this description of objects, relationships, constants, and keywords to generate a database scheme and to generate matching rules for constants and keywords.

4. To obtain data from a Web document, we invoke a record extractor that separates an unstructured Web document into individual record-size chunks, cleans them of markup-language tags, and presents them as individual unstructured record documents for further processing [EJN99].

5. We invoke recognizers that use the matching rules generated by the parser to locate and extract data in the unstructured records. The extraction algorithms use proximity heuristics to correlate extracted keywords to extracted constants and use cardinality constraints in the conceptual-model description to determine how to construct records.

Once the data is extracted and placed in the database, it can be queried using a standard database query language. (Figure 3, for example, shows partial results of a query for the year, make, model, and price of cars as extracted from a raw HTML document.) To make our approach general, we fix in advance: the parser, the Web record extractor, the keyword recognizer, the constant recognizer, and the database record generator; we change only the conceptual-model description as we move from one application domain to another.

To measure the success of our data-extraction work, we computed recall and precision ratios for each attribute for each application. We achieved recall ratios in the range of 90% and precision ratios near 98% for both car ads and job ads [ECLS98]. For obituaries, a much more complex challenge, recall ratios ranged from 70% to 100%, and precision ratios ranged from 93% to 100% (except for names of relatives, which dropped to 71%) [ECJ+98]. When applied without change to world-wide obituaries from Ireland, Sri Lanka, New Zealand, and India, these results continued to hold, although there was some drop off caused by culture localizations that could be corrected within our framework.

```
Car [-> object];
Car [0..1] has Year [1..*];
Year matches [4]
    constant { extract "\d{2}";
               context "\b'[4-9]\d\b";
               substitute "^" -> "19";
               },
     { extract "\d{2}";
               context "\b'0\d\b";
               ...
...
```

Figure 2: Conceptual-Model Description (Partial).

Year	Make	Model	Price
1998	BUICK	Century	14,995
1994	DODGE		4,995
1994	DODGE	Intrepid	10,000
...			

Figure 3: Query Results (Partial).

Our work differs fundamentally from the approach others have taken, basically because we provide a document-independent target description. The most common approach to information extraction from the Web has been through page-specific wrappers, written by hand [CGMH+94, AM97, GHR97] or written using a variety of techniques, including hand-written with the aid of a toolkit [SA99], hand-coded specialized grammars [ACC+97], wrapper generators based on HTML and other formatting information [AK97, HGMC+97], page grammars [AMM97], landmark grammars [MMK98], concept definition frames [SL97], or some form of supervised learning [Ade98, AK97, DEW97, KWD97, Sod97, Fre98, CDF+98]. A disadvantage of these wrapper-generation techniques is the work required to create the initial wrapper (a disadvantage we also share in the sense that we have to create a target description), and the rework required to update the wrapper when the source document changes (a disadvantage we do not share).

The approach of [SL97] using "concept definition frames" and [CDF+98] using "an ontology describing classes and relations" are closest to our approach. Our notion of a "data frame" [Emb80] is similar to a "concept definition frame", but embodies a richer description of the data to be recognized and extracted, and our notion of an "ontology" is similar to "ontology" of [CDF+98], but goes much further in describing the application of interest. The work reported in [Bri98] is also similar to ours in the sense that it is robust with respect to source document changes. The technique in [Bri98], which extracts author/title pairs, requires very little supervision for the machine-learning approach it takes, and need not be altered either for new pages or when pages change. This approach, however, appears to be limited to very small, tightly coupled application domains such as author/title pairs for which it was used.

```
 symbol name    input string
-----------------------------
 alpha          \(*a
 beta           \(*b
 gamma          \(*g
```

Figure 4: Sample Non-electronic Input Table.

Another approach that has been used for information extraction is natural language processing (NLP) [LCF+94, CL96, Sod97] NLP approaches use tokenization, part-of-speech and sense tagging, building syntactic and semantic structures and relationships, and producing a coherent framework for extracted information fragments. Our work does not attempt to understand the text in the deep NLP sense; consequently it does not depend upon sentential elements (as deep NLP approaches do), which are often missing for Web pages of classified ads and for partially formatted data found in forms and census records.

Our approach uses a specific target description, but we are not the only ones who have suggested target descriptions. With a somewhat similar objective in mind, [DMRA97, MD99, DM99] presents Structured Maps as a modeling construct imposed over Web information sources. Similar to our target description, a semantic model is used to provide a scheme over a domain of interest, which is then populated with information elements from the Web. In another effort with a similar objective, [AMM97] introduces a data model to describe the scheme for a user view over information on the Web along with a set of languages for synthesizing the scheme for a particular application and to manage and restructure data with respect to the scheme. Our work differs from these other efforts because they do not attempt to populate their model instances automatically, populating them instead by hand, with the aid of tools, or by semiautomatic methods.

2.2 Initial Experimentation—Tables in Non-Electronic Documents

Our table processing system takes as input a scanned image of a table—say from an old science book or handbook of mathematical tables—and produces as output a table that matches attributes with values. When our table processing system succeeds, a printed table becomes computer readable. We may place the results, for example, in a relational database or in a spreadsheet and further process the table electronically. Figure 4 shows a sample input table (which we assume is literally printed as shown and has no electronic representation). Figure 5 shows the sample output for this input table as SQL code to build and populate a relational database with the information from the printed table.

Table processing proceeds in two phases:

1. Scan the table and produce a cleaned-up, normalized, internal representation for the table. This phase involves the following steps:

   (a) Scan a paper copy of the table (e.g., Figure 4) and produce a bitmap image. We assume the paper copy of the table is already zoned to contain only the table.

   (b) Represent the structure of the bitmap image in an intermediate form. The intermediate form has representations for tables with lines, without lines, and with

```
create table T1 (
symbolname char(10),
inputstring char(5)
);

insert into T1 values ('alpha', '\(*a');
insert into T1 values ('beta', '\(*b');
insert into T1 values ('gamma', '\(*g');
```

Figure 5: Sample Output Table as Generated by SQL.

mixed lines and white-space separators. In producing the representation, we account for vectorization, noise filtering, and junctions.

(c) Apply OCR (Optical Character Recognition) to each elementary cell (i.e., a cell with no lines or white-space separators) to obtain a string value for the contents of the cell.

2. Process the internal representation for the table, and produce a generic computer-readable table as output. This phase involves the following steps:

   (a) Represent the intermediate form with OCR cell values received from Phase 1 as a database of facts about the table.

   (b) Apply structure heuristics to produce a relation scheme. We may represent the result by SQL create-table statements (e.g., as in Figure 5).

   (c) Produce table values. For SQL DDL, we may represent the table values by SQL insert statements (e.g., as in Figure 5), which populate the scheme with values.

As an intermediate form between the two phases, we use an XML description, which represents the results of scanning, cleaning, organizing, and applying OCR. Figure 6 shows part of the XML that describes the scanned table. According to the XML, the table was scanned at 300 pixels per inch, has its upper left corner at (4,4), and has its lower right corner at (220,650). The basic structure of the intermediate description is an *XY-tree* [NS84], first with X-cuts (horizontal cuts) that extend across the table either at white-space gaps or at actual horizontal lines, then with Y-cuts (vertical cuts) that extend between the X-cut boundaries at white-space gaps or vertical lines, then with X-cuts that extend across the Y-cut boundaries, and so forth until we arrive at atomic cells. Atomic cells contain a string, possibly an empty string. Figure 6 shows that we have a white-space X-cut at Row 4 and an 8-pixel-thick-actual-line X-cut at Row 58. Within these boundaries, we have a white-space Y-cuts at Column 4 and 291. These boundaries form an atomic cell, that contains the text "symbol name", with its bounding box having (18,12) as its upper-left corner and (51,251) as its lower-right corner.

After loading the facts from the XML description into a knowledge structure, we process the facts with horn-clause rules. The rules determine whether the table is a vertical table, like the table in Figure 4, or is a horizontal table. Vertical tables have attributes at the top, have values in columns, and have records in rows; horizontal tables have attributes at the

```
<table>
   <pixelsPerInch>300</pixelsPerInch>
   <upperLeft>
      <rowPixel>4</rowPixel>
      <columnPixel>4</columnPixel>
   </upperLeft>
   <lowerRight>
      <rowPixel>220</rowPixel>
      <columnPixel>650</columnPixel>
   </lowerRight>
   <horizontalCut cut="H1">
      <extentOfCut>
         <columnPixelBegin>4</columnPixelBegin>
         <columnPixelEnd>650</columnPixelEnd>
      </extentOfCut>
      <line>
         <rowPixel>4</rowPixel>
         <thickness>0</thickness>
      </line>
      <line>
         <rowPixel>58</rowPixel>
 <thickness>8</thickness>
      </line>
      ...
      <veritcalCut cut="H1V1">
         <extentOfCut>
    <rowPixelBegin>4</rowPixelBegin>
    <rowPixelEnd>58</rowPixelEnd>
         </extentOfCut>
 <line>
    <columnPixel>4</columnPixel>
    <thickness>0</thickness>
 </line>
 <line>
    <columnPixel>291</columnPixel>
    <thickness>0</thickness>
 </line>
 ...
 <horizontalCut cut="H1V1H1">
    <extentOfCut>
       <columnPixelBegin>4</columnPixelBegin>
       <columnPixelEnd>291</columnPixelEnd>
    </extentOfCut>
    <elementaryCell>
       <text>symbol name</text>
       <textBoundingBox>
          <textUpperLeftRowPixel>18</textUpperLeftRowPixel>
          <textUpperLeftColumnPixel>12</textUpperLeftColumnPixel>
          <textLowerRightRowPixel>51</textLowerRightRowPixel>
          <textLowerRightColumnPixel>251</textLowerRightColumnPixel>
       </textBoundingBox>
    </elementaryCell>
 </horizontalCut>
 ...
      </verticalCut>
      <verticalCut cut="H1V2">
      </verticalCut>
   </horizontalCut>
</table>
```

Figure 6: Intermediate Table Information in XML.

left, have values in rows, and have records in columns. The rules also check for nested tables with factored attributes.

In experiments we conducted, we were able to correctly match attributes with values and produce relational equivalents for several tables [Haa98]. These tables, however, were artificially created to test our table-processing software.

[LN99] surveys the current state of the art in automated table processing. It surveys our work (described above) and the work of several other researchers. This work includes several approaches to digitized table analysis (e.g., [ACM96, DHQ95, GK95, HD95, Ito93, Kie98, KW98, PCA97, RC96, WQS95]) work on segmenting and labeling digitized pages [NKSV93], and work on modeling and interpreting tables [Wri73, GK95, WQS95]. Like [Wri73, GK95, WQS95, aDS97], we also model tables and table properties; in addition, however, we model the application's domain so that we know the real-world objects, relationships, and constraints represented by the values displayed in a table. In general, our work has many similarities with the work reported here, but it differs because we identify and extract particular information from tables as guided by an application target specification.

3 Objectives

Our objective for TIDIE is to extract information from data-rich, semistructured documents and structure the information with respect to a given target description. Our concept of a semistructured document encompasses the notion of semistructured data, which [ABS00] defines as being "schemaless" but "self-describing" and representable by a variant of OEM (the Object Exchange Model [CGMH+94, AQM+97, MAG+97]). Starting with this notion of data semistructuredness, we enlarge it to include any document where self-descriptive clues have two properties: (1) they are sufficient to match attributes and values and (2) they are sufficient to allow these attribute-value pairs to be assembled into meaningful chunks of information representable by OEM. Semistructured documents run from the high end, where attribute-value pairs and their organization are given, to the low end, where the clues are subtle and depend on a high degree of human understanding to assemble and organize attribute-value pairs. In TIDIE we exploit these human-understandable, self-descriptive clues to classify atomic data values and to organize molecular record structures. Further, we seek to exploit these clues in a document-independent way, so that our techniques apply robustly over the full range of semistructured documents.

We classify the particular self-descriptive clues we wish to exploit in developing TIDIE as being linguistic, geometric, ontological, and metatextual.

- *Linguistic Clues*: lexical data values, lexical attributes, and lexical context. We can sometimes classify data values, such as vehicle-identification numbers (VINs), dates, Social-Security Numbers, dollar amounts, and university course numbers, without the aid of accompanying attribute designators. Attributes and lexical context, however, are needed when there can be ambiguity such as when integers, reals, dollar amounts, or dates play different roles in a document.

- *Geometric Clues*: patterned layout including row alignment, column alignment, nested indentation, page headers, and page footers. Row and column alignment and nested

indentation provide clues for organizing attribute-value pairs into record structures. Linguistic and geometric clues together are often sufficient to permit attribute-value pairing, especially for forms and tables.

- *Ontological Clues*: objects, relationships, cardinalities, generalization/specialization, aggregation, and general constraints. Knowing about real-world objects and their relationships and constraints can aid in both recognition and organization of data values. Ontological expectations embodied in "IS-A" and "Part-Of" relationships as well as general constraints can limit the possible choices for attribute-value pairs. Ontological organization guides the record construction.

- *Metatextual Clues*: punctuation, italics, bold, underlining, arrows, pointing fingers, lines, and boxes. Punctuation, such as sentence-boundary designations, and bounding lines and boxes limit the scope of context and help prevent spurious connections between attributes and values. Metatextual emphasis aids in distinguishing more important from less important attributes; boundaries aid in sorting out ambiguities.

TIDIE project assumptions make our task tractable, but we are careful that our assumptions do not unreasonably diminish the range of applicability. We assume (1) that the target descriptions are *ontologically narrow* and (2) that the documents we process are *data rich* and *semistructured*. These three notions defy a precise definition, but are bounded as follows.

> Target descriptions are *ontologically narrow* if the conceptual model instance describing objects, relationships, and constraints is "small." "Small" means that the conceptual model has a half dozen to a few dozen object sets (attributes), about the same number (or a few more) relationship sets (connections among the attributes), and several dozen constraints. The conceptual models we used in our initial experiments were ontologically narrow.
>
> Documents are *data rich* if they contain "many" attribute-value pairs. "Many" means that we can populate at least a few (say a half dozen or more) attribute-value pairs and their relationships in an ontologically narrow target description.
>
> Documents are *semistructured* if they contain sufficient linguistic, geometric, ontological, and metatextual clues to allow human readers to extract atomic attribute-value pairs and organize them into molecular record structures. Documents that are chaotic, ambiguous, or contain literary imagery are outside the scope of TIDIE.

We complete this section on objectives by briefly describing our current efforts and giving a brief glimpse of future possibilities. In so doing, we relate these efforts and possibilities to our objectives for TIDIE.

3.1 Current Efforts

Information Integration [BE99]. We have a detailed framework for integrating information from heterogeneous information sources. Our framework assumes that a target view is

specified ontologically and independently of any of the sources. We model both the target and all the sources in the same modeling language. For each source we generate a target-to-source mapping that tells us how to obtain target facts from source facts. As the mapping generator runs, it raises specific issues for a user's consideration. It is endowed with defaults, however, to allow it to run without user input. The integration framework is based on a formal foundation. The foundation is sufficient to prove that when a source has a valid interpretation, the part of the target obtained from the source according to a generated source-to-target mapping also has a valid interpretation. If we are given individual target-to-source mappings from several sources, our integration framework merges the data into an integrated target database. We can prove that the merged source data has a valid interpretation in the target. Whether the merged source data is minimal, however, is still an open question.

Business Reports [LCC99]. A considerable amount of clean semistructured data is internally available to companies in the form of business reports. However, business reports are untapped for data mining, data warehousing, and query processing because they are not in relational form. Business reports have a regular structure that can be reconstructed. We developed algorithms that automatically infer the regular structure underlying business reports and automatically generate wrappers to extract data and store it in a relational database.

Document Filtering [EFKR99]. Our system for data extraction from multiple-record Web documents works well [ECJ+99] when the ontology is suitable for the Web document. How do we algorithmically determine whether an ontology is suitable? To resolve this question, we devised an approach that filters documents and keep only those that are suitable. The approach is based on three heuristics: density, schema, and grouping. We encoded the first heuristic as a density function. We are exploring the vector-space model [SM83, BYRN99], probabilistic models [CLRC98], and various statistical models to capture the second and third heuristics. We have argued informally that these heuristics filter documents with respect to an ontology [EFKR99], and we are currently experimenting with Web documents to verify our informal argument.

Information Culling. Since our intent is to work at the level of atomic attribute-value pairs and molecular record structures, most of the data we seek is a small part of a larger document. To make matters worse, when multiple records appear in a document, their content may be presented in at least five formats: (1) some of their values may be factored into headings and subheadings, (2) some of their values may be linked by an off-page connector or by a sequence of off-page connectors, (3) some of their values may be presented as tables, (4) some of their values may be interspersed with records which are not of interest, and (5) some of their values may be embedded in irrelevant information. We have a two-pronged approach to locate information of interest, piece it together, and ignore the inapplicable parts of a document: (1) we identify one of the five patterns just mentioned, and (2) we reorganize the information such that the density, schema, and grouping heuristics (discussed under "Document Filtering" above) are increased. For example, if we recognize that records have been factored, we distribute the factored values into the records and thus increase the likelihood that the records correspond to the schema (second heuristic) and are grouped according to the ontology (third heuristic).

Sentence Boundary Recognition. We have a sentence-boundary detection system that is

based on end-of-sentence punctuation rules. Because a period can appear in decimals, e-mail addresses, abbreviations, initials in names, and honorifics, as well as the end of a sentence, the development of algorithmic rules to classify end-of-sentence punctuation is nontrivial. At the current experimental stage, we are using a corpus of 24,986 sentences, 24,914 of which are detected correctly. Most of the rules used in by our sentence-boundary detection system are based on English grammar, but some are data-context driven. For example, the abbreviation "St." may stand for "street" or "saint." When "St." stands for "saint," it may not appear at the end of a sentence. Our sentence-boundary detection system is an example of creating metatextual clues—boundaries over which some context information should not cross.

Record Recognition in Microfilm Documents [Tub00]. A wealth of information is locked in microfilm. Although extracting and structuring this wealth of data is overwhelming, automatic extraction and organization techniques can offer help. In the system we are building the process of capturing records from structured, tabular microfilm has two parts. First, data fields must be extracted from microfilm; and second, data fields must be organized into records. The record-recognition system focuses on the second challenge. (It assumes that the first task has been done by techniques like those in our non-electronic table processing system or by hand with the aid of a tailor-made user interface.) The record-recognition system accepts extracted attribute-value pairs and their geometric location and assembles them into records for a given ontological description.

Forms Yielding Dynamically Generated Data from the Web. In the early days of the Web, most Web pages were static. Now many Web pages are dynamic, displaying data generated on the fly from data stored in files, databases, and other repositories. To access this data, users fill in forms. How can unknown forms be automatically filled in? How can the retrieved data be organized according to a specified user view? We are investigating the following technique. For fields that have designated small finite sets of entries (radio buttons, check boxes, and pull-down lists), we automatically fill in all combinations and retrieve the data for all combinations. For text boxes, we leave the fields empty, or alternatively, we match the field with target-specified data frames and data-frame values. A postprocessor identifies records and discards duplicates. The assembled records are then processed using the data-extraction techniques we have already developed.

3.2 Vision of Possibilities

This section lists things to accomplish and issues to address. We also postulate possible uses of the technology we intend to develop.

We first bring our current efforts to fruition. In particular we propose to complete the tasks of Section 3.1. In addition:

- We wish to exploit all four types of clues (linguistic, geometric, ontological, and metatextual) in all of our efforts.
- We wish to make all heuristics, rules, and clue-processing specifications declarative. Declarative specifications pave the way (1) for easier and more active experimentation (we can alter declarative statements more easily than hard-coded procedures) and (2) for possible machine learning, where the system, rather than a human, creates the declarative statements. Although counter to prevailing thought, we question whether

it is less human intensive to prepare sufficient labeled examples to train a machine-learning algorithm or less human intensive to create an ontologically narrow application ontology with the aid of software tools [Dem]. Our anectodal experience experience tells us that a few dozen person hours is sufficient to create a reasonable application ontology. Further, hand-crafted application ontologies tend to have higher recall and precision (e.g., about 80% [NM00] verses about 90% [ECLS98] on job ads).

- We wish to follow up on six "smaller" specific ideas that have been suggested. (1) Explore the use of WordNet [Fel98] for use in integration, (2) explore theories of evidence—such as Dempster-Shafer Theory [SP90]—for detecting object identity in integration, (3) explore the use of grammars for identifying patterns of lines both in business reports and in tables, (4) explore the possibility of XML refinement for culling and reorganizing semistructured documents, (5) explore the possibility of learning geometric patterns for locating and rubber-banding potential attribute-value pairs in images of microfilm documents, and (6) explore the ramifications of dynamic target development that takes place synergistically during data extraction and integration.

As a "vision of possibility," we can see that the technology we are developing can be embedded in personal agents; in customized search, filtering, and extraction tools; and in tools that provide individually tailored views—integrated, organized, and summarized to meet individual or organizational needs.

4 Significance

By completing our current work (Section 3.1) and by following up on the specific possibilities (Section 3.2), we can accomplish a great deal. Indeed we can establish the technological basis for obtaining critical information extracted expertly, organized automatically, and summarized smartly in a usable personalized view.

Our particular angle on achieving these significant possibilities is TIDIE. TIDIE is target-based; we believe that the more a search technology lets users specify their wants, the better it can deliver what is wanted. TIDIE operates independently of a particular document; we believe that technology should *not* be based on the peculiarities of a particular document, but rather based on document-independent clues that hold robustly over all semistructured documents. The clues we propose to investigate are lexical clues, geometric clues, ontological clues, and metatextual clues. The short-term objective for TIDIE is to build a system that extracts data from semistructured source documents and integrates it with respect to a target specification.

5 Research Plan

So, how do we achieve these objectives? How do we algorithmically use lexical, geometric, ontological, and metatextual clues to filter, identify, and cull information applicable to a target specification? How do we ultimately extract and assemble semantic information from information sources and present it in a usable personalized view? How do we add a measure

of certainty to extracted information? How do we measure whether a target specification provides a "good" basis for extracting semantic information? How do we minimize human involvement in developing and maintaining target specifications and synergistically push the burden of development and maintenance as much as possible to the TIDIE system we are proposing?

We propose four specific projects to address these questions. Each project involves resolving several issues. We consider the resolution of these issues to be milestones, and we thus attach projected completion dates. Table 1 shows our performance-evaluation metrics and the techniques we intend to employ. For performance evaluation, we use recall and precision over sets of human-countable size as we have done in our initial experimentation [ECJ+99]; for development, we use standard, well-known computer-science techniques.

1. *Data Extraction from Data-Rich Web Documents.* We have made considerable headway on this project [ECJ+99], but there is much more to do. We expect to satisfactorily resolve the following issues: (a) high precision filtering with respect to a target specification (Dec. 2000), (b) accurate atomic information culling from within a Web page or from within sets of linked Web pages (Aug. 2001), (c) extraction of data behind Web forms (Jun. 2002), (d) definition and implementation of declarative processing rules and heuristics (Mar. 2003), and (e) efficiency concerns for practical application of our data-extraction technology (Sep. 2000–Aug. 2003).

2. *Revitalization of Data in Historical Documents.* We have begun to explore the analysis of both digitized tables [Haa98] and digitized microfilm documents [Tub00], and we expect to satisfactorily resolve the following issues: (a) the development of conceptual models for tables and forms (Dec. 2000), (b) geometric/linguistic/ontological/metatextual extraction of atomic information (Dec. 2001), (c) matching filled-in tables and forms with target specifications (Dec. 2002), (d) the investigation of geometric patterns for molecular record identification (Jun. 2003), and (e) synergistic user interaction for the resolution of "human-only resolvable problems" (Sep. 2000–Aug. 2003).

3. *Integration—Target-to-Source Mapping Generation.* We have outlined in detail our general approach to matching a target specification with a potential data source [BE99]. Following this outline, we expect to satisfactorily resolve the following issues: (a) a formal definition of the properties and implications of target-to-source mappings (Dec. 2000), (b) semantic matching of object- and relationship-sets in target and source specifications (Dec. 2001), (c) generation of implied source object- and relationship-sets to support target specification (Dec. 2002), (d) investigation of confidence measures for assuring the quality of target-to-source mappings (Dec. 2000–Dec. 2002), and (e) automatic identification and synergistic resolution of "human-only resolvable problems" (Jan. 2001–Aug. 2003).

4. *Integration—Merging Data.* We have described a formal basis for merging populated target-specification instances that have been (partially) populated with data from heterogeneous sources [BE99]. Building on this formal basis, we expect to satisfactorily resolve the following issues: (a) object identity (Dec. 2001), (b) global constraint satisfaction, given local constraint satisfaction (Jun. 2001), (c) transformation of values

to a common target ontology (Dec. 2002), (d) incremental updates given new source information (Jun. 2003), and (e) automatic identification and synergistic resolution of "human-only resolvable problems" (Jun. 2001–Aug. 2003).

Performance Metrics & Evaluation Techniques		
Type	Granularity	Issue Applicability
recall & precision	attribute-value pairs	2b,4c
recall & precision	object	4a
recall & precision	record	1b,1c,2d
recall & precision	document	1a
recall & precision	object sets & relationship sets	2c,3b,3c,3d
big-Oh	document set size & document size	1e
observation	human	2e,3e,4e
formal properties	object sets & relationship sets	2a,3a,4d
formal properties	1st-order rules & constraints	1d,4b

Across the four projects, the issues to be addressed have a great deal of overlap. The common, underlying philosophy of TIDIE ties these efforts together—they are all target-based and they rely on document-independent linguistic/geometric/ontological/metatextual clues. The synergy among the projects should promote a more rapid resolution of issues than if the projects were each done in isolation. Moreover, most of the issues can be pursued and resolved in parallel. There are some dependencies (2a precedes 2b–e, 3a precedes 3b–e, 3a precedes 4a-c, and 4a-c precedes 4d), but, for the most part, we have simply spread the milestones over the grant period in a way that appears doable.

Our team of PI's/Co-PI's has complementary expertise in information extraction, table analysis, database theory, ontology building, natural language analysis, and information retrieval. We represent three different departments on campus (CS, IS, and Linguistics), and we have collaborated in various combinations for many years. Our team of students currently consists of one Ph.D. student (female), three Master's students (1 female), and two undergraduates. The bulk of our request is for this student team. We plan to expand to four Ph.D. students, and we plan to maintain our internal funding and local industrial support to support a few Master's students and undergraduates. As for equipment, we have a RAID system with 140 gigabytes of storage in our lab and several workstations. As a cost-sharing measure, BYU is willing to add to and replace our current workstations (about $31,000) so that we will have the equipment we need for this project. Our RAID system currently holds about 1.2 million Web pages which we have downloaded from 50 US newspapers. We plan to expand this set of pages and use it as a testbed for TIDIE. To broaden our reach, we currently teach several graduate courses on Web-based data extraction and integration (Embley), on digital libraries (Campbell), and natural-language processing (Lonsdale), all of which are highly related to our project. To further broaden our reach, we maintain a Web site for our project (www.deg.byu.edu) which includes a demo of the current status of our extraction work and downloadable software, which we have developed. Our source code is

and will always be freely available to interested parties. We also plan to make our Web test pages available to anyone who wishes to use them.

We have already accomplished much. We have developed OSM [EKW92], have extended it with data frames, and have shown how to use it as an ontological representation for a narrow target domain of interest. Using this framework, we have built a prototype information-extraction system that performs with high recall and precision [ECJ+99]. Finally, we have begun to explore the use of these same kinds of target specifications for the integration of information from heterogeneous sources [BE99] and for the extraction of information from non-electronic tables and forms [Tub00]. Although much has been accomplished, there is much more yet to be accomplished to enable the practical use of TIDIE technology.

References

[ABS00] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann Publishers, San Francisco, California, 2000.

[ACC+97] S. Abiteboul, S. Cluet, V. Christophides, T. Milo, G. Moerkotte, and Jérôme Siméon. Querying documents in object databases. *International Journal on Digital Libraries*, 1(1):5–19, April 1997.

[ACM96] J.F. Arias, A. Chabra, and V. Misra. Interpreting and representing tabular documents. In *Proceedings of the International Conference on Pattern Recognition (ICPR'96)*, pages 600–605, San Francisco, California, June 1996.

[Ade98] B. Adelberg. NoDoSE—a tool for semi-automatically extracting structured and semistructured data from text documents. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 283–294, Seattle, Washington, June 1998.

[aDS97] D. Rus aned D. Subramanian. Customizing information capture and access. *ACM Transactions on Information Systems*, 15(1):67–101, 1997.

[AK97] N. Ashish and C. Knoblock. Semi-automatic wrapper generation for internet information sources. In *Proceedings of the CoopIS'97*, 1997.

[AM97] P. Atzeni and G. Mecca. Cut and paste. In *Proceedings of the 16th ACM PODS*, pages 144–153, May 1997.

[AMM97] P. Atzeni, G. Mecca, and P. Merialdo. To weave the Web. In *Proceedings of the Twenty-third International Conference on Very Large Data Bases*, pages 206–215, Athens, Greece, August 1997.

[AQM+97] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. Wiener. The Lorel query language for semistructured data. *International Journal on Digital Libraries*, 1(1), April 1997.

[BE99] J. Biskup and D.W. Embley. Extracting information from heterogeneous information sources using ontologically specified target views. http://ls6-www.informatik.uni-dortmund.de/issi/publications/1999.html.en, 1999.

[Bri98] S. Brin. Extracting patterns and relations from the World Wide Web. In *Proceedings of the WebDB Workshop (at EDBT'98)*, 1998.

[BYRN99] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, Menlo Park, California, 1999.

[CDF+98] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, pages 509–516, Madison, Wisconsin, July 1998.

[CGMH+94] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J Widom. The TSIMMIS project: Integration of heterogeneous information sources. In *IPSJ Conference*, pages 7–18, Tokyo, Japan, October 1994.

[CL96] J. Cowie and W. Lehnert. Information extraction. *Communications of the ACM*, 39(1):80–91, January 1996.

[CLRC98] F. Crestani, M. Lalmas, C.J. Van Rijsbergen, and I. Campbell. Is this document relevant? ... probably: A survey of probabilistic models in information retrieval. *ACM Computing Surveys*, 30(4):528–552, December 1998.

[Dem] Homepage for byu data-extraction group. URL: http://www.deg.byu.edu.

[DEW97] R.B. Doorenbos, O. Etzioni, and D.S. Weld. A scalable comparison-shopping agent for the World-Wide Web. In *Proceedings of the First International Conference on Autonomous Agents*, pages 39–48, Marina Del Rey, California, February 1997.

[DHQ95] S. Douglas, M. Hurst, and D. Quinn. Using natural language processing for identifying and interpreting tables in plain text. In *Proceedings of the Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, pages 535–545, Las Vegas, Nevada, April 1995.

[DM99] L. Delcambre and D. Maier. Models for superimposed information. In P.P. Chen, D.W. Embley, J. Kouloumdjian, S.W. Liddle, and J.F. Roddick, editors, *Proceedings of the Workshop on the World Wide Web and Conceptual Modeling (WWWCM'99)*, volume LNCS 1727, pages 264–280, Paris, France, November 1999. Springer Verlag.

[DMRA97] L.M.L. Delcambre, D. Maier, R. Reddy, and L. Anderson. Structured maps: Modeling explicit semantics over a universe of information. *International Journal on Digital Libraries*, 1(1):20–35, April 1997.

[ECJ+98] D.W. Embley, D.M. Campbell, Y.S. Jiang, Y.-K. Ng, R.D. Smith, S.W. Liddle, and D.W. Quass. A conceptual-modeling approach to extracting data from the Web. In *Proceedings of the 17th International Conference on Conceptual Modeling (ER'98)*, pages 78–91, Singapore, November 1998.

[ECJ+99] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record Web pages. *Data & Knowledge Engineering*, 31(3):227–251, November 1999.

[ECLS98] D.W. Embley, D.M. Campbell, S.W. Liddle, and R.D. Smith. Ontology-based extraction and structuring of information from data-rich unstructured documents. In *Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM'98)*, pages 52–59, Washington D.C., November 1998.

[EFKR99] D.W. Embley, N. Fuhr, C.-P. Klas, and T. Roelleke. Ontology suitability for uncertain extraction of information from multi-record web documents. In *Proceedings of the Workshop on Agenten, Datenbanken und Information Retrieval (ADI'99)*, Rostock-Warnemuende, Germany, 1999.

[EJN99] D.W. Embley, Y.S. Jiang, and Y.-K. Ng. Record-boundary discovery in Web documents. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, pages 467–478, Philadelphia, Pennsylvania, 31 May - 3 June 1999.

[EK85] D.W. Embley and R.E. Kimbrell. A scheme-driven natural language query translator. In *Proceedings of the 1985 ACM Computer Science Conference*, pages 292–297, New Orleans, Louisiana, March 1985.

[EKW92] D.W. Embley, B.D. Kurtz, and S.N. Woodfield. *Object-oriented Systems Analysis: A Model-Driven Approach*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[Emb80] D.W. Embley. Programming with data frames for everyday data items. In *Proceedings of the 1980 National Computer Conference*, pages 301–305, Anaheim, California, May 1980.

[Emb89] D.W. Embley. NFQL: The natural forms query language. *ACM Transactions on Database Systems*, 14(2):168–211, June 1989.

[Fel98] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachussets, 1998.

[Fre98] D. Freitag. Information extraction from html: Application of a general machine learning approach. In *Proceedings of AAAI/IAAI*, pages 517–523, 1998.

[GHR97] A. Gupta, V. Harinarayan, and A. Rajaraman. Virtual database technology. *SIGMOD Record*, 26(4):57–61, December 1997.

[GK95] E.A. Green and M.S. Krishnamoorthy. Model-based analysis of printed tables. In *Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR'95)*, pages 214–217, Montréal, Canada, August 1995.

[Haa98] T.B. Haas. The development of a prototype knowledge-based table-processing system. Master's thesis, Brigham Young University, April 1998.

[HD95] O. Hori and D.S. Doermann. Robust table-form structure analysis based on box-driven reasoning. In *Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR'95)*, pages 218–221, Montréal, Canada, August 1995.

[HGMC+97] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo. Extracting semistructured information from the Web. In *Proceedings of the Workshop on Management of Semistructured Data*, Tucson, Arizona, May 1997.

[Ito93] K. Itonori. A table structure recognition based on textblock arrangement and ruled line position. In *Proceedings of the Second International Conference on Document Analysis and Recognition (ICDAR'93)*, pages 765–768, Tsukuba Science City, Japan, October 1993.

[Kie98] T.G. Kieninger. Table structure recognition based on robust block segmentation. In *Proceedings of Document Recognition V (IS&T/SPIE Electronic Imaging'98)*, volume 3305, pages 22–32, San Jose, California, January 1998.

[KW98] W. Kornfeld and J. Wattecamps. Automatically locating, extracting and analyzing tabular data. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 347–348, Melbourne, Australia, August 1998.

[KWD97] N. Kushmerick, D.S. Weld, and R. Doorenbos. Wrapper induction for information extraction. In *Proceedings of the 1997 International Joint Conference on Artificial Intelligence*, pages 729–735, 1997.

[LCC99] S.W. Liddle, D.M. Campbell, and C. Crawford. Automatically extracting structure and data from business reports. In *Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM'99)*, pages 86–93, Kansas City, Missouri, November 1999.

[LCF+94] W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. Evaluating an information extraction system. *Journal of Integrated Computer-Aided Engineering*, 1(6), 1994.

[LN99] D. Lopresti and G. Nagy. Automated table processing: An (opinionated) survey. In *Proceedings of the Third IAPR Workshop on Graphics Recognition*, pages 109–134, Jaipur, India, September 1999. IAPR is International Association for Pattern Recognition. Submitted to IJDAR (International Journal of Document Analysis and Recognition).

[MAG+97] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: A database management system for semistructured data. *SIGMOD Record*, 26(3):54–66, September 1997.

[MD99] D. Maier and L. Delcambre. Superimposed information for the internet. In S. Cluet and T. Milo, editors, *Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB'99)*, Philadelphia, Pennsylvania, June 1999.

[MMK98] I. Muslea, S. Minton, and C. Knoblock. STALKER: Learning extraction rules for semistructured, Web-based information sources. In *Proceedings of AAAI'98: Workshop on AI and Information Integration*, Madison, Wisconsin, July 1998.

[NKSV93] G. Nagy, M. Krishnamoorthy, S. Seth, and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):737–747, 1993.

[NM00] U.Y. Nahm and R.J. Mooney. A mutually beneficial integration of data mining and information extraction. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00)*, Austin, Texas, 2000. Submitted.

[NS84] G. Nagy and S. Seth. Hierarchical representation of optically scanned documemts. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 347–349, 1984.

[PCA97] C. Peterman, C.H. Chang, and H. Alam. A system for table understang. In *Proceedings of the Symposium on Document Image Understanding Technology (SDIUT'97)*, pages 55–62, Annapolis, Maryland, April/May 1997.

[RC96] M.A. Rahgozar and R. Cooperman. A graph-based table recognition system. In *Proceedings of Document Recognition III (IS&T/SPIE Electronic Imaging'96)*, volume 2660, pages 192–203, San Jose, California, January 1996.

[SA99] A. Sahuguet and F. Azavant. Looking at the Web through XML glasses. In *Proceedings of the Fourth International Conference on Cooperative Systems (CoopIS'99)*, Edinburgh, Scotland, UK, September 1999.

[SL97] D. Smith and M. Lopez. Information extraction for semi-structured documents. In *Proceedings of the Workshop on Management of Semistructured Data*, Tucson, Arizona, May 1997.

[SM83] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[Sod97] S. Soderland. Learning to extract text-based information from the World Wide Web. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 251–254, Newport Beach, California, August 1997.

[SP90] G. Shafer and J. Pearl, editors. *Readings in Uncertain Reasoning*. Morgan Kaufmann, Los Altos, California, 1990.

[Tub00] K. Tubbs. Automatically creating records from extracted data fields of genealogical microfilm. Proposal Submitted to BYU ORCA (funded), 2000.

[WQS95] T. Watanabe, Q.L. Quo, and N. Sugie. Layout recognition of multi-kinds of table-form documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):432–445, 1995.

[Wri73] P. Wright. Understanding tabular displays. *Visible Language*, 7:351–359, 1973.