Enabling Search for Facts and Implied Facts in Historical Documents

David W. Embley, Spencer Machado, Thomas Packer, Joseph Park, Andrew Zitzelberger
Department of Computer Science
Brigham Young University
Provo, Utah 84602, USA

Stephen W. Liddle, Nathan Tate
Information Systems Department
Marriott School of Management
Brigham Young University
Provo, Utah 84602, USA

Deryle W. Lonsdale
Department of Linguistics and English Language
Brigham Young University
Provo, Utah 84602, USA

ABSTRACT

Building a database of facts extracted from historical documents to enable database-like query and search would reduce the tedium of gleaning facts of interest from historical documents. We propose a solution in which historical documents themselves constitute the stored database. In our solution, we use information-extraction techniques to produce a conceptualized external annotation of facts found in each document, and we superimpose the conceptualization over the document collection. The annotation process populates the conceptualization producing a repository of extracted facts, and a reasoner obtains inferred facts from these extracted facts. Our query interface accepts free-form queries and converts them to formal queries over the extracted and inferred facts. Displayed results include, in addition to standard query results, images of original documents with results highlighted along with reasoning chains for inferred facts grounded in these highlighted facts. Along with giving the implementation status of our proof-of-concept prototype, we present results for extraction accuracy and efficiency and point to current and future work needed to enable a practical solution for the envisioned historical-document database.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; E.0 [**Data**]: General

General Terms

Design,Theory

Keywords

information extraction, facts in historical documents, inferred facts

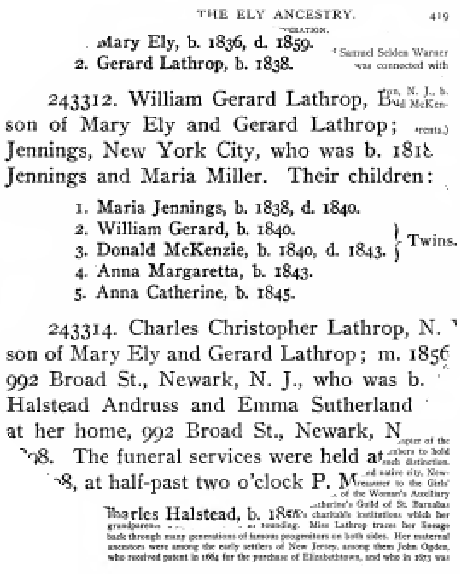
1. INTRODUCTION

Historians, both professional and avocational, are interested in gleaning facts from historical documents. Examples:

- When was William Lathrop born?
- Who are the grandchildren of Mary Ely?
- Which of Mary Ely's grandchildren died before reaching the age of eight?

Documents such as Page 419 of *The Ely Ancestry* [4] in Figure 1 contain answers to these queries, but searching for them manually is tedious. Of value are keyword searches over documents that have been run through an OCR engine (e.g., Page 419 of the Ely ancestry in Figure 2), but can we do better? Can we make facts and implied facts database-like searchable? And, so that untrained users can pose the queries themselves, can we effectively support free-form queries for these facts? Further, along with giving answers to queries, can we explain why the answers satisfy the queries: can we give reasoning chains and display original documents with facts of interest highlighted?

Answering these questions demands resolving a number of challenging issues that have been the subject of a considerable amount of research: (1) Automated fact extraction from text, a challenging sub-area of information extraction [18, 19, 23], (2) OCR recognition accuracy [15], (3) Question answering [5, 7, 13], (4) Keyword search and ranking algorithms as pioneered by Google and Yahoo!, and (5) Inference in logic, particularly description logics [3]. In addition to these general problems, the queries above applied to the Ely page in Figures 1 and 2 point to some specific problems needing resolution: (1) Names in queries may not directly match names in the text. Neither of the two occurrences of "William Lathrop" in the magnified part of Figure 1 is a direct match: both have a middle name and one has an implied surname. Further, as a common problem, the name is ambiguous—which William Lathrop? (2) Automatically sorting out stated family relationships in complex sentences such as the ones describing the Lathrop families in Figure 1 is non-trivial. (3) Implied information requires reasoning: of interest, but not stated in Figure 1, for example, are inferable facts, such as the surname of the Lathrop children,


THE ELY ANCESTRY. 419

. Mary Ely, b. 1836, d. 1859.
2. Gerard Lathrop, b. 1838.

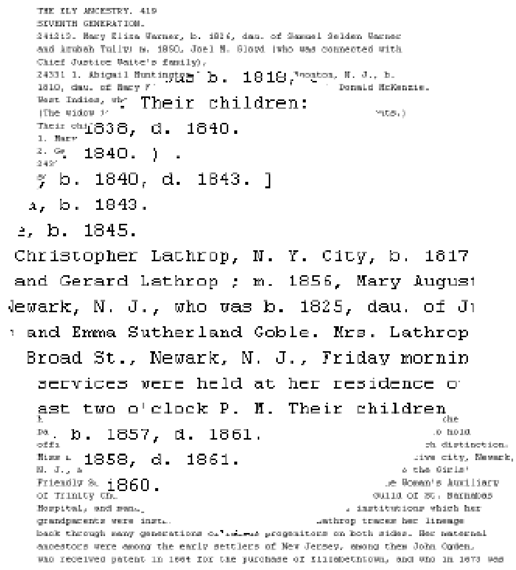
243312. William Gerard Lathrop,
son of Mary Ely and Gerard Lathrop;
Jennings, New York City, who was b. 181
Jennings and Maria Miller. Their children:

1. Maria Jennings, b. 1838, d. 1840.
2. William Gerard, b. 1840.
3. Donald McKenzie, b. 1840, d. 1843. } Twins.
4. Anna Margaretta, b. 1843.
5. Anna Catherine, b. 1845.

243314. Charles Christopher Lathrop, N.
son of Mary Ely and Gerard Lathrop; m. 1856
992 Broad St., Newark, N. J., who was b.
Halstead Andruss and Emma Sutherland
at her home, 992 Broad St., Newark, N
98. The funeral services were held at
8, at half-past two o'clock P. M.

harles Halstead, b. 185

Figure 1: Sample Page from The Ely Ancestry.


THE ELY ANCESTRY. 419
SEVENTH GENERATION.

b. 1818,
Their children:
1838, d. 1840.
1840.) .
b. 1840, d. 1843.]
, b. 1843.
, b. 1845.
Christopher Lathrop, N. Y. City, b. 1817
and Gerard Lathrop ; m. 1856, Mary August
Newark, N. J., who was b. 1825, dau. of J
and Emma Sutherland Goble. Mrs. Lathrop
Broad St., Newark, N. J., Friday mornin
services were held at her residence o
ast two o'clock P. M. Their children
b. 1857, d. 1861.
1858, d. 1861.
i860.

Figure 2: OCR of Sample Page in Figure 1.

their grandparents, and their gender. (4) The information extraction must be tolerant of OCR errors (e.g., the *i860* at the bottom of the magnified part of Figure 2 is really the year *1860*) (5) Expectations that all facts (stated and implied) can be anticipated and therefore extracted and inferred is not reasonable. But help can still be provided by a combination of keyword and database-query search leading to pages in documents that humans can read to obtain desired facts (e.g., Where was Maria Jennings born?).

To address these issues, and as a way to enable search for facts and implied facts in historical documents, we propose *WoK-HD*.[1] In our WoK-HD, we automate the construction of a conceptual structure superimposed over images of historical documents that enables the document repository to serve as a queriable repository of facts. To provide a firm foundation for our WoK-HD we adopt and adapt fundamental tenets from the philosophical disciplines of ontology, epistemology, logic, and linguistics and to make it computationally viable we integrate these ideas with conceptual modeling fundamentals for information systems. Thus, the conceptual structure is a linguistically-grounded ontology (as described in Section 2), enabling it to commit to the existence of objects and their properties based on text obtained from document images via an OCR engine. Epistemologically, the conceptual structure, populated with extracted facts, is a formal predicate-calculus theory, which immediately enables logic reasoners to infer facts that are implied, but not directly stated in the historical documents (as described in Section 3). The application of linguistically-grounded ontologies to free-form user queries leads to the automatic generation of formal queries over the populated conceptual structure (as described in Section 4). In addition to returning results to free-form user queries, the WoK-HD also provides epistemological truth authentication: when asked, it displays reasoning chains for any inferred facts and also displays original document images with relevant extracted facts highlighted (also described in Section 4).

[1]We call our proposed system *WoK-HD* (*W*eb *o*f *K*nowledge for *H*istorical *D*ocuments) because it is part of our larger WoK project [12] in which we superimpose a web of data (in the semantic web sense [24]), over a general collection of documents—in this case over historical-document images with text made machine readable by an OCR engine.

Our contribution in this paper is the WoK-HD implementation, which we describe here. Prior work has focused on extraction from multiple-record web documents [10], free-form query processing [2], the use of 3rd-party reasoners over extracted genealogical data [25], named-entity recognition in OCR'ed historical documents [16], and advocating the WoK vision [12]. However, this is the first time we have been able to focus on fact extraction in historical documents and the first time we have been able to put the various pieces together in a single implemented system to create a WoK-HD—the superimposition of extraction ontologies, free-form query processing, and result authentication over a data repository of historical documents. In Section 5 we give our current implementation status, and we report the results of some initial experimentation. Although we have accomplished much, many opportunities for further work still await us (Section 6). We conclude in Section 7.

2. FACT EXTRACTION

Facts are instantiated first-order-logic predicates. Examples:

- *Person*(*osmx17*) is a one-place predicate, where *osmx17* is the object identifier for some person.[2]
- *Person*(*osmx17*) *has Name*("*William Gerard Lathrop*") is a two-place predicate, written here using infix notation with whitespace allowed, rather than in the more common prefix notation: *hasName*(*osmx17*, "*William Gerard Lathrop*").
- *Person*(*osmx17*) *married Person*(*osmx193*) *on Marriage Date*(*1837*) is a three-place predicate.

[2]All our object identifiers have the form $osmx(\backslash d)^*$—"osm" for Object-oriented Systems Modeling [11], the conceptual-modeling language we use, and "x" for XML, the language in which we represent our populated conceptual models.

Predicate names are ontological statements of what objects can exist for one-place predicates and statements of what relationships among objects can exist and how existing objects can relate for n-place predicates ($n \geq 2$). Instantiated predicates therefore are statements of fact—epistemological knowledge assertions.

2.1 Ontologies as Frameworks for Facts

Ontology is the study of existence. Its most fundamental question is "What exists?" We answer computationally for our WoK-HD by declaring a conceptual-model instance. The conceptual model we use for the WoK-HD is *fact-oriented*, meaning that each declared object set corresponds to a one-place first-order-logic predicate and each declared n-ary relationship set corresponds to an n-place predicate. In the context of our WoK-HD, we call our fact-oriented conceptual-model instances *ontologies* because they are intensional statements about what can exist.

Epistemology is the study of knowledge. One of its fundamental questions is "What is knowledge?" For our WoK-HD, we answer computationally by saying that a populated conceptual-model instance constitutes knowledge. WoK-HD ontologies are frameworks for facts, which, when instantiated, constitute atomic epistemological knowledge statements. Figure 3 is a screenshot of our Ontology Workbench with an ontology on the left and a document to be annotated on the right. When a user clicks on the go button, the workbench automatically annotates the document and adds it to the WoK-HD.

In the ontology in Figure 3, boxes denote object sets—dashed boxes for lexical object sets and solid boxes for non-lexical object sets. Elements of lexical object sets are text strings (e.g., "Lathrop" could be an element of the lexical object set *Surname*, and both "William" and "Gerard" could be elements of *Given Name*). Elements of non-lexical object sets are object identifiers (e.g., *osmx17* could be an element of the non-lexical object set *Person*, and *osmx443* could be an element of Name). Object-set names are one-place predicate names.

Lines connecting object sets denote binary relationship sets, and diamonds with n incident lines connecting to object sets denote n-ary relationship sets. Names for binary relationship sets with reading direction arrows next to relationship lines are a space-delimited concatenation of the tail-side object-set name, the name associated with the reading-direction arrow, and the head-side object-set name (e.g., *Person died on Death Date*). Names for n-ary relationship sets are specified fully and must include the object-set name for each of its connections. Relationship sets without specified names have default names: an alphabetized hyphen-delimited concatenation of the space-elided object-set name for each connection (e.g., MarriageDate-Person-Person for the ternary relationship set in Figure 3). Relationship-set names are n-place predicate names, which we write in infix notation (e.g., *Person*(x) *has Birth Date*(y)).

Ontological constraints come in several varieties. A white triangle denotes a generalization/specialization, an is-a relationship; generalizations attach to the apex of a triangle and specializations attach to the base (e.g., in Figure 3 *Child* is a specialization of *Person*). An is-a relationship constrains the set of objects in a specialization to be a subset of the objects in a generalization. A black triangle denotes aggregation and indicates that an aggregate object includes its connected component objects. Formally, however, a black triangle is just a template for a collection of *is part of* relationship sets. Thus, the aggregation in Figure 3 stands for two relationship sets whose names are *Given Name is part of Name* and *Surname is part of Name*. Every relationship-set/object-set connection has participation constraints, either explicitly given or given by default. The participation constraint *1:2* associated with the connection of the object set *Child* and the relationship set *Child has parent Person* constrains child objects to associate with either one or two person objects (no more and no less). Participation constraints with variables must satisfy given constraints over the variables. Thus, in Figure 3, since $a + b > 0$, name aggregates must have either one or more given names or a surname or both. In the absence of explicit participation constraints, decorations on connections determine cardinality constraints. Arrowheads constrain relationship sets to be functional from tail(s) to head(s) (e.g., a person can be born on only one date and a marriage takes place on only one date). Circles on connections denote optional participation, so that in our ontological specification in Figure 3, for example, a person need not participate in a marriage and need not have a death date (e.g., because the date may be unknown), but must have a name.

2.2 Linguistically Grounded Ontologies

Besides "What is knowledge?" another fundamental question of Epistemology is "How is knowledge acquired?" For our WoK-HD, we answer by grounding our ontologies linguistically [6]. *Linguistics* is the study of human language, and one of its fundamental questions is "How does language convey meaning?" To provide for meaning conveyance, we assign a data frame [9] to each object set and each relationship set in an ontology. In artificial intelligence, a *frame* is a data structure for representing a stereotyped situation (e.g., a room or a child's birthday party) [14]. A *data frame* is a data structure for representing a data item such as a date—i.e., everything about a date: how a date appears in text in all of its various forms (e.g., "May 12, 1835" or "Easter 1874"), how a date is represented internally (e.g., as a Julian date, 1847032 for February 1, 1847), and what operations apply (e.g., *between*(x, y, z):*Boolean* for date x is between dates y and z). A data frame ties ontology components to language and thus grounds them linguistically.

We call linguistically grounded ontologies *extraction ontologies* because their data frames enable our extraction system (*OntoES*)[3] to extract facts from a document (e.g., from the Ely page in Figure 3) and to populate an ontology (e.g., the conceptual-model instance in Figure 3) with these facts. In ontological terms, data frames are about *ontological commitment*. Because extraction ontologies populate three different types of conceptualizations—lexical object sets, non-lexical object sets, and relationship sets—data frames specify ontological commitment in three different ways.

[3]As the tab in the Ontology Workbench in Figure 3 shows, we call the system that invokes extraction ontologies on documents *OntoES* (***Onto**logy **E**xtraction **S**ystem*).

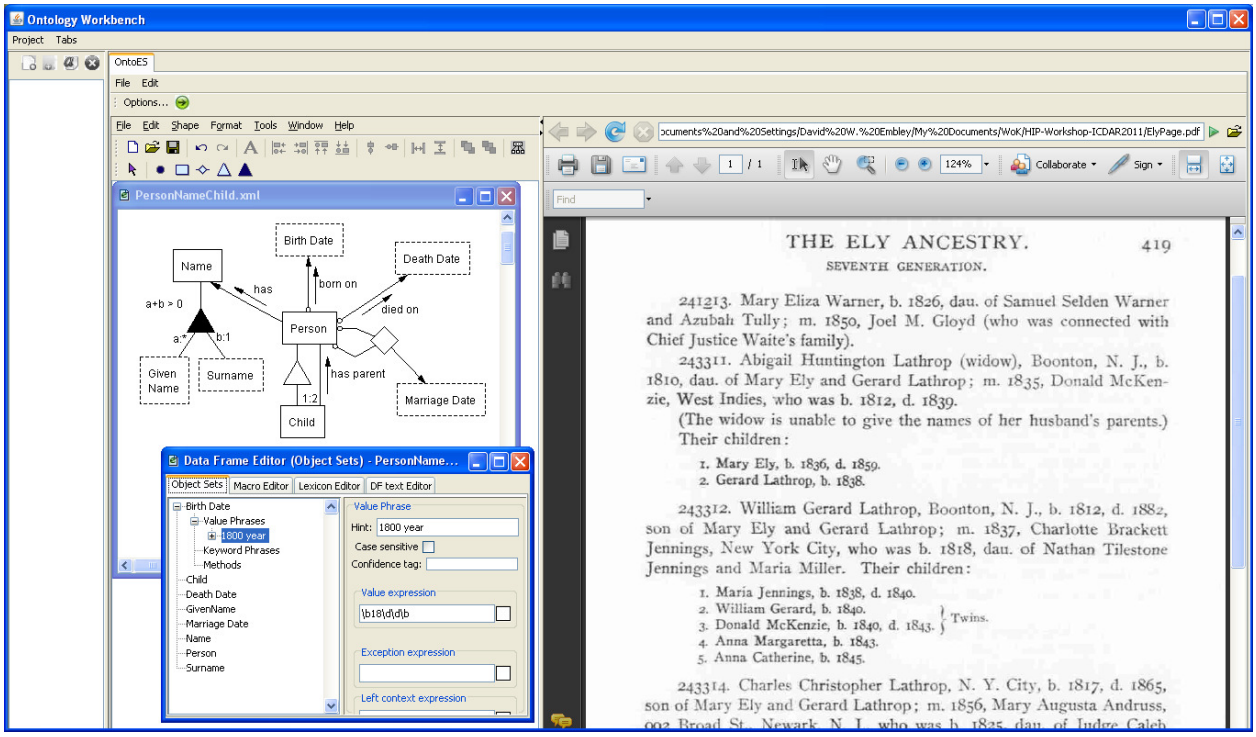


Figure 3: Screenshot of Workbench with Extraction Ontology Ready to Apply to Ely Page

.

Ontological Commitment for Lexical Object Sets. The elements in lexical object sets are strings that represent themselves. Thus, we merely need to recognize a textual string as potentially belonging to a lexical object set to attain ontological commitment. In OntoES, we mainly use regular expressions as recognizers. In Figure 3, for example, in the *Data Frame Editor* window, the regular expression *\b18\d\d\b*, recognizes years in the 1800s—more *Value expressions* can be added, as needed. The data-frame interface in Figure 3 also shows some additional recognition features. Under the *Lexicon Editor* tab, users can declare lexicons, which are simple lists of values (e.g., country names); as such they are special kinds of regular-expression recognizers, although we process them differently for efficiency reasons. Under the *Macro Editor* tab, users can declare and name regular expressions that are useful as components of more complex regular expressions. *Exception expressions* (which users can add in the *Value Phrase* form) eliminate strings that a value expression may recognize but are not wanted. *Left* and *Right context expression*s and *Keyword Phrases* can help ensure the correctness of extracted values. In addition to recognizing lexical elements that belong to an object set, a data frame also converts values into an internal representation (especially for use in applicable *Methods*), provides a standard string representation for display, and stores annotation information for provenance purposes (a reference to a cached page from which a value is extracted and the location on the page of the extracted value).

Ontological Commitment for Non-Lexical Object Sets. Elements in non-lexical object sets are object identifiers. As is typical in philosophical ontology, it is non-trivial, in general, to determine when to commit to an object's existence—to determine when OntoES should generate an object identifier. In our earlier work on multiple-record web documents [10] (e.g., for car ads or obituaries), we assumed that each record was for an object (e.g., each car ad for a car and each obituary for a deceased person). In general, however, we must answer the question, "What lexical information provides sufficient evidence for the extraction ontology to commit to the existence of an object?" Data frames for non-lexical object sets provide for declarations of ontological commitment, i.e., let users state what lexical evidence must exist for generating an object. In many documents, and particularly in historical documents, proper nouns denote existence. In the Ely page in Figures 1–3, the proper noun "William Gerard Lathrop" denotes the existence of a person and the proper noun "Boonton, N. J." denotes the existence of a location. Further, the constraints in the ontology in Figure 3 require that a person have a name and that a name (which is itself non-lexical) have either one or more given names or a surname or both. Hence, we declare in the *Person* data frame that the existence of a new *Name* object commits the ontology to generate a new *Person* object, and the existence of name strings in the text being processed commits the ontology to generate a new *Name* object. As an example of another kind of ontological commitment for non-lexical object sets, if we were to choose to model *Marriage Event* as a non-lexical object the ontology in Figure 3 with related object sets *Person* (twice) and *Marriage Date*, the existence of two people and a marriage date would imply the existence of a marriage event.

Ontological Commitment for Relationship Sets. Recognizers in OntoES for relationship sets are regular expressions with slots for lexical or non-lexical objects. Examples:

{Person}\sb\.\s{Birth Date}
{Child}.$^{+}$son\sof.$^{+}$and\s{Person}

where braces enclose references to connected object sets. Thus, ontological commitment for relationship sets depends on ontological commitment for object sets. Interestingly, ontological commitment for object sets can also depend on ontological commitment for relationship sets. When the *son of* regular expression above matches a textual phrase like "William Gerard Lathrop ... son of ... and Gerard Lathrop", this match also signals the ontological commitment of a new *Child* object and, since *Child* is a specialization of *Person*, also the propagation of the new *Child* object to the *Person* object set.[4]

3. INFERRED FACTS

Because our populated ontologies constitute predicate-calculus theories, we can immediately apply any logic inference engine to obtain inferred facts.[5] In our Ontology Workbench, a user can click on the *Tools* menu (see Figure 3) to bring up a *Rule Editor* that lets a user declare rules based on predicates and rule heads of other declared rules in an ontology. Thus, for example, based on the ontology in Figure 3, we can declare a rule for *grandparent of*:

Person(x) *is grandparent of* *Person*(y) :-
Child(y) *has parent* *Person*(z),
Child(z) *has parent* *Person*(x).

When a user declares rules, the system adds structure to the ontology: for this *grandparent-of* rule, for example, it adds a recursive *Person-Person* relationship set. Then, after OntoES extracts basic facts, it runs the inference rules and populates the added structures. It would, for our example here for instance, insert a relationship from the person identifier for Mary Ely to the person identifier for Maria Jennings and place the relationship in the *Person is grandparent of Person* relationship set.

From the Ely page in Figure 1, a reader can infer a number of facts. A reader can easily discern, for example, that Mary Ely is the grandmother of the five Lathrop children: Maria, William, Donald, Anna Margaretta, and Anna Catherine. For some "easily discernible" facts it is possible to write simple inference rules as just described, but for others there may be better ways and for still others additional facilities may be needed to obtain all the facts "easily discernible" by a reader. For example, should we extract the fact that the surname of the grandchildren is "Lathrop", which in the Ely page is stated only indirectly as the last name of the father of the family, or should we infer it? And, for example, how can we obtain gender information so that we know that Mary is the grandmother, and not the grandfather? The obvious "son of" and "dau. of" clues can help, but in the Ely page, these clues are explicit only for the parents, not the grandparents and not the children.

Through the use of features already available in OntoES in combination with inference rules, it is possible for users to resolve these issues and others like them. For gender, as an example, the inference is indirect either through gender-specific relationships or through knowledge of gender-specific given names. For gender-specific relationships, for example, we can add the specialization *Son* in addition to *Child* in the ontology in Figure 3 and add inference rule *Person*(x) *has Gender*("*Male*") :- *Son*(x). And for gender-specific names we can add the object set *Female Given Name* to the ontology along with the inference rule *Person*(x) *has Gender*("*Female*") :- *Person*(x) *has Name*(y), *Given Name*(z) *is part of Name*(y), *Female Given Name*(z). For distinguishing given names and surnames, one possible resolution is to make use of canonicalization methods, which, as mentioned earlier, is a way to convert extracted strings into internal representations. When, for example, a recognizer picks up the full name of a person, a canonicalization method can divide it into given names and a surname. And, in addition, when the surname is implicit, as it is for the children of a family in the Ely page, the following rule for surnames could be added: *Surname*(x) *is part of Name*(y) :- *Person*(z) *has Name*(y), *Child*(z) *has parent* *Person*(w), *Person*(w) *has Gender*("*Male*").

4. QUERY INTERFACE

Suppose a user enters the free-form query: *grandchildren of Mary Ely*. Figure 4 shows the results along with the authentication displayed when a user clicks on *Maria Jennings* in the table of returned results. The authentication is a display of the reasoning chain explaining why the system believes that Maria Jennings is a grandchild of Mary Ely. Highlighted in the displayed Ely page are the extracted base facts supporting the reasoning chain.

The motivation for authentication traces its roots to Plato who insists that knowledge should be justified [17]. "Knowledge" without some sort of truth authentication can be confusing and misleading. For our WoK-HD, we thus attempt to establish truth via provenance and authentication. And, although we cannot guarantee that rules and facts in sources are genuine, we can expose them and let users decide for themselves whether the results returned are valid.

The motivation for supporting free-form queries is clear: untrained users are incapable of formulating structured queries, and even trained users are incapable when neither the type of database nor the schemas of the data are known. Although necessary, processing free-form queries automatically is challenging. To explain our approach for the WoK-HD project, consider the query: *Mary Ely's grandchildren who died before reaching the age of eight.* The key idea is to apply extraction ontologies to queries. Linguistically, meaning conveyance is a matter of matching query words and phrases to object and relationship sets (given or inferred) and to methods applicable to these object and relationship sets. Then, the system can generate the equivalent of *select-project-join* queries over conceptual-model instances by joining over the subgraph of matched object and relationship

[4]Note that care must be taken not to generate extra objects by invoking multiple rules for object commitment for the same recognized textual string. Thus, for example, although there is evidence that "William Gerard Lathrop" denotes a person via both *Person has Name* and *Child is-a Person*, OntoES only generates one person object. Different appearances of the same name denoting the same person is another matter—and is to be resolved through determining object identity, not by fact extraction (see Section 6).

[5]In our implementation we generate OWL ontologies, populate them with RDF triples, and reason with the Pellet inference engine.

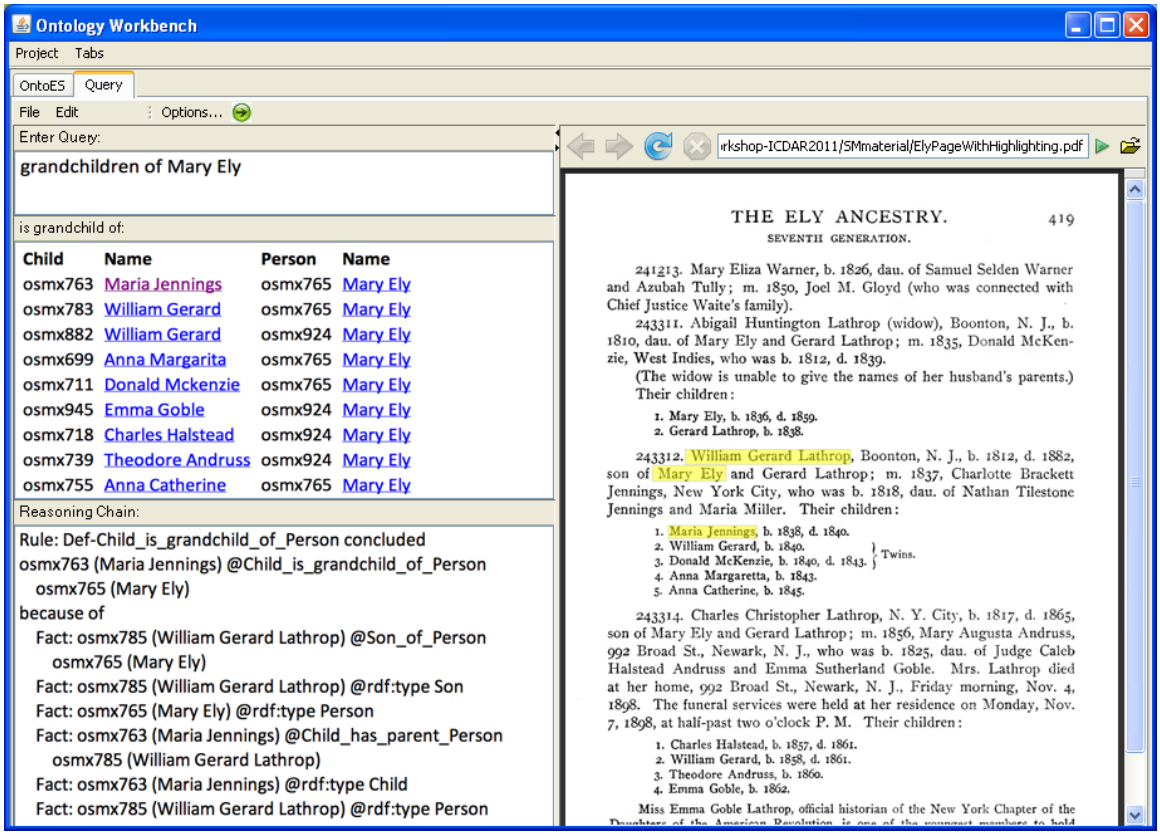


Figure 4: Query Results with Reasoning Chain Grounded in Extracted Facts

sets, selecting both by doing equality restrictions for identified lexical-object-set instances and by applying identified Boolean methods, and projecting on matched lexical object sets in the subgraph. For example, the system can obtain the following select, project, and join expressions for the query about Mary's grandchildren by applying an extraction ontology like the one in Figure 3, albeit augmented to include inference rules and to have a rich set of recognizers not only for object and relationship sets, but also for methods and inferred object and relationship sets.

select with: *Given_Name = "Mary" ∧ Surname = "Ely" ∧ Age_in_Years(Birth_Date, Death_Date) < 8*

project on: *Given Name*(*of grandchild*), *Surname*(*of grandchild*), *Birth Date*(*of grandchild*), *Death Date*(*of grandchild*), *Given Name*(*of grandparent*), and *Surname*(*of grandparent*).

join over: *Given Name is part of Name*, *Surname is part of Name*, *Person has Name*, *Person has grandchild Person* (an inferred relationship set), *Person born on Birth Date*, and *Person died on Death Date*.

Derivation of *Age_in_Years* requires sophisticated data-frame recognizers as detailed in [2]. Determination of projection on the names of the grandchildren is interesting because it involves the inverse of ontological commitment—recognizing that a name should be written to identify a person. Other projections are for all mentioned lexical object sets in the select or join statements. Implied also are projections for non-lexical object sets. Joins are over a path among all mentioned object sets. Multiple uses of the same relationship set must observe appropriate natural-join renaming conventions. Ambiguity resolution for cyclic hypergraphs requires either path recognition from phrases in the query via relationship-set data-frame recognizers or user interaction.

5. EVALUATION

We have largely implemented the WoK-HD as described here, and we are diligently working to complete the basic, full-line implementation. Users can declare extraction ontologies with an ontology editor implemented within the Ontology Workbench as Figure 3 shows. Specifically, they can declare conceptual-model instance and specify data frames for both object sets and relationship sets including instance recognition, canonicalization of recognized values, applicable method declaration, and recognizers for method applicability. OntoES runs and populates declared ontologies with facts as described in Section 2. Users can invoke a rule editor through the Ontology Workbench and declare inference rules (although we still need to allow for declaring more sophisticated rules). At the completion of an OntoES run, the system can call an inference engine, which produces all facts inferred from extracted facts according to declared inference rules as described in Section 3. The system is also able to evaluate free-form queries ([2]). However, integrating previous work on free-form queries into the Ontology Workbench is only partial, as is work on authentication display. Thus, we are unable to provide more than the mock-up that appears in Figure 4 even though much of the implementation

described in Section 4 is working in separate tools.

Based on our implementation, we declared an extraction ontology similar to the one in Figure 3, except that it also includes *Son* and *Daughter* specializations of *Person* and *Son of Person* and *Daughter of Person* relationship sets, and it does not include the marriage relationship set (the code for n-ary relationship sets, $n > 2$, is not yet complete). We ran it against the Ely page in Figure 2 and populated the ontology with 103 facts. OntoES extracted these 103 facts in about 10 seconds. Checked manually, these results have a recall ratio of 0.91 (97 correct facts of the 107 ground-truth facts) and a precision ratio of 0.94 (only six false positives) for a combined harmonic-mean F-measure of 0.93. We also declared inference rules to find grandchildren and ran our inference engine and found 31 of the 40 grandchild facts for the three families on the Ely page (which can be seen most clearly in Figure 4). Failure to find the additional nine was a result of the OntoES's failure to extract all base facts. Inferred facts are always correct with respect to the given base facts, although users, of course, may specify nonsense rules and rules that do not correspond to standard expectations (which is why reasoning-chain authentication is important). We also ran the extraction ontology against Pages 420–426 of *The Ely Ancestry* and extracted 859 facts. We checked recall and precision manually, yielding 0.59 and 0.71 respectively, which is lower than for Page 419, but this was expected since our recognizers were created with family descriptions of the Ely page in Figures 1–4 in mind. The majority of the missed facts were linked to not correctly recognizing names and bad OCR. When this happened, the system became confused and used the closest available name or date to form a relationship.

6. CURRENT AND FUTURE WORK

Our current task is to complete the work of integrating all code within our Ontology Workbench and to make the full-line from extraction through inference, query processing, and authentication operational. Our future work, which we have already begun, is to make various components work better. Our efforts are particularly directed to three major enhancements: (1) hybrid keyword and semantic search, (2) semi-structured text readers, and (3) integration of fact-filled ontologies and determining object identity within and across fact-filled ontologies.

Hybrid Keyword and Semantic Search—HyKSS. HyKSS [27] integrates standard keyword processing with semantic search. In addition to returning results for free-form queries as described here, HyKSS ranks pages from the repository in which semantic results and keyword hits are found. It uses a combined keyword and semantic ranking score in which the relative weights of keywords and semantics depend on the proportion of each in the user query. HyKSS also has an advanced-search facility, in which it dynamically generates a form depending on the ontology (or ontologies) applicable to a user request, prepopulates the form with what it understands from the user request, and allows a user to alter the content of the form—in particular in a way to be able to specify disjunctive and negation queries, as well as standard conjunctive queries. Integrating HyKSS into our Ontology Workbench is underway.

Semi-structured Readers. Knowledge engineering, by hand, is expensive. We thus have sought for ways (e.g., [20, 21]) and continue to seek for ways to generate extraction ontologies in an unsupervised or semi-supervised fashion by taking advantage of the semi-structured nature of data-rich documents. The lists in the Ely page in Figures 1–4 are examples. From an ordinary form specified by a user and after filling in the form for one list entry (e.g., the family description for the William Gerard Lathrop family in the middle of the Ely page), we can generate an extractor for the list entry. To the extent other list entries have the same pattern, we can also extract from them. The challenge is to have the system also be able to dynamically adjust its list-entry extractor in the face of variations, asking for occasional help only when it sees that variations are ambiguous or contain information beyond the data filled in for the initial list entry. We further see that, using techniques similar to those of Divali, et al. [8], this automated list reader should be able to generate an extractor in an unsupervised fashion from (possibly noisy and likely incomplete) extraction results obtained from OntoES. In addition to cutting the cost of knowledge engineering, we conjecture that we can also increase extraction accuracy. Exploiting semi-structured patterns has a better chance of being tolerant of OCR errors (e.g., of accepting the *i800* in Figure 2) and is less dependent on dictionaries for named-entity extractors (e.g., of accepting the erroneous "c" as an "e" in *Tilcstone*)

Integration and Object Identity. Extraction ontologies designed with respect to the way facts are conveyed in a document or generated automatically according to semi-structured patterns in text are not likely to be organized in an ideal way. For the extraction ontology used for our evaluation, for example, we extracted names as they appeared in the text, but would prefer to have names decomposed as in the ontology in Figure 3. Further, we extracted *son of* and *dau. of* facts, but would prefer to only have *child of* facts along with gender information. We can use techniques developed in previous work [26] to automate the generation of schema mappings and data-transformation rules. Besides transforming facts to conform to a predeclared ontology, we may wish to integrate ontologies to create a larger application ontology (particularly for ontologies generated automatically from semi-structured sources such as tables as we have described in prior work [22]). An additional challenge, particularly for historical documents, is to resolve object identity—e.g., to determine that all the grandmother Mary Ely's on the Ely page in Figures 1–4 are the same, but the one child Mary Ely in the Abigail Huntington Lathrop family is different. Our approach is to compare the extracted attribute information for each identified person to resolve object identity as we have done in previous work [1].

7. CONCLUDING REMARKS

We have largely achieved our objective a creating a WoK-HD (a web of knowledge super-imposed over a database of images of historical documents). In the pilot field experiment we ran, OntoES extracted hundreds of family facts from eight pages of the on-line copy of the 1902 *The Ely Ancestry* [4] with an F-measure accuracy of 0.68. All inferred grandchild facts were necessarily correct with respect to the extracted base facts. The WoK-HD enables search for both facts and implied facts through its free-form query inter-

face. Results returned, in addition to the facts that satisfy a query, include display authentication information, which consists of reasoning chains grounded in facts highlighted in original document images. The full-line implementation is nearly complete, but we still have work to do to integrate free-form query processing and authentication display into our WoK-HD. Nevertheless, the results we already see are encouraging and show the practical viability of a WoK-HD.

8. ACKNOWLEDGMENTS

This research was supported in part by the National Science foundation under grants #0414644 and #0083127. We also acknowledge the many BYU students—too numerous to list—who have worked on OntoES and the WoK project.

9. REFERENCES

[1] R. Al-Kamha and D.W. Embley. Grouping search-engine returned citations for person-name queries. In *Proceedings of the ACM Sixth International Workshop on Web Information and Data Management (WIDM 2004)*, pages 96–103, Washington, DC, November 2004.

[2] M. Al-Muhammed and D.W. Embley. Ontology-based constraint recognition for free-form service requests. In *Proceedings of the 23rd International Conference on Data Engineering (ICDE'07)*, pages 366–375, Istanbul, Turkey, April 2007.

[3] F. Baader and W. Nutt. Basic description logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter 2, pages 43–95. Cambridge University Press, Cambridge, UK, 2003.

[4] M.S. Beach, W. Ely, and G.B. Vanderpoel. *The Ely Ancestry*. The Columer Press, New York, New York, 1902. on-line book: http://www.archive.org/details-/elyancestrylinea00beac.

[5] M.W. Bilotti, P. Ogilvie, J. Callan, and E. Nyberg. Structured retrieval for question answering. In *Proceedings of SIGIR 2007*, pages 351–358, 2007.

[6] P. Buitelaar, P. Cimiano, P. Haase, and M. Sintek. Towards linguistically grounded ontologies. In *Proceedings of the 6th European Semantic Web Conference (ESWC'09)*, pages 111–125, Heraklion, Greece, May/June 2009.

[7] M.J. Cafarella. Extracting and querying a comprehensive web database. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR'09)*, Monterey, California, 2009.

[8] N. Dalvi, R. Kumar, and M. Soliman. Automatic wrappers for large scale web extraction. *Proceedings of the VLDB Endowment*, 4(4):208–218, January 2011.

[9] 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.

[10] 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, 1999.

[11] 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.

[12] D.W. Embley, S.W. Liddle, and D.W. Lonsdale. *Conceptual Modeling Foundations for a Web of Knowledge*, chapter 15, pages 477–516. Springer, Heidelberg, Germany, 2011.

[13] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A.A. Kalyanpur, A. Lally, J.W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty. Building Watson: An overview of the DeepQA project. *AI Magazine*, Fall, 2010.

[14] M. Minsky. A framework for representing knowledge. In P.H. Winston, editor, *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill, 1975.

[15] G. Nagy. On the frontiers of OCR. *Proceedings of the IEEE*, 40(8):1093–1100, 1992.

[16] T.L. Packer, J.F. Lutes, A.P. Stewart, E.K. Ringger D.W. Embley, K.D. Seppi, and L.S. Jensen. Extracting person names from diverse and noisy ocr text. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data Documents (AND 2010)*, pages 19–26, Toronto, Ontario, Canada, October 2010.

[17] Plato. *Theaetetus*. BiblioBazaar, LLC, Charleston, South Carolina, about 360BC. (translated by Benjamin Jowett).

[18] D. Sànchez. A methodology to learn ontological attributes from the web. *Data & Knowledge Engineering*, 69:573–597, 2010.

[19] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[20] C. Tao and D.W. Embley. Automatic hidden-web table interpretation, conceptualization, and semantic annotation. *Data & Knowledge Engineering*, 68(7):683–703, July 2009.

[21] C. Tao, D.W. Embley, and S.W. Liddle. FOCIH: Form-based ontology creation and information harvesting. In *Proceedings of the 28th International Conference on Conceptual Modeling (ER2009)*, pages 346–359, Gramado, Brazil, November 2009.

[22] Y.A. Tijerino, D.W. Embley, D.W. Lonsdale, Y. Ding, and G. Nagy. Toward ontology generation from tables. *World Wide Web: Internet and Web Information Systems*, 8(3):261–285, 2005.

[23] J. Turmo, A. Ageno, and N. Català. Adaptive information extraction, July 2006.

[24] Semantic Web. www.w3.org/standards/semanticweb.

[25] C. Woodbury. Automatic extraction from and reasoning about genealogical records: A prototype. Master's thesis, Department of Computer Science, Brigham Young University, August 2010.

[26] L. Xu and D.W. Embley. A composite approach to automating direct and indirect schema mappings. *Information Systems*, 31(8):697–732, December 2006.

[27] A. Zitzelberger. HyKSS: Hybrid keyword and semantic search. Master's thesis, Department of Computer Science, Brigham Young University, Provo, Utah, USA, July 2011.