FROntIER: A Framework for Extracting and Organizing Biographical Facts in Historical Documents

by

Joseph Park

A thesis proposal submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

February 2013

FROntIER: A Framework for Extracting and Organizing Biographical Facts in Historical Documents

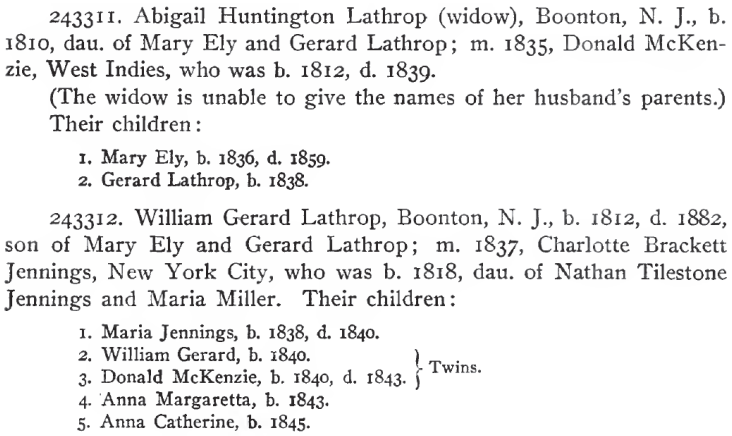
Joseph Park

February 2013

Abstract. The tasks of entity recognition through ontological commitment, fact extraction and organization in conformance to a target schema, and entity deduplication have all been examined in recent years, and systems exist that can perform each individual task. A framework combining all these tasks, however, is still needed to accomplish the goal of automatically extracting and organizing biographical facts about persons found in historical documents into disambiguated entity records. We propose FROntIER (**F**act **R**ecognizer for **Ont**ologies with **I**nference and **E**ntity **R**esolution) as a framework to recognize and extract facts using an ontology and organize facts of interest by inferring implicit facts using inference rules, a target ontology, and entity resolution. Our evaluation consists of precision and recall over facts of interest, which include person entities and their respective attributes, and clustered entities. We also propose a predictive evaluation of the time and effort required to use FROntIER to extract and organize facts of interest at a determined level of precision and recall for a corpus consisting of 50,000+ family history books in the LDS Church's online collection.

1 Introduction

Historians, genealogists, and others have great interest in extracting facts about persons and places found in historical documents and organizing these facts into disambiguated entity records. Figure 1, for example, shows part of a page from *The Ely Ancestry* [BEV02]. Facts of interest include those explicitly stated such as *William Gerard Lathrop was born in 1812*, *married Charlotte Brackett Jennings in 1837*, and *is the son of Mary Ely*. In addition to explicitly stated facts, implicit facts are also of interest. These include the fact that *William Gerard Lathrop has gender Male*, inferred from the stated fact that he is a son, and *Maria Jennings has surname Lathrop*, inferred from cultural tradition and the stated fact that her father has the surname Lathrop. Implicit facts also include disambiguating references to objects: the first and third Mary Ely mentioned are the same person, but not the same person as the second Mary Ely mentioned.



243311. Abigail Huntington Lathrop (widow), Boonton, N. J., b. 1810, dau. of Mary Ely and Gerard Lathrop; m. 1835, Donald McKenzie, West Indies, who was b. 1812, d. 1839.

(The widow is unable to give the names of her husband's parents.)

Their children:

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

243312. William Gerard Lathrop, Boonton, N. J., b. 1812, d. 1882, son of Mary Ely and Gerard Lathrop; m. 1837, Charlotte Brackett Jennings, New York City, who was b. 1818, dau. of Nathan Tilestone Jennings and Maria Miller. Their children:

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

Figure 1: An Excerpt from Page 419 of *The Ely Ancestry*.

Automating the process of extracting stated facts, inferring implicit facts, and resolving object references is a difficult task. Sarawagi [Sar08] surveys much of the work of the last decade or so that has been done in automated information extraction of facts from unstructured and semi-structured text. For inferring implicit facts, work dates back to Aristotle and is typified nowadays by the work in description logics [BCM+03], which describes research on methods for defining first-order logics, proving soundness and decidability, and inferring facts from existing facts. To help disambiguate object references—solve the record linkage or entity resolution problem—researchers often resort to the use of statistical methods, which include machine learning algorithms [Chr12]. Though much has been accomplished and still more can be done to thoroughly examine these issues, what is lacking most is tying them together into a unified, synergistic whole—a framework.

In answer to this lack of a unifying framework, we propose FROntIER (**F**act **R**ecognizer for **Ont**ologies with **I**nference and **E**ntity **R**esolution) as a framework to automatically extract and organize facts from historical documents into disambiguated entity records. FROntIER makes use of extraction ontologies [ECJ+99, ELL11] to automatically extract stated facts of interest using regular expression patterns and dictionaries. Once stated facts of interested have been recognized and properly associated with an extraction ontology, FROntIER disambiguates objects, infers

additional facts about these objects, and organizes the objects and facts about these objects with respect to a target ontology. FROntIER's extraction ontologies allow users to model text and layout as it appears in historical documents, while FROntIER's target ontologies model knowledge of interest to be gleaned by historians—facts both directly and indirectly stated. To see the difference, compare the target ontology in Figure 2, which is an ontological view of biographical facts of a person, against the extraction ontology in Figure 3, which models how explicitly stated biographical facts appear in *The Ely Ancestry*. FROntIER uses pattern-based extractors to identify the existence of objects and their interrelationships according to the particular layout in the text document, and uses logic rules to organize extracted facts in a target ontology. FROntIER, for example, extracts the stated "son of" and "dau. of" facts into the *Son-Person* and *Daughter-Person* relationship sets in Figure 3 and then uses the inference rules "if Son, then male" and "if Daughter, then female" to populate the *Person-Gender* relationship set in Figure 2. Inference and organization also include entity resolution, which proceeds based on extracted and inferred facts. The first-mentioned Mary Ely in Figure 1, for example, is the grandmother of the second-mentioned Mary Ely, and therefore cannot be the same Mary Ely.

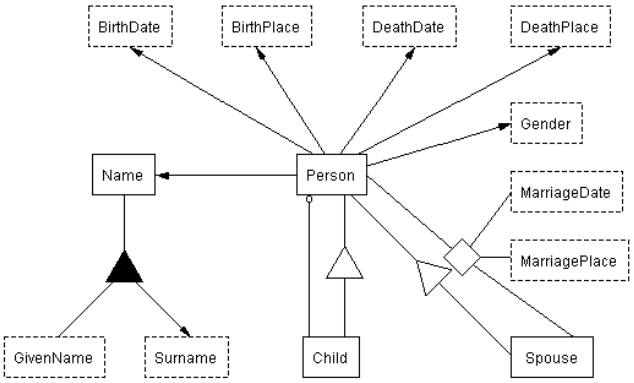


Figure 2: Target Ontology of Desired Biographical Facts.

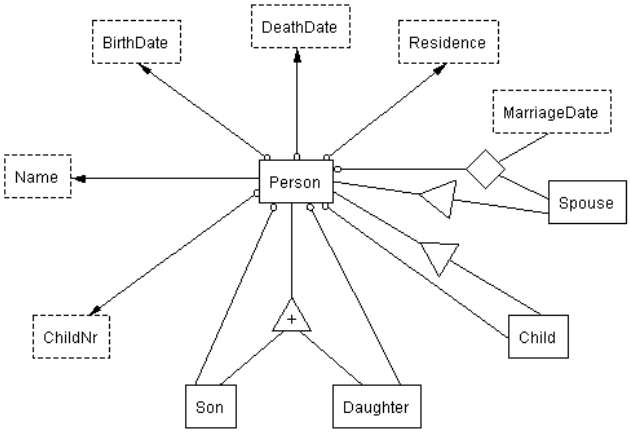


Figure 3: Extraction Ontology of Stated Biographical Facts of a Person in *The Ely Ancestry*.

The expected contribution of the thesis is the construction of a unified framework for extracting and organizing facts that will include:

1. Provisions for users to express relationship-based regular-expression extractors and record-based regular-expression extractors (in addition to the already existing entity-based regular-expression extractors);
2. Provisions for users to state object existence rules for identifying the existence of objects such as people;
3. Provisions for users to specify inference rules for obtaining inferred facts; and
4. Provisions for automatic, fact-based entity resolution.

With these framework provisions, FROntIER will be able to extract and organize both stated and implied facts found in OCRed historical documents. We plan to demonstrate these expected contributions by showing the precision and recall of extracted facts of interest, inferred facts of interest, and clustered entities for a sampling of a corpus of 50,000+ online historical books.

2 Related Work

Much work has been done in extracting and organizing facts of interest from documents, but only few of these efforts focus on both automatic fact extraction and record linkage (e.g. [GX09], [BGH09],[BBC+10], [BHH+11]). These systems, however, lack strong enough extraction capabilities and the use of inferred facts as well as extracted facts for doing record linkage. Our framework, FROntIER, strengthens weaknesses in extraction capabilities, adds the ability to infer implied facts of interest, and enables better attribute-based record linkage.

Much more effort has been spent on improving techniques to solve the individual tasks of FROntIER. Sarawagi [Sar08] has written a book that surveys current information extraction techniques. Turmo et al. [TAC06] have surveyed information extraction techniques, focusing more on statistical methods. Mishra and Kumar [MK11] have surveyed various semantic web reasoners and languages. For the use of inference, Baader et al. [BCM+03] have written a book on description logics. For resolving object references, Christen [Chr12] published a book on techniques for data matching, record linkage, and entity resolution. Herzog [HSW07] has published a book on deterministic and probabilistic record linkage techniques. Each of these books and surveys reference many dozens of research papers.

For FROntIER we select, build on, and synergistically combine this prior work, as follows:

- Our framework extends the capabilities of systems developed by the Data Extraction Research Group at Brigham Young University. Embley et al. [ECJ+99] developed a system, OntoES, for ontology-driven extraction with the aid of regular expression based recognizers over HTML pages. Liddle et al. [LHE03] built a development environment for the construction of ontologies called Ontology Editor. Wessman et al. [WLE05] further refined these systems by adding wrappers and facades to facilitate the development of ontologies and the organization of data.
- For the framework, we add the ability to infer implied facts by adding the Jena reasoner[1]. Our framework allows for the construction of inference rules that follow the syntax for Jena

[1] http://jena.apache.org/

rules, and we use the constructed inference rules with the Jena reasoner over extracted facts to organize facts with respect to a target ontology. Our framework also allows for user-defined predicates for use in inference rules by extending the built-ins framework provided by Jena.

- We include Duke[2], an off-the-shelf entity resolution tool, in our framework to aid in resolving entities. We generate *owl:sameAs* relationships between entities found in equivalence classes output by Duke.

3 Thesis Statement

FROntIER is an effective framework for ontology-based extraction of biographical facts of persons in historical documents, organizing facts with respect to a target ontology, and performing entity resolution to produce disambiguated entity records.

4 Project Description

The FROntIER framework has two key components: (1) extraction ontologies and (2) organization rules. We discuss the details of each in the subsections below, explaining how each will be built in order to complete the FROntIER framework.

As an overview, however, we first explain how the components fit together to comprise the FROntIER framework. Our target application is historical documents, which are OCRed pages in PDF format, and the input in FROntIER. FROntIER first invokes OntoES, our **Onto**logy **E**xtraction **S**ystem [ECJ$^+$99], which extracts information from pages of text documents and populates an ontology with recognized objects, object properties, and relationships among objects and object properties. The output of OntoES is an XML document containing these objects and relationships, which is converted into RDF[3] triples (in an OWL[4] ontology) to be processed by the Jena reasoner. Organization rules transform RDF triples into triples that conform to a target ontology and also disambiguate entities, producing *owl:sameAs* relationships to denote that entities are coreferent.

4.1 Extraction Ontologies

An extraction ontology is a linguistically grounded conceptual model. Figure 3 is an example of the conceptual model component of an extraction ontology. Figure 4 illustrates some of the linguistic grounding for the conceptualization in Figure 3. As extraction ontologies constitute the first key component of FROntIER, we briefly explain their details and say what we must develop as part of the thesis to complete the extraction ontology component of the FROntIER framework.

In a conceptual model diagram (e.g. Figures 2 and 3), each box represents an object set, or class of objects. Object sets can either be lexical (represented with dashed lines) or non-lexical (represented with solid lines). Lexical object sets contain strings whereas non-lexical object sets contain surrogates that denote real-world objects. Line segments connecting object sets denote relationship sets, which are usually binary, meaning they only connect two concepts together, but can also be n-ary ($n > 2$). For example, the line segments connecting the *Person*,

[2]http://code.google.com/p/duke/
[3]http://www.w3.org/RDF/
[4]http://www.w3.org/TR/owl2-overview/

MarriageDate, and *Spouse* object sets in Figure 3, which are intersected by a diamond shape, denote a ternary relationship set. Relationship sets can be functional, optional, or both as well as nonfunctional and mandatory. Arrowheads on the range side of relationship sets denote functional relationship sets and unfilled circles on the domain side denote optional participation of objects in relationships. An unfilled triangle denotes generalization/specialization with the generalization, or object set that represents the hypernym, connected to the apex of the triangle and the specializations, or object set that represents the hyponyms, connected to the base. The set of specializations of a generalization may be disjoint (represented by a '+' symbol as are *Son* and *Daughter* in Figure 3) or complete (represented by a '∪' symbol) or both disjoint and complete, constituting a partition. A black triangle denotes aggregation with the holonym object set connected to the apex of the triangle and the meronym object sets connected to the base.

Lexical Object Sets:
- Name
  - **external representation:** \b{FirstName}\s{LastName}\b
  - **external representation:** \b{FirstName}\s[A-Z]\w+\b
    - **left context:** \d{1,2}\.\s
  - ...
- Residence
  - **external representation:** \b{City},\s{State}\b
  - ...
- BirthDate
  - **external representation:** \b{Month}\.?\s*(1\d|2\d|30|31|\d)[.,]?\s*\b[1][6-9]\d\d\b
    - **left context:** b\.\s
    - **right context:** [.,]
    - **exclusion:** \b(February|Feb\.?)\s*(30|31)\b|...
  - **external representation:** \b[1][6-9]\d\d\b
    - **left context:** b\.\s
    - **right context:** [.,]
  - ...

Non-lexical Object Sets:
- Person
  - **object existence rule:** {Name}
  - ...
- Son
  - **object existence rule:** {Person}[.,]?.{0,50}\s[sS]on\b
  - ...

Relationship Sets:
- Person-BirthDate
  - **external representation:** ^\d{1,3}\.\s{Person},\sb\.\s{BirthDate}[.,]
  - ...
- Son-Person
  - **external representation:** {Son}[.,]?.{0,50}\s[sS]on\s+of\s.*?\s{Person}
  - ...
- Person-MarriageDate-Spouse
  - **external representation:** {Person}[.,]?.{0,50};\s*m[.,]\s{MarriageDate}[,]?\s*{Spouse}
  - ...

Ontology Snippets:
- ChildRecord
  - **external representation:** ^(\d{1,3})\.\s+([A-Z]\w+\s[A-Z]\w+) (,\sb\.\s([1][6-9]\d\d))?(,\sd\.\s([1][6-9]\d\d))?\.
  - **predicate mappings:** *Person-ChildNr*(x,1); *Person-Name*(x,2); *Child*(x); *Person-BirthDate*(x,4); *Person-DeathDate*(x,6)

Figure 4: Example Data Frames for Concepts in the Ontology in Figure 3.

The linguistic component of an extraction ontology consists of four types of instance recognizers—recognizers for lexical object sets, non-lexical object sets, relationship sets, and designated ontology snippets. Instance recognizers are embedded in data frames [Emb80]—abstract data types tied to concepts in an extraction ontology that, in addition to instance recognizers, contain operators that manipulate data values [EZ10]. Recognizers for the four types of data frames are similar, but are distinct in some characteristics. We explain each in turn. (Data frames for lexical object sets have been part of OntoES since its inception [ECJ$^+$99]. Data frames for non-lexical object sets, relationship sets, and ontology snippets are part of the development work for this thesis.)

Lexical object-set recognizers identify lexical instances in terms of external representations, context, exclusions, and dictionaries. External representations are regular expressions for specifying how instances may appear in text. For example, one of the external representations for *BirthDate* in Figure 4 is "\b[1][6-9]\d\d\b", representing years between 1600 and 1999. Left context is a regular expression that matches text found immediately before an instance pattern, and likewise, right context is a regular expression that matches text found immediately after an instance pattern. Examples are "b\.\s" as left context and "[.,]" as right context for *BirthDate* in Figure 4 for recognizing values for *BirthDate* in phrases such as "b. 1836," in Figure 1. Exclusions are regular expressions that match external representations; however, recognized exclusions are not accepted as valid instances. Thus, in Figure 4 dates such as *February 30* are invalid instances. Dictionaries are regular expressions where each entry in the dictionary is delimited by an OR ('|'), e.g. for *Name* in Figure 4 "(Abigail|Mary|William|...)" could be part of the *FirstName* dictionary. In general, braces around a name—e.g. {FirstName}—refer to a regular expression defined elsewhere.

Non-lexical object-set recognizers identify non-lexical objects through object existence rules. Object existence rules identify text, such as a proper noun, that designates the existence of an object. An example is a person's name. In Figure 4 "{Name}" references *Name*. When a name is recognized, OntoES generates a *Person* object and associates it with the recognized name. Object existence rules for non-lexical specializations identify roles for objects in its generalization. For example, "{Person}[.,]?.{0,50}\s[sS]on\b" in Figure 4 references *Person* and establishes the person recognized in the object-existence rule as a son.

Relationship-set recognizers identify phrases that relate objects. For example, the external representation for *Person-BirthDate* in Figure 4 is "^\d{1,3}\.\s{Person},\sb\.\s{BirthDate}[.,]" and represents a phrase that relates a person to a birth date. To process the recognizer, OntoES replaces "{Person}" and "{BirthDate}" with strings previously recognized for the *Person* and *BirthDate* object sets resulting in a regular expression such as "^\d{1,3}\.\s(Maria Jennings|William Gerard|...),\sb\.\s(1838|1840|...)[.,]" which OntoES uses, for example, to relate Maria Jennings to 1838 and William Gerard to 1840—two of the *Person-BirthDate* relationships that appear in Figure 1.

Ontology-snippet recognizers identify text patterns that provide instances for several object and relationship sets. Data frames for ontology snippets consist of external representations and predicate mappings. External representations are regular expressions with capture groups that map captured instances to ontology predicates—the object and relationship sets in the ontology. Variables for the mappings denote non-lexical objects, and integers denote captured lexical objects. As an example, suppose the ontology-snippet recognizer named *ChildRecord* in Figure 4 is applied to the first child list in Figure 1. The recognizer for *ChildRecord* would identify the patterns

"(1). (Mary Ely) (, b. (1836))(, d. (1859))." and "(2). (Gerard Lathrop) (, b. (1838))." where the parenthesized expressions represent captured groups numbered left to right by appearance of a left parenthesis. Predicate mappings for *ChildRecord* would generate the following objects and relationships for the ontology in Figure 3 for Mary Ely, the 5th person mentioned in Figure 1: *Child*($Person_5$); *Person-ChildNr*($Person_5$, "1"); *Person-Name*($Person_5$, "Mary Ely"); *Person-BirthDate*($Person_5$, "1836"); *Person-DeathDate*($Person_5$, "1859"); plus, as implied by referential-integrity and generalization/specialization constraints, *Person*($Person_5$), *Name*("Mary Ely"), *BirthDate*("1836"), and *DeathDate*("1859").

4.2 Organization Rules

FROntIER uses rules to organize facts in conformance to a target ontology (e.g. Figure 2). The kind of organization rules that FROntIER uses include canonicalization, inference, and entity-resolution rules. We briefly explain each kind of rule and say what needs to be developed as part of the thesis to complete the organization-rules component of FROntIER.

Canonicalization rules homogenize recognized instance values so that they can be manipulated and compared. For example, values for *BirthDate* in Figure 3 such as "1836" and "1832", which are strings, are canonicalized into an internal data type such as *Date*. Abbreviated *Name* values such as "Sam'l" and "Geo." can become "Samuel" and "George", and *Residence* values such as "New York City" and "Boonton, N.J." can be homogenized to a common format to become "New York, NY" and "Boonton, NJ". (Although canonicalization has been part of OntoES from its inception, constructing and executing canonicalization for transforming non-canonicalized source data to canonicalized target data is new and part of this thesis.)

In order to do inference, we convert target object and relationship instances into RDF triples so that we can use the Jena reasoner to reason over the triples. To conform with RDF syntactic requirements, we must normalize our ontologies as we convert them. We convert lexical object sets into non-lexicals with a *Value* property and convert n-ary relationship sets ($n > 2$) into binary relationships connected to a non-lexical that represents the n-ary relationship set. As a result, all relationship sets are binary between two RDF classes, and each lexical object set has a property value associated with its RDF class. Consider for example, the quaternary relationship set *Person-MarriageDate-MarriagePlace-Spouse* in Figure 2. The lexical object sets *MarriageDate* and *MarriagePlace* become non-lexicals with *MarriageDateValue* and *MarriagePlaceValue* properties, respectively. Then we create a non-lexical object set *PersonMarriageDateMarriagePlaceSpouse* and form binary relationship sets *PersonMarriageDateMarriagePlaceSpouse-Person*, *PersonMarriageDateMarriagePlaceSpouse-MarriageDate*, *PersonMarriageDateMarriagePlaceSpouse-MarriagePlace*, and *PersonMarriageDateMarriagePlaceSpouse-Spouse* between the newly created non-lexical object set and the four non-lexical object sets involved in the quaternary relationship set.

In FROntIER inference rules specify schema mappings between a source ontology and a target ontology. Figure 5 shows several sets of rules in the Jena syntax. Each rule set corresponds to a particular source-to-target transformation. For example, the first rule set in Figure 5 copies the *Person* instances in the ontology in Figure 3 to *Person* instances in Figure 2. This kind of rule performs a direct schema mapping. The second rule set in Figure 5, which also perform a direct schema mapping, copies *BirthDate* instances as well as *Person-BirthDate* instances in the

```
1 [(?x rdf:type source:Person) -> (?x rdf:type target:Person)]

2 [(?x rdf:type source:BirthDate),(?x source:BirthDateValue ?bv)
  -> (?x rdf:type target:BirthDate),(?x target:BirthDateValue ?bv)]
  [(?x source:Person-BirthDate ?y) -> (?x target:Person-BirthDate ?y)]

3 [(?x rdf:type source:Spouse) -> (?x rdf:type target:Spouse)]
  [(?x rdf:type source:MarriageDate),(?x source:MarriageDateValue ?mv)
  -> (?x rdf:type target:MarriageDate),(?x target:MarriageDateValue ?mv)]
  [(?x rdf:type source:PersonMarriageDateSpouse)
  -> (?x rdf:type target:PersonMarriageDateMarriagePlaceSpouse)]
  [(?x source:PersonMarriageDateSpouse-Person ?y)
  -> (?x target:PersonMarriageDateMarriagePlaceSpouse-Person ?y)]
  [(?x source:PersonMarriageDateSpouse-MarriageDate ?y)
  -> (?x target:PersonMarriageDateMarriagePlaceSpouse-MarriageDate ?y)]
  [(?x source:PersonMarriageDateSpouse-Spouse ?y)
  -> (?x target:PersonMarriageDateMarriagePlaceSpouse-Spouse ?y)]

4 [(?n source:NameValue ?nv),(?n rdf:type source:Name),
  regex(?nv, '\b([A-Z][a-z]+)\b\s\b([A-Z][a-z]+)\b', ?x, ?y),makeTemp(?gx),makeTemp(?gy)
  -> (?gx rdf:type target:GivenName),(?gx target:GivenNameValue ?x),
     (?n target:Name-GivenName ?gx),(?gy rdf:type target:GivenName),
     (?gy target:GivenNameValue ?y),(?n target:Name-GivenName ?gy)]

5 [(?x source:Person-Name ?n),(?n source: NameValue ?nv), isMale(?nv),makeTemp(?gender)
  -> (?x target:Person-Gender ?gender),(?gender rdf:type target:Gender),
     (?gender target:GenderValue 'Male'^^xsd:string)]
```

Figure 5: Example Inference Rules for Organizing Facts with Respect to a Target Ontology.

ontology in Figure 3 to *BirthDate* instances and *Person-BirthDate* instances in Figure 2. Observe that the original lexical *BirthDate* instance is now a *BirthDateValue* instance and a property of a *BirthDate* object instance. More complicated schema mappings are possible such as the third rule set in Figure 5 which copies instances in the ternary relationship set *Person-MarriageDate-Spouse* in Figure 3 to *Person-MarriageDate-MarriagePlace-Spouse* instances in Figure 2. Notice that there is no *MarriagePlace* object instance associated with the quaternary relationship set. No marriage place is given, nor can it be inferred.

Our inference rules are constrained to the set of constructs supported by the Jena framework. Conveniently, the Jena framework defines a set of built-in predicates that is extendible. The fourth rule set in Figure 5 splits a name into given name components. The predicates 'regex' and 'makeTemp' are built-ins for splitting a string and assigning variables to its captured groups and for generating a unique identifier for an object, respectively. For extending the set of built-ins, the Jena framework allows the implementation of a *builtin* interface, and we implement this interface for each user-defined built-in. For example, the fifth rule set in Figure 5 infers the gender of a person as male; the user-defined built-in 'isMale' refers to a predefined statistical table to determine whether a name is for a male [Sch12].

FROntIER's entity-resolution rules use facts for entities in populated target ontologies as input and generate *owl:sameAs* relationships as output. FROntIER can use any off-the-shelf or specially developed fact-based entity resolver. We do not plan to study entity resolution techniques

as part of this thesis. Instead, we plan to use Duke[5], an off-the-shelf entity resolver.

In order to use Duke, we convert the RDF triples output by Jena into a comma-separated value file (which can be viewed as a table of entity records). Currently, the conversion from RDF triples to comma-separated value file is hand-specified—once for each target ontology within a domain. For the target ontology in Figure 2, we produce RDF as follows: convert non-lexicals with a *Value* property into table attributes such as *BirthDate* with a *BirthDateValue* property into the attribute *BirthDate*; collapse the aggregate *Name* into *GivenName1*, *GivenName2*, etc. up to the maximum observed number of given names and a *Surname*; convert the quaternary relationship set *Person-MarriageDate-MarriagePlace-Spouse* into *MarriageDate*, *MarriagePlace*, and *Spouse* attributes; calculate the maximum observed cardinality for *Person-MarriageDate-MarriagePlace-Spouse* instances and *Person-Child* instances to produce the attributes *MarriageDate1*, *MarriagePlace1*, *Spouse1*, *Child1*, etc. up to the maximum number of instances for each, respectively; generate the attributes *Father* and *Mother* for *Child-Person* instances (the inverse of *Person-Child*); and convert non-lexicals without a *Value* property into attributes with lexical values such as converting *Child1* into *Child1GivenName1*, *Child1GivenName2*, and *Child1Surname*.

```
Person,GivenName1,GivenName2,Surname,BirthDate,DeathDate,BirthPlace,DeathPlace,Gender,
MarriageDate1,MarriagePlace1,Spouse1GivenName1,Spouse1GivenName2,Spouse1Surname,
Child1GivenName1,Child1GivenName2,Child1Surname,FatherGivenName1,FatherGivenName2,
FatherSurname,MotherGivenName1,MotherGivenName2,MotherSurname,

Person_2,Mary,,Ely,,,,,Female,,,Gerard,,Lathrop,Abigail,Huntington,Lathrop,,,,,,,
Person_5,Mary,Ely,McKenzie,1836,1859,,,Female,,,,,,,,,Donald,,McKenzie,Abigail,Huntington,Lathrop,
Person_8,Mary,,Ely,,,,,Female,,,Gerard,,Lathrop,William,Gerard,Lathrop,,,,,,,
Person_3,Gerard,,Lathrop,,,,,Male,,,,,,Abigail,Huntington,Lathrop,,,,,,,
Person_9,Gerard,,Lathrop,,,,,Male,,,,,,William,Gerard,Lathrop,,,,,,,
Person_7,William,Gerard,Lathrop,1812,1882,,,Male,1837,,Charlotte,Brackett,Jennings,Anna,Margaretta,
        Lathrop,Gerard,,Lathrop,Mary,,Ely,
Person_14,William,Gerard,Lathrop,1840,,,,Male,,,,,,,,,William,Gerard,Lathrop,Charlotte,Brackett,Jennings,
```

Figure 6: Example Comma-separated Value File of Persons in Figure 1.

The Duke entity resolver uses a configuration file to set attribute comparators and parameter values. We use Jaro-Winkler similarity for comparing name components, Levenshtein edit distance for places, and exact match comparisons for all other attributes. For parameter values, each attribute has a low value for when two attribute-value pairs do not match and a high value for when they do match. Duke combines the values to produce a probability that two entities are the same. To demonstrate, consider Figure 6 which is a comma-separated-value file of the attributes for some of the people in Figure 1. Intuitively, names are moderately discriminative so we set a value of 0.7 if two person's first given name match and a value of 0.01 if they do not match. Gender is not so helpful in disambiguating persons when they share the same gender (we set a value of 0.61 if the genders match) but it is very discriminating if they do not share the same gender (we set a value of 0.01 if the genders do not match). Similarly, we set other parameter values according to expected

[5]http://code.google.com/p/duke/

significance within the domain. After running Duke over the example file, it concludes that the probability that the first and third Mary Ely are the same entity is 0.995 and the probability that the two Gerard Lathrops are the same is 0.954. Duke concludes that the second Mary Ely does not match with the other Mary Elys (probability 0.02 for both, which is low because the second Mary Ely has the last name McKenzie, and we set the value for a mismatching surname to be 0.01). The two William Gerard Lathrops also do not match (probability 0.266) because they have different parents (we set a value of 0.01 for both differing mothers and differing fathers).

5 Validation

We plan to use 50,000+ scanned, OCRed books provided by the LDS Church for our corpus. We wish to be able to predict how well the FROntIER framework will work on the entire 50,000+ corpus of books. In particular, we wish to estimate the time required and the sophistication level required to gather all facts from the entire corpus and at what level of accuracy. Applying standard statistical techniques for sampling [Wei02], we know that if we randomly select 200 pages from the corpus of books (treated as a corpus of pages), we can estimate these measurements with 95% confidence to within a 7% margin of error. To estimate the time required to gather all facts from the entire corpus, we will calculate the time required to build recognizers to extract from and annotate the 200 randomly selected pages to estimate the time required per page and calculate an estimate of time required over the entire corpus. To determine the sophistication level required, we will define a hierarchy of complexity of features for extraction-ontology recognizers and organization rules based on intuition and the interdependencies between components and analyze regular-expression recognizers and organization rules with respect to the hierarchy. To estimate the accuracy of extraction, we focus on evaluating the following: (1) extracted facts of interest, (2) inferred facts of interest, and (3) clustered entities from entity resolution.

1. For extracted facts, we will construct recognizers for an extraction ontology to extract facts over the 200 randomly selected pages. We will produce a gold standard of facts by annotating these pages using a form annotator with fields for stated facts of interest in the form and calculate precision and recall values between the extracted facts and the gold standard facts.
2. Similarly, for inferred facts, we will construct inference rules to infer implicit facts for a target ontology. Then, implied facts will be generated using FROntIER over the 200 randomly selected pages. We will produce a gold standard of implied facts by annotating these pages using the form annotator with fields for implied facts of interest in the form and calculate precision and recall values between the implied facts and the gold standard implied facts.
3. FROntIER will generate *owl:sameAs* relationships for coreferent entities in the 200 randomly selected pages using its entity resolver and will produce clusters of entities using generated *owl:sameAs* relationships. We will produce gold standard *owl:sameAs* relationships over coreferent entities in these pages, yielding clusters of same-as entities. We will then calculate precision and recall values over FROntIER-generated clusters and gold standard clusters. It is possible that none or very few of the 200 randomly selected pages would have coreferent entities. If not, then we will additionally find several pages like the page in Figure 1 that do have coreferent entities and make a separate test to determine whether the attribute-based entity resolver is behaving as expected.

6 Thesis Schedule

Complete Coding	by March 2013
Finish Experiments	by April 2013
Finish Writing Thesis	by June 2013
Revise and Defend Thesis	August 2013

References

[BBC+10] L. Blanco, M. Bronzi, V. Crescenzi, P. Merialdo, and P. Papotti. Redundancy-driven web data extraction and integration. In *Proceedings of the 13th International Workshop on the Web and Databases*, pages 1–6, New York, NY, USA, June 2010.

[BCM+03] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[BEV02] M.S. Beach, W. Ely, and G.B. Vanderpoel. *The Ely Ancestry*. The Calumet Press, 1902.

[BGH09] R. Baumgartner, G. Gottlob, and M. Herzog. Scalable web data extraction for online market intelligence. *Proceedings of the VLDB Endowment*, 2(2):1512–1523, 2009.

[BHH+11] D. Burdick, M. Hernández, H. Ho, G. Koutrika, R. Krishnamurthy, L. Popa, I.R. Stanoi, S. Vaithyanathan, and S. Das. Extracting, linking and integrating data from public sources: A financial case study. *IEEE Data Engineering Bulletin*, 34(3):60–67, 2011.

[Chr12] P. Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, 2012.

[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, 1999.

[ELL11] D.W. Embley, S.W. Liddle, and D.W. Lonsdale. Conceptual modeling foundations for a web of knowledge. In D.W. Embley and B. Thalheim, editors, *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*, chapter 15, pages 477–516. Springer, 2011.

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

[EZ10] D.W. Embley and A. Zitzelberger. Theoretical foundations for enabling a web of knowledge. In *Proceedings of Foundations of Information and Knowledge Systems*, pages 211–229, Sofia, Bulgaria, February 2010.

[GX09] J. Gardner and L. Xiong. An integrated framework for de-identifying unstructured medical data. *Data & Knowledge Engineering*, 68(12):1441–1451, 2009.

[HSW07] T.N. Herzog, F.J. Scheuren, and W.E. Winkler. *Data Quality and Record Linkage Techniques*. Springer, 2007.

[LHE03] S.W. Liddle, K.A. Hewett, and D.W. Embley. An integrated ontology development environment for data extraction. In *Proceedings of 2nd International Conference on Information Systems Technology and its Applications*, pages 21–33, Kharkiv, Ukraine, June 2003.

[MK11] R.B. Mishra and S. Kumar. Semantic web reasoners and languages. *Artificial Intelligence Review*, 35(4):339–368, 2011.

[Sar08] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[Sch12] P. Schone. Personal communication, 2012.

[TAC06] J. Turmo, A. Ageno, and N. Català. Adaptive information extraction. *ACM Computing Surveys*, 38(2):1–47, 2006.

[Wei02] N.A. Weiss. *Introductory Statistics*. Addison-Wesley, 2002.

[WLE05] A. Wessman, S.W. Liddle, and D.W. Embley. A generalized framework for an ontology-based data-extraction system. In *Proceedings of 4th International Conference on Information Systems Technology and its Applications*, pages 239–253, Palmerston North, New Zealand, May 2005.