Pragmatic Quality Assessment for Automatically Extracted Data

Scott N. Woodfield[1], Deryle W. Lonsdale[2], Stephen W. Liddle[3], Tae Woo Kim[1], David W. Embley[1,4], and Christopher Almquist[1]

[1] Department of Computer Science
[2] Department of Linguistics and English Language
[3] Information Systems Department
Brigham Young University, Provo, Utah 84602, USA

[4] FamilySearch International, Orem, Utah 84097, USA

Abstract. Automatically extracted data is rarely "clean" with respect to pragmatic (real-world) constraints—which thus hinders applications that depend on quality data. We proffer a solution to detecting pragmatic constraint violations that works via a declarative and semantically enabled constraint-violation checker. In conjunction with an ensemble of automated information extractors, the implemented prototype checks both hard and soft constraints—respectively those that are satisfied or not and those that are satisfied probabilistically with respect to a threshold. An experimental evaluation shows that the constraint checker identifies semantic errors with high precision and recall and that pragmatic error identification can improve results.

Keywords: quality data, data cleaning, automated information extraction, declarative constraint specification, automated integrity checking, conceptual-model-based extraction ensemble.

1 Introduction

Automated information-extraction systems (and sometimes even humans) can extract erroneous (even ridiculous) information. Unless extracted information about entities, values, and relationship assertions among entities and values is correct, applications that depend on the information being correct—such as search, marketing, advertising, and hinting applications—quickly degrade.

Perhaps the most important aspect of data quality is whether the data satisfies real-world constraints—formally, *pragmatic constraints*. In our proposed solution to assessing the quality of automatically extracted data, we begin by aligning internal conceptual-model constraints—formally, *semantic constraints*—with pragmatic constraints. Realizing that pragmatic constraints may be probabilistic and both hard and soft and that verification of accuracy may require supporting documentation, we semantically enrich conceptual models with constraint specification based on probability distributions, and we add the possibility of

attaching supporting documentation to every object and relationship assertion [1]. Then, contrary to standard practice in business database systems, we allow an ensemble of automated extractors to populate the conceptual schema with data that may violate declared integrity constraints. Checking incoming data against declared constraints is straightforward—indeed, is fully automatic based on the declarations alone. Deciding how to handle constraint violations, however, is application-dependent.

Although these augmented conceptual models are generally applicable for use with machine-learned or rule-encoded expert information-extraction systems, our implemented prototype, Fe6,[5] focuses on family-history applications.[6] In Fe6 we handle constraint violations by flagging them red, yellow, or green depending on the severity of the violation and allow adjudication users to correct errors. Interestingly, because constraint specification is declarative in Fe6 conceptual models, handlers that send warning messages to adjudication users for constraint violations can all be generated automatically.

Figures 1 and 2 show an example. In the text snippet in Figure 1, observe that Reverend Ely's children belong to two different mothers: Elizabeth who died in 1871 and Abbie, whom Reverend Ely married subsequently. The automated extraction in Figure 2 has the children all belonging to Elizabeth, but Francis, the last child in the list, was born after Elizabeth died. The automatic extraction engines, which are blind to semantics, regularly make these kinds of (ridiculous) mistakes. Semantic constraint checkers, however, can assess the extracted information and catch constraint violations. Handlers generate messages and flag potentially erroneous filled-in form-fields with a "circle-?" warning icon. When an adjudication user clicks on the icon, a message like the one in Figure 2 pops up to warn the user of potential constraint violation. (Note that the message refers to birth dates, which are not present in the family-composition form in Figure 2. They are, however, extracted onto another form.)

Contributions of the paper include:

1. the addition of probabilistic constraints and of documentation for assertions in a populated model instance;
2. the automatic generation of constraint checkers and constraint handlers; and
3. an experimental validation of constraint-checker precision and recall in the context of an ensemble of information-extraction engines applied to OCRed pages of family-history books.

We explain the details of these contributions as follows. Section 2 elucidates these contributions in the context of related work. Section 3 describes the application system, highlighting the augmented conceptual model and the means by which constraint checkers and handlers can be generated. Section 4 gives the results

[5] Fe6: **F**orm-based **e**nsemble with **6** pipeline phases that accepts an OCRed document as input and generates a conceptualization of document-asserted facts as output.

[6] Increased usage of on-line genealogical sites such as Ancestry.com and FamilySearch.org and increased participation in conferences such as RootsTech.org illustrate the growing interest in these applications.

243327. Rev. Ben Ezra Stiles Ely, Ottumwa, Ia., b. 1828, son of Rev. Ezra Stiles Ely and Mary Ann Carswell; m. 1848, Elizabeth Eudora McElroy, West Ely, Mo., who was b. 1829, d. 1871, dau. of Abraham McElroy and Mary Ford Radford; m. 2nd, 1873, Abbie Amelia Moore, Harrison, Ill., who was b. 1852, dau. of Porter Moore and Harriet Leonard. Their children:

1. Elizabeth B., b. 1849.
2. Ben-Ezra Stiles, b. 1856.
3. George Everly Montgomery, b. 1858, d. 1877.
4. Laura Elizabeth, b. 1859.
5. LaRose DeForest, b. 1861.
6. Charles Wadsworth, b. 1863.
7. Mary Anita, b. 1865.
8. Francis Argyle, b. 1876.

Fig. 1. Text Snippet from *The Ely Ancestry*[2], Page 421.

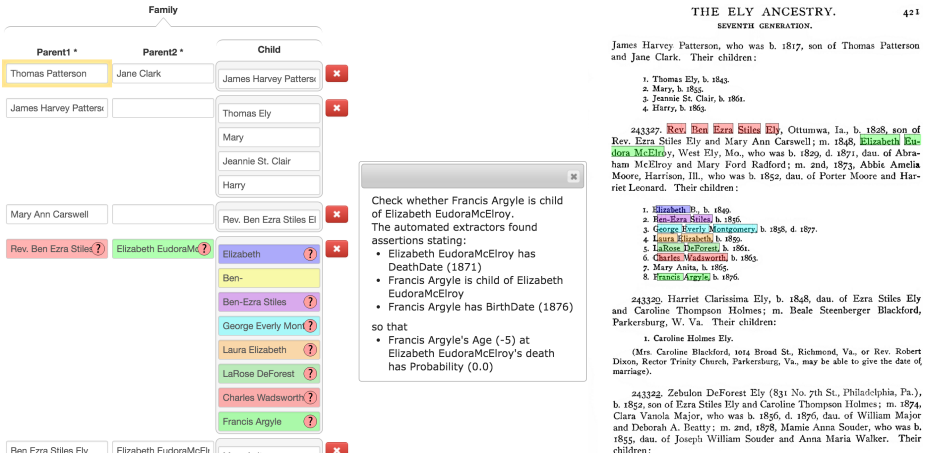

Fig. 2. Screenshot of Constraint Violation: Child Born After Mother's Death.

of an experimental evaluation over a blind test set consisting of more than a thousand automatically extracted assertions from twelve pages taken from three different OCRed family-history books. Section 5 summarizes, draws conclusions, and foreshadows future work.

2 Related Work

Related work is found in the intersection of three disciplines: error detection in information extraction, data cleaning in database systems, and data quality in conceptual modeling.

Information extraction (IE) systems process input text and typically store the results in some kind of database format. Often this extracted content undergoes

a further stage called data cleaning, where inconsistencies and errors are detected and repaired. While data formats and checking methods vary, rules often serve to discover these anomalies. Sample data cleaning approaches include the following:

- A hybrid rule-based/statistical algorithm called LEIBNITZ achieves state-of-the-art performance for building and checking typed functional relations like *'was born in'*, *'died in'*, and *'lived in'* after information extraction has taken place [3].
- A hybrid Ontology-Based IE system consists of two extensions: (i) an ensemble of IE systems that each treat the input text, and (ii) an ontology-based error/contradiction detection system that identifies assertions where the various components' outputs are in disagreement or domain-inconsistent [4].
- The SemantiClean system uses ontology-based reasoning to perform data cleaning on IE output [5]. The Pellet reasoner is run against the IE system's RDF data to perform consistency checking, including collecting provenance information for the assertions.
- A system developed under the PHEME project performs IE and then uses an OWL ontology for biographical knowledge to interpret, encode, and check temporal events and relations like *marriedTo* and *dateOfBirth* [6]. Rule schemas written in the Protegé ontology editor allow for detecting contradictory assertions.

In harmony with these initiatives, Fe6 constraint checkers also apply standard reasoners to detect domain-specific pragmatic errors using ontology-based techniques. Fe6, however, deals with a wider array of input text types and focuses more specifically on the family-history domain.

More generally, database content, even if not derived from IE, often must undergo this type of scrutiny. Various approaches and tools have been developed to perform data cleaning in large-scale database systems, as illustrated in surveys of the field [7–9]. Fe6 constraint-checking techniques differ from the traditional approach that works in a relational database context. Fe6 allows contradictory facts to be captured, and then the system reasons probabilistically over such facts.

Data quality is a primary concern in information systems and conceptual modeling. Indeed, a significant aspect of the activity of conceptual modeling is to identify integrity constraints such as cardinality constraints [10]. However, constraint enforcement alone is insufficient; quality also depends on the particular design and production processes that lead to the capturing of specific data associated with an information system [11]. Researchers have also explored questions related to the quality of conceptual model instances themselves because errors at the schema level cascade to the data level (e.g. [12]). Conceptual modeling researchers have proposed various frameworks for assessing model quality (e.g. [13–16]) from which some level of data quality will presumably follow [17, 18]. Fe6 constraint checkers directly address data quality in ontological conceptualizations by aligning conceptually declared semantic constraints with pragmatic real-world constraints and then checking asserted fact-instances proposed for inclusion in a populated model instance.

3 Application System

To serve their customers, family-history web sites such as FamilySearch.org and Ancestry.com provide search and hinting facilities over a large[7] collection of data about individuals and families. They populate their searchable data stores mostly by crowd-sourcing. Hundreds of thousands of volunteers painstakingly fill-in forms with data copied from images displayed on a computer screen. Most of the images are of handwritten data, often in pre-created forms (e.g. census records, birth certificates, death certificates, and military records). Some of the images, however, are typeset or typewritten such as are newspaper obituaries and family-history books.[8] To extract genealogical data from these printed sources, providers are turning to OCR and automated information-extraction techniques to make this data available for search and hinting.

Fe6 consists of an ensemble of extractors designed to span the space from fully unstructured text to highly semi-structured text. Extracted data from a page of a document (e.g. Page 421 of *The Ely Ancestry* in Figure 2) is distributed to a form (e.g. the "Family" form in Figure 2). An adjudicator checks the filled-in form for correctness and makes corrections as necessary. As an aid to checking, hovering over a record in the form highlights fields as Figure 2 shows and also displays warning icons on fields for which the system has detected a semantic constraint violation. Clicking on an icon pops open a display window explaining the violation.

3.1 Conceptualization

An evidence-based conceptual model [1] serves as the formal foundation for Fe6 applications. Figure 3 shows an example—a conceptualization with its predicates, constraints, and documenting evidence.

The diagram in Figure 3 graphically represents a logic database schema [19]. Object sets, depicted as named rectangular boxes, are one-place predicates (e.g. $Person(x)$). Relationship sets, depicted by lines connecting object sets, are n-place predicates (e.g. $Person(x)$ *has* $BirthDate(y)$). Observe that predicates are in infix form and that predicate names come directly from the text and reading direction arrows in the diagram.

Constraints can be hard (returning only either *satisfied* or *not satisfied* when checked) or soft (returning a *probability of being satisfied* when checked). The conceptual-model diagram in Figure 3 has 28 hard participation constraints specifying a minimum and maximum number of times an object may participate in a relationship set. Each object-set/relationship-set connection has one participation constraint as denoted by the decorations on the ends of the connecting lines. The 2's in Figure 3 explicitly specify participation constraints that override decoration-specified participation constraints—each specifies that children

[7] FamilySearch International, for example, has information about more than a billion deceased individuals.

[8] FamilySearch International has in its collection many millions of newspaper obituaries and has scanned and placed online more than 200,000 family-history books.

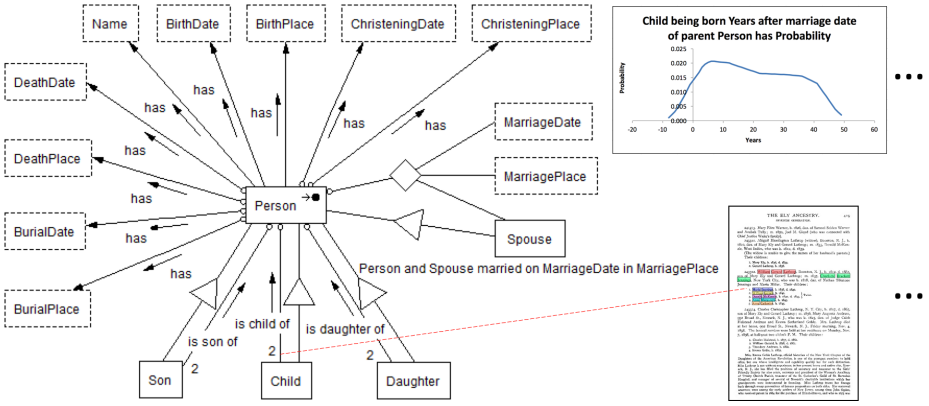

Fig. 3. Depiction of Conceptual Model Features

have two parents. The diagram also shows 4 hard subset constraints (denoted by triangles on connecting lines) specifying that the objects in an object set must be a subset of the objects in another object set—children and spouses are also persons. In addition, Figure 3 shows one of many possible soft constraints as a probability distribution (*Child being born Years after marriage date of parent Person has Probability*). Figure 3 indicates, as well, that evidence can be associated with (and in Fe6 is associated with) every predicate assertion instance (e.g. *Child is child of Person* statements found in a document).

3.2 Hard Constraints

The conceptual-model diagram itself declaratively specifies hard cardinality constraints [10]. For example, it specifies that a person has at most one death date. The *Person* side of the *Person has DeathDate* relationship set has an "o" ("o" for "optional") on its connection and thus allows for no death date. The *DeathDate* side of the relationship set has an arrowhead, which specifies that the relationship from *Person* to *DeathDate* is functional (at most one death date).

Figure 4 shows the adjudicator user interface with extracted data, warning icons, messages, and original text for the case of more than one death date having been extracted for Jesse Harwood. The generated message explains that in addition to the date displayed in the form, the extraction ensemble also extracted another death date for Jesse. On examination of the document, the user would discover that the second death date is for Jesse's wife and thus that the date extracted (which happens to be the right choice) should be kept as the death date but that the erroneous death place should be removed.

The declaration of a participation constraint is sufficient to generate code that both checks for participation constraint violations and handles them. In a populated model instance, counting the number of times an object participates in a relationship set is straightforward, as is checking whether the count

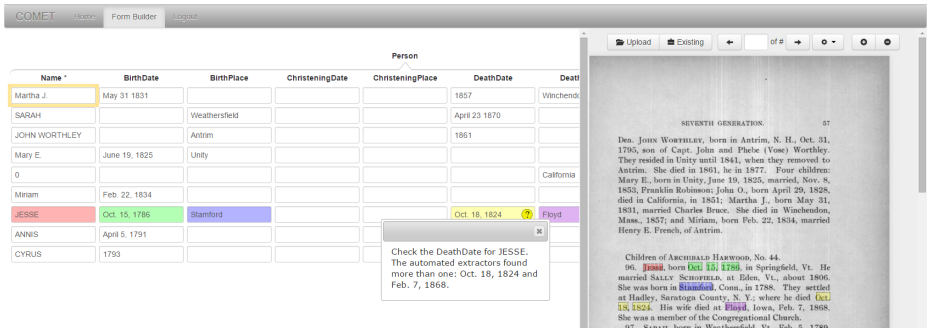

Fig. 4. Screenshot of Constraint Violation (taken from *A Genealogical History of the Harwood Families*[20], Page 57): Two Death Dates.

is within a *min–max* range. Similarly, generating a handler that names the object sets involved and lists the violating objects in a statement template is also straightforward. In family-history applications, the appearance of a name creates a person object, and thus the name can be retrieved and substituted in the template in place of *Person* in the *Person has DeathDate* relationship set in the example in Figure 4.

3.3 Soft Constraints

Soft constraints are based on probability distributions. Since the conceptual model is foundationally predicate calculus, constraint rules can all be Datalog-like implications [21]. The antecedents of an implication are predicates in the model or derived from these predicates or from given probability distributions, and the single consequent gives the probability of a condition being satisfied. For example, we can write a rule about the length of time after a parent's marriage date a child is born:

$$
\begin{aligned}
&\mathit{Child}(x_1)\ \mathit{is\ child\ of}\ \mathit{Person}(x_2),\\
&\mathit{Person}(x_1)\ \mathit{has\ BirthDate}(x_3),\\
&\mathit{Person}(x_2)\ \mathit{and\ Spouse}(x_4)\ \mathit{married\ on\ MarriageDate}(x_5)\ \mathit{in\ MarriagePlace}(x_6),\\
&\mathit{Years}(x_7) = \mathit{Years}(\mathit{YearOf}(x_3) - \mathit{YearOf}(x_5)),\\
&\mathit{child\ being\ born\ Years}(x_7)\ \mathit{after\ marriage\ date\ of\ parent\ has\ Probability}(x_8)\\
&\Rightarrow\\
&\mathit{Child}(x_1)\ \mathit{being\ born\ Years}(x_7)\ \mathit{after\ marriage\ date\ of\ parent}\ \mathit{Person}(x_2)\ \mathit{has}\\
&\mathit{Probability}(x_8).
\end{aligned}
$$

Any probability that fails to meet a user-specified threshold is a constraint violation. Violations tell us that one or more of the antecedents must be incorrect. Figure 5 shows an example in which an incorrect marriage date[9] has been

[9] The date is actually a "p." date, the date a proclamation of the marriage was posted, which serves as a reasonably accurate approximation of a marriage date.

assigned to a couple. As a result, the birth of their son, James, happened 58 years before their marriage—a highly unlikely occurrence.[10] (In another form, a "Family" form like the one in Figure 2, James has been correctly extracted as a child of Jackson, Robert and Isobel King.)

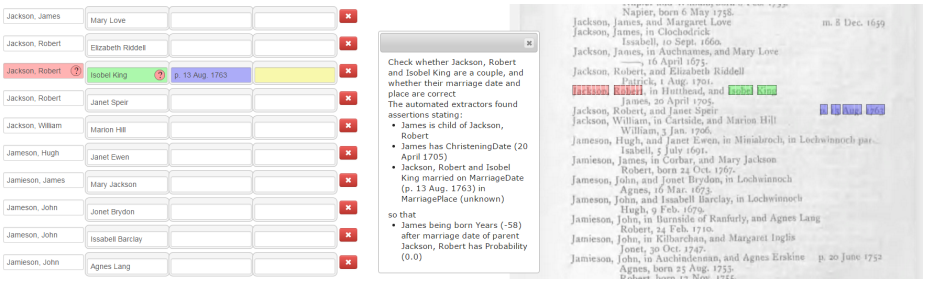

Fig. 5. Screenshot of Constraint Violation (taken from *Index to The Register of Marriages and Baptisms in the Parish of Kilbarchan*[22], Page 55): Birth (-58) Years After Marriage.

Each possible constraint violation has an application-dependent handler. Interestingly, given only the Datalog rule, both the code to check for a violation and the code to handle a violation can be generated automatically. The checker code need only run its usual interpreter on the given Datalog statement, which in essence creates a relational table in which each tuple is the join of all predicate instances that satisfy the Datalog statement. These tuples are then fed one at a time to the handler. Given a user-chosen threshold for constraint violation, the handler fills in a message template with extracted instance data found to be in violation. The handler generator substitutes textual instance values for variables in unary predicate-statement phrases (such as $BirthDate(x)$) and formats them for ease of reading. Since non-textual objects (such as *Person* instances and *Child* instances) come into existence by the principle of ontological commitment, the handler generator replaces unary person predicates with the person's name—the trigger for committing the extraction ontology [23] to recognize the existence of a person.

4 Experimental Evaluation

We designed an experiment to test three hypotheses:

H1 The constraint checker identifies all errors with semantic inconsistencies.
H2 Errors the constraint checker finds correspond to adjudicators' corrections.

[10] The probability distribution for this example comes from a snapshot of the public ancestral tree available on FamilySearch.org consisting of information on over 900 million deceased individuals and their families.

H3 Removing assertions flagged by the constraint checker as possible extraction errors improves precision.

For the experiment we selected three books: *The Ely Ancestry* [2] (sample page snippet in Figure 2), *The Register of Marriages and Baptisms in the Parish of Kilbarchan* [22] (sample page snippet in Figure 5), and *A Genealogical History of the Harwood Families* [20] (sample page snippet in Figure 4). As a development test set, we chose three pages from each book. On these nine pages, we identified extraction errors with semantic inconsistencies made by the ensemble of extractors. For soft errors, we wrote Datalog rules over probability distributions, that would find each of these errors. These soft constraints plus the hard max-participation constraints in the conceptual model in Figure 3 became the fixed set of constraints for the blind test set. The blind test set consisted of the four pages in each book located 1/5, 2/5, 3/5, and 4/5 of the way through the book (although we took a subsequent page if the page turned out to be a picture page as happened in three cases and also if the page contained essentially no genealogical information as happened in one case). Tables 1, 2, and 3 show the statistical data we gathered from the blind test set for testing the hypotheses.

To test Hypothesis **H1**, we ran the data extracted by the ensemble through the constraint checker and produced the pre-adjudication list of semantic constraint violations. To obtain the ground truth about any actual violations in the data, we used the adjudicator interface to correct the extraction from the twelve blind test pages and then ran the constraint checker on the ground truth to produce the post-adjudication list of constraint violations. We also identified all errors in the blind test set that a constraint checker should catch regardless of whether a rule for the constraint had been identified in the development test set. Table 1 shows the precision, recall, and F-score. True positives are violations in the pre-list that did not appear in the post-list, and false positives are those that appeared in both lists. The total number of positives is the list of all violations identified in the blind test set—both those caught from the rules discovered for the development test set plus those that would have been caught by new rules needed to catch additional semantic violations in the blind test set.

The 54 dev-set rules were all either date-based such as mother's age relative to child's birth and age at death (including negative ages, meaning the impossible

Table 1. Constraint Violations Correctly Discovered and Reported.

	Dev-Set Rule	New Rule	%		
Book	Violations	Violations	Precision	Recall	F-score
Ely	41	8	100	84	91
Kilbarchan	12	3	100	80	89
Harwood	1	2	100	33	50
Overall	54	13	100	81	90

Total number of dev-set rules: 10
Total number of new rules needed for blind test set: 5

Table 2. Semantic Errors Marked by the Constraint Checker.

Book Form	Form Fields Filled	Fields with Warning Icons	Icons on Erroneous Data
Ely	388	81	53
Person	207	29	8
Couple	63	5	1
Family	118	47	44
Kilbarchan	680	19	10
Person	266	5	0
Couple	169	6	6
Family	245	8	4
Harwood	81	0	0
Person	55	0	0
Couple	22	0	0
Family	4	0	0
Overall	1149	100	63

death before birth) or relation-based such as person is parent of self. The 13 new rules encountered in the blind test set extended relation-based violations with, for example, person married self, female married female,[11] and person's parents are parents-in-law of person's child, and also added a new type of violation, given name or surname consisting of all digits. The 100% precision in Table 1 indicates that given a rule, the constraint checker identifies violations with high accuracy. The relatively high 81% recall indicates that violations that actually occur can be caught with a small number of rules (12 in the experiment). With the current version of the constraint checker, Hypothesis **H1** (stated with *all*) does not hold, but the results provide assurance that most violations can be caught with relatively few rules.

To test Hypothesis **H2**, we assigned each of 28 students in a sophomore-level linguistics class four adjudication tasks. An adjudication task consisted of correcting the information filled-in automatically by the ensemble of extraction engines in one of three forms: (1) *Person* with birth and death information (see Figure 4); (2) *Couple* with person, spouse(s), and marriage date and place information (see Figure 5); and (3) *Family* with parents and a list of children (see Figure 2). We created batches of four tasks for each student adjudicator to evenly cover the tasks of the blind test set. Due to a glitch in the save software, we lost 18 of the 112 tasks, and we lost another 11 tasks by students not completing their assignment, leaving us with 83 tasks to evaluate. Of these 83 tasks, Table 2 shows statistics about the tasks. Then, using a random sampling of completed tasks covering as many of the 36 book-form-page combinations as possible (only 27

[11] Observe that gender is not in the conceptual model in Figure 3. Gender is inferred based on first given name from a list of 2.2 million name/gender-frequency pairs obtained from FamilySearch data.

Table 3. Accuracy (%): Precision, Recall, and F-score.

	Ensemble Extraction			Extraction with Suspect Assertions Retracted			Student Adjudicator Results		
Book	Prec.	Rec.	F-s.	Prec.	Rec.	F-s.	Prec.	Rec.	F-s.
Ely	58.8	48.5	53.1	55.7	40.2	46.7	54.0	47.2	50.4
Person	67.4	72.5	69.8	61.7	55.0	58.1	54.3	52.8	53.5
Couple	55.2	25.4	34.8	55.2	25.4	34.8	62.5	53.6	57.7
Family	36.4	30.0	32.9	40.0	30.0	34.3	0.0	0.0	N/A
Kilbarchan	84.8	80.1	82.4	85.8	80.1	82.8	95.7	93.5	94.6
Person	100	95.0	97.4	100	95.0	97.4	100	97.2	98.6
Couple	64.8	63.0	63.9	69.0	67.1	68.5	87.0	85.5	86.2
Family	75.8	68.5	71.9	75.8	68.5	71.9	96.2	94.4	95.3
Harwood	29.0	20.0	23.7	24.4	20.0	22.0	69.8	67.3	68.5
Person	33.3	30.0	31.6	33.3	30.0	31.6	80.0	80.0	80.0
Couple	22.2	15.4	18.2	20.0	15.4	17.4	58.3	53.9	56.0
Family	0.0	0.0	N/A	0.0	0.0	N/A	54.6	50.0	52.2
Overall	71.4	62.4	66.6	70.4	59.4	64.4	83.5	79.6	81.5

due to losses, but enough to cover every book-form combination), we computed the accuracy of student-adjudicated work with respect to the ground truth as the third column in Table 3 shows. We obtained these performance measures for each page by randomly selecting one student-completed task for each of the three forms for the page, merging the results, and computing accuracy scores with respect to the ground truth.[12]

The 1149 in Table 2 is a count of the extracted unary assertions—the filled form fields. The constraint checker marked 100 of these filled form fields with warning icons. A record, which is one grouping of filled-in form fields (e.g. the record with highlighted fields in Figure 1), may or may not have some of its fields marked with warning icons. The ensemble extracted 468 records. Of these, 63 contained one or more fields with warning icons and contained data in the record that was indeed erroneous. Many records which were not erroneous also contained fields marked with warning icons. These, of course, should not have been altered, but the 63 should have all been corrected. In an attempt to verify **H2**, we checked to see how many of these 63 records were corrected by student adjudicators. We expected all of them to be corrected, but due to lost tasks and incomplete work, we were unable to gather enough evidence to directly verify **H2**. We were able, however, to see that student adjudicators accurately corrected erroneous records and added missing records in ensemble-extractor-filled forms.

Observe in Table 3 that the student-adjudicator F-score for *Kilbarchan* is 94.6% (a typical good score for language extraction tasks) and that it is considerably better than the ensemble-extraction F-score of 82.4%. The student-

[12] Because of lost tasks, it was not possible to compute performance measures for two *Ely* pages and one *Kilbarchan* page.

adjudicator F-score for *Harwood*, 68.5%, is also much better than the ensemble-extraction F-score, 23.7%. Compared to *Kilbarchan*, *Harwood* tasks are much more difficult because both humans and machines must "read and understand" the text in order to draw conclusions. The results for *Ely* were puzzling until we looked closer. We discovered that the exacting requirement for record match—every filled field in the record having exactly the same text located at exactly the same place in the page with all OCR errors fixed and being exactly in accord with some punctilious instructions—made it hard for human adjudicators to "get it right." The exacting requirements were further exacerbated by the *Ely* author's style—an unconventional mixture of structured and unstructured English, unlike *Kilbarchan* which is nearly fully structured and *Harwood* which is near-ordinary narrative English.

To better see how the student adjudicators corrected extraction errors in *Ely*, we assessed the results by hand giving credit for edits that were understandable and essentially correct despite not correcting OCR errors or not including titles and punctuation such as parentheses around nicknames and maiden surnames or internal punctuation in dates and place names. In this case *Ely* precision increased from the 54.0% reported in Table 3 to 90.5%, recall from 47.2% to 79.2%, and F-score from 50.4% to 84.5%.

To test Hypothesis **H3**, we retracted every ensemble-extracted assertion that appeared as an antecedent predicate assertion in a Datalog rule that detected a constraint violation. Thus, for example, we retracted assertions like the erroneous assertion in Figure 5 that Robert and Isobel were married about 13 Aug 1763, and along with it (unfortunately) any accompanying assertions, like the two correct assertions that James is a child of Robert and that James is a child of Isobel. Because at least one assertion must be incorrect for any constraint violation, the number of false positives should decrease.[13] The middle column of Table 3 shows the results.

The results show that **H3** is unfounded. Rather than increase, overall precision decreased from 71.4% to 70.4%. Although the number of false positives did decrease, so also did the number of true positives, leading to an overall decrease in precision. A positive lesson learned from testing **H2** is that the simple fix of discarding all suspect assertions is not helpful. Instead, we must investigate and solve the harder problem of deciding which suspect assertions should be discarded.

5 Summary and Concluding Remarks

Conceptual-model-based pragmatic quality assessment of automatically extracted data has several desirable properties. Being based on a formal conceptual model whose underlying semantics is predicate calculus makes the specification of constraints and constraint processing declarative. Hence:

[13] In Fe6 applications precision is far more important than recall. An overwhelming number of misleading hints and search results can confuse and discourage customers.

- Code that checks for and handles any and all model-specified participation-constraint violations need only be written once (or can even be generated).
- Adding a probability-distribution constraint requires only the writing of an appropriate Datalog rule.
  - Code to check and handle any rule need only be written once (or can be generated).
  - Rules can be added, modified, and retracted dynamically.
- To the extent user-specified Datalog rules reflect real-world pragmatics, constraint checkers can identify semantically inconsistent extraction errors. The checker does not, however, know which of the extracted fact assertions in antecedent predicates is in error.
- With access to large genealogical data repositories such as those owned by FamilySearch International, probability distributions for rule consequent statements are readily obtainable.

These properties reduce the effort required of a system administrator who has the responsibility to create both constraint checkers and constraint-violation handlers. By pointing out possible errors adjudicators can receive "just-in-time" messages for their task of correcting erroneously extracted fact assertions.

We have identified three areas for future work: (1) Discover how to intelligently retract antecedent assertions of extracted assertions that violate pragmatic rules. (2) Resolve ensemble tool conflicts by choosing the highest probabilistic interpretation of the asserted facts. (3) Use discovered constraint violations as feedback to improve the extractors. The ensemble extraction results in Table 3 are not yet satisfactory for Fe6 applications. They are, however, ideal for testing the constraint checker.

References

1. D.W. Embley, S.W. Liddle, and S.N. Woodfield. A superstructure for models of quality. In *Advances in Conceptual Modeling: Proceedings of the ER 2014 Workshops*, volume LNCS 8823, pages 147–156, Atlanta, Georgia, USA, October 2014.
2. G.B. Vanderpoel, editor. *The Ely Ancestry: Lineage of RICHARD ELY of Plymouth, England, who came to Boston, Mass., about 1655 & settled at Lyme, Conn., in 1660.* The Calumet Press, New York, New York, 1902.
3. T. Lin, Mausam, and O. Etzioni. Identifying functional relations in web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1276. Association for Computational Linguistics, 2010.
4. F. Gutierrez, D. Dou, S. Fickas, D. Wimalasuriya, and H. Zong. A hybrid ontology-based information extraction system. *Journal of Information Science*, pages 1–23, 2015.
5. J. Dolby, J. Fan, A. Fokoue, A. Kalyanpur, A. Kershenbaum, L. Ma, W. Murdock, K. Srinivas, and C. Welty. Scalable cleanup of information extraction data using ontologies. In *Proceedings of The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, pages 100–113. Springer Verlag, Berlin, Germany, 2007.

6. H.-U. Krieger and T. Declerck. An OWL ontology for biographical knowledge: Representing time-dependent factual knowledge. In *Proceedings of the First Conference on Biographical Data in a Digital World*. CEURS-WS.org, July 2015. http://ceur-ws.org/Vol-1399.
7. E. Rahm and H.H. Do. Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, 23(4):3–13, 2000.
8. J.I. Maletic and A. Marcus. Data cleansing: Beyond integrity analysis. In *Proceedings of the Conference on Information Quality*, pages 200–209, 2000.
9. H. Müller and J.-C. Freytag. *Problems, Methods, and Challenges in Comprehensive Data Cleansing*. Professoren des Inst. Für Informatik, 2005.
10. S.W. Liddle, D.W. Embley, and S.N. Woodfield. Cardinality constraints in semantic data models. *Data & Knowledge Engineering*, 11(3):235–270, 1993.
11. Yair Wand and Richard Y. Wang. Anchoring data quality dimensions in ontological foundations. *Commun. ACM*, 39(11):86–95, 1996.
12. Daniel L. Moody. Metrics for evaluating the quality of entity relationship models. In *Conceptual Modeling - ER '98, 17th International Conference on Conceptual Modeling, Singapore, November 16-19, 1998, Proceedings*, pages 211–225, 1998.
13. P. Assenova and P. Johannesson. Improving quality in conceptual modelling by the use of schema transformations. In *Proceedings of the 15th International Conference on Conceptual Modeling*, pages 277–291, Cottbus, Germany, October 1996.
14. R. Schütte and T. Rotthowe. The guidelines of modeling - an approach to enhance the quality in information models. In *Proceedings of the 17th International Conference on Conceptual Modeling*, pages 240–254, Singapore, November 1998.
15. J. Akoka, L. Berti-Equille, O. Boucelma, M. Bouzeghoub, I. Comyn-Wattiau, M. Cosquer, V. Goasdoué-Thion, Z. Kedad, S. Nugier, V. Peralta, and S.S. Cherfi. A framework for quality evaluation in data integration systems. In *ICEIS 2007 - Proceedings of the Ninth International Conference on Enterprise Information Systems*, pages 170–175, Funchal, Madeira, Portugal, June 2007.
16. Jordi Cabot and Ernest Teniente. Incremental integrity checking of UML/OCL conceptual schemas. *Journal of Systems and Software*, 82(9):1459–1478, 2009.
17. I. Comyn-Wattiau, J. Akoka, and L. Berti-Equille. La qualité des systèmes d'information. *Ingénierie des Systèmes d'Information*, 15(6):9–32, 2010.
18. Stuart E. Madnick, Richard Y. Wang, Yang W. Lee, and Hongwei Zhu. Overview and framework for data and information quality research. *J. Data and Information Quality*, 1(1):2:1–2:22, June 2009.
19. D.W. Embley and A. Zitzelberger. Theoretical foundations for enabling a web of knowledge. In *Proceedings of the Sixth International Symposium on Foundations of Information and Knowledge Systems (FoIKS'10)*, pages 211–229, Sophia, Bulgaria, February 2010.
20. W.H. Harwood. *A Genealogical History of the Harwood Families, Descended from Andrew Harwood, Whose English home was in Dartmouth, Devonshire, England, And who emigrated to America, and was living in Boston, Mass., in 1643*. Published by Watson H. Harwood, M.D., Chasm Falls, New York, third edition, 1911.
21. H. Gallaire and J. Minker, editors. *Logic and Data Bases, Symposium on Logic and Data Bases*. Advances in Data Base Theory. Plenum Press, New York, 1978.
22. F.J. Grant, editor. *Index to The Register of Marriages and Baptisms in the PARISH OF KILBARCHAN, 1649–1772*. J. Skinner & Company, LTD, Edinburgh, Scotland, 1912.
23. 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.