Pragmatic Quality Assessment for Automatically Extracted Data

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

[1] Brigham Young University, Provo, Utah 84602, USA
[2] 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,[3] focuses on family-history applications. 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.

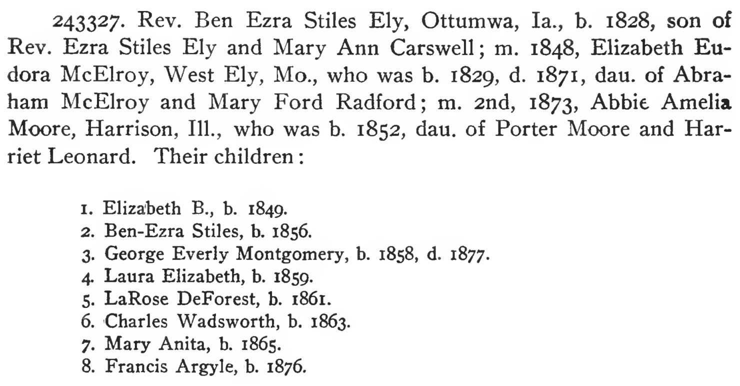
Figures 1 and 2 show an example. In the text snippet in Figure 1, observe that Reverend Ely's children belong to two different mothers: Elizabeth who died in 1871 and Abbie, whom Reverend Ely married subsequently. The automated extraction in Figure 2 has the children all belonging to Elizabeth, but Francis, the last child in the list, was born after Elizabeth died. The automatic extraction engines, which are blind to pragmatics, regularly make these kinds of 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(s). (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.)

The Fe6 constraint checker primarily contributes to increasing data quality, a major concern in information systems and conceptual modeling. Conceptual modeling researchers have proposed various frameworks for assessing model quality (e.g. [2]) from which some level of data quality will presumably follow. 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. Moreover, the Fe6 approach to constraint checking harmonizes well with work on information-extraction systems in which inconsistencies and errors are detected and repaired (e.g. [3]). It also harmonizes well with work on data cleaning for database systems [4], but extends this work by allowing contradictory facts to be captured and then reasoning probabilistically over facts to increase data quality.

2 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 collection of data about individuals and families. They populate their searchable data stores mostly

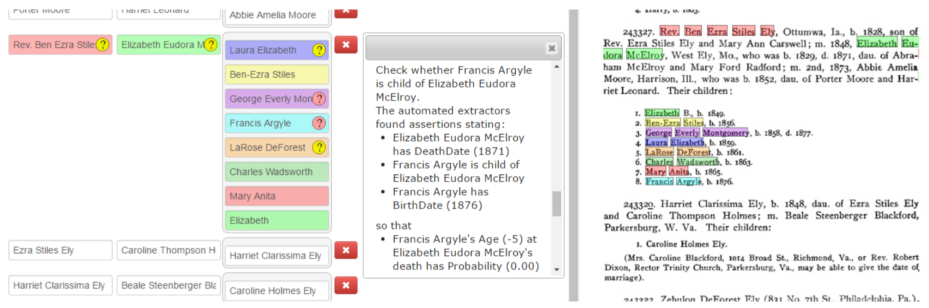
[3] Fe6: **F**orm-based **e**nsemble with **6** pipeline phases that accepts an OCRed document as input and generates a conceptualization of document-asserted facts as output.



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*[5], Page 421.



Porter Moore	Harriet Leonard	Abbie Amelia Moore
Rev. Ben Ezra Stile ?	Elizabeth Eudora M ?	Laura Elizabeth ? Ben-Ezra Stiles George Everly Mon ? Francis Argyle ? LaRose DeForest ? Charles Wadsworth Mary Anita Elizabeth
Ezra Stiles Ely	Caroline Thompson H	Harriet Clarissima Ely
Harriet Clarissima Ely	Beale Steenberger Bla	Caroline Holmes Ely

Check whether Francis Argyle is child of Elizabeth Eudora McElroy.
The automated extractors found assertions stating:
- Elizabeth Eudora McElroy has DeathDate (1871)
- Francis Argyle is child of Elizabeth Eudora McElroy
- Francis Argyle has BirthDate (1876)

so that
- Francis Argyle's Age (-5) at Elizabeth Eudora McElroy's death has Probability (0.00)

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.

243320. Harriet Clarissima Ely, b. 1848, dau. of Ezra Stiles Ely and Caroline Thompson Holmes; m. Beale Steenberger Blackford, Parkersburg, W. Va. Their children:

1. Caroline Holmes Ely.

(Mrs. Caroline Blackford, 1014 Broad St., Richmond, Va., or Rev. Robert Dixon, Rector Trinity Church, Parkersburg, Va., may be able to give the date of marriage).

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

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. 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.

2.1 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.

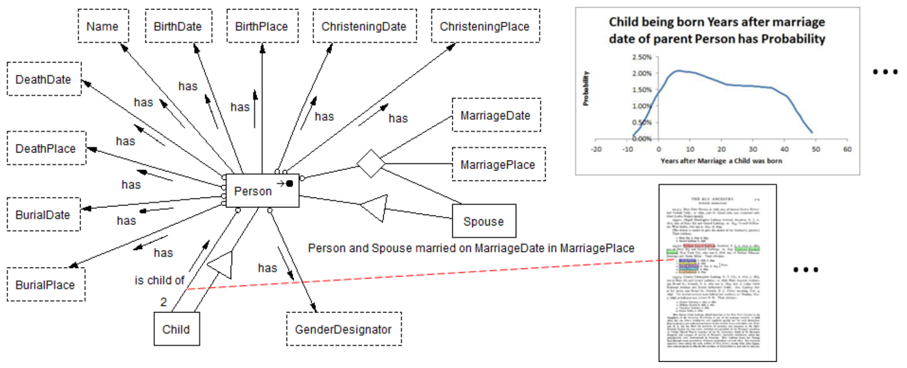


Fig. 3. Depiction of Conceptual Model Features

The diagram in Figure 3 graphically represents a logic database schema. 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 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).

2.2 Hard Constraints

The conceptual-model diagram itself declaratively specifies hard cardinality constraints [6]. 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).

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 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.

2.3 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. 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:

Child(x_1) *is child of Person*(x_2),
Person(x_1) *has BirthDate*(x_3),
Person(x_2) *and Spouse*(x_4) *married on MarriageDate*(x_5) *in MarriagePlace*(x_6),
Years(x_7) = *Years*(*YearOf*(x_3) − *YearOf*(x_5)),
child being born Years(x_7) *after marriage date of parent has Probability*(x_8)
$\Rightarrow$
Child(x_1) *being born Years*(x_7) *after marriage date of parent Person*(x_2) *has Probability*(x_8).

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.

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 to recognize the existence of a person.

3 Experimental Evaluation

We designed an experiment to test the constraint checker: (1) How well does it identify errors with semantic inconsistencies? (2) Can extraction accuracy be improved by intelligently removing assertions flagged by the constraint checker as possible extraction errors?

For the experiment we selected three books: *The Ely Ancestry* [5] (sample page snippet in Figure 2), *The Register of Marriages and Baptisms in the Parish of Kilbarchan* [7], and *A Genealogical History of the Harwood Families* [8]. 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).

To determine how well the constraint checker identifies errors, we ran the extraction ensemble on the development test pages, identified the semantic errors encountered, and wrote rules to catch these errors—14 rules in addition to the model-specified participation-constraint rules in Figure 3. Table 1 shows the results of applying the constraint checker to the twelve blind test pages. Overall, the ensemble extracted 479 records consisting of 1201 filled-in fields. The constraint checker marked 239 of these fields as possibly being in error as a result of finding violations of the 14 probabilistic semantic inference rules and the max-participation-constraint rules in the conceptual model. Looking for additional rules that would have caught errors in the blind test set that did not occur in the development test set, we found three—person names consisting of all digits and two kinds of improbable in-law relationships.

After ground-truthing the extraction for the blind test pages, we again applied the constraint checker to determine whether it would incorrectly identify and erroneously mark fields as possible semantic errors. Table 1 shows the precision, recall, and F-score. True positives are fields marked as possible errors in pre-ground-truthing forms that were not marked in the post-ground-truthing forms. False positives are those marked fields that appeared in both pre- and post-ground-truthing forms. The total number of positives is the number of true-positive marked fields plus the number of unmarked fields that would have been

Table 1. Fields Marked as Potential Errors by the Constraint Checker.

				Erroneously				
		Filled	Marked	Marked	unMarked		%	
Book	Records	Fields	Fields	Fields	Fields	Prec.	Rec.	F-Score
Ely	159	410	127	3	25	97.7	83.6	90.1
Kilbarchan	276	694	108	12	0	90.0	100	94.7
Harwood	44	97	4	0	0	100	100	100
Overall	479	1201	239	12	25	94.1	90.5	92.3

Total number of development test-set rules: 14
Total number of new rules needed for blind test set: 3

marked had the constraint checker encoded the additional three rules for semantic errors in the blind test set that did not apply to the development test set. In our experiment the constraint checker was 100% accurate except in a few instances in the Kilbarchan and Ely books where it encountered parents of the same child supposedly having the same gender. Gender is inferred from gender designators such as “son of”, “Mrs.”, etc. or in the absence of a gender designator by a large list of name/gender-frequency pairs.

Table 2 shows the results of our efforts to determine how well the constraint checker could repair erroneously extracted data. Overall, the ensemble extracted information with an F-score of 83.5%. Retracting all suspect assertions improved precision by 4.8 percentage points at the expense of a large drop in recall (11.1 percentage points) and a drop in F-score of 4.6 percentage points. Intelligently retracting just those assertions that are certainly or heuristically identifiable as being erroneous, improved precision slightly to 88.2% without dropping recall by much, but enough to cause a slight drop in the F-score of 0.7 percentage points.

Assertions identifiable as certainly erroneous are those from rules with exactly one antecedent assertion such as “parent of self” and “spouse of self”. Based on text layout, we heuristically chose to reject assertions violating participation constraints in which the lexical reading distance between the objects being related is more distant than the closest. Thus, for example, when the extractors declared two death dates for an individual, we kept only the date closest to the person’s name.

4 Concluding Remarks

Being based on a formal conceptual model whose underlying semantics is predicate calculus makes the specification of constraints and constraint processing declarative. To the extent user-specified inference rules reflect real-world pragmatics, constraint checkers can identify semantically inconsistent extraction errors. Except in a few cases, however, the checker does not know which of the extracted assertions in antecedent predicates is in error. In general, determining which one(s) of several possible antecedent assertions is in error is non-trivial.

Table 2. Accuracy (%): Precision, Recall, and F-score.

	Ensemble Extraction Results			All Suspect Assertions Retracted			Identifiable Erroneous Assertions Retracted		
Book	Prec.	Rec.	F-s.	Prec.	Rec.	F-s.	Prec.	Rec.	F-s.
Ely	81.2	65.1	72.2	83.6	44.0	57.6	77.0	59.1	66.8
Person	83.8	93.3	88.3	82.7	75.3	78.8	83.8	93.3	88.3
Couple	78.6	35.5	48.9	84.6	35.5	50.0	84.0	33.9	48.3
Family	78.0	56.8	65.7	86.7	16.0	27.1	61.1	40.7	48.9
Kilbarchan	91.9	90.5	91.2	97.3	85.2	90.8	95.3	91.1	93.2
Person	100	96.4	98.2	100	89.2	94.3	100	94.2	97.0
Couple	87.7	87.7	87.7	94.4	93.2	93.8	88.7	86.3	87.5
Family	85.6	85.6	85.6	96.0	76.0	84.8	94.2	90.4	92.2
Harwood	79.1	79.1	79.1	80.5	76.7	78.6	81.0	79.1	80.0
Person	96.3	86.7	91.2	96.2	83.3	89.3	96.3	86.7	91.2
Couple	75.0	60.0	66.7	85.7	60.0	70.6	85.7	60.0	70.6
Family	25.0	66.7	36.4	25.0	66.7	36.4	25.0	66.7	36.4
Overall	87.3	80.1	83.5	92.1	69.0	78.9	88.2	78.1	82.8

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. 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.
3. 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.
4. E. Rahm and H.H. Do. Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, 23(4):3–13, 2000.
5. 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.
6. S.W. Liddle, D.W. Embley, and S.N. Woodfield. Cardinality constraints in semantic data models. *Data & Knowledge Engineering*, 11(3):235–270, 1993.
7. 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.
8. 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.