Varish Mulwad, Tim Finin and Anupam Joshi, Interpreting Medical Tables as Linked Data to Generate Meta-Analysis Reports, 15th IEEE Int. Conf. on Information Reuse and Integration, IEEE, August 2014.

Interpreting Medical Tables as Linked Data for Generating Meta–Analysis Reports

Varish Mulwad, Tim Finin and Anupam Joshi
University of Maryland, Baltimore County
Baltimore, Maryland, USA
{varish1, finin, joshi}@cs.umbc.edu

Abstract

Evidence-based medicine is the application of current medical evidence to patient care and typically uses quantitative data from research studies. It is increasingly driven by data on the efficacy of drug dosages and the correlations between various medical factors that are assembled and integrated through meta–analyses (i.e., systematic reviews) of data in tables from publications and clinical trial studies. We describe a important component of a system to automatically produce evidence reports that performs two key functions: (i) understanding the meaning of data in medical tables and (ii) identifying and retrieving relevant tables given a input query. We present modifications to our existing framework for inferring the semantics of tables and an ontology developed to model and represent medical tables in RDF. Representing medical tables as RDF makes it easier for the automatic extraction, integration and reuse of data from multiple studies, which is essential for generating meta–analyses reports. We show how relevant tables can be identified by querying over their RDF representations and describe two evaluation experiments: one on mapping medical tables to linked data and another on identifying tables relevant to a retrieval query.

1 Introduction

Evidence–based medicine (EBM) is commonly defined as "the conscientious, explicit, and judicious use of current best evidence in making decisions about the care of individual patients" [19]. EBM analyzes questions such as efficacy of drug dosages, correlation between various medical factors or correlation between drugs by performing meta–analyses (i.e., systematic reviews) over evidence and data previously published in scientific literature and clinical trial studies. The goal is to find, integrate and analyze available high-quality quantitative data to inform clinical and health care related decisions.

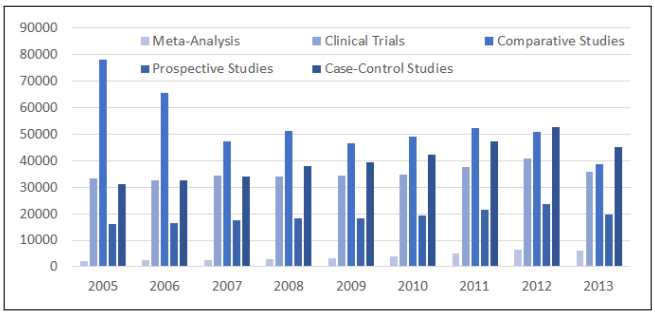

Figure 1. Number of meta–analysis, clinical trials, comparative studies, prospective studies and case–control studies published between 2005 and 2013.

EBM has been gaining traction for the past several years. A search on PubMed[1] for publication type "Meta–Analysis" shows, while only 272 meta–analyses reports were published in 1990, more than 6600 meta–analyses reports were published in 2013. Organizations such as The Cochrane Collaboration[2] have a dedicated set of medical researchers whose primary goal is to perform and publish systematic reviews on a number of health care related issues and to keep them updated as new medical research findings become available.

The process of generating a meta–analysis report is still largely manual. Medical researchers start with keyword search on systems like MEDLINE[3] which often lead to thousands of initial set of studies. Researchers carefully analyze each study reducing the result set to a few hundred studies or less which are included in the final meta–analysis. Often a two stage filtering is done in which studies are accepted or rejected first based on the title and abstract

[1] http://www.ncbi.nlm.nih.gov/pubmed
[2] http://www.cochrane.org/
[3] http://nlm.nih.gov/bsd/pmresources.html

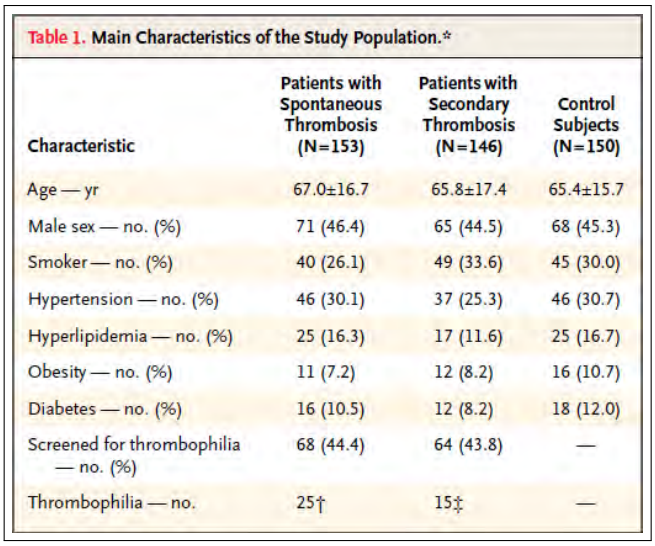

Table 1. Main Characteristics of the Study Population.*

Characteristic	Patients with Spontaneous Thrombosis (N=153)	Patients with Secondary Thrombosis (N=146)	Control Subjects (N=150)
Age — yr	67.0±16.7	65.8±17.4	65.4±15.7
Male sex — no. (%)	71 (46.4)	65 (44.5)	68 (45.3)
Smoker — no. (%)	40 (26.1)	49 (33.6)	45 (30.0)
Hypertension — no. (%)	46 (30.1)	37 (25.3)	46 (30.7)
Hyperlipidemia — no. (%)	25 (16.3)	17 (11.6)	25 (16.7)
Obesity — no. (%)	11 (7.2)	12 (8.2)	16 (10.7)
Diabetes — no. (%)	16 (10.5)	12 (8.2)	18 (12.0)
Screened for thrombophilia — no. (%)	68 (44.4)	64 (43.8)	—
Thrombophilia — no.	25†	15‡	—

Figure 2. Table found in [18] is typical of those found in medical research reports.

and then later filtered after a close examination. A meta–-analysis of the correlation between *cardiovascular risk factors* and *venous thromboembolism* [1], for example, started with an initial search yielding 1949 studies which were downselected to just 22 for the final analysis and report. Figure 1 gives insight into how tedious the process is. While the number of meta-analysis reports published each year is growing, they are out-paced by the number of clinical trials, comparative studies, prospective studies and case–control studies that potentially provide evidence.

Key information required to produce meta–analyses reports is often obtained from tables like the one shown in Figure 2. Consider an analysis of the correlation between *Obesity*, a cardiovascular risk factor and *venous thromboembolism*. Conclusions about the correlation are derived by first identifying relevant studies such as [18, 17] and then by extracting and integrating results from these individual studies. In this example, information such as number of individuals that suffer from obesity and venous thromboembolism (23/299) and number of individuals that suffer only from obesity (16/150) is of key interest. Such information, as seen from Figure 2, is encoded in tables published in medical studies.

The automatic discovery and interpretation of tables in a medical research report gives strong evidence that (i) the report includes empirical data and (ii) the degree of relevance to the question. The tables themselves provide the raw study data that will eventually be integrated. Inferring the semantics of tables and producing a linked data representation delivers the data in a form that facilitates aggregation, mapping and integration.

In the remainder of the paper we describe our framework for inferring the semantics and meaning of tables published in medical research studies and representing it as RDF linked data. We discuss the changes we made to our previous framework for inferring the semantics of general tables [16] in order to deal with the challenges posed by medical tables. The resulting system maps header cells from medical tables to appropriate concepts from existing ontologies and further represents the inferred semantics of the data using a custom ontology for medical tables [13]. We demonstrate how structured SPARQL queries over this representation of medical tables can improve and assist in the process of identifying relevant studies, extracting key data and integrating the data for meta–analyses reports.

2 Interpreting Medical Tables

A system producing meta–analyses reports benefits from a deeper understanding of the semantics of medical tables during the discovery phase as well as during extraction and integration phases. Inferring the semantics of column and row headers provides an idea of various medical factors and patient groups compared in the table. Mapping header cells to appropriate concepts from an ontology enhances the process of discovery of relevant studies. For example, mapping the string Obesity to appropriate concepts from SNOMED CT[24] or UMLS[21], will not only allow structured queries during discovery, but will also allow to infer additional knowledge that Obesity and other row headers are also cardiovascular risk factors via relations in the ontology. Inferring metadata encoded in column headers provide information on the size of groups used in comparison which is required during the integration process. Similarly, understanding the data cell values is critical for integration of data from several medical studies. Data cells in medical tables as seen in Figure 2 have complex representation, whose meaning can be inferred by understanding the metadata encoded in row headers (and sometimes column headers or even table captions).

Automatically generating meta–analysis report would involve three steps: Find – Extract – Integrate (FEI), followed by an analysis which produces conclusions and recommendations. The *Find* step is the process of searching for and identifying relevant studies to be included in the meta–analysis; *Extract* the extraction of relevant data from selected studies; and *Integrate* the integration of the extracted data to produce a dataset for analysis. Inferring the semantics of medical tables is vital for all three steps. Producing an overall interpretation of a medical table requires (i) inferring the metadata encoded in the header cells as well as inferring and normalizing data cells (ii) inferring the semantics of header cells and mapping them to appropriate concepts from an ontology and (iii) generating an appropri-

ate RDF representation of medical tables for easy discovery, extraction and integration.

3 Approach

We extend our previous domain independent framework [15] [16] to infer the semantics of medical tables and generate linked data representations, as shown in Figure 3. An input table first goes through a pre-processing module *Normalize*, which handles the idiomatic patterns typically found in medical tables. It also infers header cell metadata as well as normalizes the content in data cells. Once the table is normalized, the *Query and Rank* (Q&R) module queries linked open data sources and medical knowledge bases to generate an initial ranked list of concepts for each column and row header. The *Joint Inference* module uses a probabilistic graphical model to jointly infer the semantics and map row and column headers to appropriate concepts from the ranked list. A final step produces a linked data representation using our MTO ontology [13] and relevant domain ontologies.

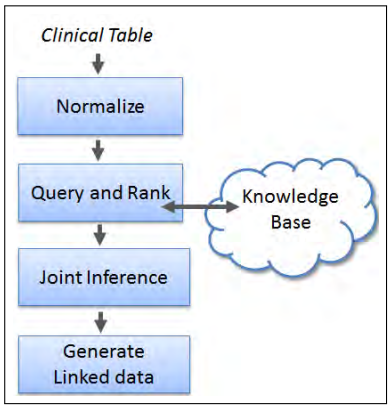

Figure 3. Our framework for generating linked data from a medical table has four steps.

Normalization. Medical tables do not have the simple structure of most tables found on the web (webtables), i.e., a rectangular array of data cells (with optional column headers) where each cell holds a single value. Medical tables often exhibit of both column and row headers, between whom the data is enclosed. Non-header cells are *data cells* that typically represent values for the relationship between the respective column and row headers. For example, the value *46* from the column of *Patients With Spontaneous Thrombosis* and row header *Hypertension* in Figure 2 leads to the interpretation that 46 of the 153 patients with *spontaneous thrombosis* also suffer from *hypertension*.

Typically, content in header and data cells in webtables is simple in nature; it either consists of strings that can be directly mapped to a class or a entity or literal values such as numbers that can be mapped as values of a property. The content found in header and data cells in medical tables, however, are represented using idiomatic patterns often encoding additional metadata in header cells or representing the literals in data cells as "complex objects". Consider the column header *Patients With Spontaneous Thrombosis (N=153)*; not only it represents a concept (*Spontaneous Thrombosis*), but also encodes additional metadata *(N=153)* which is useful for interpreting the values in the column. Similar encoding can be observed in row headers. Data cells are also complex objects. Consider the data cell *46 (30.1)* from the row with header *Hypertension* in Figure 2; from the metadata in its row header, one can interpret, *46* represents the raw number and *30* its percentage of the total. Interpreting medical tables thus requires significant pre–processing in which the content in the header cells have to be normalized to extract strings to be mapped to concepts and decompose and interpret data cells to generate an accurate semantic representation.

The *Normalize* module first processes the header cells. The content in column and row header cell can be parsed into two distinct parts: a *query string* denoting a concept such as a disease, drug, or patient characteristic and *metadata* that describes how the data in the column/row should be interpreted, e.g., giving units of measure (*kg*), statistical properties (*avg*) or a schema for non-atomic values (*no. (%)*). For example, the fourth row header *Hypertension – no. (%)* would produce the query string *Hypertension* and metadata *no. (%)*. We develop a set of regular expression patterns that cover the most common cases observed in published medical data tables to extract query string and metadata. In most medical tables, the query string and metadata are either separated by a hyphen [query string - metadata], a comma [query string , metadata] or the metadata follows the query string in parenthesis [query string (metadata)]. Every column and row header is matched against these three patterns to extract query string and metadata. If the header content does not match any of the patterns, the content is treated as query string. In our current example, the header *Hypertension – no. (%)* will match with [query string - metadata] extracting *Hypertension* as query string and *no. (%)* as metadata.

The extracted metadata can require further processing as it encodes information regarding how values in the respective column or row should be interpreted. For example, the metadata *no. (%)* conveys the message that values of the form *a (b)* in the given row should be interpreted as *a* representing the raw number and *b* representing its percentage of the total. While metadata in medical tables can encode vast variety of information, they can be generalized to a limited set of common patterns such as generalizing *no. (%)* to *a (b)*. The Normalize module further processes metadata by mapping it to a pattern from a set of generalized metadata patterns. We identify a set of common metadata patterns which include: *a (b) (c)* (e.g. mean (standard deviation)

(range)); *a (b)* (e.g. no. (%)); *a +/-b* (e.g. mean standard +/- deviation); *a/b* (e.g. Male/Female). Every metadata extracted from the content in header cells is mapped against one of four metadata patterns. Thus, the extracted metadata from the *Hypertension* row header, *no. (%)* is normalized as *a (b)*. The generalized metadata pattern is further used to interpret the values in the respective columns and rows. Header cells representing patient groups used in the study, encode number of the patients in the group as part of the header metadata. The Normalize module uses an additional rule $n = x$ pattern, where x represents the number of patients in respective patient group.

The content in data cells is processed by the Normalize module by using the generalized header cell metadata patterns. The data cell content is also mapped against one of the four metadata patterns. For example, the data cell content *46 (30.1)* will get mapped to the pattern *a (b)*. Once the data cell is mapped, the Normalize module checks for the same pattern in the respective column or row header. If the same pattern is discovered, it is used to decompose the data cell content. Again, in the case of *46 (30.1)*, its pattern *a (b)* will match with its row header pattern *a (b)* , which is used by the Normalize module to decompose and interpret *46* as *no.* and *30.1* as *%*.

Query and Rank. The Query and Rank (Q&R) module generates a ranked list of candidate classes for every query string in row and column headers. We assess three different knowledge bases (KBs), two domain specific ones, UMLS Metathesaurus [21] and SNOMED CT [24], and one general purpose one, DBpedia [2]. KBs in the health care domain are still maturing and our goal behind assessing different KBs is to compare the coverage and strengths of each in the context of medical tables.

SNOMED CT (Systematized Nomenclature of Medicine–Clinical Terms) is a clinical terminology consisting of more than 311,000 medical concepts organized in a hierarchy with more than 1,360,000 links or semantic relationships between them. For every column and row header query string, the Q&R module first executes an exact match query against all concepts and their synonyms in SNOMED CT. If no results are returned, Q&R executes a free text search query over an index of SNOMED CT concepts. The order in which the concepts are returned, both for exact match and free text search queries, is retained by Q&R to rank the concepts. For example, the query string *Age* results in a direct match with two concepts *Age (qualifier value)* and *Current chronological age (observable entity)* as results; whereas the query string *Diabetes* leads to a text search query with *Diabetic cataract associated with type I diabetes mellitus*, *Diabetic oculopathy associated with type I diabetes mellitus*, *Diabetic retinopathy associated with type I diabetes mellitus*, etc. as results.

UMLS Metathesaurus (Unified Medical Language System Metathesaurus) has of large number of concepts related to clinical, health care, and biomedical domains assembled by combining information from over 150 different clinical, biomedical, health care related terminologies, vocabularies and code–sets, including SNOMED CT. Like SNOMED CT, concepts within the UMLS Metathesaurus are connected via a set of relationships. We use the UMLS API *findConcept* service to generate candidate concepts. For every query string, Q&R module queries the API which returns a ranked list of concepts matching the query string. For example, for the input query *Hypertension*, the API returns concepts such as *Hypertensive disease*, *Hypertension Adverse Event*, *No hypertension*, *Hypertension with complications*, etc.

DBpedia is a general purpose KB whose classes and properties are good for describing the semantics of general tables. Unlike SNOMED CT and UMLS Metathesaurus, the row and column headers from medical tables can be mapped to DBpedia instances rather than classes. For every query string, Q&R generates a ranked list of candidate instances by querying against Wikitology [27], a hybrid knowledge base that combines structured and unstructured information from Wikipedia, DBpedia, Freebase and Yago. Q&R matches the input query against Wikipedia article's title, redirects, first sentence and contents(body). The query for *Hypertension*, for example, returns ranked list of instances *Idiopathic_intracranial_hypertension*, *Pulmonary_hypertension* and *Hypertension*. Q&R module reranks the results returned by Wikitology using a previously developed *entity ranker* [16] trained on a set of string similarity (Levenshtein distance, Dice score) and popularity metrics (predicted google page rank, article length, Wikitology score).

Joint Inference. A typical table provides a number of correlations between its elements. Column headers often represents type or a class and values in that column represent instances of that type. The class assigned to the column header depends on instances linked in the column and vice–versa. Column headers themselves are interrelated with each other and so are values in a single row. These dependencies between various elements in a table are captured by modeling the assignment problem as a joint assignment or a joint inference problem. We jointly infer the semantics of tables by representing a table as a undirected Markov network. The column headers and data cells represent variables in the graph and edges indicate dependencies between the variables.

These dependencies, however, are less common in medical tables. Row headers are fairly independent of each other.

```
# Triples representing the sixth row related to Obesity
mto:Observation_6 mto:hasVariable umls:Obesity;
    mto:hasPatientGroupObservation mto:PGroupObs_61;
    mto:hasPatientGroupObservation mto:PGroupObs_62;
    mto:hasPatientGroupObservation mto:PGroupObs_63.
mto:PGroupObs_61 mto:hasPatientGroup mto:PGroup_t11;
    mto:hasObservationValue mto:Value_611 .
    mto:hasObservationValue mto:Value_612 .

# Triples representing the first patient group (second column header)
mto:PGroup_t11 mto:numberOfIndividuals "153"^^xsd:integer;
    mto:hasGroupAttribute umls:Venous_Thrombosis.

# Triples representing the value in first cell of Hypertension row
mto:Value_611 mto:hasRawValue "11"^^xsd:string;
    mto:Type "no."^^xsd:string.
mto:Value_612 mto:hasRawValue "7.2"^^xsd:string;
    mto:Type "%."^^xsd:string.
```

Figure 4. Subset of RDF for data in Figure 2. The prefixes 'mto' is for our Medical Tables Ontology; 'umls' is for UMLS Metathesaurus and 'xsd' is for XML schema. Complete representation can be found at [14]

.

Correlation between column headers can be captured. Column headers in medical tables consists of patient groups and statistical tests comparing the groups. Thus, if a header cell in a medical table gets mapped to a statistical test, then it is likely that other headers are various patient groups. This correlation can be captured by inserting edges between column headers. Row headers, in certain cases, can be disambiguated with the help of values in the row. Consider the values in the row *Age*: *67*, *65.8*, and *65.4*. One can infer from the range of the values alone, in the context of the medical domain, that they represent age of people. Such evidence can be captured by inserting edges between row headers and values in the respective rows. For the purposes of this paper, we do not implement the Joint Inference module for medical tables. Column and row headers are mapped to the top ranked concept from the list of each concepts generated by the Q&R module.

RDF Generation. Once data cells are normalized and interpreted, and the column and row headers are mapped to appropriate concepts, the *Generate RDF* module generates a RDF linked data representation of the table. Unlike webtables, medical tables have a complex representation with the data cell values representing the relationship between the row and column headers. We develop the *MTO* ontology [13] to model and represent information encoded in such medical tables.

A typical medical table represents information about a set of observations related to the question analyzed in the study. A study of the correlation between *Atherosclerosis* and *Venous Thrombosis* will include medical tables reporting on observations that helps one analyze if the correlation exists or not. Each row in such a table can be interpreted as representing one observation. We add a class **Observation** to represent every row in a medical table. Every observation is typically associated with different aspects of the study such as patient groups, characteristics associated with patients in such groups and statistical analysis tests comparing the groups.

The characteristics associated with patients in the group can vary from information about demographics and habits such as gender, age and smoking habits to information about the diseases patients may or may not be suffering to data on vitals such as blood pressure, sugar level and hemoglobin. We combine all of these into a single group and refer to it as *Variables* associated with the observation. Typically, such variables appear in the row headers of the table. The association between an observation and its variable is captured by the property **hasVariable**.

Patient groups refer to group of individuals with different characteristics used for comparisons in the study. For example, a study comparing the correlation between *venous*

thrombosis and *obesity* will consist of group of individuals that suffer from both the diseases and a control group which includes individuals that suffer only from *obesity*. Another study might include patient groups, each on a different drug dosage to study the effect of dosages on a particular disease. The relation between a observation and a patient group is captured via the **hasPatientGroupObservation** property.

Finally, the table often reports statistical analysis performed to compare different groups using tests such as Odds ratio, Hazard Ratio, and p-value. The correlation between an observation and a statistical test is captured via the **hasStatisticalTestObservation** property.

Instances of the class **StatisticalTestObservation** capture test information with two key components: the type of statistical test (e.g., odds ratio, hazard ratio) via the property **hasStatisticalTest** and an associated result or value via the **hasObservationValue** property. The domain of the property hasObservationValue is StatisticalTestObservation or PatientGroupObservation and its range is the class **Value**, whose instances are used to capture the normalized and decomposed content from data cells. The property **hasRawValue** captures the actual value from the table cell and **hasType** captures its interpretation, with both having domain Value and range **string**.

Similarly, instances of the class **PatientGroupObservation** are used to capture observation value associated with the patient group. We define a property **hasPatientGroup** to associate patient group observation with a patient group; and use previously hasObservationValue to associate the actual observation value. The domain of hasPatientGroup is the class PatientGroupObservation and its range is the class **PatientGroup**. The class PatientGroup is used to capture information related to the group. Depending upon the study, this information can vary from a disease from which individuals suffer (e.g., *Venous Thrombosis*) to drugs that individuals are taking. We define the property **hasGroupAttribute** to capture this information. The domain of this property is the class **PatientGroup**. Additional metadata related the group such as number of individuals in the group is also captured via the instances of class PatientGroup. We define the property **numberOfIndividuals** to capture this metadata, with domain PatientGroup and range **xsd:integer**.

Observation instances are associated with a **Table** through the property **tableObservation**. **Tables** are linked to a study using the property **hasTable**. We also record additional metadata associated with the study such as name of the study, publication information associated with the study such as authors, date of publication and journal. We use the properties **studyTitle**, **studyAuthor**, **publicationDate** and **publicationJournal**.

The *Generate RDF* module uses the MTO ontology to represent inferred semantics of medical tables as RDF linked data. A subset of RDF linked data generated from the table in Figure 2 is shown in Figure 4. Every row in the table is represented as instance of the *Observation* class. Linking the row header string *Hypertenion* to *umls:Hypertension* provides additional information that the header type is disease. Concepts such as diseases are often characteristics or attributes of a patient group and are linked with a observation using the *hasVariable* property. The identified patient groups in column headers are linked with the observation using the *hasPatientGroup* property. Instances of the *PatientGroup* represent additional information about the group. The data cells are linked with the observation using *hasPatientGroupObservation* property. Every data cell is linked to the observation using instances of the *PatientGroupObservation*. The normalized data cells are represented using instances of the *Value* class and linked to patient group observations using *hasObservationValue* property.

(a) Find studies correlating *venous thrombosis* and *obesity*.

```
SELECT ?study WHERE {
 ?patGroup mto:hasGroupAttribute
    umls:Venous_Thrombosis.
 ?patObsGroup mto:hasPatientGroup ?patGroup.
 ?obs mto:hasPatientGroupObservation ?patObsGroup;
    mto:hasVariable umls:Obesity;
    mto:observationInTable ?table.
 ?table mto:tableInStudy ?study.}
```

(b) Get all observations for a given study.

```
SELECT ?obs WHERE {
 mto:study3 hasTable ?table.
 ?table mto:tableObservation ?obs.}
```

Figure 5. Example SPARQL queries demonstrating how *FEI* process can be enhanced.

4 Find–Extract–Integrate

Once medical tables are represented as RDF linked data, we can execute SPARQL queries over the data to automate the process of discovery, extraction and integration of data from multiple studies. One of the most common queries of interest for meta–analysis reports is finding comparative studies for pairs of factors, e.g., correlation between *obesity* and *venous thrombosis* or drug interaction between *clopidogrel* and *proton pump inhibitors*. With available semantics, studies for abstract queries such as correlation between *cardiovascular risk factors* and *venous thrombosis* can be eas-

(c) Find studies in which *hypertension* is one of the patient group characteristics.

```
SELECT ?study WHERE {
  ?obs mto:hasVariable umls:Hypertension .
  ?table mto:tableObservation ?obs .
  ?study mto:hasTable ?table . }
```

(d) Find studies correlating *venous thrombosis* and *obesity* published after May 2013.

```
SELECT ?study WHERE {
  ?patGroup mto:hasGroupAttribute
     umls:Venous_Thrombosis .
  ?patObsGroup mto:hasPatientGroup ?patGroup .
  ?obs mto:hasPatientGroupObservation ?patObsGroup;
     mto:hasVariable umls:Obesity .
  ?table mto:tableObservation ?obs .
  ?study mto:hasTable ?table;
     mto:publicationDate ?date .
  FILTER (?date >= "20130501"^xsd:date) }
```

Figure 6. Results can be filtered on number of parameters such as publication date, medical factors and diseases.

ily discovered. It is also possible to extract tables or parts of data in tables for a given study. Meta–analyses reports are often updated as new literature and evidence is published, requiring researchers to periodically search for relevant new studies. Results can be restricted based on a range of properties, including publication date and venues, size of the patient groups, etc. Figures 5, 6 show SPARQL queries for the some of the mentioned use–cases.

Automating the process of generating meta–analyses reports involves building and executing the right set of SPAQRL queries based parameters of the study. Consider the use-case of generating a meta–analysis report that studies the correlation between cardiovascular risk factors and venous thrombosis. An automated system first would find *Find* all relevant previous studies that compared various cardiovascular risk factors and venous thrombosis. This task can be achieved by constructing and executing a query like (a) in Figure 5. Once all the relevant studies are retrieved, the system extracts relevant data from tables associated with studies. Data *Extraction* again can be done via SPARQL query like the one shown in (b) in Figure 5. After data is extracted, this system can run defined statistical tests and *Integrate* the results. These results would be key to produce the final meta–analysis report.

While we presented a system that attempts to automatically generate meta–analyses reports, a sensitive domain such as healthcare which demands highest levels of accuracy would benefit from a person-in-the-loop approach. An interactive framework will allow medical researchers to inspect and correct inferred semantics of medical tables and allow to perform structured search over data inferred from tables. Systems such as Graph of Relations [8] can be used to allow users to construct SPARQL queries using natural language terms and graphical interfaces.

5 Evaluation

We present preliminary evaluations of how well our prototype framework (described in section 3) performs in inferring the semantics of medical tables and its utility in discovering relevant studies associated with a meta–analysis report.

Evaluating inferred semantics of medical tables. We begin by collecting two datasets: one of meta–analysis reports and associated studies used to generate them and another of the medical tables extracted from these studies. A dataset of six different meta–analyses reports was provided to us by our collaborators at the University of Maryland, School of Medicine. We further obtained publicly available medical studies used to generate each meta–analysis report and manually extracted tables embedded in these documents.

The extracted medical tables go through the *Normalize*, *Query and Rank* and *Generate RDF* modules finally leading to RDF linked data as output. In this section, we specifically focus on the evaluation of *Query and Rank* step which is crucial in inferring the semantics of tables. In the absence of the *Joint Inference* step, the top ranked class (or instance) from the ranked list of candidates is assigned to the column and row headers. Thus, instead of evaluating *Q&R* on query strings parsed by *Normalize*, we randomly choose 25 query string terms[4] from our dataset. For each query string, we obtain ground truth by mapping[5] it to an appropriate concept (or instance) for each of the three sources – SNOMED CT, UMLS Metathesaurus, and DBpedia.

For each query string, a ranked list of candidate concepts (or instances in case of DBpedia) are generated as described in section 3. The number of candidates in the ranked list was restricted to 25 for UMLS and DBpedia and 100 in the case of SNOMED. The module was able to correctly link (i.e., correct concept at rank 1) 14 out of 25 for SNOMED CT; 12 out of 25 for UMLS and 7 out of 25 for DBpedia. In cases where it was not able to link to the right concept, a majority appeared between the ranks 2 and 5 (7 for SNOMED CT; 5 for UMLS and 5 for DBpedia). *Q&R* failed to discover the correct concept in its ranked list only three times for

[4] A list of the query terms is available online at [14].

[5] The mapping was performed by authors of the paper

SNOMED, but eight times for UMLS and eleven times for DBpedia.

The UMLS API failed to find the right concepts in its ranked list largely in cases where query string consisted of a modifier (e.g., *no diabetes*, *treatment with statins*, *active cancer*) or query string that used abbreviations (e.g., *recurrent VTE*). In addition to the cases where query string included modifiers, mapping concepts to DBpedia often failed because of its broad coverage. For example, query terms such as *age* and *race* were mapped to a number of instances in DBpedia and the limited context provided by the table made disambiguation challenging. If we increase the number of candidates in the ranked list for DBpedia from 25 to 100, the number of concepts not found is reduced from 11 to 7, with the correct concepts appearing at ranks 19, 23, 76 and 98.

Restricting or filtering the set of DBpedia instances that *Q&R* queries against will improve accuracy. For example, it might be useful to query for instances under Wikipedia categories such as *Human Anatomy*, *Medicine*, *Demographics*. In the case of the SNOMED CT knowledge base, the exact match query was invoked 17 times, whereas a text search query was required only eight times, indicating large overlap between terms used in medical research literature and ones used in the medical knowledge sources. In our limited dataset, we also noticed cases where SNOMED failed to find the right concept, but UMLS succeeded. For example, terms such as *control groups* or statistical tests such as *odds ratio, p–value* are present in UMLS but not in SNOMED. The broad coverage of UMLS, which draws on a combination of 150 different datasets, is both boon and a bane. UMLS often contains slight variations for the same concept extracted from different sources, which makes disambiguation more challenging.

Evaluating *Find*. We selected a meta–analysis report and identified individual studies used to generate the report. Our selected meta–analysis report analyzed correlation between various cardiovascular risk factors such obesity, diabetes, hypertension and venous thrombosis [1]. For each risk factor, different set of studies were used to produce a conclusion on whether correlation exists or not. We extract tables from these studies and represent them as RDF Linked Data using our framework. We start with the assumption that tables were normalized; thus framework skips *Normalize* and begins with the *Q&R* module.

Retrieval queries were executed against a set of triples generated from the following tables: Table 1 from [17] (t1-Paganin), Table 1 from [18] (t1-Prandoni), Table 2 from [7] (t2-Frederiksen), and Table 1 from [6] (t1-Deguchi). Figure 7 lists out the medical factors used in the evaluation. The query column lists the UMLS concept associated with the factor used in a SPARQL query to retrieve the tables,

Factor	*Query*	*Expected,Retrieved*
Obesity	Obesity	2, 2
Hypertension	Hypertensive_disease	2, 2
Diabetes	Diabetes_Mellitus	2, 1
Smoking	Smoking	2,1

Figure 7. Find performance for different cardiovascular risk factors. Last column indicates number of expected and retrieved tables.

with the last column showing the number of expected and retrieved tables. In the context of the evaluation, SPARQL query of the form (c) in Figure 6 was executed. This evaluation is promising: relevant observations were retrieved for both *obesity* and *hypertension* and in no cases irrelevant sets of observations retrieved. However, in the case of *diabetes* and *smoking*, the query failed to retrieve all relevant observations due to errors in linking in the *Q&R* module. In the case of *diabetes* and *smoking*, the relevant query strings were mapped to concepts *smoker* and *diabetes* in *t1-Prandoni* whereas the concept used in retrieval query were *diabetes_mellitus* and *smoking*. The case of *smoker* vs. *smoking* is interesting: the latter is represented as an *individual behavior* in UMLS whereas the former is represented as a *finding*. However both concepts seem accurate to describe whether a person is a smoker. Similar ambiguity arises in the cases like *triglyceride*: is the correct concept for *triglyceride*, *Triglyceride – Biologically Active Substance* or *Triglyceride level – finding*. These challenges will have to be either tackled in *Q&R* by disambiguating to similar type of entity (i.e., always prefer type *Finding*) or in retrieval phase by executing multiple queries for related concepts.

6 Related Work

We present related work from three different threads of research. The first focused on inferring the semantics of data found in tables, the second on automating the generation meta–analysis reports and third on using ontologies to model clinical trials and other medical research studies.

Recent research has focused on inferring the semantics of tables, but most, including our own [16] has focused on inferring the semantics of tables found on the web (webtables). Wang et al. [31] begin by identifying a single 'entity column' in a table and, based on its values and rest of the column headers, associate a concept from the Probase knowledge base with the table. Ventis et al. [30] assign multiple class labels (or concepts) to table columns and identify relations between the 'subject' column and the rest of the

columns in the table. Limaye et al. [12] use a graphical model which maps every column header to a class from a known ontology, links table cell values to entities from a knowledge-base and identifies relations between columns. Szekely et al. [28] present an interactive tool to convert tabular data into RDF. The tool uses a conditional random field model to suggest initial mappings to users; if they wish to, users are allowed to change the mapping by selecting another suggestion or picking a new term directly from the ontology.

Webtables typically contain column headers and their data cells contain strings, many of which refer to entities in a knowledge base. Medical tables present unique challenges not present in webtables, thus making it necessary to modify and adapt existing techniques. Medical tables have a unique structure with header cells in both row and columns and data cells consisting almost exclusively of numerical data, often with several numbers per cell. Our modified framework is able to normalize medical tables, query different sources and model and represent them in RDF.

Research has also focused on taking steps towards automating evidence–based medicine and generating meta–analysis reports. Cohen et al. [3] present a design for end-to-end text-mining pipeline to automate the process of generating and updating meta–analysis report. Their pipeline consists of searching, classifying, grouping and ranking medical research papers to produce systematic reviews. ExaCT [11] is a information extraction system that searches and extracts sentences from clinical trials and other related studies that best match the clinical trial characteristics provided by the user. It aims to facilitate in the process of identifying relevant studies for producing evidence reports.

Researchers have produced systems to create summaries of medical papers [20] and also from medical paper abstracts [26] to be used in meta–analysis studies. Others have have applied machine learning techniques to reduce the number of search query results a medical researcher must analyze to collect relevant studies [10, 4]. However, this entire body of work has focused on analyzing free text; to the best of our knowledge no work has focused on analyzing and inferring information encoded in tables in medical research literature. Our approach can not only find relevant studies, but can also extract and integrate the data to produce meta–analysis reports.

Related work has also focused on using ontologies and other Semantic Web technologies in assisting clinical trial management. Frameworks such as the ObTiMA System [25] and Epoch [22] allow management of ongoing clinical trial by providing tools to researchers to capture data and represent data as RDF. Both provide ontologies useful for representing data of ongoing clinical trials as RDF. ORCe [23] is a general purpose ontology allowing users to model various aspects related to clinical trials such as study design, study protocol, statistical concepts related to the study analysis. ADDIS [29] presents an ontology to ground various clinical trials in a common data schema, facilitating search and integration. ADDIS further provides users a semi-automatic decision support software that allows importing studies, representing them in RDF and producing evidence reports. LinkedCT [9] triplifies data sources such as ClinicalTrials.gov and Dameron et al. [5] designed an ontology to model and reason about patient eligibility in clinical trials. While existing work has covered the breadth in clinical trial management using Semantic Web technologies, our ontology has focused on a very specific task – modeling and representing medical tables published in medical research papers.

7 Conclusions

Evidence-based medicine is increasingly vital in health care decision making. Producing evidence reports and meta–analysis reports in still largely manual and can benefit from better tools that can assist in the process of generating these reports. Existing work has focused on automatically analyzing free text, largely ignoring tables which encode key information required to identify relevant studies as well as generate meta–analysis reports.

We presented a framework for inferring the semantics of tables published in medical research papers and modeling and representing them as RDF linked data. We demonstrated the benefits RDF medical tables provide in the process of finding, extracting and integrating data from individual studies to produce meta–analysis reports. A preliminary evaluation showed promising results, but leaves room for extension and improvement. We believe our framework can address the challenges in inferring the semantics of medical tables and it is a step in the direction of building a framework for automating the process of generating evidence reports.

Our future work will include exploiting table captions and descriptions which often hold information helpful or even essential to understand a table, incorporating a person-in-the-loop architecture to identify incorrect semantics inferred by the framework and performing a thorough evaluation across different areas in medical research.

Acknowledgement.

This research was supported by NSF awards 1228198, 1250627 and 0910838 and a gift from Microsoft Research.

References

[1] W. Ageno, C. Becattini, T. Brighton, R. Selby, and P. W. Kamphuisen. Cardiovascular risk factors and venous throm-

boembolism a meta-analysis. *Circulation*, 117(1):93–102, 2008.

[2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009.

[3] A. Cohen, C. Adams, J. Davis, C. Yu, P. Yu, W. Meng, L. Duggan, M. McDonagh, and N. Smalheiser. Evidence-based medicine, the essential role of systematic reviews, and the need for automated text mining tools. In *1st Int. Health Informatics Symposium*, pages 376–380. ACM, 2010.

[4] A. M. Cohen, K. Ambert, and M. McDonagh. Cross-topic learning for work prioritization in systematic review creation and update. *Journal of the American Medical Informatics Association*, 16(5):690–704, 2009.

[5] O. Dameron, P. Besana, O. Zekri, A. Bourdé, A. Burgun, M. Cuggia, et al. Owl model of clinical trial eligibility criteria compatible with partially-known information. In *SWAT4LS*, 2012.

[6] H. Deguchi, N. M. Pecheniuk, D. J. Elias, P. M. Averell, and J. H. Griffin. High-density lipoprotein deficiency and dyslipoproteinemia associated with venous thrombosis in men. *Circulation*, 112(6):893–899, 2005.

[7] J. Frederiksen, K. Juul, P. Grande, G. B. Jensen, T. V. Schroeder, A. Tybjærg-Hansen, and B. G. Nordestgaard. Methylenetetrahydrofolate reductase polymorphism (c677t), hyperhomocysteinemia, and risk of ischemic cardiovascular disease and venous thromboembolism: prospective and case-control studies from the copenhagen city heart study. *Blood*, 104(10):3046–3051, 2004.

[8] L. Han, T. Finin, and A. Joshi. GoRelations: An Intuitive Query System for DBpedia. In *Proc. Joint Int. Semantic Technology Conf.*, LNCS. Springer, Dec. 2011.

[9] O. Hassanzadeh, A. Kementsietsidis, L. Lim, R. J. Miller, and M. Wang. Linkedct: A linked data space for clinical trials. *arXiv preprint arXiv:0908.0567*, 2009.

[10] H. Kilicoglu, D. Demner-Fushman, T. C. Rindflesch, N. L. Wilczynski, and R. B. Haynes. Towards automatic recognition of scientifically rigorous clinical research evidence. *Journal of the American Medical Informatics Association*, 16(1):25–31, 2009.

[11] S. Kiritchenko, B. de Bruijn, S. Carini, J. Martin, and I. Sim. Exact: automatic extraction of clinical trial characteristics from journal publications. *BMC medical informatics and decision making*, 10(1):56, 2010.

[12] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. In *Proc. 36th VLDB*, 2010.

[13] Medical Table Ontology. http://ebiquity.umbc.edu-/ontology/mto/v1/, May 2014.

[14] Medical Tables Linked Data Project Page. http://ebiq.org/j/101, May 2014.

[15] V. Mulwad, T. Finin, and A. Joshi. A Domain Independent Framework for Extracting Linked Semantic Data from Tables. In *Search Computing - Broadening Web Search*, pages 16–33. Springer, July 2012. LNCS v. 7538.

[16] V. Mulwad, T. Finin, and A. Joshi. Semantic message passing for generating linked data from tables. In *12th Int. Semantic Web Conf.*, pages 363–378. Springer, 2013.

[17] F. Paganin, A. Bourde, J.-L. Yvin, R. Genin, J.-L. Guijarro, A. Bourdin, and C. Lassalle. Venous thromboembolism in passengers following a 12-h flight: a case-control study. *Aviation, space, and environmental medicine*, 74(12):1277–1280, 2003.

[18] P. Prandoni, F. Bilora, A. Marchiori, E. Bernardi, F. Petrobelli, A. W. Lensing, M. H. Prins, and A. Girolami. An association between atherosclerosis and venous thrombosis. *New England Journal of Medicine*, 348(15):1435–1441, 2003.

[19] D. Sackett, W. Rosenberg, J. Gray, R. Haynes, and W. Richardson. Evidence based medicine: what it is and what it isn't. *Bmj*, 312(7023):71, 1996.

[20] A. Sarker, D. Mollá, and C. Paris. An approach for query-focused text summarisation for evidence based medicine. In *Artificial Intelligence in Medicine*, pages 295–304. Springer, 2013.

[21] P. L. Schuyler, W. T. Hole, M. S. Tuttle, and D. D. Sherertz. The UMLS metathesaurus: representing different views of biomedical concepts. *Bulletin of the Medical Library Association*, 81(2):217, 1993.

[22] R. D. Shankar, S. B. Martins, M. J. O'Connor, D. B. Parrish, and A. K. Das. Epoch: an ontological framework to support clinical trials management. In *Proceedings of the international workshop on Healthcare information and knowledge management*, pages 25–32. ACM, 2006.

[23] I. Sim, S. W. Tu, S. Carini, H. P. Lehmann, B. H. Pollock, M. Peleg, and K. M. Wittkowski. The ontology of clinical research (ocre): An informatics foundation for the science of clinical research. *Journal of biomedical informatics*, 2013.

[24] M. Q. Stearns, C. Price, K. A. Spackman, and A. Y. Wang. SNOMED clinical terms: overview of the development process and project status. In *AMIA Symposium*, page 662. AMIA, 2001.

[25] H. Stenzhorn, G. Weiler, M. Brochhausen, F. Schera, V. Kritsotakis, M. Tsiknakis, S. Kiefer, and N. Graf. The obtima system-ontology-based managing of clinical trials. *Stud Health Technol Inform*, 160(Pt 2):1090–1094, 2010.

[26] R. L. Summerscales, S. Argamon, S. Bai, J. Huperff, and A. Schwartzff. Automatic summarization of results from clinical trials. In *International Conference on Bioinformatics and Biomedicine*, pages 372–377. IEEE, 2011.

[27] Z. Syed and T. Finin. Creating and Exploiting a Hybrid Knowledge Base for Linked Data. In *Agents and Artificial Intelligence*, pages 3–21. Springer, April 2011.

[28] P. A. Szekely, C. A. Knoblock, F. Yang, X. Zhu, E. E. Fink, R. Allen, and G. Goodlander. Connecting the smithsonian american art museum to the linked data cloud. In *10th Int. Conf. on the Semantic Web: Semantics and Big Data (ESWC)*, pages 593–607, 2013.

[29] G. Van Valkenhoef, T. Tervonen, T. Zwinkels, B. De Brock, and H. Hillege. Addis: a decision support system for evidence-based medicine. *Decision Support Systems*, 55(2):459–475, 2013.

[30] P. Venetis, A. Halevy, J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, and C. Wu. Recovering semantics of tables on the web. In *Proc. 37th VLDB*, 2011.

[31] J. Wang, B. Shao, H. Wang, and K. Q. Zhu. Understanding tables on the web. Technical report, Microsoft Research Asia, 2011.