Data Driven Knowledge Acquisition Method for Domain Knowledge Enrichment in the Healthcare

Sujan Perera, Cory Henson, Krishnaprasad Thirunarayan, Amit Sheth
Ohio Center of Excellence in Knowledge-enabled Computing (Kno.e.sis)
Wright State University, Dayton OH, USA
{sujan, cory, tkprasad, amit}@knoesis.org

Suhas Nair
ezDI, LLC
suhas.nair@ezdi.us

***Abstract*—Semantic computing technologies have matured to be applicable to many critical domains, such as life sciences and health care. However, the key to their success is the rich domain knowledge which consists of domain concepts and relationships, whose creation and refinement remains a challenge. In this paper, we develop a technique for enriching domain knowledge, focusing on populating the domain relationships. We determine missing relationships between the domain concepts by validating domain knowledge against real world data sources. We evaluate our approach in the healthcare domain using Electronic Medical Record(EMR) data, and demonstrate that semantic techniques can be used to semi-automate labour intensive tasks without sacrificing fidelity of domain knowledge.**

***Index Terms*—knowledge acquisition, semantic technology, Electronic Medical Record**

I. Introduction

Semantic computing has already begun to play a significant role in the health care domain[1][2]. Experts have realized the significant value of using semantic technologies to solve the heterogeneity challenge[3] and improve our understanding of health care data. The availability of rich domain knowledge base (DKB)(collection of facts as entities/concepts and relationships) is a key to the success of semantic solutions[6].

DKB consists of domain concepts and their relationships. These relationships that associate meaning between concepts play a crucial role in realizing the full potential of semantic computing[7]. Unfortunately, the existing standard vocabularies in healthcare domain like SNOMED[4] and ICD[5] concentrate on classification and organization of concepts, but do not focus on many interesting relationships that exist between concepts[10]. However, creating rich DKB requires overcoming challenges such as, the unavailability of domain experts, the difficulty in obtaining a community agreement, the inherently dynamic nature of domain knowledge, the difficulty of assessing the completeness of coded domain knowledge, the lack of expressiveness of knowledge representation languages, and the shortage of resources (time and labour) and people with necessary expertise.

Typical process of manually creating DKB requires huge effort from the ontologists (i.e., knowledge engineers) and the domain experts. The task of gathering domain knowledge is similar to the requirements elicitation problem in the Software Development Life Cycle (SDLC). This task is inherently incomplete due to knowledge gaps that exist between ontologists and domain experts. An ontologist may be unaware of the kind of knowledge to gather from a domain expert, and a domain expert may be unaware of the knowledge they should provide to solve the problem in a technically sound manner. Furthermore, the resulting domain knowledge is often subjective, as it largely depends on the experiential knowledge of a small group. It is not reasonable to assume that a single person or a small group of people would have complete knowledge of a domain like healthcare, or even one of its subareas (e.g., cardiology). Therefore, the DKB built with the inputs from experts will suffer from shortcomings such as inefficiency in capturing expert knowledge, subjectivity, and incompleteness.

To address these issues, we propose a systematic method of identifying missing relationships between concepts in DKB, and generating suggestions to fill those gaps. We demonstrate its effectiveness in the health care domain. It starts by building DKB with the input from the healthcare professionals and uses this initial DKB to explain EMR documents. We call EMR document "explainable/consistent", when the symptoms in it can be explained by the disorders in it otherwise, it is "unexplainable/inconsistent". We have identified four possible reasons (refer Section III-B4) for initial DKB to fail in explaining EMR documents, including the absence of required relationships between domain concepts. Our approach uses these "unexplainable" documents to spot the gaps in DKB and generate suggestions to fill those gaps. This is a bootstrapping approach since we used the DKB to enhance itself. Ultimately, our approach can compensate for the subjectivity and incompleteness of a DKB.

Contributions of this paper include the following:

1) It proposes a method to validate DKB by using real-world data sources,
2) It proposes a bootstrapping/semi-supervised method for finding missing relationships in a given DKB,
3) It proposes a convenient method to elicit missing domain knowledge from experts (i.e., find and rank missing relationships based on statistical properties in the dataset).

In Section II we discuss the factors which motivate our approach for domain knowledge acquisition. Section III discusses each step of the proposed method in detail. Section IV demonstrates it using an example. Section V discusses

978-1-4673-2560-8/12/$31.00 ©2012 IEEE

the implementation details. Evaluation results are presented in Section VI and VII. Section VIII contains related work and Section IX concludes the paper.

II. Motivation

This work was motivated by the need to retrieve and organize patient records based on relationships among concepts. For example, while a patient's EMR document may not explicitly mention the circulatory disorder such as *myocardial infarction (MI)* during diagnosis, if such a patient has been prescribed medications generally given to the *MI* patients, or if a patient has symptoms generally caused by *MI*, we would like this patient to be included in the result set for a query for *MI* patients. This requires explicit representation of domain knowledge that relates a disease to its symptoms and medications, and incorporation of rich DKB into our search application. Furthermore, we improve the query results by using the DKB to remedy syntactic gaps in the data sets, e.g., retrieval of the patients with *shortness of breath*, ideally, should include the patients with *dyspnea* (*dyspnea* is synonym for *shortness of breath*), but this is possible only if the application "knows" that both terms refer to the same concept. The DKB that will emerge from this process is expected to cover disorders, symptoms, medications, and procedures in cardiology domain as concepts, and the relationships that exist between these concepts. For example:

1) A symptom *is caused by* a disorder,
2) A medication *is prescribed for* a disorder,
3) A procedure *is conducted for* a disorder.

The identification of the related domain concepts was done by identifying the entities in EMR documents with the help of domain expert. But learning and populating the relationships among these concepts is still a tedious task. For example, a domain expert is required to enumerate all the symptoms against each disorder and link two concepts if a relationship exists. Assume that there are 50 disorders and 100 symptoms, then there are 5000 possibilities. So, this approach is labour intensive, and results are subjective. This observation motivates the search for better methodology to acquire domain knowledge.

III. The Approach

We propose a question answering mechanism to acquire domain knowledge based on the IntellegO ontology[8]. Questions are generated by leveraging the DKB and EMR documents. We focus on how to discover missing domain knowledge in the form of relationships between disorders and symptoms. The same method is general enough to be applicable to other relationships mentioned above.

A. Ontology of Perception

Perception is the process of interpreting observations of the environment to derive situational awareness; or, in other words, the process of translating low-level observations into high-level abstractions. People have evolved sophisticated mechanisms to efficiently perceive their environment; machines, however, continue to struggle with this task. IntellegO is an ontology that provides a formal semantics of machine perception by defining the informational processes involved in translating observations into abstractions. The ontology is encoded in set-theory and has been used in various applications, including a weather-alert service[8] and a fire-detecting robot[24].

Diagnosis is a function of Perception. Medical professionals derive the disorder (abstraction) by examining the symptoms (low level signals). EMR documents contain the knowledge involved in the informational process of converting the symptoms into disorders. We used Intellego ontology because it nicely aligns with the characteristics of the knowledge that we represent in healthcare domain and improves the interoperability of knowledge, and supports the reasoning we required.

This paper provides an alignment between the IntellegO ontology and the cardiology DKB, and introduces the new concept of *'intellego:coverage'*[1]. The semantics of *'intellego:coverage'* is defined in Section III-B3.

The proposed methodology uses a subset of concepts (*'intellego:entity'*, *'intellego:quality'*, *'intellego:percept'* and *'intellego:explanation'*) and the *'intellego:inheres-in'* relationship from IntellegO ontology. Here we discuss the semantics of these primitives.

Let us take *hypertension* and an associated symptom, *chest pain*. *Hypertension* is an *'intellego:entity'* and *chest pain* is an *'intellego:quality'*. The *chest pain 'intellego:quality'* is an inherent property of *hypertension 'intellego:entity'*. In general, *'intellego:entity'* is an object or event in the world and *'intellego:quality'* is an inherent property of an *'intellego:entity'*. *'intellego:inheres-in'* is the relationship between *'intellego:quality'* and *'intellego:entity'*.

The perception process begins by observing a few qualities. From these observations, it derives entities which can explain the observed qualities. *Chest pain* can be explained by the presence of *hypertension*[2]. Set of observed *'intellego:quality'* (e.g., *chest pain*) are members of the class *'intellego:percept'* and *'intellego:entity'* (e.g., *hypertension*) which can explain the *'intellego:percept'* are members of the class *'intellego:explanation'*. *'intellego:explanation'* and *'intellego:percept'* are sub-classes of *'intellego:entity'* and *'intellego:quality'* respectively.

B. Data Driven Methodology

We have developed a generic, reproducible methodology for systematically detecting missing knowledge from an existing DKB and suggest likely relationships to 'fill in the gaps'.

Throughout this process, we assume that the EMR documents are consistent, i.e., the symptoms appearing in the documents are accounted for by the disorders in it. This is not always true, but still sufficient to achieve our goals.

The proposed method uses an initial DKB, which is built from the initial knowledge available on the relationship between disorders and symptoms. The following steps summarize the method.

[1] intellego prefix specifies terms from the IntellegO ontology

[2] Note that there may be multiple entities that can explain the observed qualities (e.g., *chest pain* can be explained by *hypertension*, *cardiomyopathy*, *coronary artery disease* and a host of other disorders), but for simplicity of the example, we assume *chest pain* can be explained only by *hypertension*.

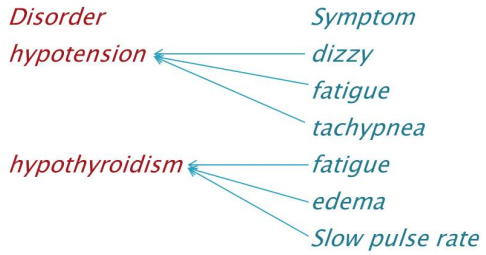

Fig. 1. inheres-in Relationships between Disorders and Symptoms

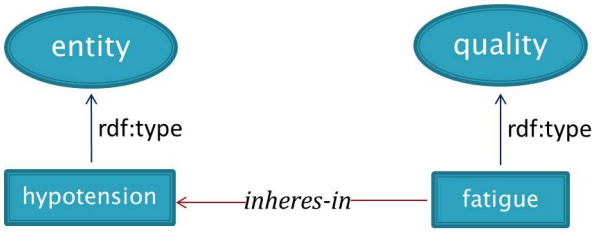

Fig. 2. Knowledge Representation in IntellegO

1) Build initial DKB,
2) Semantically annotate the EMR documents with concepts from the DKB,
3) Generate the *'intellego:coverage'* for document,
4) Identify inconsistent EMR documents[3],
5) Suggest candidate relationships between disorders and symptoms that can rectify these inconsistencies,
6) Provide specific questions to ask the domain experts about the correctness of the candidate relationships,
7) Update the DKB based on the feedback.

We now discuss each step in detail.

1) Build Initial Domain Knowledge Base

The DKB consists of a bi-partite graph relating symptoms to disorders[4], as depicted in Figure 1. The initial DKB is constructed with the minimum involvement from domain experts. The alignment of IntellegO ontology and cardiology DKB results in a mapping between disorders from the cardiology DKB to *'intellego:entity'* and symptoms from the cardiology DKB to *'intellego:quality'*. Figure 2 illustrates how the instances of cardiology DKB are annotated with concepts in IntellegO.

In addition to the *'intellegO:inheres-in'* relationship, the DKB also uses the *owl:sameAs* relationship to assert the equivalency of two individuals (e.g., *shortness of breath* is same as *dyspnea*). This helps to overcome some heterogeneity (due to synonyms). We represent each concept with its SNOMED ID[5].

2) Data Annotation

After building the initial DKB, the next step is to semantically annotate the EMR documents. Semantic annotation is the process of mapping terms in a document with concepts defined in the DKB. Semantic annotation is used to make unstructured data machine understandable; i.e., to make the semantics explicit. This section describes how the EMR documents are annotated.

```
<problem v="fatigue" code="SNM:84229001">
   <certainty v="no"/>
   <degree v="high degree"/>
   <sectname v="report history of present illness item"/>
   <code v="SNM:84229001"/>
</problem>
```

Fig. 3. XML Element 'problem' Describe the Properties of *fatigue*

An EMR is a semi-structured document that has several sections, including:

- Current diagnosis – list of disorders afflicting the patient,
- Allergies - list of medications that the patient is allergic to,
- Medications - current list of medications the patient is taking,
- Patient history - medical history of the patient,
- Review of systems - observations of each bodily system (cardiovascular, neurological, etc),
- Physical Examination - symptoms and readings relevant to the health condition (blood pressure, heart rate, etc...),
- Impression and Recommendation - the doctor's interpretation of the condition and conclusion.

Our method uses an existing Natural Language Processing (NLP) entity spotting tool such as MedLEE[9], to convert a semi-structured document to a structured document. This tool identifies disorders, symptoms, medications, and procedures mentioned within the EMR document, and converts these terms to XML elements. These XML elements provide information that further describe the concept, e.g., the XML element 'problem' represents the information about a disorder or a symptom, such as degree (severity), certainty (confidence that the EMR document conveys about a disorder or a symptom), the body location, the section of the EMR document in which the concept found, the SNOMED code of the concept (Figure 3).

The proposed method uses the disorders and the symptoms mentioned in current diagnosis, review of systems, and physical examination sections within the EMR, because these sections contain information most relevant to the patient's current status. Each symptom which appears in these sections is annotated as *rdf:type 'intellego:percept'* and each disorder is annotated as *rdf:type 'intellego:explanation'*. Recall that observed *'intellego:quality'* is also *'intellego:percept'* and *'intellego:entity'* that can explain a set of *'intellego:percept'* is also *'intellego:explanation'*. Furthermore, the disorders in a consistent EMR document should explain the symptoms in the document.

3) Generate 'intellego:coverage'

Coverage can be described as the aggregation of *'intellego:quality'* that can be accounted for by *'intellego:explanation'*.

[3]whenever we say inconsistent EMR, we mean EMR document is inconsistent w.r.t DKB, - i.e DKB could not explain the EMR document

[4]Throughout this paper the relationship/link between disorders and symptoms, refers to the bi-partite graph.

[5]However. throughout this paper, we use string representation of a concept to improve readability

Formally, coverage is defined as,

$$coverage : Powerset(E) \rightarrow Powerset(Q)$$

$$\text{coverage(F)} = \{q \in Q | q \text{ intellego:inheres-in e } \wedge \text{ e} \in F\}$$

Here q is an instance of quality, e is an instance of entity, Q is 'intellego:quality' and E is 'intellego:entity'.

The following example shows an encoding of coverage class in Manchester syntax[6] for an EMR document which reports *atrial fibrillation*, *hypertension* and *diabetes* as disorders.

Class: coverage
  EquivalentTo:
    (intellego:inheres-in value atrial fibrillation)
    or (intellego:inheres-in value diabetes)
    or (intellego:inheres-in value hypertension)
  SubClassOf:
    intellego:quality

4) Identify Inconsistent EMRs

The next step compares the set of *'intellego:coverage'* with the set of *'intellego:percept'* in order to discover discrepancies between the DKB and EMR documents. For a consistent EMR document, *'intellego:percept'* should be a subset of *'intellego:coverage'*, i.e., observed symptoms should be a subset of all possible symptoms that can be explained by the disorders present. Whenever our method finds a symptom in the set of *'intellego:percept'* that is not in the *'intellego:coverage'* set, it flags it as inconsistent. We have identified the following situations which may cause inconsistency:

- The text extraction can introduce errors,
  The input to the proposed method is structured text from entity spotting tool. This conversion can introduce errors. For example, we have observed that occasionally it does not interpret negation correctly.
  E.g., the phrase 'Denies any history of depression' can incorrectly result in a XML element that does indicate a presence of depression. This causes unexpected symptoms to appear XML document.
- Some combination of disorders can produce symptoms that are not inherent in any individual disorders, The DKB does not represent complex situations where multiple disorders can manifest new symptoms over and beyond those caused by individual disorders through complex interactions over time. This scenario can present extra symptoms in *'intellego:percept'* than *'intellego:coverage'*.
- Irrelevant observations,
  An EMR document can contain symptoms that are not used for diagnosis causing it to be inconsistent (symptoms in it are not explained by the disorders in it).
- Missing domain knowledge (i.e., missing relations between *'intellego:entity'* and *'intellego:quality'*).
  The accuracy and completeness of the *'intellego:coverage'* depends on the accuracy and completeness of the DKB. If the DKB lacks a relationship, then the generated *'intellego:coverage'* set can be incomplete.
  E.g., Assume that a *hypertension* patient has *edema*. But the relationship between *hypertension* and *edema* (*edema 'intellego:inheres-in' hypertension*) is missing in the DKB. This leads to generate incomplete *'intellego:coverage'* for the above patient's EMR document, therefore *'intellego:percepts'* is not a subset of *'intellego:coverage'*.

[6]http://www.w3.org/TR/owl2-manchester-syntax/

Since our intention is to build a rich DKB about the disorders, symptoms and their relationships, our focus is on remedying missing domain knowledge.

5) Identify Candidate Relationships

The identification of inconsistent EMR documents (w.r.t DKB) indicates gaps in DKB. The next step is to spot the missing relationships, that would rectify the inconsistencies. This is achieved by collecting the set of disorders that appear with unexplained symptom which then serve as candidates for completion. These candidates are ranked based on number of times they co-occur with the unexplained symptom in inconsistent EMR documents.

6) Generate Questions

Candidate relationship identification allows us to generate questions for the domain expert in order to acquire the missing domain knowledge, e.g., if symptom A is found to be unexplained in x number of documents and disorder B appeared with A in y (<=x) number of such documents, A becomes a candidate and the question "Does A inheres-in B?" can be formulated.

The final step is to get the expert feedback on the proposed relationships between the symptoms and disorders, and update the DKB. Several iterations of these Q&A steps are necessary to improve the DKB.

Our method is applicable to the knowledge in bi-partite graph form. Bi-partite graph knowledge design pattern is a common pattern for design of the environmental knowledge. For example, a widely used ontology like SSN[25][26] uses the relation *ssn:isPropertyOf* between property and feature. Furthermore, this pattern is aligned with the concepts in the DOLCE Ultra Lite ontology (a upper level ontology)[27] in order to enable the integration with other ontological knowledge on the web. These observations suggest wider applicability of our method.

IV. Use Case

We demonstrate our approach to convenient domain knowledge acquisition on a concrete example. The EMR document considered has the following disorders and symptoms:

Disorders:	Symptoms:
atrial fibrillation	chest pain
hypertension	weight gain
diabetes	numbness
hypertrophy	

Recall that, both disorders and symptoms are represented with XML element 'problem' in structured EMR documents. Hence, the structured version of the EMR document does not distinguish between symptoms and disorders. We use the DKB to distinguish between the symptoms and disorders (The DKB

has symptoms labelled as *'intellego:quality'* and disorders as *'intellego:entity'*). Once we categorize the concepts, all the disorders found in the EMR document are annotated as *'intellego:explanation'*, and the symptoms as *'intellego:percept'* (Recall from Section III-A that observed qualities are *'intellego:percept'* and possible entities that can explain the percepts are *'intellego:explanation'*. The disorders in a consistent EMR document should be able to explain the symptoms in it). The next step is to generate the *'intellego:coverage'* for the disorders within the EMR. The definition of *'intellego:coverage'* class for this particular EMR is,

Class: coverage
EquivalentTo:
(intellego:inheres-in value atrial fibrillation)
or (intellego:inheres-in value diabetes)
or (intellego:inheres-in value hypertension)
or (intellego:inheres-in value hypertrophy)
SubClassOf:
intellego:quality

Once the coverage class is defined, an OWL-DL reasoner is used to infer the instances of the *'intellego:coverage'* class (i.e., all the symptoms inheres-in disorders in definition of *'intellego:coverage'*). The following is the set of instances of *'intellego:coverage'* for the above example.

dystonia tongue, cough, respiratory distress,nausea, vomiting, shortness of breath, tiredness, pain face, weight gain, ischemia, headache, dizzy, pain calf, retention leg, chest pain, cerebrovascular accident, weight loss, syncope, distress, pain calf, polydipsia, discomfort in chest, fatigue, edema extremity, flushing, and neuropathy

By comparing the *'intellego:coverage'* set and *'intellego:percepts'* set, we see that *numbness* is not listed in the *'intellego:coverage'* set but is listed in the *'intellego:percept'* set. This alerts the system of an inconsistency. From this inconsistency, we hypothesize that at least one disorder – among *atrial fibrillation*, *diabetes*, *hypertension*, and *hypertrophy* – should have a relationship with *numbness*. The DKB and the semantics of the *'intellego:coverage'* enable us to generate this hypothesis. We are now able to formulate the following questions to the domain expert.

1) Does *numbness* inheres-in *atrial fibrillation*?
2) Does *numbness* inheres-in *diabetes*?
3) Does *numbness* inheres-in *hypertension*?
4) Does *numbness* inheres-in *hypertrophy*?

This approach to acquire domain knowledge is better than the naive way of enumerating all possibilities. Here it was found that *numbness* is a symptom of all these disorders; this knowledge was missing in the DKB, causing the alert to be generated.

V. Implementation Details

We used Java to implement the proposed methodology, and the OWL API[7] was used to interact with the ontology. The Pellet reasoner[8] was used for the reasoning task of finding the instances of class *'intellego:coverage'*.

[7]http://owlapi.sourceforge.net/
[8]http://clarkparsia.com/pellet/

Symptom	# times unexplained
gastroesophageal reflux	246
edema	236
depression	190
angina	142
shortness of breath	118
syncope	91
chest pain	80
weight gain	67
discomfort in chest	66
headache	63
fatigue	50

TABLE I
SYMPTOMS NOT EXPLAINED BY A DISORDER

Disorder	# times
hyperlipidemia	134
hypertension	131
atrial fibrillation	95
coronary artery disease	64
diabetes	57
hypothyroidism	48
chronic obstructive pulmonary disease	28
peripheral vascular disease	26
ischemia	23
cardiomyopathy	20
transient ischemic attack	19
cerebrovascular accident	18

TABLE II
DISTRIBUTION OF DISORDERS CO-OCCURRING WITH *edema*

VI. Evaluation

We evaluated the proposed method using a set of de-identified EMR documents[9] collected from different hospitals about different patients. By using the diverse set of documents, the subjective nature of the acquired knowledge can be minimized. The corpus consists of 1,535 EMR documents and their structured (XML) counterpart. We randomly selected a subset of these documents to extract the domain concepts, to build the initial DKB. By examining these, the domain expert identified 54 cardiology related disorders and 128 symptoms. Afterwards, the domain expert identified 323 relationships between these symptoms and disorders. Then we used the SNOMED knowledge base to identify the instance equivalency to populate the *owl:sameAs* relationship. The outcome is used to build the initial DKB as described in Section III-B1.

The proposed method was executed against all 1,535 EMR documents using the initial DKB. Table I summarizes the frequently occurring symptoms which were found to be unexplained by the disorders in the document. The second column consists of the number of times each symptom was not accounted for by a disorder in the same document (e.g., there are 236 EMR documents mention *edema*, but without an associated "cause" disorder).

The next step is to find the disorders that are good candidates to account for unexplained symptoms. This is done by analyzing the co-occurring disorders with the unexplained symptom in inconsistent documents, e.g., we analyze each

[9]IRB allows us to use the data in this study, but does not allow us to release it for public use yet

of the 236 documents where *edema* is found to be unexplained and aggregate the disorders that appear in each of the corresponding documents. According to our hypothesis, there should be at least one disorder which accounts for *edema* in each of these documents.

Table II summarizes the results of this step for *edema*. The first row shows that *hyperlipidemia* co-occurs as a disorder 134 times among the 236 times that *edema* is found to be unexplained. Similarly, the 2nd row indicates that *hypertension* co-occurs 131 times within the 236 cases. It is natural to think that either *hyperlipidemia* or *hypertension* may have relationship with *edema*. However, we then consulted the domain expert to determine the relationship between *edema* and all other candidate disorders. The feedback from the domain expert improved the fidelity of the DKB further.

We define the notion of *link-accuracy* in order to measure the effectiveness of the proposed method. By *link-accuracy* we measure how effective the method is in finding candidate disorders to unexplained symptoms.

$$\textit{link-accuracy} = \frac{\textit{number-of-correct-links}}{\textit{total-number-of-proposed-links}} * 100$$

The 100% *link-accuracy* means all proposed disorders for unexplained symptom were correct while 0% *link accuracy* indicates that the our method was unable to find at least one correct candidate disorder.

The experiment was conducted with the initial DKB and 1535 EMR documents. We found 40 symptoms that were unexplained and discovered at least one correct relationship for 39 of them. Figure 4 visualizes the *link-accuracy* for each symptom. As depicted in the figure, the links proposed for the six symptoms (*shortness of breath*, *fatigue*, *flushing*, *tiredness*, *chill*, and *nausea*) have 100% *link-accuracy*, while only 88% of the proposed links for *edema* were correct. However all links suggested for *urinary tract infection* were incorrect.

We modified the initial DKB according to the expert feedback. Figure 5 depicts the comparison of unexplained instance count for each symptom in initial DKB and modified DKB. As expected, the number of unexplained instance count was reduced for almost all symptoms. The number of unexplained instances for *edema* was 222 in initial DKB, and as a result of this experiment, we have been able to find 30 new *'intellego:inheres-in'* relationships of *edema* with candidate disorders. With these new relationships, the unexplained instance count for *edema* is reduced to 12 (Figure 5). Furthermore, the unexplained instance count for 20 symptoms reaches 0. However, the count remains the same in modified DKB for *urinary tract infection*.

The effectiveness of our method can also be evaluated using the effectiveness of the questions being generated. As mentioned above, initial DKB had 54 disorders and 128 symptoms, hence 6,912 (54*128) possible relationships can exist. The proposed method found 1,643 unexplained instances from dataset when ran with the initial DKB (note that one document can have multiple unexplained symptoms). Based on this unexplained instance set, our method generated only 625 questions (instead of 6,912) and 394 of them were answered positively by domain expert. Hence the modified DKB has 394 new relationships between *'intellego:entity'* and *'intellego:quality'*. As a result of this, the unexplained document count reduced to 417. This shows that added relationships helped to explain 75% of unexplained documents, justifying the effectiveness of the generated questions.

In order to prove the effectiveness of our method we executed same experiment on a new dataset consisting of 677 EMR documents. Running the experiment using this data set and the initial DKB resulted in 391 unexplained documents. But there were only 101 unexplained documents found when we ran it with modified DKB. This demonstrates the effectiveness of the knowledge acquisition methodology, since discovered knowledge in the first iteration has been able to cater for 75% of unexplained documents. More importantly, there were only 12 new questions generated by running the new dataset with modified DKB. This observation shows that the effort required from the domain expert significantly reduces in subsequent iterations as completeness and fidelity of DKB improves gradually.

VII. Discussion

The unexplained instance count for all symptoms which has *link-accuracy* value 100% in Figure 4 reach 0 in modified DKB (Figure 5). *link-accuracy* level 100% means, all links suggested for this symptoms were correct. Recall that these potential links were found from the disorders co-occur with this symptoms in inconsistent documents. Since each suggested disorder linked with the symptom in modified DKB, previously found inconsistencies are no longer exists.

In addition to the symptoms with 100% *link-accuracy* level in Figure 4, there are other symptoms which have no inconsistent instances with the modified DKB. This result is possible since it is not necessary for an unexplained symptom to be linked with all candidate disorders in order to cater for all inconsistent instances.

In order to explain the latter case, we present the following example, observed during our evaluation. The symptom *aphasia* is found to be unexplained in four documents and there are seven distinct disorders found in these documents. The seven disorders are *cerebrovascular accident*, *hypertension*, *peripheral vascular disease*, *hyperlipidemia*, *atrial fibrillation*, *hypotension* and *coronary artery disease*. Our method suggests seven potential *'intellego:inheres-in'* relationships for *aphasia* (one for each disorder). But according to the domain expert, only five disorders (*cerebrovascular accident*, *hypertension*, *atrial fibrillation*, *hypotension*, and *coronary artery disease*) can be linked to *aphasia* (*link-accuracy* 71%). Furthermore, it is found that *cerebrovascular accident* present in all four documents. So adding the relationship between *aphasia* and *cerebrovascular accident* will rectify all inconsistencies with respect to *aphasia*, since *aphasia* now gets included in *'intellego:coverage'* in all four documents.

The number of unexplained instance count remains the same for *urinary tract infection*. The *link-accuracy* for *urinary tract*

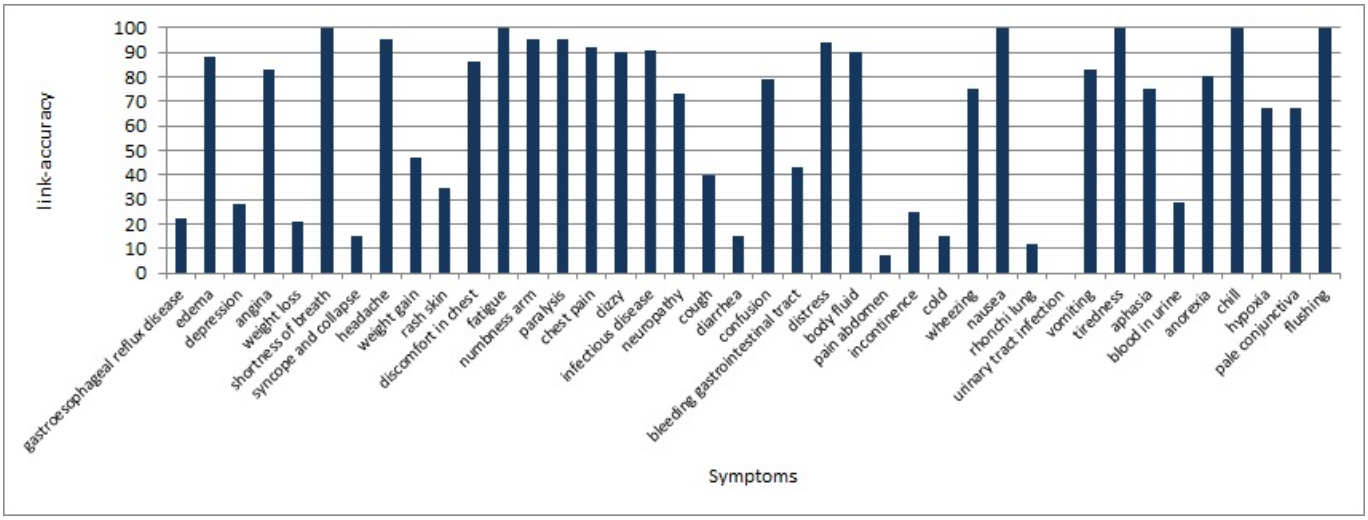

Fig. 4. *link-accuracy* of Proposed Links

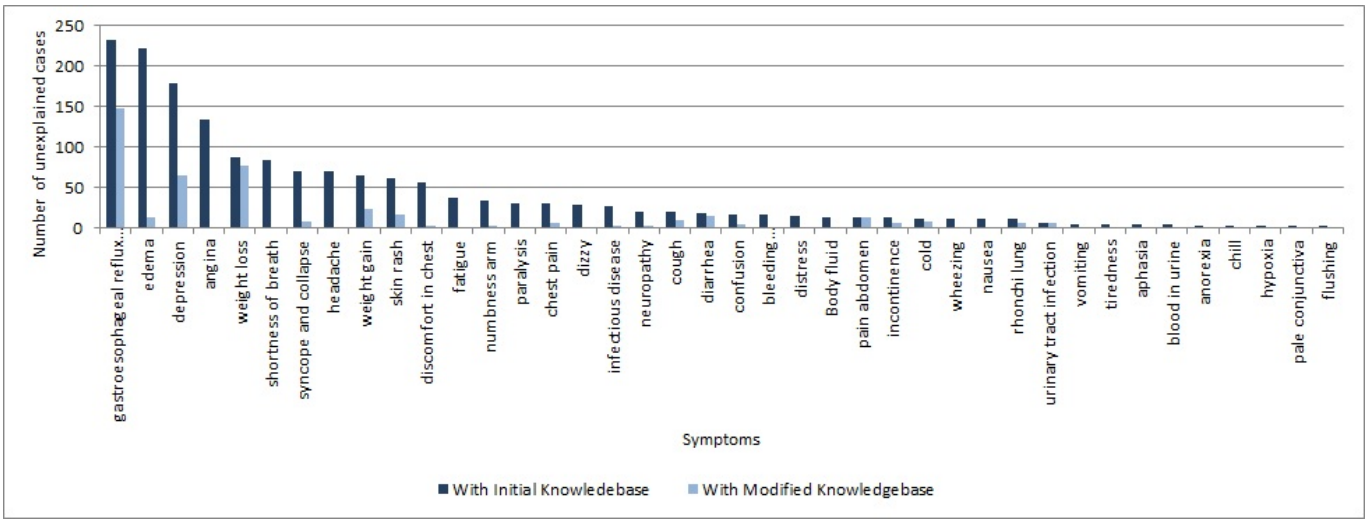

Fig. 5. Comparison of Knowledge Bases

infection is 0%, which means that all suggested disorders for *urinary tract infection* were wrong. Hence no new knowledge about *urinary tract infection* is added to the modified DKB, and inconsistencies were not resolved.

VIII. Related Work

Ontology enrichment is a subarea of ontology learning. Automated ontology development and learning has two sub tasks: 1) identify the domain concepts to be added or removed and 2) identify relationships between the concepts to be added or removed. The former task addresses questions such as: When is it necessary to add/remove new concept/existing concept to/from the ontology? What are the concepts to be added/removed to/from the ontology? Where should the new concept be placed in the ontology? Multiple techniques have been proposed to address these questions[11][12][13], but just adding concepts may not necessarily improve the ontology, unless necessary relationships are populated. On this paper, we concentrated on the task of identifying the domain relationships between the concepts. Specifically, identifying missing relationships in existing DKB.

Relationships in a DKB can be categorized as taxonomic and non-taxonomic where subclass/superclass relationship is considered as taxonomic and domain specific relationships (e:g *chest pain* is a symptom of *hypertension*) are considered as non-taxonomic. Different methods have been proposed to discover taxonomic relationships[16][17][18]. Here we were interested in discovering domain specific relationships. The techniques in the literature to find domain relationships can be categorized as rule based techniques, machine learning techniques and NLP techniques.

A comprehensive ontology evaluation framework proposed in[15] uses Scarlet[14] to find the relationships among the concepts. Scarlet[14] uses multiple rules to derive taxonomic relationships as well as domain specific relationships. While these rules are capable of deriving taxonomic relationships by integrating multiple ontologies, it cannot derive domain specific relationships. Scarlet can only find such relationship only if some other existing ontology express such knowledge.

Our approach differs from this since it identifies and remedies missing domain relationships.

People use freely available knowledge(peer reviewed publications, books, articles etc) in the form of unstructured documents to glean the domain ontologies. The most popular techniques to learn the ontologies from the text corpus are based on the NLP and Machine Learning (ML)[19][20][21][22][23]. These methods rely on named entity identification methods[21], pre-defined template patterns[21], lexical syntactic properties of the free text (like frequency of words appearing together[21][20], position of the words[19], and frequency of verbs appearing with the lexical terms[20][22]). But all these ML methods require of training datasets, while NLP method suffers from the text parsing errors.

Our method differs from the above methods, since it learns the relationships in DKB by validating it against the real world data sources. But it complements the existing corpus based techniques, since it can guide these techniques by providing specifics about what information to look for in the text (e.g., look for the existence of relationship between *edema* and *hypertension*). This will help existing algorithms to have better focus on domain knowledge exploration.

IX. Conclusion

The data-driven approach is a convenient way to fill the gaps in incomplete domain knowledge and it significantly reduces the effort required by the domain expert. The overall *link-accuracy* of the formulated questions is 63% (i.e., 63 out of 100 suggested relationships were correct), which shows that the proposed method is able to spot the missing knowledge with good accuracy. It frees the domain expert to focus only on relevant portion of the domain knowledge and improves the understanding between ontologists and domain experts. This method enables us to aggregate knowledge from various sources created by different domain experts, thereby reducing subjectivity and creating a comprehensive DKB.

References

[1] A. Sheth, S. Agrawal, J. Lathem, N. Oldham, H Wingate, P. Yadav, K. Gallegher. "Active Semantic Electronic Medical Records,in The Semantic Web: Real World Applications from Industry." Springer,pp. 123-140, 2008.

[2] C. Ogbuji, E. Blackstone, C. Pierce. "A Semantic Web Content Repository for Clinical Research". Internet:http://www.w3.org/2001/sw/sweo/public/UseCases/ClevelandClinic/, Oct. 2007 [May. 25, 2012].

[3] A. Ryan, P. Eklund. "A framework for semantic interoperability in healthcare: a service oriented architecture based on health informatics standards." Studies In Health Technology And Informatics,vol. 136, pp. 759-764, 2008.

[4] S. Schulz, R. Cornet. "SNOMED CT's Ontological Commitment." In Proc. ICBO: International Conference on Biomedical Ontology; National Center for Ontological Research, 2009, pp. 5558.

[5] World Health Organization, "International Classification of Diseases." Internet:http://www.who.int/classifications/icd/en/, 2004 [May. 25, 2012].

[6] A. Maedche, S. Staab. "Ontology learning for the Semantic Web." Intelligent Systems, IEEE, vol. 16(2), pp. 72-79, 2001.

[7] A. Sheth, I. Arpinar, V. Kashyap. "Relationships at the Heart of Semantic Web: Modeling, Discovering, and Exploiting Complex Semantic Relationships" Enhancing the Power of the Internet (Studies in Fuzziness and Soft Computing), vol. 139, pp. 63-94. 2004.

[8] C. Henson, K. Thirunarayan, A. Sheth. "An Ontological Approach to Focusing Attention and Enhancing Machine Perception on the Web." Applied Ontology, vol. 6(4), pp. 345-376, 2011.

[9] C. Friedman, P. Alderson, J. Austin, J. Cimino, S. Johnson. "A general natural-language text processor for clinical radiology." Journal of the American Medical Informatics Association, vol. 1(2), pp. 161174, 1994.

[10] S. Bowman. "Coordination of SNOMED-CT and ICD-10: Getting the Most out of Electronic Health Record Systems." Journal of the American Health Information Management Association, vol. 76(7), pp. 60-1, 2005.

[11] I. Novalija, D. Mladenic. "Ontology extension towards analysis of business news." Informatica (Slovenia), vol. 34(4), pp. 517-522, 2010.

[12] H. Packer, N. Gibbins, N. Jennings. "An on-line algorithm for semantic forgetting." in Proc IJCAI, 2011, pp. 2704-2709.

[13] F. Wu, D. Weld. "Automatically refining the wikipedia infobox ontology." In Proc WWW, 2008, pp 635-644.

[14] M. Sabou, M. DAquin, E. Motta, "Exploring the Semantic Web as Background Knowledge for Ontology Matching." Journal on Data Semantics XI, vol. 5383, pp. 156-190, 2008

[15] F. Zablith, "Evolva: A comprehensive approach to ontology evolution." In Proc 6th European Semantic Web Conference (ESWC) PhD Symposium, 2009, pp 944-948.

[16] M. Hearst, "Automatic Acquisition of Hyponyms from Large Text Corpora." In Proc Fourteenth International Conference on Computational Linguistics, 1992.

[17] S. Caraballo. "Automatic construction of a hypernym-labeled noun hierarchy from text." In Proc of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, pp. 120-126, 1999.

[18] S. Cederberg, D. Widdows. "Using LSA and noun coordination information to improve the precision and recall of hyponymy extraction." In Proc CoNLL, pp. 111-118, 2003.

[19] D. Faure, C. Nedellec, "A corpus-based conceptual clustering method for verb frames and ontology acquisition." In Proc LREC workshop on adapting lexical and corpus resources to sublanguages and applications, pp. 5-12, 1998.

[20] M. Kavalec, A. Maedche, V. Svatek, "Discovery of Lexical Entries for Non-taxonomic Relations in Ontology Learning." SOFSEM Theory and Practice of Computer Science, vol. 2932, pp. 17-33, 2004

[21] M. Ciaramita, A. Gangemi, E. Ratsch, S. Jasmin, R. Isabel. "Unsupervised Learning of Semantic Relations between Concepts of Molecular Biology Ontology." In proc International Joint Conference on Articial Intelligence, pp. 659-664, 2005.

[22] A. Schutz, P. Buitelaar, "RelExt: A Tool for Relation Extraction from Text in Ontology Extension." In Proc 4th International Semantic Web Conference, pp. 593-606, 2005.

[23] B. Rosario, M. Hearst. "Classifying semantic relations in bioscience text." In Proc. 42nd Annual Meeting of the Association for Computational Linguistics, 2004.

[24] P. Desai, C. Henson, P. Anatharam, A. Sheth. "Demonstration: SECURE – Semantics Empowered resCUe Environment." In Proc 4th Intl. Workshop on Semantic Sensor Networks, pp. 110-113, co-located with the 10th International Semantic Web Conference (ISWC 2011), 23-27 October 2011, Bonn, Germany, 2011.

[25] L. Lefort, C. Henson, K. Taylor, P. Barnaghi, M. Compton, O. Corcho, R. Garcia-Castro, J. Graybeal, A. Herzog, K. Janowicz, H. Neuhaus, A. Nikolov, and K. Page. "Semantic Sensor Network XG Final Report." W3C Incubator Group Report, June 28, 2011. www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628.

[26] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, K. Taylor. "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group." Journal of Web Semantics: Science, Services and Agents on the World Wide Web, North America, O, May 2012.

[27] http://www.loa.istc.cnr.it/ontologies/DUL.owl