XOntoRank: Ontology-Aware Search of Electronic Medical Records

Fernando Farfán [†1], Vagelis Hristidis [†2], Anand Ranganathan [‡3], Michael Weiner, MD [#4]

[†] *School of Computing and Information Sciences, Florida International University*
Miami, Florida
U. S. A.
[1] ffarfan@cis.fiu.edu
[2] vagelis@cis.fiu.edu

[‡] *IBM T.J. Watson Research*
Yorktown Heights, New York
U. S. A.
[3] arangana@us.ibm.com

[#] *Regenstrief Institute, School of Medicine, and Center for Aging Research, Indiana University*
Indianapolis, Indiana
U. S. A.
[4] mw@cogit.net

***Abstract*— As the use of Electronic Medical Records (EMRs) becomes more widespread, so does the need for effective information discovery within them. Recently proposed EMR standards are XML-based. A key characteristic in these standards is the frequent use of ontological references, i.e., ontological concept codes appear as XML elements and are used to associate portions of the EMR document with concepts defined in a domain ontology.**

A rich corpus of work addresses searching XML documents. Unfortunately, these works do not make use of ontological references to enhance search. In this paper we present the XOntoRank system which addresses the problem of ontology-aware keyword search of XML documents with a particular focus on EMR XML documents. Our current prototypes and experiments use the Health Level Seven (HL7) Clinical Document Architecture (CDA) Release 2.0 standard of EMR representation and the Systematized Nomenclature of Human and Veterinary Medicine (SNOMED) ontology, although the presented techniques and results are applicable to any EMR hierarchical format and any ontology that defines concepts and relationships.

I. Introduction

The National Health Information Network (NHIN) and its data-sharing building blocks, RHIOs (Regional Health Information Organizations), are encouraging the widespread adoption of Electronic Medical Records (EMR) for all hospitals within five years. A key component of this effort is the standardization of EMR. To date, there has been little or no effort to define methods or approaches to search such documents effectively.

One of the most promising standards for EMR manipulation and exchange is Health Level 7's [1] Clinical Document Architecture (CDA) [2], which leverages a semi-structured (XML) format, and ontologies to specify the structure and semantics of EMRs for the purpose of Electronic Data Interchange (EDI).

In this paper we present the XOntoRank system, which addresses the problem of facilitating ontology-aware information discovery within a corpus of XML-based EMR documents. By information discovery [3], [4] we mean the extraction of relevant pieces of data from a database given a user query. Information discovery can be viewed as an extension of traditional Information Retrieval (IR), which ranks the relevance of unstructured documents given a keyword query. Hence, given a question (query) and a set of EMRs, we need to find the entities (typically subtrees) that match the query, and rank them according to their "goodness" with respect to the query. The success of Web search engines has shown that keyword queries are a useful and intuitive approach to information discovery. Therefore, we focus on keyword queries in this paper.

A large corpus of work (e.g. [5], [6], [7], [8]) addresses keyword search of XML documents, where the query keywords are matched to XML nodes and a minimal tree containing these nodes is returned. A variety of ranking techniques are used, ranging from the size of the result-trees to adaptations of Information Retrieval (IR) scoring. Investigators have explored ontologies (e.g. [9], [10]) for XML querying; we compare them to our work in Section VIII.

For example, consider the query *"Bronchial Structure Theophylline"* and a CDA document such as the one in Figure 1, which is explained in detail in Section II. The phrase *"Bronchial Structure"* does not appear in this document. Hence, most traditional XML-based keyword search systems will not return any results. However, this document contains an ontological reference to an *"Asthma"* concept defined in SNOMED (in Line 39, Figure 1). The SNOMED ontology further defines a *"finding-site-of"* relationship between *"Asthma"* and *"Bronchial Structure"* (as shown in Figure 2). Hence, based on the definitions in the ontology, a result tree connecting the *"Asthma"* node of Line 39 and the *"Theophylline"* node of Line 50 can be created as output.

The use of ontological definitions allows us to perform semantic search on the XML documents. We no longer require an exact match between keywords in the query and in the document, but we can make use of the domain ontology to infer a semantic relationship between keywords in the query and terms in the document. This allows returning more results than would otherwise be returned with an exact-match requirement. This paper makes the following contributions:

1) Introduce the problem of ontology-aware keyword search among XML-based EMR documents, which can be extended to general XML documents.
2) Define the semantics of what constitutes a result and how the results are ranked for the problem of ontology-aware keyword search within the EMR. We leverage previous work related to searching XML data.
3) Develop a set of techniques to compute the degree of association between ontological concepts that take into account both taxonomic *is-a* links as well as more general semantic relationships between concepts. This is a core component of our ranking framework.
4) Create and experimentally evaluate algorithms to answer efficiently ontology-aware keyword queries in EMRs. These algorithms were tested with real EMR data acquired from a local hospital.

We note that our study does not address the important privacy issues involved in accessing patient information, as required by HIPAA [11]. The policies and principles described in [12] could work as a starting point in achieving Hippocratic information discovery.

The rest of this paper is organized as follows: Section II presents background knowledge. Section III defines the problem and its semantics. Alternative approaches to compute the semantic relevance of an ontological concept to a keyword are presented in Section IV. In Section V we present the architecture. Section VI presents the algorithms to implement the approaches of Section IV. Section VII presents the experimental evaluation of XOntoRank. Section VIII presents previous work and we conclude in Section IX.

II. Background

HL7: Health Level Seven (HL7) [1] is a not-for-profit organization that provides standards for interoperability in the healthcare industry, mainly focused on clinical and administrative data. HL7 is an American National Standards Institute (ANSI) -accredited Standards Developing Organization (SDO) that includes providers, vendors, payers, consultants, government groups and other entities interested in developing clinical and administrative standards for healthcare.

HL7 standards specify a series of flexible standards to facilitate the communication between heterogeneous systems and vendors, allowing information to be shared and processed in a uniform and consistent manner. During the years, HL7 has developed Conceptual Standards (i.e. HL7 RIM),

```
1 <? xml version="1.0" ?>
2 <ClinicalDocument xmlns="urn:hl7-org:v3" xmlns:voc="urn:hl7-org:v3/voc"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:hl7-org:v3 CDA.ReleaseTwo.Committee.2004.xsd"
   templateId="2.16.840.1.113883.3.27.1776">
3   <id extension="c266" root="2.16.840.1.113883.3.933" />
5   <author>
6     <time value="20040407" />
7     <assignedAuthor>
8       <id extension="KP00017" root="2.16.840.1.113883.3.933" />
9       <assignedPerson>
10        <name>
11          <given>Juan</given>
12          <family>Woodblack</family>
13          <suffix>MD</suffix>
14  </name></assignedPerson></assignedAuthor></author>
15  <recordTarget>
16    <patientRole>
17      <id extension="49912" root="2.16.840.1.113883.3.933" />
18      <patientPatient>
19        <name>
20          <given>FirstName</given>
21            <family>LastName</family>
22            <suffix>Jr.</suffix>
23        </name>
24        <administrativeGenderCode code="M" codeSystem="2.16.840.1.5.1"/>
25        <birthTime value="20020924"/>
26      </patientPatient>
27      <providerOrganization>
28        <id extension="M345" root="2.16.840.1.113883.3.933"/>
29  </providerOrganization></patientRole></recordTarget>
30  <component>
31    <StructuredBody>
32      <component>
33        <section>
34          <code code="10160-0" codeSystem="2.16.840.1.113883.6.1"
             codeSystemName="LOINC"/>
35          <title>Medications</title>
36          <entry>
37            <Observation>
38              <code code="84100007" codeSystem="2.16.840.1.113883.6.96"
                 codeSystemName="SNOMED CT" displayName="Medications"/>
39              <value xsi:type="CD" code="195967001" codeSystem=
                 "2.16.840.1.113883.6.96" codeSystemName="SNOMED CT"
                 displayName="Asthma">
40                <originalText><reference value="m1"/></originalText>
41          </value></Observation></entry>
42          <entry>
43            <Observation>
44              <code code="84100007" codeSystem="2.16.840.1.113883.6.96"
                 codeSystemName="SNOMED CT" displayName="Medications"/>
45              <value xsi:type="CD" code="32398004" codeSystem=
                 "2.16.840.1.113883.6.96" codeSystemName="SNOMED CT"
                 displayName="Bronchitis">
46                <value xsi:type="CD" code="91143003" codeSystem=
                   "2.16.840.1.113883.6.96" codeSystemName="SNOMED CT"
                   displayName="Albuterol" />
47          </value></Observation></entry>
48          <entry>
49            <SubstanceAdministration>
50              <text><content ID="m1">Theophylline</content>20 mg every
                 other day, alternating with 18 mg every other day. Stop
                 if temperature is above 103F.</text>
51              <consumable>
52                <manufacturedProduct>
53                  <manufacturedLabeledDrug>
54                   <code code="66493003" codeSystem="2.16.840.1.113883.6.96"
                      codeSystemName="SNOMED CT" displayName="Theophylline"/>
55              </manufacturedLabeledDrug></manufacturedProduct></consumable>
56          </SubstanceAdministration></entry>
57      </section></component>
58      <component>
59        <section>
60          <code code="11384-5" codeSystem="2.16.840.1.113883.6.1"
             codeSystemName="LOINC"/>
61          <title>Physical Examination</title>
62          <component>
63            <section>
64              <code code="8716-3" codeSystem="2.16.840.1.113883.6.1"
                 codeSystemName="LOINC"/>
65              <title>Vital Signs</title>
66              <text>
67                <table>
68                  <tr>
69                    <th>Temperature</th>
70                    <td>36.9 C (98.5 F)</td>
71                  </tr>
72                  <tr>
73                    <th>Pulse</th>
74                    <td>86 / minute</td>
75              </tr></table></text>
76              <entry>
77                <Observation>
78                  <code code="50373000" codeSystem="2.16.840.1.113883.6.96"
                     codeSystemName="SNOMED CT" displayName="Body height"/>
79                  <effectiveTime value="200404071430"/>
80                  <value xsi:type="PQ" value="1.77" unit="m" />
81      </Observation></entry></section></component></section></component>
82 </StructuredBody></component></ClinicalDocument>
```

Fig. 1. HL7 CDA Sample Document

Document Standards (i.e. HL7 CDA), Application Standards (i.e. HL7 CCOW) and Messaging Standards (i.e. HL7 v2.x and v3.0). These standards define the language, structure and data types that participate in the integration of heterogeneous systems [13].

SNOMED CT: The International Systematized Nomenclature of Human and Veterinary Medicine (SNOMED) [14] has matured into a comprehensive set of over 150 000 records in twelve different chapters or axes. *SNOMED Clinical Terms (SNOMED CT)* is a universal health care terminology and infrastructure. The SNOMED CT (simply termed "SNOMED" in the rest of the paper) structure is concept-based; each concept represents a unit of knowledge, having one or more natural language terms that can be used to describe the concept. Every concept has relationships with other concepts, including hierarchical *"is-a"* relationships as well as other relationships that describe clinical attributes.

Figure 2 shows a sub graph of the SNOMED ontology graph. At the moment, SNOMED contains more than 325 000 concepts, with 800 000 terms in English, 350 000 in Spanish and 150 000 in German. Also, there are 1 200 000 relationships connecting these terms and concepts. SNOMED terms are referenced in CDA documents by their numeric codes.

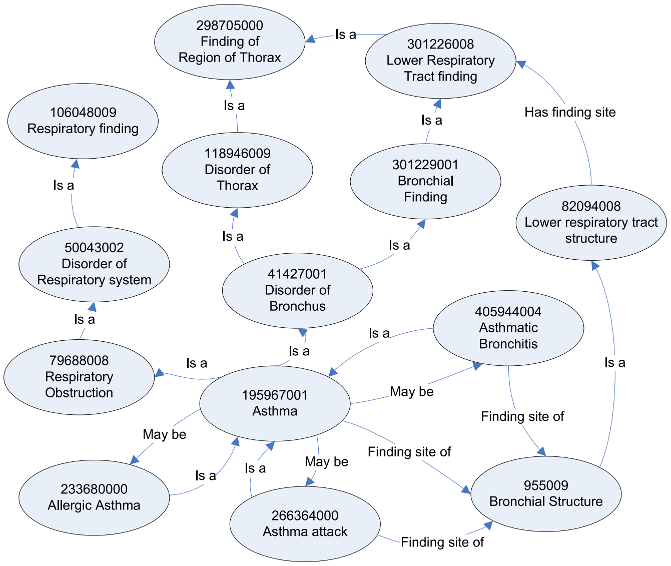


Fig. 2. Subgraph of SNOMED Ontology.

Clinical Document Architecture (CDA): CDA is an XML-based document markup standard that specifies the structure and semantics of clinical documents, such as discharge summaries and progress notes, for the purpose of exchange. It is an ANSI-approved HL7 standard, intended to become the de facto standard for electronic medical records. Figure 3 [15] shows a fragment of the CDA's Object Model that represents the semantic constructs of the Reference Information Model (RIM) [16], depicting the connection from a document section to a portion of the CDA clinical statement model with nested CDA entries. The colors in Figure 3 identify these classes with the core classes of RIM (Red for Act specializations, blue for Participations, green for Entities, yellow for Roles and pink for Relationships).

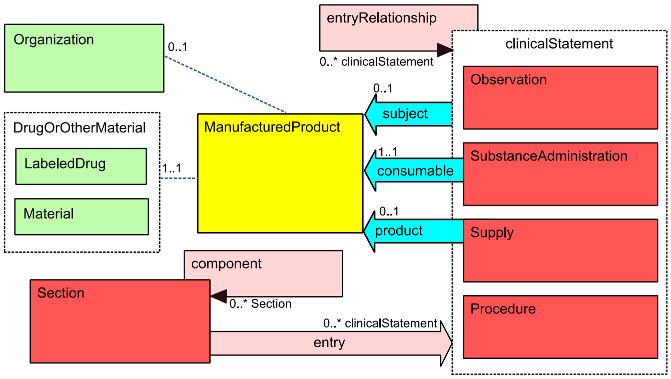


Fig. 3. Fragment of CDA Object Model.

Figure 1 depicts a sample CDA document, D_1, which is wrapped by the *"ClinicalDocument"* element, as it appears in Line 2. The CDA header (Lines 3-29) identifies and classifies the document, and provides information about the participants (patient and providers). The CDA body (Lines 30-82), which is wrapped by the *"StructuredBody"* element, is the core of the document. It can be either an unstructured segment or an XML fragment. Every information unit is allocated as a section under a component, following the class diagram of Figure 3. A section can represent any of the entities under the clinicalStatement superclass, and hence we find several sections such as Observations (Lines 37 and 43) and SubstanceAdministration (Line 49).

We focus on structured CDA documents, which provide a better opportunity for high-quality information discovery. Traditional Information Retrieval (IR) approaches [17], [18] can be applied to the unstructured scenario. Note how certain XML elements in D_1 reference concepts of SNOMED. For example, Line 39 in Figure 1 references the *SNOMED* (as system *code="2.16.840.1.113883.6.96"*) concept with *code="84100007"*.

III. Problem Definition and Semantics

XML data: Our data collection is a set $D = \{T_1, \ldots, T_n\}$ of XML documents. We view an XML document as a labeled tree T. Each node $v \in T$ has:

a. A textual description $v.text$, which is the concatenation of its tag name, attribute names and values, and text content, and
b. An optional ontological reference $v.onto$, which typically consists of an integer code $v.onto.system$ for the referenced ontological system (e.g., SNOMED) and an integer code $v.onto.concept$ for the specific concept (e.g., *"Asthma"*).

Nodes with ontological reference are called *code nodes*. The set of ontological systems referenced by nodes in D is called ontological systems collection $O = \{O_1, \ldots, O_s\}$.

For instance, the node of Line 39 in Figure 1 has *v.text=“value xsi:type=“CD” code=“195967001” codeSystem=“2.16.840.1.113883.6.96” codeSystemName=“SNOMED CT” displayName=“Asthma”*, $v.onto.system$ = 2.16.840.1.113883.6.96, and $v.onto.concept$ = 195967001. Note that some attribute values like code strings are not included in $v.text$ since these are unlikely to be used in a query keyword or in ontology reference words from. An expert specifies the attributes that should not be included in the textual description.

In the algorithms presented in this paper we ignore ID-IDREF edges as well as inter-document references, since we build on tree search algorithms. However, the techniques we use to incorporate ontological information are straightforwardly applicable to graph search algorithms as well (i. e. when ID-IDREF edges are considered [8]).

Keyword Search: A keyword query Q is a set $\{w_1, \ldots, w_m\}$ of keywords. Previous work, which ignores ontological references, has generally defined the results as subtrees of the XML documents that contain all query keywords (see Section VIII for an overview of related work). In this work we adopt the result semantics of XRANK [6], which is a popular representative of this class of works, and extend it to account for ontological references. Any other system could be extended in a similar way. The key extension is that instead of requiring keywords to be contained in the nodes of the result subtree, we require that the result subtree has nodes *associated* with every query keyword. Let $NS(v, w)$ (Node Score), whose computation is explained later, be the association degree of a node v with respect to a keyword w which is directly contained in v or is associated to v through an ontology. The result of Q for a document $T \in D$ is defined as follows. Let

$$\begin{aligned} R_0 = \{v | v \in T \wedge \\ \forall w \in Q \exists u \in (Desc(v) \cup v)(NS(u, w) > 0)\} \end{aligned}$$

be the set of elements that are, themselves or through their descendant nodes, associated to all query keywords of Q. $Desc(v)$ is the set of descendants of v in T.

The result of the query Q is defined as:

$$\begin{aligned} Result(Q) = \{v | \forall w \in Q, \exists u \in (Desc(v) \cup v) \\ (NS(u, w) > 0 \wedge \neg \exists t \in Desc(v)(t \in R_0))\} \end{aligned} \quad (1)$$

Intuitively, a result v is an element that has sub-elements associated with each of the query keywords, but no sub-element is associated with all keywords. Note that $Result(Q)$ is a subset of R_0. The latter condition ensures we do not generate non-specific results.

For instance, if query *q=[“asthma”, “medication”]* is executed on the document of Figure 1, we get the XML fragment depicted in Figure 4, being the most specific sub-element in the CDA document that contains both terms in the query. Note that in the case, both terms are actually contained in the XML fragment. In general, though, the terms need not be in the fragment, but may be associated with nodes in the fragment through the ontology.

```
<Observation>
  <code code="84100007" codeSystem="2.16.840.1.113883.6.96"
      codeSystemName="SNOMED CT" displayName="Medications"/>
  <value xsi:type="CD" code="195967001" codeSystem=
      "2.16.840.1.113883.6.96" codeSystemName="SNOMED CT"
      displayName="Asthma">
    <originalText><reference value="m1"/></originalText>
</value></Observation>
```

Fig. 4. XML Fragment representing the answer to query *q=[“asthma”, “medications”]*.

Score of results: As mentioned above, $NS(v, w)$ is non-zero if a node v directly contains w or is associated to w through an ontological system. This score is propagated to other nodes of the XML document as follows. The *propagated score* $PS(v, w, u)$ of an element v with respect to keyword w, assuming that a sub-element u of v has $NS(u, w) > 0$, is

$$PS(v, w, u) = decay^l \cdot NS(u, w) \quad (2)$$

where $l = distance(v, u)$ is the number of containment edges between v and u. $Decay$ is set between 0 and 1 to account for the specificity of a result.

Given that multiple sub-elements of v may be associated with w, we use the following formula for the overall score of v given w

$$Score(v, w) = max_{u \in Desc(v) \cup v} PS(v, w, u) \quad (3)$$

Other monotonic aggregation functions are also possible. The score of a result element v for Q is

$$Score(v, Q) = \sum_{w \in Q} Score(v, w) \quad (4)$$

Again other monotonic aggregation functions are possible.

Association degree of node to keyword: The association degree $NS(v, w)$ of node $v \in T$, $T \in D$ with respect to a keyword w, given documents collection D and an ontological systems collection O is a combination of its IR score with respect to w and its ontological association to w.

$$NS(v, w) = max \begin{pmatrix} IRS(v.text, w), \\ OS_{v.onto.system}(CN(v.onto), w) \end{pmatrix} \quad (5)$$

where $IRS(d, w)$ is the IR score of a document d given keyword w within the collection D. D is an implicit input to $IRS(\cdot)$ since popular IR functions [17], [19], [20] use the document frequency (*df*) which is computed over D. We view each XML element as a document to apply the IR function. In our experiments we use the BM25 [19] function.

$OS_{v.onto.system}(u, w)$ is the association degree (*OntoScore*) of a node (concept) $u \in O_i$, where O_i is specified by $v.onto.system$, to keyword w, and is computed by exploiting the relationships in O_i, as explained in detail in Section IV.

$CN(v.onto)$ returns the concept node with code $v.onto.concept$ in the ontological system specified by $v.onto.system$. For instance, consider the document of Figure 1 and the ontological system of Figure 2. $CN(v.onto)$ for the code element v of Line 39 in Figure 1 will return the concept node *"Asthma"* identified with the code 195967001 in Figure 2. $IRS(\cdot)$ and $OS(\cdot)$ are normalized to $[0, 1]$.

The intuition of (5) is that a node v may be associated with a keyword w either through its textual description $v.text$ or through its ontological reference $v.onto$. We then pick the strongest one. The $OS(\cdot)$ term of a non-code node is 0. Again, alternative monotonic aggregation functions are possible.

For instance, for the keyword w=*"Asthma"* assuming node v of Line 39 in Figure 1 has $IRS(v.text, w) = 0.3$ and its related SNOMED node u has $OS_{SNOMED}(u, w) = 0.5$, its $NS(v, w)$ would be 0.5.

IV. Semantic Relevance of Ontological Concepts to Keywords

A key component of XOntoRank is the derivation of semantic relevance of a concept v in the ontology to a query keyword w. Since nodes in an XML document may refer to concepts in the ontology, this derivation essentially quantifies the semantic relevance of an XML element to a query keyword based on terminological definitions in the ontology.

The Semantic Web community has developed various mechanisms to determine semantic similarity of concepts in an ontology (see Section VIII for a description of Related Work). However, most existing measures do not use relationship information between concepts in a general manner. The main advantage of ontologies like SNOMED over simpler taxonomies is that they describe various kinds of relationships between concepts, which can be used to calculate relevance measures.

We view the ontology as a graph, where the nodes in the graph represent concepts, and edges represent relationships between concepts. Our approach for calculating the semantic relevance of a concept to a query keyword is inspired by the idea of authority flow. Initially, each concept in the ontology is granted a certain authority based on how strongly it is related to w, as measured by its IR score. Authority then flows from these concepts to other concepts in the ontology based on certain rules. Note that the authority flow occurs in a recursive fashion and hence, it can affect descendants and not only direct children of the involved elements.

In this section, we examine various strategies for directing the flow of authority, based on different views of the ontology. For simplicity of presentation we consider a single ontology O_0 and omit the O_0 subscript at $OS()$. We use the overloaded function $OS(v, w, x)$ to represent the relevance of concept v to keyword w due to the occurrence of w in another node x in the ontology. It is:

$$OS(v, w) = max_{x \in O_0}(OS(v, w, x)) \qquad (6)$$

Other monotonic aggregation functions are possible.

A. View Ontology as Undirected, Unlabeled Graph

This strategy treats the ontology as an undirected graph, with no distinction among the different kinds of relationships between concepts. Based on this view, we define $OS(v, w, x)$ as:

$$OS(v, w, x) = IRS(x, w) \cdot decay^{l} \qquad (7)$$

where $l = distance(v, x)$ and $0 \leq decay \leq 1$.

B. View Ontology as Taxonomy

This strategy only considers the taxonomic portion of the ontology, i.e. we only consider *is-a* links between concepts for calculating $OntoScore$. The *is-a* links form a Directed Acyclic Graph *(DAG)*, since cycles are not permitted based on subclass relationships. $OS(v, w, x)$ is computed recursively using (6) and the following two cases:

i **x is a superclass of v,** i.e., there is a path from v to x in the DAG formed by the *is-a* links. In this case,

$$OS(v, w, x) = IRS(x, w)$$

The intuition behind this definition is that since x is a superclass of v, any query for x is completely and logically satisfied by v. For example, let v be *"Asthma"*, w be *"Bronchus"* and x be *"Disorder of Bronchus"* (*"DOB"*) in the ontology fragment of Figure 2. It is *OS("Asthma", "Bronchus", "DOB") = IRS("DOB", "Bronchus")*. An extreme case of this rule is when x is the same as v. In this case, $OS(v, w, v) = IRS(v, w)$.

ii x **is a direct subclass of** v**,** i.e. there is an *is-a* link from x to v. In this case,

$$OS(v, w, x) = IRS(x, w) \cdot (1/n)$$

where n is the number of subclasses of v. The intuition behind this definition is that since x is a subclass of v, any query for x is partially satisfied by v. Our heuristic for calculating the extent of the partial satisfaction is based on the number of subclasses of v, similarly to the authority flow distribution in [21]. For example, let v be *"Disorder of Bronchus"*, w be *"Asthma"* and x be *"Asthma"* in Figure 2. In the actual ontology, the concept *"Asthma"* has 26 direct subclasses. Hence, in this case, *OS("Disorder of Bronchus", "Asthma", "Asthma") = IRS("Asthma", "Asthma") *(1/26)*.

C. Including the Relationships between Concepts

To handle different kinds of relationships, we interpret concepts and relationships in SNOMED using description logics [22]. Many biomedical ontologies, including SNOMED, belong to a category of Descriptions Logics called $\mathcal{EL}^{+}$ [23]. Concepts in this logic are defined as follows:

$$C ::= A|T|C \sqcap D|\exists r.C \qquad (8)$$

where A ranges over atomic concept names
T is the top concept
r ranges over relationship names
C, D are concept names
$\sqcap$ is the concept intersection operator

The $\exists r.C$ construct is an existential quantification operator that declares the existence of a relationship (or role) to a concept C. We can also view $\exists r.C$ as a concept where every instance of the concept is related by role r to an instance of a concept C. We call such a concept an *existential role* *restriction*, since it describes a constraint or restriction on the values of a relationship. (8) describes the different ways in which a concept can be defined in the $\mathcal{EL}^+$ logic. The $\mathcal{EL}^+$ logic also defines subclass (or concept inclusion) relationships between concepts as $C \sqsubseteq D$.

Some examples of $\mathcal{EL}^+$ expressions from Figure 2 are:

$$\begin{aligned} \text{Disorder of Thorax} &\sqsubseteq \text{Finding of Region of Thorax} \\ \text{Asthma Attack} &\sqsubseteq \text{Asthma} \\ &\sqcap \exists\text{Finding-site-of.Bronchial Structure} \end{aligned}$$

Consider the last statement, which says that *"Asthma Attack"* is a concept that is a subclass of Asthma and that has a *finding-site-of* relationship to the *"Bronchial Structure"* concept. In other words, any instance of *"Asthma Attack"* (e.g. the *"Asthma Attack suffered by"* a specific patient) is also an instance of *"Asthma"* and is found in some instance of *"Bronchial Structure"*.

This description logic view allows us to describe every concept as a subclass of a set of atomic concepts or existential role restrictions. Hence, we can reduce a graph with different kinds of relationships into one that has only subclass or *is-a* relationships.

For example, consider an ontology graph fragment depicted in Figure 5. A description logic view of this ontology would appear as shown in Figure 6. The dotted links between concepts represent *is-a* links, meant to indicate the relationship between a concept X and a $\exists r.X$ for any role r.

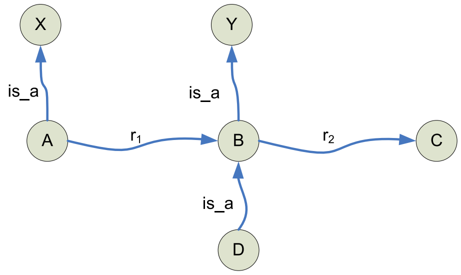


Fig. 5. Sample Ontology Fragment.

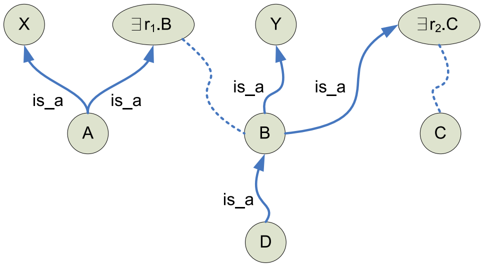


Fig. 6. Ontology's Description Logic View.

We now calculate $OS(v, w, x)$ in this logically transformed ontology graph using an extension of the strategy of Section IV-B. In particular, if there is a "dotted link" between x and v, i.e. one of x or v is of the form C, and the other is of the form $\exists r.C$, then,

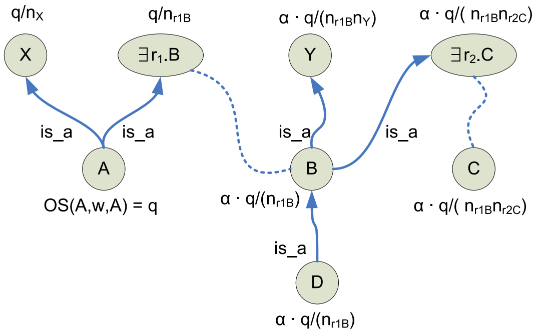


Fig. 7. OntoScore Propagation. n_i is the number of subclasses of node i.

$$OS(v, x, w) = OS(x, w) \cdot \alpha \tag{9}$$

Here, α represents the decay in semantic relevance when traversing a dotted link between a concept C and a role restriction $\exists r.C$.

As an example, assuming that $OS(A, w, A) = q$, then the *OntoScore* would propagate as shown in Figure 7 to different nodes in the ontology.

We provide a syntactic name to the concepts corresponding to existential relationship restrictions so as to allow calculating $IRS(x, w)$ when x is a role restriction concept of the form $\exists r.C$. The syntactic name in our implementation is *"Exists"+r+C*. For example, the relationship *"finding site of"* between *"Asthma Attack"* and *"Bronchial Structure"* in Figure 2 gives rise to the new existential role restriction named *"Exists finding site of Bronchial Structure"*.

V. Architecture and System Overview

In this section we present the architecture and overview of the XOntoRank system.

A. XOntoRank Architecture

Figure 8 shows the architecture of XOntoRank, which is divided into two stages. The pre-processing phase consists of the Index Creation Module, which takes as input the corpus of XML-formatted EMR documents to be indexed (CDA in our experiments), the ontological system(s) referenced in the EMR documents and the set of all keywords (the vocabulary) to be indexed.

The Index Creation Module generates the *XOntoRank Dewey Inverted Lists (XOnto-DILs)* which are inspired from the Dewey Inverted Lists of XRANK [6]. XRANK is based on *ElemRank*, a variation of the PageRank algorithm that exploits the structure and containment edges of XML documents. The key difference is that instead of $ElemRank(v)$ we store $NS(v, w)$, that is, the relevance score of node v with respect to keyword w given the XML documents and the ontological systems, defined in (5). *ElemRank* could be incorporated in $NS(v, w)$ but our CDA documents have no ID-IDREF edges and hence *ElemRank* would make no difference.

For example, Figure 9 shows the Dewey ID's generated for a subset of the document of Figure 1. We have truncated

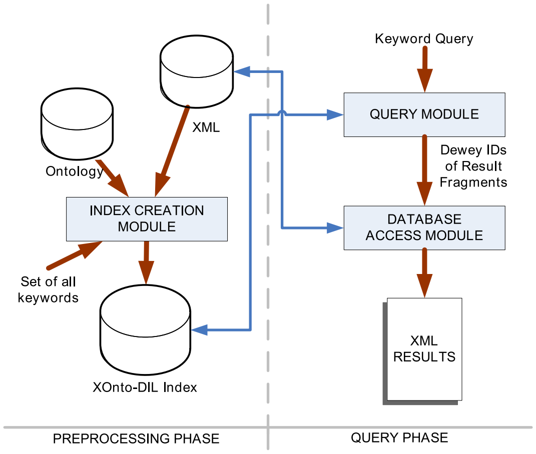


Fig. 8. XOntoRank Architecture.

the prefix in the Dewey ID's for space constraints. Figure 10 shows a fragment of the *XOnto-DIL* for the same document. Note that the first component of each Dewey ID is the document ID. The process to build *XOnto-DILs* is described in detail in Section VI-B.

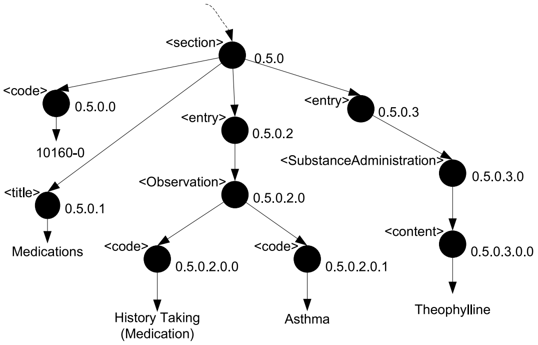
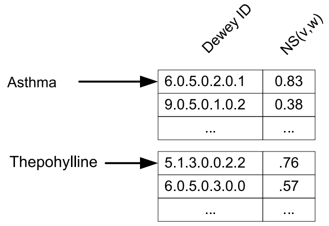


Fig. 9. Dewey IDs for CDA Document.



	Dewey ID	NS(v,w)
Asthma	6.0.5.0.2.0.1	0.83
	9.0.5.0.1.0.2	0.38
	...	...
Thepohylline	5.1.3.0.0.2.2	.76
	6.0.5.0.3.0.0	.57
	...	...

Fig. 10. Dewey Inverted List for CDA Document.

During the query phase, the Query Module inputs the user keyword query and executes XRANK's DIL algorithm using the XOnto-DILs generated in the pre-processing phase. The Database Access Module then obtains the appropriate XML fragments addressed by the resulting Dewey ID's.

B. Building the XOnto-DILs

In this section we describe how the *XOnto-DIL*s are computed for the various semantics described in Section IV. We compute *XOnto-DIL*s for all words in the Vocabulary, defined as the union of words in the ontological systems $O_1, \ldots, O_s$ and in documents in D. As above, we assume there is a single ontological system O_0. *XOnto-DIL*s are computed in three stages:

Full-text Indexing: First, we build a full-text index of the CDA documents and the ontology. This phase is common to all the algorithms, and computes the TF-IDF score.

OntoScore Computation Stage: Second, we build an *OntoScore Hash Map* M, that stores the $OS(v,w)$ for every pair (v,w) of concept node v and keyword w with $OS(v,w) > threshold$, where *threshold* is a predefined value used to improve the efficiency of building M. We chose a *threshold* that could give us a balance of space and quality. The details of computing M, as well as the criteria to choose *threshold* are presented in Section VI.

DIL Creation: Finally, we compute the XOnto-DILs for the documents in D. The $NS(v,w)$ for each pair (v,w) of node $v \in T_i$, $T_i \in D$, $w \in Vocabulary$ is computed by (5), where $OS(CN(v.onto), w)$ is retrieved from Hash Map M. We show how M is computed in the next section.

VI. OntoScore Computation Algorithms

In the next sections we show how the Hash Map M is computed during the OntoScore stage for each of the OntoScore computation methods described in Section IV.

A. Ontology as Undirected Graph

If a node $v \in O_i$ can be reached from multiple concept nodes $u_1, \ldots, u_x$, then we assign to u the maximum score that any of $u_1, \ldots, u_x$ would assign. Again other aggregation functions are possible.

$$OS(v,w) = max_{i=1\ldots x}(OS(v,w,u_i)) \qquad (10)$$

The algorithm to compute the Hash Map M in the *OntoScore* phase is depicted in Algorithm 1.

An inefficiency of Algorithm 1 is that it does breadth-first-search (BFS) starting from all nodes that contain keyword w (Line 4). This can potentially lead to traversing the same node multiple times, once for each BFS instance. This can be avoided using the following observation:

Observation 1: *If multiple BFS instances arrive at a node, then we only need to propagate one value, which corresponds to the aggregate function, that is, we merge the met BFS expansions into one with the aggregate node score.*

The reason is that the score propagates by multiplying by decay for each level. Hence, if v has score $f(OS_i, OS_j)$ where $f(\cdot)$ is the combining function (max in (10)), a node u with distance l from v will have score $f(OS_i, OS_j) \cdot decay^l$. If we would ignore this observation and do the BFS expansions independently, u would get score $f(OS_i \cdot decay^l, OS_j \cdot decay^l)$. The two quantities are equal for any reasonable combining function $f(\cdot)$ like max, sum, and $product$.

The above observation is implemented by doing the following changes to Algorithm 1. We replace Line 4 by 4' and insert Lines 6.1, 6.2 after Line 6.

```
Input: Vocabulary $V$, SNOMED Ontology graph $O$
Output: Hash Map $M$ with key: pair $(v, w)$ where $v$
        is concept node id and $w$ is keyword, and
        value: $OS(v, w)$
1  foreach keyword $w$ in $V$ do
       /* Find all concept nodes in $O$
          that contain $w$                        */
2      $S \leftarrow getRootSet(w, O)$;
3      foreach concept $s \in S$ do
4          do BFS from $s$;
5          foreach accessed concept node $v$ do
6              Compute $OS(v, w)$;  /* By Eq. 7 */
               /* If expanding from $u$ to $v$,
                  $OS(v, w) = OS(u, w) \cdot decay$    */
7              if $M.get(v, w) < OS(v, w)$ then
8                  $M.put((v, w), OS(v, w))$;
9              else
10                 Stop BFS expansion for $v$;
11             end
12         end
13     end
14 end
```

Algorithm 1: Compute OntoScore Hash Map.

```
4'  do BFS in parallel from $s$;
6.1 if $v$ already has an $OS$ score then
6.2     Stop expanding $v$ for expansion instance that produced
        the smallest $OS(v, w)$;
```

Note that to do BFS in parallel we insert all nodes in S in the BFS queue and then do BFS as usual. To halt the expansion of a node v (Line 6.2 in the correction above) that has already been processed and its adjacent nodes C have already been inserted in the queue, we maintain pointers from v to C in the queue, and remove from the queue the nodes in C when v's expansion is halted.

B. Ontology as Taxonomy

As mentioned in Section IV-B, we restrict the links used to compute *OntoScore*, by only considering the *is-a* and *inverse-is-a* edges in SNOMED. Hence, the first modification is to change the loop in Line 3 of Algorithm 1 to restrict the BFS to only follow these two types of relationships, capturing only the taxonomic portion of the ontology.

We also modify the way in which $OS(v, w)$ is computed (Line 5 of Algorithm 1), replacing the formula in (7) by the cases exposed in Section IV-B. In particular, if we expand from node u with $OntoScore OS(u, w)$ to node v, then:

- if $u \xrightarrow{is-a} v$ then $OS(v, w) = \frac{OS(u,w)}{InDegree_{is-a}(v)}$
- if $u \xleftarrow{is-a} v$ then $OS(v, w) = OS(u, w)$

where $InDegree_r(v)$ is the number of incoming relationship edges of type r.

The rest of the algorithm stays as specified in Algorithm 1, using the same threshold constraints and the same optimization described in *Observation 1*.

C. Ontology as Collection of Relationships

In this case, as mentioned in Section IV-C, all relationship edges are considered. We enumerate below how the expanded nodes are assigned *OntoScores* without having to physically create the ontological graph with the existential role restrictions described in Section IV-C. The assigned *OntoScores* are equal to the ones computed by building the ontological graph described in Section IV-C.

Hence, the BFS expansion is the same as in Section V-A. The *OntoScore* computation of Line 5 is changed as follows, to reflect the approach described in Section IV-C. If we expand from node u with *OntoScore* $OS(u, w)$ to node v, then:

- if $u \xrightarrow{is\_a} v$ then $OS(v, w) = \frac{OS(u,w)}{InDegree_{is\_a}(v)}$
- if $u \xleftarrow{is\_a} v$ then $OS(v, w) = OS(u, w)$
- if $u \xrightarrow{r} v, r \neq is\_a$ then $OS(v, w) = a \cdot \frac{OS(u,w)}{InDegree_r(v)}$
- if $u \xleftarrow{r} v, r \neq is\_a$ then $OS(v, w) = a \cdot OS(u, w)$

Note that the denominator $InDegree_r(v)$ is the in-degree of the existential role restriction $\exists r.v$.

VII. Experiments

In this section we experimentally evaluate the XOntoRank system and show the feasibility of both the Preprocessing and Query phases. The experiments were performed on a Pentium 4, 2.8 GHz PC with 1GB RAM. XOntoRank was implemented in Java JDK 5.0, using DOM for XML parsing and Microsoft SQL Server 2000 for the persistent storage of indexes. To access and navigate SNOMED CT, which takes multiple GBs of disk space, we used the API provided by the National Library of Medicine (NLM) Unified Medical Language System (UMLS) [24]. This API provides the necessary methods to query the ontology and dictionary and obtain the concept code and display name for a particular string. We used this API as a black box in both the preliminary CDA document generation and the Index Creation Module of XOntoRank.

In Section VII-A we quantify the differences in the ranking for the alternative OntoScore computation techniques of Section IV. We also present results of a user survey that we performed with the aid of a medical doctor and researcher. In Section VII-B we measure the performance of the XOntoRank system in terms of index creation and query execution times. Some screenshots of the XOntoRank system are available at the project homepage [25]. The system was not made available to the public due to patient record privacy concerns.

CDA Documents Generation: We developed a program to convert automatically the relational anonymized EMR database of the Cardiac Division of a local hospital into a set of XML CDA documents. Each CDA document represents the medical record of a single patient conglomerating all her hospitalization entries. 3 492 such documents were created, each being on average 47KB with 1 133 XML elements. Ontological references were inserted for every XML node whose value matched one of the concepts in SNOMED. This

TABLE I

NUMBER OF RESULTS MARKED AS RELEVANT FOR EACH QUERY. USER MARKS UP TO 5 RESULTS.

	Query	*XRANK*	*Graph*	*Taxonomy*	*Relationships*
q_1	“cardiac” “arrest”	5	5	5	5
q_2	“cardiac” “coarctation”	5	5	5	5
q_3	“neonatal” “cyanosis”	3	3	0	3
q_4	“carbapenem” “ibuprofen”	0	3	0	3
q_5	“supraventricular arrhythmia” “pericardial effusion”	0	0	1	0
q_6	“regurgitant flow” “amiodarone”	0	1	1	2
q_7	“supraventricular arrhythmia” “acetaminophen”	0	0	0	0
	AVERAGE	1.875	2.429	1.714	2.571

resulted in 2 454 CDA documents with ontological references to SNOMED with an average of 151 references per document.

A. Quality Results

We performed two quality experiments. The first one compares the distances between the result lists of the proposed search approaches for a real query workload, and the second one is a proof-of-concept user survey which compares the user satisfaction for these approaches. The four approaches –baseline plus the three described in Section IV– are denoted as *XRANK* (baseline, no use of ontology), *Graph* (Section IV-A), *Taxonomy* (Section IV-B), and *Relationships* (Section IV-C).

Distance between Top-k lists: We performed a series of two-keyword queries obtained from domain expert collaborators. The second column of Table I shows a sample of these queries. Note that some keywords are phrases enclosed in quotes. We use the top-k Kendall Tau [26] measure to determine the distance between the lists and hence test the effects of each individual algorithm. Table II reports the Kendall Tau values for $k = 20$ and penalty parameter $p = 0.5$ (see [26] for definition of p), normalized over 20 queries. We observe the large distance between the result of *Graph* and the *Relationships* algorithm; this was expected since the expansion on the ontology graph achieved by the *Graph* algorithm is less restricted than the *Relationships* algorithm, which extends the *Taxonomy* expansion. For this reason, the distance between *Taxonomy* and *Relationships* lists is small.

TABLE II

NORMALIZED KENDALL TAU VALUES FOR FOUR APPROACHES.

	XRANK	*Graph*	*Taxonomy*	*Relationships*
XRANK	0.000	0.171	0.101	0.209
Graph	0.171	0.000	0.116	1.000
Taxonomy	0.101	0.116	0.000	0.171
Relationships	0.209	1.000	0.171	0.000

Quality Survey: We conducted a survey to determine the quality of each of the four algorithms we presented. Given the specialized nature of our medical records dataset, which come from a children's cardiac clinic, it is hard to find many users to properly evaluate the results. Hence, we chose to only report, as a proof of concept, the results of a survey on a single domain expert–medical doctor and researcher knowledgeable in this area–instead of involving non-expert users who could degrade the reliability of the results.

The results of the survey are shown in Table I. For each query, we presented to the user the union of the top-5 results from each of the four algorithms. The user was asked to select up to 5 results that he found relevant to the query. For this experiment, we set *decay* to 0.5, *threshold* to 0.1 and α to 0.5.

For queries q_1 and q_2, the top-5 results obtained by *XRANK* are also the top-5 results for the ontology-enabled algorithms, because the query keywords appear frequently in the CDA documents. For q_3, *XRANK* only generated three results –all of which were marked as relevant–, but only one of these appear in the top-5 list of the other three algorithms. For the remaining queries, *XRANK* does not produce any results, since there is no CDA document with direct occurrences of both keywords (or phrases). In contrast, the ontology-enabled algorithms find relevant results to the queries by mapping the keyword's concept to other concepts present in the documents. For q_4, both *Graph* and *Relationships* algorithms produce the same results by expanding through non-taxonomical edges in the SNOMED ontology.

For q_5, only the *Taxonomy* algorithm produced a result that was considered “relevant” by the domain expert. This result did not reach the top-5 of *Graph* and *Relationships* algorithms, because the expansion through non-taxonomical concepts produced more compact results –single XML elements that mapped a concept to both query keywords– with higher score, but those were not considered relevant by the domain expert.

For q_6, the *Relationships* algorithm produces better results, because it combines the results of both the *Graph* and *Taxonomy* algorithms; the expansion over the ontology for the *Graph* algorithm decayed before it could reach the taxonomical result found by the *Taxonomy* and *Relationships* algorithms.

Note that in some cases, the semantic knowledge represented by the ontology might not be sufficient to provide high quality Information Retrieval over EMR's. For instance, consider query q_7 =[*“supraventricular arrhythmia” “acetaminophen”*]. The scores of zero for the ontology-assisted algorithms in Table I are due to the following reason: All the results of these algorithms map the concept *“acetaminophen”* to the concept *“aspirin”*. In the context of *pain control*, these two concepts are indeed related, because they both provide relief of pain. But in this specific case, the keyword *“supraventricular arrhythmia”* implies that the target context of this query is not *pain control* but *cardiology*, and in this context, however, these drugs are generally unrelated. *“Aspirin”* has cardiac benefits that are not seen with *“acetaminophen”*, due to the differing properties of the two drugs.

The findings of Table I are summarized as follows. The quality of *Relationships* and *Graph* is generally superior to the baseline *XRANK* algorithm, which means that when the keywords are not present in a document, the ontology-enhanced algorithms are capable of finding "good" results to satisfy the given queries. The *Taxonomy* algorithm can be slightly worse than *XRANK*, since the former could return results where a query keyword is matched to a far ancestor concept, because *Taxonomy* does not penalize the ontology expansion when following *is-a* (parent) edges.

B. *Performance Results*

Pre-processing phase: Building XOnto-DIL lists for all keywords in the SNOMED ontology was not feasible given that they are in the order of millions, the keywords vocabulary cannot be extracted from the provided SNOMED API, and the API is slow given that it is IO-intensive (note that SNOMED is a multi-gigabyte ontology). Note that there is a method to get all occurrences of a specific keyword, but there is no vocabulary of all keywords in the database. Hence, we indexed a subset of this universe of keywords which let us execute a large number of queries and estimate reliable projections of index execution time. In particular we built XOnto-DIL lists for all the keywords in the CDA documents and for all keywords contained in a concept, up to 2 relationships away from a concept referenced in a CDA document (more than 400 unique concepts are referenced in our CDA collection). The above rules translated to the indexing of more than 40 000 keywords directly present in the documents and more that 100 000 concepts from the SNOMED ontology. To navigate SNOMED efficiently, we loaded the appropriate fragment in main memory, thus reducing the access to SNOMED flat files. However, the SNOMED navigation was still too slow. In the future, we plan to work on more efficient ways to navigate the ontology to build the XOnto-DIL lists, as discussed in Section IX. We set *decay* to 0.5, *threshold* to 0.1 and α to 0.5.

Table III presents the average creation time, average number of postings (rows in Figure 10) and size of a XOnto-DIL list of a keyword for each of the four approaches. For the average creation time, we exclude the time taken to navigate the SNOMED ontology, since it can take up to several minutes for frequent keywords, given the current implementation of the SNOMED API.

We observe that the average creation time for *Taxonomy* is much larger than *Graph*. This is due to the fact that the expansion in *Graph* decays continuously, whereas the expansion for *Taxonomy* decays quickly only for descendants, but may expand indefinitely for parent relationships. We also see how the *Graph* and both *Relationships* approaches generate the largest number of XOnto-DIL entries, given the fact that the navigation does not decay for the one direction of *is-a* edges. We observe a high difference between the number of postings for the *Taxonomy* approach compared to the *Relationships* algorithm, giving evidence of the large number of concepts mapped through the ontology graph. Note that the size of the XOnto-DIL entries can be reduced by appropriately adjusting the *threshold* and/or *decay* parameters.

TABLE III

AVERAGE SIZE FOR XONTO-DIL ENTRIES.

Algorithm	*Per Keyword*		
	Avg. Creation Time (ms)	*Postings*	*Size (KB)*
XRANK	1.0	1 435.7	39.3
Graph	4 143.5	20 906.7	571.7
Taxonomy	10 743.5	5 511.9	150.7
Relationships	13 485.3	46 979.5	1 284.6

Query Phase: Figure 11 presents the average execution times for queries with varying number of keywords, for $k = 10$. The time for *Relationships* algorithm is higher due to the larger number of nodes in the XML document that are ontologically related to the query keywords.

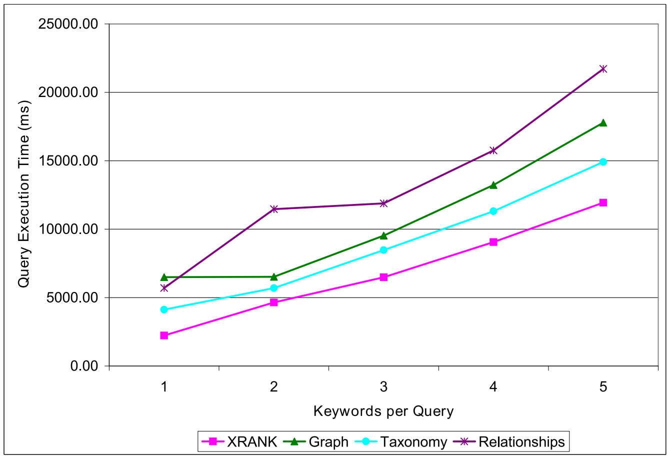


Fig. 11. Average Execution Time for Keyword Queries with Varying Number of Keywords.

VIII. RELATED WORK

Authority Flow: XOntoRank uses related techniques to our previous work in ObjectRank [21], [27]. The principle of authority flow is the main concept under these two systems, but there are also key differences between them. As a first difference, ObjectRank performs IR over relational databases, whereas XOntoRank works over XML tree documents —with no associations among them— augmented with the graph of the referenced ontologies. In ObjectRank, the source of authority is the data nodes that contain the keywords; in XOntoRank, the source of authority also includes XML elements that link to the ontology, without directly containing the keywords. The second key difference is that we do not apply iterative PageRank-style authority propagation in this work (as we do in ObjectRank). Applying ObjectRank on the ontology graph would be an alternative option, but we chose to use a one-pass BFS expansion algorithms for scalability purposes, given the size of SNOMED and the number of unique keywords for which ObjectRank should be precomputed.

Searching XML documents: The following works perform keyword search on XML documents without considering any external knowledge, such as ontologies. XSEarch [7] ranks the results, taking into consideration both the degrees of the semantic relationship and the relevance of the keyword. We found that XSEarch would not be an appropriate framework to base XOntoRank, since their "interconnection relationship" would not work well in the particular case of CDA documents. XIRQL [5] utilizes a strategy different to XSEarch's to compute its ranking, defining index units, specific entity types that can be indexed and used for tf-idf computation. Schema-free XQuery [28] refines the work of XSEarch by utilizing meaningful lowest common ancestors instead of the concept of interconnected nodes. Cohen et al. [29] improve even further this approach by including the schema into the framework and discovering interconnection information. Xu and Papakonstantinou [30] define a result as a "smallest" tree, that is, a subtree that does not contain any subtree that also contains all keywords. Hristidis et al. [31] group structurally similar tree-results to avoid overwhelming the user. XKeyword [8] operates on an XML graph (with ID-IDREF edges) and returns subtrees of minimum size.

Query Expansion: Various query expansion strategies (e.g. [32]) have been proposed for general as well as biological documents search. For instance, the QEEF framework [33] uses the UMLS ontology to suggest additional terms. [34], [35], [10], assign weights on the ontology edges by comparing the distributions of the contents of the two nodes and of their combination on a very large dataset like the Web. This approach, which complements our work, is too time-consuming for large ontologies like SNOMED. The ontological associations are exploited by expanding the XXL query. It differs from our approach in which XXL considers symmetric associations between ontology concepts, whereas we use the authority flow model. [9], [36] expand the query by matching the ontology to the document DTD. All the above techniques are proposed for structured XML queries. For our case of keyword queries, query expansion is not appropriate, since it leads to non-minimal results (see [4] for a definition of a minimal keyword search result) — the same concept appears multiple times in a result.

Semantic similarity: In Information Retrieval, two approaches have addressed the problem of computing similarity between two concepts. Initially, statistical correlations between terms were exploited [37]. With the conception of ontologies and semantic networks like WordNet [38], a graph-oriented approach was adopted, focusing on the number, depth and direction of the edges between two concepts [39]. A more recent approach has combined these two techniques [40], [41] by taking into account the graph structure and statistics.

In the Semantic Web, various approaches have been suggested to measure semantic similarity between different artifacts. Most similarity measures such as [42], [43] focus only on subsumption relations (i.e. hierarchical *"is-a"* links in an ontology). Maguitman et al. [44] propose an information theoretic measure of similarity that also considers non-hierarchical links. However, their approach requires the presence of a large number of instances to determine the similarity between concepts. In the medical domain, most ontologies, including SNOMED, only describe concepts and not instances. Hence, their approach cannot be used. The notion of authority flows is also similar to the spreading activation scheme that is used in information retrieval [45] and web mining [46]. A novel aspect of our approach is the use of strategies based on description logics and the spreading of activation from the ontology into the XML documents.

IX. Conclusions and Future Work

We have introduced the problem of ontology-aware keyword search on XML-based EMR documents, which contain references to clinical ontological concepts. We defined semantics for this problem, where the ontological references, as well as the relationships within the ontology are used in creating and ranking the query results. Alternative views of the ontology were considered. We created efficient algorithms, building on previous work, to generate the top-k query results. The algorithms were evaluated experimentally, showing that the precision and recall of our algorithm is better than the baseline algorithm.

A critical future direction is the optimization of the index creation process. Our current index creation approach relies on the API and data provided by [14], which are based on flat files. Implementing approximation and early pruning techniques, as well as in-memory representations of the ontology graphs, may prove useful in scaling to larger ontologies and datasets.

Acknowledgment

This project was supported in part by the National Science Foundation Grants IIS-0534530 and IIS-0811922. Dr. Weiner is supported by grant 5K23AG020088 from the National Institute on Aging.

References

[1] "Health Level Seven XML," http://www.hl7.org/special/Committees/xml/index.cfm, 2008.

[2] "HL7 Clinical Document Architecture, Release 2.0 (2004)," http://lists.hl7.org/read/attachment/61225/1/CDA-doc%20version.pdf, 2007.

[3] H. A. Proper and P. D. Bruza, "What is information discovery about?" *J. Am. Soc. Inf. Sci.*, vol. 50, no. 9, pp. 737–750, 1999.

[4] V. Hristidis and Y. Papakonstantinou, "DISCOVER: Keyword search in relational databases," 2002. [Online]. Available: citeseer.ist.psu.edu/hristidis02discover.html

[5] N. Fuhr and K. Großjohann, "XIRQL: a query language for information retrieval in XML documents," in *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 2001, pp. 172–180.

[6] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram, "XRANK: ranked keyword search over XML documents," in *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2003, pp. 16–27.

[7] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv, "XSEarch: a semantic search engine for XML," in *VLDB'2003: Proceedings of the 29th international conference on Very large data bases*. VLDB Endowment, 2003, pp. 45–56.

[8] V. Hristidis, Y. Papakonstantinou, and A. Balmin, "Keyword proximity search on XML graphs," 2003, in ICDE. [Online]. Available: citeseer.ist.psu.edu/hristidis03keyword.html

[9] M. S. Kim and Y.-H. Kong, "Ontology-DTD Matching Algorithm for Efficient XML Query," in *FSKD (2)*, ser. Lecture Notes in Computer Science, L. Wang and Y. Jin, Eds., vol. 3614. Springer, 2005, pp. 1093–1102.

[10] R. Schenkel, A. Theobald, and G. Weikum, “Semantic Similarity Search on Semistructured Data with the XXL Search Engine,” *Inf. Retr.*, vol. 8, no. 4, pp. 521–545, 2005.

[11] “Health Insurance Portability and Accountability Act.” http://www.hipaa.org/, 2008.

[12] K. LeFevre, R. Agrawal, V. Ercegovac, R. Ramakrishnan, Y. Xu, and D. DeWitt, “Limiting disclosure in hippocratic databases,” in *VLDB ’04: Proceedings of the Thirtieth international conference on Very large data bases*. VLDB Endowment, 2004, pp. 108–119.

[13] “California Clinical Data Project: Setting Standards.” http://www.chcf.org/documents/CCDPProjectOverview.pdf, 2008.

[14] “SNOMED Clinical Terms (SNOMED CT),” http://www.snomed.org/snomedct/index.html, 2008.

[15] R. Dolin, L. Alschuler, C. Beebe, P. Biron, S. Boyer, D. Essin, E. Kimber, T. Lincoln, and J. Mattison, “The HL7 Clinical Document Architecture.” *J Am Med Inform Assoc*, vol. 8, no. 6, pp. 552–69.

[16] “HL7 Reference Information Model.” http://www.hl7.org/library/datamodel/RIM/C30204/rim.htm, 2008.

[17] G. Salton, *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[18] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison Wesley, May 1999. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&amp;path=ASIN/020139829X

[19] S. E. Robertson and S. Walker, “Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval,” in *SIGIR ’94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: Springer-Verlag New York, Inc., 1994, pp. 232–241.

[20] A. Singhal, “Modern Information Retrieval: A Brief Overview,” *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 24, no. 4, pp. 35–42, 2001. [Online]. Available: http://singhal.info/ieee2001.pdf

[21] A. Balmin, V. Hristidis, and Y. Papakonstantinou, “ObjectRank: Authority-based keyword search in databases,” in *VLDB, 2004*, 2004. [Online]. Available: citeseer.ist.psu.edu/balmin04objectrank.html

[22] F. Baader, *The Description Logic Handbook : Theory, Implementation and Applications*. Cambridge University Press, January 2003. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&amp;path=ASIN/0521781760

[23] F. Baader, C. Lutz, and B. Suntisrivaraporn, “Efficient Reasoning in $\mathcal{EL}^{+}$,” in *Proceedings of the 2006 International Workshop on Description Logics (DL2006)*, ser. CEUR-WS, 2006.

[24] “NLM Unified Medical Language System,” http://www.nlm.nih.gov/research/umls/, 2008.

[25] Florida International University School of Computing and Information Sciences, “XOntoRank Project Homepage,” 2008, http://dsrl.cs.fiu.edu/projects/ir_biomedical/xontorank_homepage/.

[26] R. Fagin, R. Kumar, and D. Sivakumar, “Comparing top k lists,” in *SODA ’03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003, pp. 28–36.

[27] V. Hristidis, H. Hwang, and Y. Papakonstantinou, “Authority-based keyword search in databases,” *ACM Trans. Database Syst.*, vol. 33, no. 1, pp. 1–40, 2008.

[28] Y. Li, C. Yu, and H. V. Jagadish, “Schema-free XQuery,” in *VLDB’2004: Proceedings of the Thirtieth international conference on Very large data bases*. VLDB Endowment, 2004, pp. 72–83.

[29] S. Cohen, Y. Kanza, B. Kimelfeld, and Y. Sagiv, “Interconnection semantics for keyword search in XML,” in *CIKM ’05: Proceedings of the 14th ACM international conference on Information and knowledge management*. New York, NY, USA: ACM, 2005, pp. 389–396.

[30] Y. Xu and Y. Papakonstantinou, “Efficient keyword search for smallest LCAs in XML databases,” in *SIGMOD ’05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2005, pp. 527–538.

[31] V. Hristidis and Y. Papakonstantinou, “Keyword proximity search in XML trees,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 525–539, 2006, member-Nick Koudas and Member-Divesh Srivastava.

[32] J. Xu and W. B. Croft, “Query expansion using local and global document analysis,” in *SIGIR ’96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 1996, pp. 4–11.

[33] D. Wollersheim and W. J. Rahayu, “Using Medical Test Collection Relevance Judgements to Identify Ontological Relationships Useful for Query Expansion,” in *ICDEW ’05: Proceedings of the 21st International Conference on Data Engineering Workshops*. Washington, DC, USA: IEEE Computer Society, 2005, p. 1160.

[34] A. Theobald, “An Ontology for Domain-oriented Semantic Similarity Search on XML Data,” in *BTW*, ser. LNI, G. Weikum, H. Schöning, and E. Rahm, Eds., vol. 26. GI, 2003, pp. 217–226.

[35] R. Schenkel, A. Theobald, and G. Weikum, “Ontology-Enabled XML Search,” in *Intelligent Search on XML Data*, ser. Lecture Notes in Computer Science, H. M. Blanken, T. Grabs, H.-J. Schek, R. Schenkel, and G. Weikum, Eds., vol. 2818. Springer, 2003, pp. 119–131.

[36] M. S. Kim, Y.-H. Kong, and C. W. Jeon, “Remote-Specific XML Query Mobile Agents,” in *DEECS*, ser. Lecture Notes in Computer Science, J. Lee, J. Shim, S. goo Lee, C. Bussler, and S. S. Y. Shim, Eds., vol. 4055. Springer, 2006, pp. 143–151.

[37] M. E. Lesk, “Word-word association in document retrieval systems,” *American Documentation*, vol. 20, no. 1, pp. 27–38, 1969.

[38] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&amp;path=ASIN/026206197X

[39] R. Rada, H. Mili, E. Bicknell, and M. Blettner, “Development and application of a metric on semantic nets,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 1, pp. 17–30, Jan/Feb 1989.

[40] D. Lin, “An Information-Theoretic Definition of Similarity,” in *ICML*, J. W. Shavlik, Ed. Morgan Kaufmann, 1998, pp. 296–304.

[41] P. Resnik, “Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 95–130, 1999. [Online]. Available: citeseer.ist.psu.edu/resnik99semantic.html

[42] L. Li and I. Horrocks, “A software framework for matchmaking based on semantic web technology,” 2003. [Online]. Available: citeseer.ist.psu.edu/li03software.html

[43] J. W. Kim and K. S. Candan, “CP/CV: concept similarity mining without frequency information from domain describing taxonomies,” in *CIKM ’06: Proceedings of the 15th ACM international conference on Information and knowledge management*. New York, NY, USA: ACM, 2006, pp. 483–492.

[44] A. G. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani, “Algorithmic detection of semantic similarity,” in *WWW ’05: Proceedings of the 14th international conference on World Wide Web*. New York, NY, USA: ACM, 2005, pp. 107–116.

[45] F. Crestani, “Application of Spreading Activation Techniques in InformationRetrieval,” *Artif. Intell. Rev.*, vol. 11, no. 6, pp. 453–482, 1997.

[46] F. Gelgi, S. Vadrevu, and H. Davulcu, “Improving Web Data Annotations with Spreading Activation,” in *WISE*, ser. Lecture Notes in Computer Science, A. H. H. Ngu, M. Kitsuregawa, E. J. Neuhold, J.-Y. Chung, and Q. Z. Sheng, Eds., vol. 3806. Springer, 2005, pp. 95–106.