Domain Specific Multi-stage Query Language for Medical Document Repositories

S. Bhalla
Model Driven Querying
Introduction

- **Specialized Domains**
  - Biomedical, agriculture, medical/healthcare
  - Require → Effective search and query mechanisms
  - Insufficient → Search-engine like Key-words based searches

- **Medical domain**
  - ✓ Complex
  - ✓ Example,
    - **Query** → Information about → "general AIDS information" → medical search tool, such as PubMed
    - **Result** → 1000s of documents ≤ Different aspects of AIDS are displayed
      Such as, treatment, drug therapy, transmission, diagnosis, and history
  - ✓ Medical professionals → Specific technical articles (particular topic)
  - ✓ General public → General information (disease or medicine).

- **How to retrieve medical query related information**
Introduction (1)

- Medical Information
  - Knowledge → Evolved over 10s of years
  - Contains → Well defined terms and processes

- Available on the Web
  - Patient Specific Information
  - Knowledge-based Information

Web Documents

Medical Literature

EHRs

Patient-encounter Recordings
Introduction (2)

- **Complexity** → Knowledge-based Resources

- **Heterogeneous** → End-user groups
  - Patients, researchers, doctors and other experts

- **Variation** → Information Requirements
  - Patient-treatment, self-diagnosis, general health information

- **Structure** → Medical Documents
  - Scientific papers, encyclopedias and other literature
  - Unique, well-defined
Introduction (3): Specialized Documents

- **Case of medical encyclopedias**
  - Comprehensive medical guide → Patients and clinicians
  - Authoritative source → NLM (National Library of Medicine)
  - Paper based resources → Electronic format
  - Frequently referred → Medical domain users
  - Example, MedlinePlus, WebMD, ADAMS, Merriam-Webster Medical Dictionary
Introduction (4): Why Query

- **External knowledge base** → Clinicians
- **Evidence based medicine**
- **During** → different stages of point-of-care
- **Assessment plan of treatment**
  - Patient diagnosis
- **Improve Quality of Care**
  - Authoritative information required
- **Self Diagnosis**
  - During → Early appearance of symptoms
  - Personal Knowledge → Patients and their relatives
The Underlying Structure

The Hierarchical Structure

- Topic of the Document
- Subtopics
  - Subtopic 1
  - Subtopic 2
  - Subtopic n
- Content
  - Content topic 1
  - Content topic 2
  - Content topic n
- Miscellaneous/Related Content

Flow of Contents ➔ Organized as stages of point-of-care
Introduction (5): The End Users

- **Variable**
  - Demographical Characteristics
  - Tasks/Purpose
  - Computer/Domain Expertise

- **Practitioners and Researchers**
  - **Well-versed**
    - Domain knowledge and terminologies
  - **Require**
    - Precise, complete, accurate and timely results

- **Patients and their relatives**
  - **NOT Well-versed**
    - Domain knowledge and terminologies
  - **Require**
    - General information
The Medical Queries

- **Evidence-based Queries**
  - **Intent**: Diagnostic
  - **Raised by**: Clinicians/Experts
  - **Target resources**: Online Medical Repositories (e.g. medical encyclopedia)
  - **Example**: “Cases where *helicobacter bacteria* causes *peptic ulcer*”

- **Hypothesis-directed Queries**
  - **Intent**: Non-diagnostic
  - **Raised by**: Novice users/patients
  - **Target resources**: Online Medical Repositories (e.g. medical encyclopedia)
  - **Example**: “Treatment in case of *high fever* and *dizziness*”
The Query Flows

- **Occurrence** → Evidence-based and hypothesis-directed queries
- **Represent** → Stages of information seeking
- **Comprise** → Varying levels of query complexity

1. **Simple**
   - Cases = ?
   - where
   - Causes = "Fever"

2. **Medium**
   - Causes = "Fever"
   - Symptoms = "Vomiting"
   - Medication = ?

3. **Complex**
   - Cases = ?
   - where
   - Symptoms = "Fever"
   - AND
   - Symptoms = "High Blood Pressure"

4. **Recursive**
   - Causes = "Vomiting"
   - More Symptoms = "Fever"
   - Remedies = ?

Clinician's Knowledge Base = Web document Repository Layout
**Query:** Find chances of "Cancer Risk" in patients showing symptom "Sleep Deprivation" and have been exposed to "Radiation" (but not "Environmental Toxins" and does not have "Genetic Disorder").

- **Results** → Large in number, irrelevant
- **Failure** → Keyword search, domain-specific search tools
- **Require** → Precise and easy-to-use database style query methods
  - **Key steps:**
    1. Schema → understandable by users
    2. Identify → Resources to query
    3. Identify → Granularity of results
Bridging the Gap

- **Aim: Effective Online Medical Information**
  - Transform $\rightarrow$ Document Repository $\rightarrow$ User-Level Schema
  - Enable $\rightarrow$ High-level Query Language
  - Target Audience $\rightarrow$ Skilled and semi-skilled users
  - Utilize $\rightarrow$ Query capabilities of a database query language

- **Assist Domain Experts $\rightarrow$ Using Query language**
  - Facilitate
    - In-depth Queries
    - Granular Results
Querying the New Way

**Resource**
- **WebMD**
- **MedlinePlus**

**Search Method**
- Google
- Yahoo!

**Medical Expert**
- Traditional Method

**Results returned**
- Lack specificity
- Long list of full documents
- Trustworthiness of resources → unknown

**User-level Schema**

**High-level Query language**

**Query Method**

**Medical Expert**
- Proposed Method

**Results returned**
- Specific, granular
- Segments of documents → query criteria
- Trustworthy/Authoritative sources only
Proposed Approach
Key Features

- **User-Level Schema**
  - Universal, concept-level schema
  - Attributes
    - Understandable → Domain experts and novice users
    - Query-able
  - Granular results

- **Multi-stage Query Language**
  - Map multi-stage diagnostic process → Step-by-step Query Flow
  - Interactive Querying → View Results → Add concept
    - Continuous query refinement
  - Supported Queries → Simple, Medium, Complex, Recursive
Outline

Two-step framework

- Offline Process ➔ Create User-level schema
- Online Process ➔ Enable Multi-stage Query Language

Offline Process

- Use ➔ Web segmentation algorithm, Domain concepts
- Result ➔ Automatic creation of a User-level schema

Online Process

- Enable ➔ Multi-stage Query Language
- Use ➔ User-level schema
- Results ➔ Granular, segment-level, context-based
The Method

Two-step Framework
Data Model
Data Model

- Tree Structured Repository

Example: MedlinePlus Medical Encyclopedia
Data Model (1)

- Document Repository is a collection of documents, \( R = \sum_{i=1}^{n} d_i \)

- A document \( d_i \in R \) is a collection of segments, \( d_i = \sum_{j=1}^{m} f_j \)

- A segment \( f_j \in d_i \) is a pair of subtopic label and content enclosed, \( f_j = (s_j + c_j) \)

- A query on a single/multiple documents can be defined as,
  - \( Q = \) return all segments \( f \in \sum d_i \) where \( s \in \) queried concepts and \( c \) satisfies entered values

- The result of this query can be defined as,
  - \( Res = \) all segments \( f \in \sum d_i \)

- The subtopic label considers the user context for query
Data Model (2): The Schema

- Attributes $\rightarrow$ Diagnostic concepts/terms
- Understandable by expert and novice users
- Do not change frequently

MedlinePlus Medical Encyclopedia Articles

![Diagram](chart.png)
Data Model (3): A XML Document

- Document corresponding to “Aarskog Syndrome”

**Example:** MedlinePlus Medical Encyclopedia
### Query: Find if "Oxygen therapy" work for the treatment of "Chronic Respiratory Failure" and symptoms are "Lethargy" OR "Shortness of breath".

<table>
<thead>
<tr>
<th>Advanced search: MedlinePlus</th>
<th>Proposed Method</th>
</tr>
</thead>
</table>
| ![MedlinePlus](image)       | **SELECT** attribute = “Disease_name”  
WHERE  
Attribute “Disease_name” = “Chronic Respiratory Failure”  
AND  
Attribute “Treatment” = “Oxygen therapy”  
AND  
Attribute “Symptoms” = “Lethargy”  
**OR**  
Attribute “Symptoms” = “Shortness of breath” |
Data Model (5): Granular Results

- Each result is a segment, combination of:
  - Concept/context in query
  - Item of concern (content enclosed in a segment)
An Example

Query: Find other symptoms where “chronic kidney failure” is caused by “anemia”

- Queried segment $\rightarrow$ Symptoms
- Segments in Query $\rightarrow$ Causes = “anemia” and Disease_name = “Chronic kidney failure”

- Result Segment $\rightarrow$ Symptoms
- Context $\rightarrow$ disease_name = “chronic kidney failure” & causes = “anemia” (SYMPTOM - segment)
Next Step ➔ Multi-stage Query Language
Proposed Query Language (1)

- **XQBE** ➔ **Medical document repositories**
  - Create queries ➔ Drag and drop interface
  - **Query**: “Find cases where a person is inflicted with “peptic ulcer” due to “helicobacter pylon bacteria”

- **Attributes** ➔ **understandable by end users**
  1. **Case** = `disease_name`
     - **Value** = ??
  2. **Due to** = **Causes**
     - **Value** = “helicobacter pylon bacteria”
  3. **Inflicted with** = **Symptoms**
     - **Value** = “peptic ulcer”

- **Query Effort**
  - Minimal learning curve
  - Computer-expertise ➔ not required
Proposed Query Language (2)

- **Multi-stage Query-by-Concept**
  - Concept → Query-able attribute
  - Topic, sub-topic, medical concept

- **Query Effort**
  - Dynamic selection of attributes
  - No computer expertise

- **Query Process**

An Example: Cases where fever is caused due to infliction of Pneumonia and Tuberculosis
Another Example

- **Query:** Find cases where 3 clinical concepts ("cough", "no sore-throat", and "had no sterol injection") occur in context of symptoms occur along with a concept having sub-key (i.e. "non sterol injection at the left side")

- **Possible → XQBE on Specialized Medical Repositories** ✔️

- **Possible → Multi-stage Query-by-concept Query Language** ✔️
Evaluation Plan

- **Data Sets**
  - **MedlinePlus document repository**
    - Health topics (900+), encyclopedia (4000+), drugs (12000+)
  - **Set of Queries**
    - 50 test queries (multi-staged)
      - Using diseases and medication etc.

- **Quantitative Studies**
  - **Evaluation Metrics**
    - Accuracy of segment extraction (schema creation) → Precision and Recall
    - Reduction in search space

- **Qualitative Studies**
  - **Usability Studies**
    - Actual End-users
    - Query Performance
Initial Achievements

- HTML → XML as per proposed model
- XQuery on XML
- Integration with XQBE
- Query by concept → Enumeration using paper and pencil
Challenges

- **Scalability** $\rightarrow$ Similarly structured repositories
- **List** $\rightarrow$ Query operations needed
- **Implementation** $\rightarrow$ Above query operations
- **Query Language** $\rightarrow$ User Interface
Related Work

- **Domain-specific Information Retrieval**
  - Similarity and popularity based models **→** Insufficient for domain experts
  - “Information granulation” needs to be considered in huge document repositories

- **Form-based Query Interfaces**
  - Easy-to-use
  - Limited access to the database
  - Complex queries **→** large number of forms
  - Varying medical concepts **→** large number of fields in forms

- **Beyond single page web search results**
  - Provide granular results for user’s search
  - Return segments from multiple or related web documents as results

- **High-level Graphical Query Languages**
  - Easy-to-use and understand
  - Little or no programming effort required by the user
  - Common languages **→** QBE, XQBE
Summary and Conclusions

- Proposed → Multi-stage Query Language

1. Aim → Effective online medical resources

2. Key feature → User-Level Schema

3. Facilitates → Granular/Context-based Results

4. Support → Healthcare Experts

5. Minimize → Learning curve for novice users

6. Reduce → Dependency on general-purpose search engines

- Provide → Web user level activity → no or little programming effort → Healthcare experts
References (1)

References (2)

Questions