Proceedings of the 5th IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2011) 2011:512-519.

Semantic Predications for Complex Information Needs in Biomedical Literature

Delroy Cameron*, Ramakanth Kavuluru*, Olivier Bodenreider†, Pablo N. Mendes*, Amit P. Sheth*
and Krishnaprasad Thirunarayan*
* *Kno.e.sis Center, Wright State University, Dayton, OH 45435, USA*
† *National Library of Medicine, Bethesda MD 20894, USA*
{*delroy, rama, pablo, amit, prasad*}*@knoesis.org*, {*olivier*}*@nlm.nih.gov*

***Abstract*—Many complex information needs that arise in biomedical disciplines require exploring multiple documents in order to obtain information. While traditional information retrieval techniques that return a single ranked list of documents are quite common for such tasks, they may not always be adequate. The main issue is that ranked lists typically impose a significant burden on users to filter out irrelevant documents. Additionally, users must intuitively reformulate their search query when relevant documents have not been not highly ranked. Furthermore, even after interesting documents have been selected, very few mechanisms exist that enable document-to-document transitions. In this paper, we demonstrate the utility of assertions extracted from biomedical text (called semantic predications) to facilitate retrieving relevant documents for complex information needs. Our approach offers an alternative to query reformulation by establishing a framework for transitioning from one document to another. We evaluate this novel knowledge-driven approach using precision and recall metrics on the 2006 TREC Genomics Track.**

***Keywords*-semantic predications, question answering, background knowledge, literature-based discovery, text mining**

I. Introduction

Many scientific researchers are interested not only in *whether* things are connected, but *how* they are connected and the *effects* of various physiological, environmental, chemical and other conditions. In the biomedical domain, evidence for Literature-Based Discovery (LBD) arising from such complex information needs, comes from Don R. Swanson's scientific discoveries. Through extensive searching, exploring, and manual perusal of biomedical literature, Swanson hypothesized that 1) patients suffering from Raynaud's Syndrome "might benefit from dietary fish oils rich in eicosapentaenoic acid" [1] and that 2) migraine headaches are linked to magnesium deficiency [2]. However, Swanson's intensive manual approach to open-domain complex information needs will not scale in today's information age. PubMed alone indexes over 20 million biomedical articles. Instead, a hyperlink-driven approach, as evidenced by the World Wide Web (WWW), has become the simplest and quickest way of finding information. Nearly 50% of all queries on the web are informational [3].

While effective in many search scenarios, using traditional web-centric approaches to satisfy complex information needs in scientific literature presents many challenges. The main issue is that while scientific documents may be inherently linked through a hyperlinked citation network, their actual content is almost always devoid of links. This absence of a mechanism for *linking content* is contrary to the "memory extender (or memex) vision" outlined by Vannevar Bush in 1945 [4]. Bush explained that the human brain navigates an information space using associations. He noted that "With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain." This process, called *trail blazing*, will likely be disrupted if annotations that allow transitions among documents based on content are not adequately provided [5].

The second issue is that the few attempts at linking scientific content through semantic annotations [6]–[8], have not been widely adopted. Instead, users still largely engage in the two-step process of 1) searching for relevant documents, then 2) sifting through large volumes of content for actual information relevant to their interests. This is known as the *search-and-sift paradigm* [5] and is unsuitable for search when answers span multiple documents. Consider for example, the question: *"How do mutations in the Presenilin-1 (PS1) gene affect Alzheimer's disease (AD)?"* The complete answer to this question spans several documents. Potentially, each document discusses a different aspect of how a PS1 mutation affects AD. For example, it has been reported in two PubMed documents that:

…mutations in PS1 lead to Alzheimer's disease by increasing the extracellular levels of [amyloid peptide 42] A42. (*Source: PMID10652366*)

…familial early onset Alzheimer's disease is caused by point mutations in the amyloid precursor protein gene on chromosome 21, in the presenilin 2(PS2)1 gene on chromosome 1, or, most frequently, in the presenilin 1(PS1) gene on chromosome 14 … (*Source: PMID9013610*)

A scientist who poses a query such as *"Presenilin1 chromosomes,"* will be frustrated by a system that does not return both documents. Obviously, such a situation can arise specifically due to syntactically different manifestations of the same concept. PS1 and Presenilin1 are the same concept semantically but not lexically. Although both fragments contain the common phrase *"Alzheimer's disease,"* some amount of searching-and-sifting and also query reformulation would be required to obtain all relevant documents. In fact, query reformulation itself can be problematic. It has been well established that users perform better at recognition than recall [9]. It is easier to identify an actor in a movie, if shown photographs of actors rather than arbitrarily guessing.

We therefore envisage retrieval of relevant documents for complex information needs, under circumstances where complementary or alternative approaches to query reformulation are available. In particular, suppose fragment 2 above was retrieved first; then since it is known that PS1 and Presenilin1 are the same concept semantically, fragment 1 and fragment 2 can be connected using the assertion that 1) *chromosome 14 finding_site_of Presenilin1* from fragment 2 and the assertion that 2) *PS1 associated_with Alzheimer's disease* from fragment 1. These two statements (*chromosome 14 finding_site_of Presenilin1* and *PS1 associated_with Alzheimer's disease*) are examples of semantic predications. If such predications can be extracted from scientific literature, they can be used as an alternative to hyperlinks. While this connection between the two fragments does not directly answer our question, it intuitively leads to documents that do.

Semantic predications therefore offer a mechanism for transitioning from one document to another, while also serving as contextual hints during exploration. Fundamentally the predications enable a paradigm shift away from the classical *bag-of-words* document model, to a predication-based model in which documents can be perceived as a *set-of-predications* that capture a semantic summary of the document. This view establishes the basis for a completely graph-based simulation of exploration of a document space. The idea is that by traversing the predications connecting the documents in the document space we can mimic user activity in a real world system. Such automation is significant, because it bears significance in finding implicit connections among predications. This is quite an intriguing prospect considering Swanson's manually driven discoveries [1], [2], [10], [11].

The predication-based framework can therefore be used for Information Retrieval (IR) or Question Answering (QA), by exploiting the presence of predications in documents. In this work, we take a first step by using the predications to show that documents that answer complex information needs can be connected. Hence, while fundamentally addressing QA, we also highlight the implications of this approach on LBD. Herein lies the novelty in our contribution.

We describe our approach in Section II, then discuss the dataset to which our experiments were applied in Section III. The algorithm used to generate our results is covered in Section IV and we describe the experimental results in Section V. Section VI covers related work.

II. Approach

We formulate the problem of finding documents that satisfy complex information needs using semantic predications, as one of *reachability*.

A. Reachability

Reachability [12] refers to the existence of a path from one vertex to another in a directed graph. Such a path may be obtained using ordered pairs of vertices. In Figure 1 for example, vertex (f) is reachable from vertex (a) through the set of ordered pairs (a,b),(b,c),(c,d),(d,e),(e,f). In this work, our approach is to exploit the labeled edges between vertex pairs, which together form semantic predications, to establish reachability of documents. Hence, we extend the notion of vertex reachability to *document reachability*. Specifically, a document d_j is reachable from a document d_i if d_j contains an entity vertex (v_j) and d_i contains another entity vertex (v_i) such that there is a path from vertex (v_i) to vertex (v_j) using the semantic predications. In Figure 1, document d_8 is reachable from d_1 using the ordered pairs above, consequently covering all documents $\{d_1 \ldots d_8\}$.

In order to demonstrate document reachability, our system requires four components: 1) a corpus of documents, 2) a set of questions and corresponding answer documents 3) a graph of predications (hereafter predications graph) and 4) an algorithm for reaching relevant documents. We selected the 2006 Text REtrieval Genomics Corpus (TREC) Track[1,2], which focusses on "retrieval of passages" as well as full text documents given various questions, to demonstrate our approach. In order to avoid ambiguity, we use the term *text item* instead of document, where a text item may be a paragraph or the concatenation of paragraphs. Details of the dataset and experiments are covered in Sections III and V. Next we discuss the construction of the predications graph.

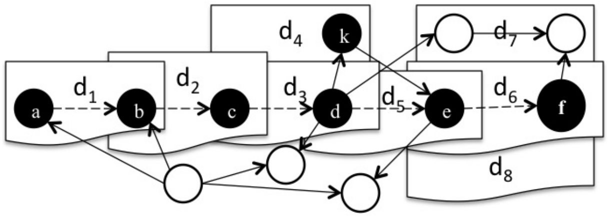

Figure 1. Document Reachability Framework

B. Predications Graph

Recall that a predication is an assertion extracted from the biomedical text. Since assertions are also triples, a predication can therefore be expressed in the canonical $subject \rightarrow predicate \rightarrow object$ form or as equally in infix form as $predicate(subject, object)$ according to first-order

[1] TREC - http://ir.ohsu.edu/genomics/2006protocol.html

[2] Experiments-http://knoesis.wright.edu/researchers/delroy/trec2006.html

logic. In either case, the predicate expresses a relationship the subject and object. Consequently, a text item can therefore be represented both in terms of its natural language content as well as a set-of-predications using formal notation. Using set notation, let $S(d_i)$ be the set of predications associated with text item d_i. If t denotes a predication and D is the set of text items $\{d_1, d_2 \ldots, d_n\}$ then,

$$\text{For any } t = (s_t, p_t, o_t), \text{let } D(t) = \{d \mid t \in S(d)\} \qquad (1)$$

be the set of corresponding text items that contain the predication t and $S(D)$ be the set of all predications associated with text items in D. That is,

$$S(D) = \bigcup_{i=0}^{|D|} S(d_i). \qquad (2)$$

The predications in $S(D)$ for a text item set D naturally form a directed labeled graph denoted $G_{S(D)}$ in which the subject and object of each predication is a node and the predicate is a labeled edge from subject to object. This graph is called the *predications graph*. The ability to connect two nodes in this graph forms the basis of reachability. In terms of document reachability, text item d_i is reachable from d_j if and only if there exist predications $(s_i, p_i, x) \in S(d_i)$ and $(x, p_j, o_j) \in S(d_j)$. That is, there are predications that share an entity which plays the role of an object in the predication from d_i and the role of a subject in the predication from d_j. Therefore, a text item d_j is reachable from d_i if and only if there exists a *path* from d_i to d_j.

C. Text Item Exploration

In a real world implementation of our approach, a user would formulate a search query consisting of an initial starting concept c (e.g. "Presenilin1 Gene" or "Alzheimer's disease" for the question in Section I). The system would then return a set of relevant text items for c, without much emphasis on ranking. Each text item would be annotated with the predications it contains. Upon selecting a predication t_1, in some text item d_1 (presumably for which the start node c is the subject), the system would return the set $D(t_1)$, in which each text item directly contains the predication t_1. Upon selecting another predication t_2 in another text item d_2 the system would then return the set $D(t_2)$, in which each text item directly contains the predication t_2.

This sequence of activity is equivalent to traversing the predications graph $G_{S(D)}$ and Figure 1 captures this equivalence between an implemented system and our predication-based simulation. In a real system, this predication-based exploration would enable three important purposes:

1) Results from recent literature will be available for browsing based on the presence of selected predications within them. The more prior domain knowledge a user has, the more selective she can be in choosing the next option to traverse. Novice users can explore different predications in a breadth first manner and learn more about the domain before digging deeper.
2) Exploration of a document space becomes a more enriching experience, since the provenance of a predication enables users to readily assess the quality of information by examining the surrounding context of the text item in which the predication appears.
3) Users can achieve LBD by discovering new connections between entities based on predications in paths.

In our simulation, we selected a modified depth-first search (MDFS) algorithm (covered in Section IV) for predications graph traversal, along with various heuristics for pruning.

D. Background Knowledge and Knowledge Abstraction

The predication-based exploration outlined in Section II-C has the limitation that it heavily relies on the predications graph. Since the predications graph only consists of assertions extracted from natural language using linguistics-based techniques (using a tool called SemRep [13]), the quality of the predication extraction could be a bottleneck. Furthermore, even if the predication extraction quality is high, it is possible that the predications necessary to connect various text items may not be expressed as such in the text.

In Figure 1 for example, if documents d_3 and d_4 do not contain the predication represented by (c,d), then the previously covering path ρ_c=(a,b),(b,c),(c,d),(d,e),(e,f) decomposes into two paths: ρ_1=(a,b) of length 1, that spans only 2/8 documents (d_1, d_2), and ρ_2=(d,e),(e,f) that covers 4/8 documents (d_5, d_6, d_7, d_8). Documents d_3 and d_4 are now unreachable. To address such cases, background knowledge can be used to connect disjoint text items by providing additional knowledge from external sources. Revisiting the two fragments in Section I, fragment 2 contains another predication p_{f2}=*(chromosome 21q21, finding_site_of amyloid precursor protein gene)* and fragment 1 contains the predication p_{f1}=*(amyloid peptide 42, associated_with, Alzheimer's disease)*. However, since amyloid beta peptide is not the same concept as amyloid precursor protein gene the two fragments cannot be connected directly. It is known however, that p_{kb}=*(amyloid precursor protein gene, produces, amyloid peptide)* from external biomedical knowledge. Therefore, we can leverage this knowledge to make a logical leap from fragment 2 to fragment 1. This is an example of *knowledge abstraction* (specifically, *associative abstraction*, since only associative predicates are used in making the logical leap between two concepts). In this work we leverage associative abstractions in which two concepts are connected by an intermediate concept. That is, we show preference for scenarios in which an entity (b) connects entities (a) and (c), through a common relationship. For example, if the entity "cell transformation" is a terminal point our traversal, a concept such as "cell fusion," which is similar to cell transformation is a viable alternative. Both concepts are known to *coexist_with* cell physiology. We also leverage *hierarchical abstraction* in which two concepts are linked through hierarchical predicates including *is_a, parent_of, child_of, etc*.

III. Datasets

We conducted an evaluation of our approach using questions from the TREC 2006 Genomics Track. The entire searchable corpus for the TREC challenge has 162,259 full text documents, segmented into paragraphs using html paragraph <p></p> tags. Each paragraph has a beginning byte offset and the length in bytes of its enclosed text. As mentioned in Section II, we refer to an individual paragraph (also called a legal span) or a collection of paragraphs belonging to the same document as a text item. We extracted 12,641,116M (M=million) such paragraphs from the entire corpus. In the subset of the 1,381 answer documents for the 26 questions, we obtained 121,162 paragraphs. We selected only this subset of answer documents for our simulation, to avoid computational limitations.

We performed two experiments (detailed in Section V). In the first experiment, we constructed a single predications graph (using the *Jung Java*[3] library), containing all predications from all text items in the subset. This experiment measured the ability to reach the documents in any given answer set, without regard for the actual answer paragraphs individually. To improve the running time of the DFS traversal, we represented each of the 1,381 answer documents as the concatenation of only their answer paragraphs instead of using all paragraphs in the document. This predications graph contained more than 13,000 unique predications, 2,105 vertices and 16,942 edges. Nearly 240 documents however, yielded no predications, as SemRep [13] either could not parse them or they did not contain any predications. In the second experiment, we created 26 separate predications graphs; one for each question. This experiment assessed the ability to reach the 3,461 correct paragraphs within the 1,381 answer documents. Evidently, some paragraphs answer multiple related questions.

We selected the biomedical knowledge repository (BKR) as the external knowledge base for associative and hierarchical knowledge abstractions. The BKR contains more than 8M relations from the UMLS Metathesaurus and over 13M SemRep predications, extracted from the abstracts of more than 18.5M biomedical documents published between 2000$-$2010.

IV. MDFS Algorithm

As outlined in Section II, the overall approach to the reachability experiments aims to traverse the predications graph using DFS, exploring every edge, and aggregating the set of answer text items that have been reached at each step. Traversal continues until all answer text items have been found or the predications graph traversal terminates on some base condition (discussed below).

The algorithm is as follows: for a given question Q and some starting point c in the predications graph $G_{S(D)}$ and an text item answer set $\mathcal{A}_Q$ for the question, Algorithm 1 recursively visits each predication in the set. If $E(c)$ represent the predications whose subject is c. That is:

$$E(c) = \{p : p \text{ is a predication with } c \text{ as the subject}\}.$$

The algorithm visits each predication in $E(c)$ in steps 2-4.

Algorithm 1 MDFS(Concept c, Set $\mathcal{A}_Q$, Graph $G_{S(D)}$)

```
1: c.visited := **true**
2: **for all** predications $p \in E(c)$ **do**
3:     MDFS-VISIT($p$, $\rho = ([\,], \emptyset, 0)$, $\mathcal{A}_Q$, $G_{S(D)}$)
4: **end for**
```

During the visit we record 1) the path traversed from the root node, 2) the corresponding text items found, and 3) the associated precision and recall. Thus, the second parameter ρ, in the recursive MDFS-VISIT procedure (Algorithm 2) is a PathObject with three components: ρ=(path, coveredSet, PRvalues). The parameter *path*, is the sequence of predications from the root to the edge traversed before the recursive call. *coveredSet* is the set of text items reached in current *path*. *PRvalues* are precision and recall values based on *coveredSet*, the answer set $\mathcal{A}_Q$, and total number of unique text items containing the predications in the *path*. In Algorithm 2, the predication p is added to the path component of the path object ρ in line 1. In lines 2–4, new text items are determined based on already reached answers, the answer set $\mathcal{A}_Q$, and the set $D(p)$ (from Eq. (1)). The covered text items set is updated. In lines 5–7, if all answers are found, the algorithm terminates and stores the path object. In lines 8–9, nodes already explored or being explored are avoided. The algorithm marks the object of p (line 10) as visited before finding successors of predication p denoted S_p in line 11. Predications successors are obtained by selecting all predications from $G_{S(D)}$ that contain the object of predication p as their subject. In steps 12–14, if no successors exist for p in $G_{S(D)}$ or no new answer text items are found, then the algorithm resorts to abstraction (see Section II-D). In lines 15–17, each predication successor is recursively visited. Finally, after a node is completely explored, the path from the root up to the current node is recorded in line 18. We discuss the two experiments to which this algorithm was applied in the following section.

V. Experimental Results

Since the output of the reachability algorithm is a set of paths $\mathcal{P}_Q$, there is no single ranked list of documents. Instead, for each question there is an optimal path containing some number of relevant documents with some best precision and recall. In the case of precision, we measure at each

[3] Jung Java Library - http://jung.sourceforge.net/

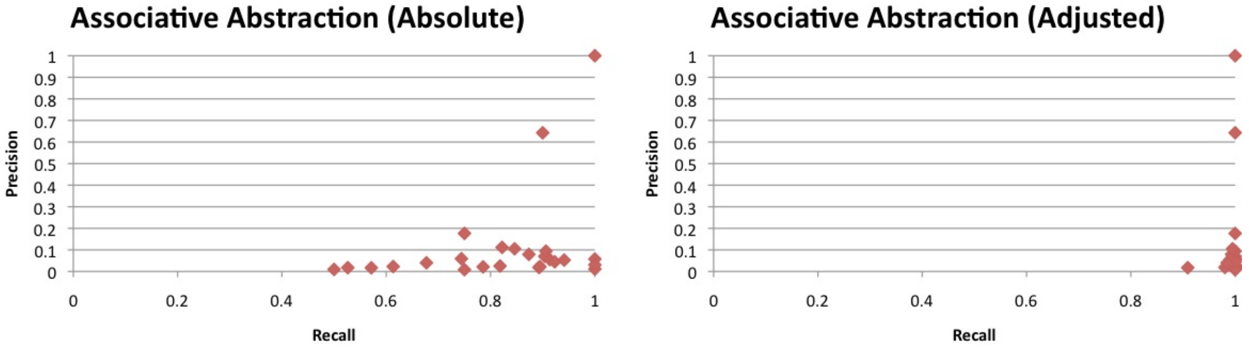

Figure 2. Precision-Recall using Associative Abstraction

Algorithm 2 MDFS-VISIT(predication p, PathObject ρ, Set $\mathcal{A}_Q$, Graph $G_{S(D)}$)

```
add predication $p$ to $\rho$.path
$newAnswers := (\mathcal{A}_Q - \rho.coveredSet) \cap D(p)$
Update $\rho.PRvalues$ for $\rho.path$ using $newAnswers$
$\rho.coveredSet := newAnswers \cup \rho.coveredSet$
**if** $(\rho.coveredSet = \mathcal{A}_Q)$ **then**
    Add $\rho$ to $\mathcal{P}_Q$ and **return**
**end if**
Let $p_o$ be the object of $p$.
**if** $(p_o.visited =$ **true**$)$ **then return end if**
$p_o.visited :=$ **true**
Get successor predications $S_p := E(p_o)$.
**if** $(newAnswers = \emptyset$ OR $E(p_o) = \emptyset)$ **then**
    Obtain successor predications $S_p$ by knowledge abstraction and record abstractions in $\rho.path$.
**end if**
**for all** predications $q \in S_p$ **do**
    MDFS-VISIT($q$, $\rho$, $\mathcal{A}_Q$, $G_{S(D)}$)
**end for**
Add $\rho$ to $\mathcal{P}_Q$
```

hop, the number of correct text items retrieved relative to the total number of unique text items retrieved. For recall, we measure at each hop, the ratio of the total number of correct text items retrieved to the total number of correct text items to be retrieved per question. More formally:

1) Precision: For some arbitrary question Q, precision is the ratio of the total number of correct text items $\mathcal{D}^c_{Q,l}$ retrieved from the answer set $\mathcal{D}_Q$ at path length l, to the total number of unique text items $\mathcal{D}_{u,l}$ retrieved at path length l.

$$Precision = \frac{\sum_{l=1}^{\mathcal{L}} \mathcal{D}^c_{Q,l}}{\sum_{l=1}^{\mathcal{L}} \mathcal{D}_{u,l}} \qquad (3)$$

2) Recall: For some arbitrary question Q, recall is the ratio of the number of correct text items $\mathcal{D}^c_{Q,l}$ retrieved from the answer set $\mathcal{D}_Q$ at path length l, to the total number of text items in the answer set.

$$Recall = \frac{\sum_{l=1}^{\mathcal{L}} \mathcal{D}^c_{Q,l}}{\mid \mathcal{D}_Q \mid} \qquad (4)$$

Since many text items that answer arbitrary questions could not be reached altogether, due either to limitations in predication extraction of absence of predication in some text items, we introduce an adjusted recall measure to better assess the effectiveness of our approach. The adjusted recall ($Recall_{adj}$) discounts text items $\mathcal{D}_{ur}$ that cannot be reached from the total number of text items in the answer set. The denominator in Equation(4) therefore becomes $\mid \mathcal{D}_Q \mid - \mid \mathcal{D}_{ur} \mid$. Additionally, as the MDFS algorithm is exhaustive, multiple paths of varying coverage can be generated for each question. To obtain a best case measure for precision and recall, we selected for each question, the path of shortest length that covers the greatest number of text items. Intuitively, if an expert user traversed such a path, it would yield the greatest number of answer text items in the fewest number of transitions.

A. Experiment 1

In the first experiment* a text item is the concatenation of all answer paragraphs in a document. The goal of this experiment was to ascertain the ability to reach answer text items without necessarily identifying the correct paragraphs within them. Table I (Row 1) shows precision and recall using no abstraction and *concept-based* text item retrieval. That is, at each hop, we retrieve only those text items containing the subject of the predication. Table I (Row 2) shows the precision and recall statistics using associative abstraction. Here we observe that the use of associative abstraction yields high recall, while precision is low. The overall 82% recall for associative abstraction, compared with close to 70% for no abstraction and 71% for hierarchical abstraction suggests that entities occurring in natural language

*No entry points exist in the predications graph for two (2) questions (Q166, & Q187) for Associative Abstraction and Hierarchical Abstraction in this experiment. Additionally, no entry points for Q177 and Q184 exists when abstraction is not used abstraction.

Table I
EXPERIMENT 1: AVERAGE PRECISION AND RECALL BY TECHNIQUE

Technique	Precision	Recall	
		Absolute	Adjusted
No Abstraction (NA)	0.205	0.707	0.846
Associative Abstraction (AA)	0.112	0.822	0.994
Hierarchical Abstraction (HA)	0.120	0.718	0.879

Table II
EXPERIMENT 2: AVERAGE PRECISION AND RECALL BY TECHNIQUE

Technique	Precision	Recall	
		Absolute	Adjusted
No Abstraction(NA)+Predication(P)	0.485	0.165	0.225
NA+P+Question Concepts(QC)	0.531	0.157	0.215
Associative Abstraction(AA)+P	0.317	0.271	0.377
AA+P+QC	0.405	0.234	0.330
Hierarchical Abstraction(HA)+P	0.497	0.174	0.239
HA+P+QC	0.543	0.166	0.230

are more frequently connected through associative predicates rather than hierarchical ones. While intuitively hierarchical predicates may be contextually relevant, the lower frequency in their usage appears to limit reachability. Figure 2 also shows pictorially that many text items are reachable using associative abstraction, but at low precision. In spite of low precision, this result confirms that answer text items can be connected using the predications.

B. Experiment 2

In the second experiment*, each paragraph is treated as an individual text item. Table II (Row 1) contains statistics for the first scenario, in which we retrieved text items based on the presence of predications in them and without using abstraction, as outlined in Algorithm 2. This result shows that while about 49% of the text items retrieved were correct, only 16.5% of all correct text items were retrieved. While the adjusted recall is higher, at 22.5% (as expected), still a large number of correct text items have not been retrieved. Such a result likely for two reasons: 1) the absence of predications in text items and 2) the absence of *direct* connections from one correct text item to another, since no abstraction is used.

In an attempt to prune the number of non-relevant text items, we added the condition that text items at each hop must also contain at least one entity from the original question in addition to the predication. Table II (Row 2) shows that for this second case we achieve higher precision (up to 53.1% from 48.5%) among the text items retrieved, but at the cost in recall, which falls from 16.5% to 15.7% We speculate that this loss is because some relevant answer text items contain related question entities instead of exact matches or synonyms. Hence, such answer text items would no longer be reachable under the constraint.

In the third case, in Table II (Row 3), we attempted to improve recall by performing associative abstraction. This created a significant increase in recall, from less than 17% to around 27% in absolute recall (and close to 38% adjusted). The accompanying decrease in precision is likely because the abstractions create links to related predications that occur in many text items in a related context, but which are ultimately not in the answer set. Table II (Row 4) shows once again that adding the question-entity constraint improves precision (from 31.7% to 40.5%) but lowers recall (down from 27.1% to 23.4%) possibly for reasons discussed above.

Similar trends are evident after applying hierarchical abstraction shown in Table II (Row 5). A slight improvement in recall over no abstraction can be observed (i.e., from 16.5% and 15.7% to 17.4%). However, applying the question-entity constraint shows a decrease in recall in Table II (Row 6), from 17.4% to 16.6%. This gives rise to an an expected increase in precision (from 49.7% to 54.3%). Figure 3 shows pictorially the individual absolute and adjusted precision and recall for the scenario described in Table II (Row 1). Five questions have above 90% recall but for all others, less than half the correct number of text items were retrieved.

Collectively our results highlight some important points. First, they indicate that ambiguity in written language imposes a major bottleneck on knowledge-driven exploration. Second, they establish that limitations in predication extraction methods further affect the accuracy and scope of our approach. Finally, the quality of background knowledge may also be a limiting factor when sufficient context does not exist or cannot be distilled from the knowledge base. Such difficulties are particularly evident when searching for fine-grained text items. It may be the case that finding answer text items at this level of granularity is inherently difficult. To support this claim, it is noteworthy that the Mean Average Precision (MAP) values reported by the TREC 2006 challenge [14] are also rather low.

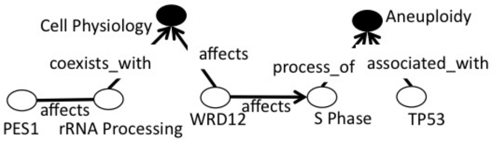

Figure 4. Predications in a Sample Path for Question 184

Nonetheless, experiment 1 shows that our knowledge-driven approach is more applicable for text items of coarser granularity. It shows that ideally, an expert user might be able to leverage the predications to transition from document to document without the need for query reformulation. Concurrently however, it raises the fundamental question of whether such paths make sense. Figure 4 shows one path that does. For the question, *How do mutations in the Pes*

*No entry points (or root nodes) exist in the predications graph for three (3) questions (Q166, Q184 & Q187) in this experiment. Results are listed for the remaining 23 questions.

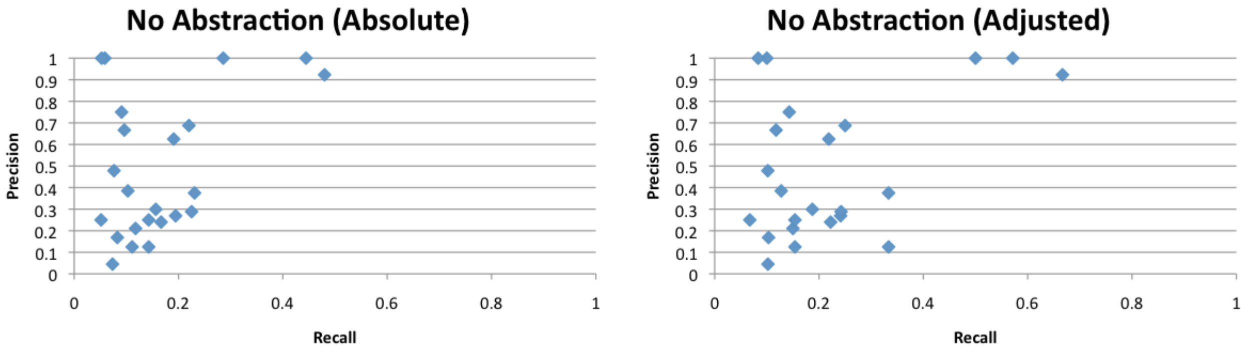

Figure 3. Precision-Recall using No Abstraction

gene affect cell growth? from the literature we observe that:

"*The contribution of pescadillo to the regulation of cell growth was further substantiated by the identification and characterization of temperature-sensitive pescadillo mutant yeast strains. Yeast expressing mutant pescadillo displayed growth arrest in the G1 or G2 phase of the cell cycle when shifted to a nonpermissive temperature.*" *(Source: PMID11071894)*

The path in Figure 4 shows that *Pescadillo affects rRNA Processing* which may affect the *S Phase*, since the *WDR12 Gene affects Cell Physiology* with which *rRNA Processing* coexists. However, the *S Phase* or synthesis phase of the cell cycle is also a process_of *Aneuploidy* which has negatively effect on cell growth. Since *DNA replication* is known to occur between he *G*1 and *G2 phases* of the *S Phase*, mutations of *Pescadillo* could lead to *G1 Phase arrest* thereby showing growth arrest. By confirmation in the text, a specific temperature sensitive mutant experiences this condition.

VI. Related Work

Many recent advances in keyword-based retrieval emerged from the annual Text REtrieval Conference (TREC). Many state-of-the-art algorithms from TREC conferences use sophisticated heuristics to expand initial user queries either by using a background knowledge base (KB), such as WordNet or UMLS, or by employing pseudo feedback [15]. While these improvements in query expansion lead to more effective document retrieval, their overall *pertinence* [16] to the larger information need is arguable. That is, relevance of a document to a search query does not often readily translate to answers that satisfy the visceral information need of the user. Therefore, the result of these search approaches is typically a long list of relevance-based ranked documents that still leaves users with the task of searching-and-sifting through them to find answers.

PubMed[4] is a web-based tool for searching the MEDLINE database of biomedical citations (most include abstracts) from more than 5000 journals, developed and maintained by the National Center for Biotechnology Information (NCBI). It returns a ranked list of results using synonym-based query expansion with a bias towards recent results and inherits the disadvantages of keyword-based search. To address this, in searching biomedical literature, well known KBs are used to cluster results into categories and to provide faceted drill-down based on hierarchies of categories. For example in GoPubMed [17], the GO has been used effectively to annotate entities in PubMed abstracts and filter them with GO hierarchy concepts as facets. XplorMed [18] and McSiBy [19] are two other systems that use MeSH classifications to cluster search results. The KBs used in these approaches like the GO or MeSH headings are fairly static and represent well known and widely accepted knowledge in biomedical domains and do not encompass recent findings and cutting edge research. While they alleviate the burden of perusing long result lists, they do not directly address the pertinence problem discussed earlier when seeking information to a particular question. In this paper we propose using semantic predications extracted from scientific literature as the backbone that guides exploratory search.

In ongoing research, we developed Scooner [20], [21], which is a concept-based, user-driven approach to exploration. This approach is predicated on the view of a document, not only as a *bag-of-words* both also as a *set-of-entities*. Background knowledge is then used to connect the entities anchored in text to support document to document transition. Unlike Scooner, this work uses predications instead of entities, largely because predications capture more relevant context than connecting documents based on entities.

VII. Conclusion

In this paper, we presented a novel knowledge-driven framework, that provides an alternative to traditional QA approaches for finding documents that answer complex information needs. Our approach is further novel because it also has implications on LBD. Through the use of semantic predications we showed that a mechanism for transitioning

[4] http://www.nlm.nih.gov/pubs/factsheets/pubmed.html

from one document to another can be achieved. Specifically, we showed that assertions extracted from biomedical literature can achieve high recall in retrieving relevant text items at coarse granularity. The use of hierarchical and associative abstraction using background knowledge can further improve recall but at a loss in precision. This loss likely stems from predications obtained through abstraction, that occur in many documents but are not of critical relevance to answer set documents. We plan in future, to develop techniques to increase precision by ranking and pruning both predications and text items.

We also showed that it is difficult to connect text items that provide exact answers to specific questions with high recall. This is largely because the predications necessary to connect various text items may not be present in the natural language text, for a variety of reasons. Such reasons include ambiguities in written language as well as nuances in expression. Additionally, the limitations of SemRep (which has 78% precision and only 50% recall) negatively impacts the recall in our experiments.

Ultimately, our focus in this work was on determining whether documents that answer complex information needs can be reached from one another, given the absence of hyperlinks. We showed that exploration among such documents is possible by leveraging semantic predications and knowledge abstractions through the use of background knowledge. In future, we will address issues of 1) ranking, 2) quality of paths, 3) ease of use of an implemented, and 4) the scalability of the underlying algorithms, for large datasets.

Acknowledgment

This work was funded by the National Library of Medicine (NLM) in the Summer 2010 and the HPCO project at the Kno.e.sis Center. Thanks to the initial efforts of Cartic Ramakrishnan, Michael Cooney, Alan Smith and Paul Fultz II. Thanks also to Jeffrey Ali Hyacinthe, May Cheh, John Nguyen, Mohammed Cyclegar, Dongwook Shin and Thomas C. Rindflesch for their support during my internship.

References

[1] D. Swanson, "Fish oil, raynaud's syndrome, and undiscovered public knowledge." *Perspect Biol Med*, vol. 30, no. 1, pp. 7–18, 1986.

[2] D. R. Swanson, "Migraine and magnesium: eleven neglected connections." *Perspect Biol Med*, vol. 31, no. 4, pp. 526–557, 1988.

[3] A. Broder, "A taxonomy of web search," *SIGIR Forum*, vol. 36, pp. 3–10, September 2002.

[4] V. Bush, "As we may think," *The Atlantic Monthly*, vol. 176, no. 1, pp. 101–108, 1945.

[5] A. P. Sheth and C. Ramakrishnan, "Relationship web: Blazing semantic trails between web resources," *IEEE Internet Computing*, vol. 11, no. 4, pp. 77–81, 2007.

[6] E. Pafilis, S. I. O'donoghue, L. J. Jensen, H. Horn, M. Kuhn, N. P. Brown, and R. Schneider, "Reflect: Augmented Browsing for the Life Scientist," *Nature Preceedings*, May 2009.

[7] A. D. Eaton, "HubMed: a web-based biomedical literature search interface," *Nucleic Acids Research*, vol. 34, pp. W745–W747, Jul. 2006.

[8] H. Kilicoglu, M. Fiszman, A. Rodriguez, D. Shin, A. M. Ripple, and T. C. Rindflesch, "Semantic medline: A web application to manage the results of pubmed searches," in *3rd Intl. Symp. for Semantic Mining in Biomedicine*, 2008, pp. 69–76.

[9] M. A. Hearst, *Search User Interfaces*. Cambridge University Press, 2009.

[10] N. R. Smalheiser and D. R. Swanson, "Linking estrogen to alzheimer's disease: An informatics approach." *Neurology*, vol. 47, no. 3, pp. 809–810, 1996.

[11] D. R. Swanson, "Undiscovered public knowledge." *Library Quarterly*, vol. 56, no. 1, pp. 103–118, 1986.

[12] Wikipedia, "http://en.wikipedia.org/wiki/reachability," 2010.

[13] T. C. Rindflesch, M. Fiszman, and B. Libbus, "Semantic interpretation for the biomedical research literature," in *Medical Informatics: Knowledge Management and Data Mining in Biomedicine*, 2005, pp. 399–422.

[14] W. Hersh, A. Cohen, P. Roberts, and H. Rekapalli, "TREC 2006 Genomics Track Overview," 2006.

[15] A. Singhal and M. Kaszkiel, "A case study in web search using trec algorithms," in *Proceedings of WWW'01*, 2001, pp. 708–716.

[16] P. Borlund, "The concept of relevance in ir," *Journal of the American Society for Information Science and Technology*, vol. 54, no. 10, pp. 913–925, 2003.

[17] H. Dietze, D. Alexopoulou, M. Alvers, L. Barrio-Alvers, B. Andreopoulos, A. Doms, J. Hakenberg, J. M´onnich, C. Plake, A. Reischuck *et al.*, "Gopubmed: Exploring pubmed with ontological background knowledge," *Bioinformatics for Systems Biology*, pp. 385–399, 2009.

[18] C. Perez-Iratxeta, P. Bork, and M. Andrade, "XplorMed: a tool for exploring MEDLINE abstracts," *Trends in biochemical sciences*, vol. 26, no. 9, pp. 573–575, 2001.

[19] Y. Yamamoto and T. Takagi, "Biomedical knowledge navigation by literature clustering," *Journal of Biomedical Informatics*, vol. 40, no. 2, pp. 114–130, 2007.

[20] D. Cameron, P. N. Mendes, A. P. Sheth, and V. Chan, "Semantics-empowered text exploration for knowledge discovery," in *48th ACM Southeast Conference*, 2010.

[21] R. Kavuluru, C. Thomas, A. P. Sheth, V. Chan, W. Wang, A. Smith, A. Sato, and A. Walters, "An up-to-date knowledge-based literature search and exploration framework for focused bioscience domains," in *2nd ACM SIGHIT International Health Informatics Symposium (to appear)*, 2012.