An Up-to-date Knowledge-Based Literature Search and Exploration Framework for Focused Bioscience Domains

Ramakanth Kavuluru*
Kno.e.sis Center
Wright State University
rvkavu2@uky.edu

Christopher Thomas
Kno.e.sis Center
Wright State University
topher@knoesis.org

Amit Sheth
Kno.e.sis Center
Wright State University
amit@knoesis.org

Victor Chan
Human Effectiveness Dir.
Air Force Research Lab
Victor.Chan@wpafb.af.mil

Wenbo Wang
Kno.e.sis Center
Wright State University
wenbo@knoesis.org

Alan Smith
Kno.e.sis Center
Wright State University
smith.706@wright.edu

ABSTRACT

In domain-specific search systems, knowledge of a domain of interest is embedded as a backbone that guides the search process. But the knowledge used in most such systems 1. exists only for few well known broad domains; 2. is of a basic nature: either purely hierarchical or involves only few relationship types; and 3. is not always kept up-to-date missing insights from recently published results. In this paper we present a framework and implementation of a focused and up-to-date knowledge-based search system, called Scooner, that utilizes domain-specific knowledge extracted from recent bioscience abstracts. To our knowledge, this is the first attempt in the field to address all three shortcomings mentioned above. Since recent introduction for operational use at Applied Biotechnology Branch of AFRL, some biologists are using Scooner on a regular basis, while it is being made available for use by many more. Initial evaluations point to the promise of the approach in addressing the challenge we set out to address.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search Process; I.2.7 [**Natural Language Processing**]: Text Analysis, Language parsing and understanding

General Terms

Design, Experimentation, Management

Keywords

knowledge-based systems, text mining, information extraction, domain models, hypothesis generation

*corresponding author

IHI'12, January 28–30, 2012, Miami, Florida, USA.

1. INTRODUCTION

The problem of information explosion in life science literature is making it increasingly difficult for researchers to meet their information needs. The citation count of bioscience journal articles accessible through NCBI's PubMed web service is currently over 20 million [14] and has been growing exponentially from 1985 at an annual growth rate $\approx 4\%$. On the other hand, the frontiers of biological and medical knowledge are stretched to an extent where, according to one estimate, just in the field of epidemiology it would take 21 hours per day for a physician to stay current [7]. On a daily basis, researchers need to lookup information spread across different articles to satisfy their information needs which range from gaining basic overviews of a topic to being able to correlate different results across different articles to generate new hypotheses.

A popular approach that addresses the information explosion problem is *focused* or *domain-specific* literature search where the user wants to search within a particular domain or topic of interest. One way of enabling focused search is to restrict the search to articles within a few important journals or those with specific terms in the *keywords* portion of an article metadata. Although this is better than searching the whole universe of articles, it is still primitive in the sense that all relevant keywords might not be known ahead of time. Also, the boundaries of life science disciplines are disappearing quickly and often articles pertaining to one topic are published in several different journals. This makes it is difficult to predict which journals contain relevant articles. For example, articles relating to the domain of "chemical and biological warfare agents" are published in different journals such as Infection and Immunity, Journal of American Medical Association, International Journal of Technology Assessment in Health Care, Microbiology and Molecular Biology Reviews, Journal of Chemical Technology and Biotechnology, and the Journal of Leukocyte Biology. This list keeps growing with time and thus restricting the search to a few journals is not ideal. This approach is not ideal also because important pieces of information from different journals that are not very relevant to the domain of interest might actually be the missing links that could lead to new discoveries. For example study of neurodegenerative diseases is observed to span the disciplines of psychiatry, neurology, microscopic anatomy, neuronal physiology,

biochemistry, genetics, molecular biology, and bioinformatics [18]. One approach to address this situation is to use the entire set of articles for searching but also use a domain-specific knowledge base (KB) as a guide in the search and exploration process to surface articles that are more related to the domain. This idea of using an underlying KB in the search process has been explored by many search systems built on PubMed citations [14]. We describe some of these systems and discuss their limitations in Section 2. We address these limitations with our approach in Section 3. In Section 4 we elaborate on the steps we follow to create focused knowledge bases. We present the knowledge-based search system, Scooner, with examples and preliminary user studies in Section 5. We discuss future improvements and make some concluding remarks in Section 6.

2. BACKGROUND

Here we are using the term "knowledge base" to represent some form of machine processable representation of information that models a domain. In this sense, controlled vocabularies or terminologies, taxonomies, and ontologies are all different types of KBs. Each piece of knowledge here is generally represented as a triple *subject* $\rightarrow$ *predicate* $\rightarrow$ *object* that connects two entities subject and object with a predicate (or relationship type). For example *dopamine* $\rightarrow$ *is_a* $\rightarrow$ *neurotransmitter* and *dopamine* $\rightarrow$ *modulates* $\rightarrow$ *brain_plasticity* are two triples with *dopamine* as the subject. An ontology is usually characterized by concept hierarchies and inter-concept relationships and constitutes the well accepted and consensual knowledge of a domain; and the instance base of concepts and inter-instance relationships in a domain often constitute the KB that is built around an ontology. The Gene Ontology (GO) [8] is the most cited KB with over 450 PubMed citations per year [2]. Medical Subject Headings (MeSH), Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT), and the Unified Medical Language System (UMLS) are other popular terminologies that are frequently cited in the literature. KBs not only serve as standalone references of the basic knowledge pertaining to a domain, but also assist in tasks such as knowledge management, data integration, decision support, information extraction, and literature search and exploration. Here we are more interested in approaches that use KBs for facilitating effective search and exploration of bioscience literature. For example in GoPubMed [5], the GO has been used effectively to annotate entities in PubMed abstracts and filter them with GO hierarchy concepts as facets. XplorMed [16] and McSiBy [24] are two other systems that use MeSH classifications to cluster search results. There are also other systems that use synonym based query expansion and rank search results based on analysis of a given set of 'query' documents (See [14] for an informative survey).

Because the construction of high quality KBs such as the GO is time consuming and requires significant human expert involvement, it is becoming increasingly important to explore computational techniques to automate it to keep pace with the large quantities and the dynamic nature of new information. The expert-curated standard KBs are also mostly static and do not cover emerging knowledge from recently published results; granted such results might not be well accepted by the community. Nevertheless, they are crucial in enabling knowledge-based search systems that facilitate literature-based hypothesis generation and knowledge discovery. That is, although manually built ontologies are of high quality (or precision), their automatic counterparts offer better coverage for effective information retrieval. We realize that manually created KBs are indispensable in knowledge management and data integration. Here we are only concerned with their effectiveness in facilitating knowledge-based search and exploration. Many of the popular KBs generally have few relationship types (eg., *is a* or *part of* in GO), which limits their utility in the search process to merely cluster results while our goal is to provide interesting new knowledge that might lead the user to interesting trails of relationships (more on this later). Finally, most of the current KBs are specific to a set of well known broad domains of interest. However, current research trends point to narrower specializations in the future and it is not clear from the state-of-the-art how one would go about building domain-specific KBs for arbitrary domains for use in knowledge-based search systems. For example, neither the NCBO ontologies[1] nor the NCBI databases contain significant information related to *biological warfare agents (BWA)*. BWA occurs as a leaf concept (0 children) in the MeSH vocabulary; it's parents include *Weapons of Mass Destruction* and *Weapons*. Even using the Metathesaurus is also not very helpful. For example, it takes a path of length at least five when starting at the concept *biological warfare agents* to get to an example agent *Francisella Tularemis*, a popular agent currently being studied by our collaborators in life sciences. Thus it is not clear how one can extract a KB for a specialized domain from existing expert-curated KBs. Next we outline our approach to address these shortcomings.

3. APPROACH AND OUTLINE

Motivated by the reasons detailed in Section 2, we started out with two major objectives: 1. Build an up-to-date KB based on a given *user-specified* focused domain 2. Build a knowledge-based search system that uses the KB in the first step as a backbone to facilitate KB-enhanced search and discovery. Our final goal is to provide superior (both in quality and speed) search and retrieval over scientific literature for life scientists that will enable them to elicit valuable information in a focused domain. Our use-case for the rest of the article is the domain of **"human performance and cognition"** (HPC) and our efforts are funded by the human effectiveness directorate of the Air Force Research Lab (AFRL) that hosts biologists interested in the HPC topic.

The first hurdle is to find a way of letting users *specify* a focused domain for our framework to create the KB. We handle this by letting users provide keywords that represent important concepts in their domain of interest. This initial seed set of keywords is input to the framework, which has the following main components.

C1 The input seed set of keywords is used to computationally carve an initial hierarchy of concepts related to the domain from Wikipedia based on its link structure and the category hierarchy on top of its articles. Concepts from this hierarchy are then mapped to any popular related expert-curated KBs. Once this initial model is approved by domain experts, it is passed on to the component C2. With C1, we facilitate a way of choosing an arbitrary domain of focus for the search system. In the

[1]http://www.bioontology.org/

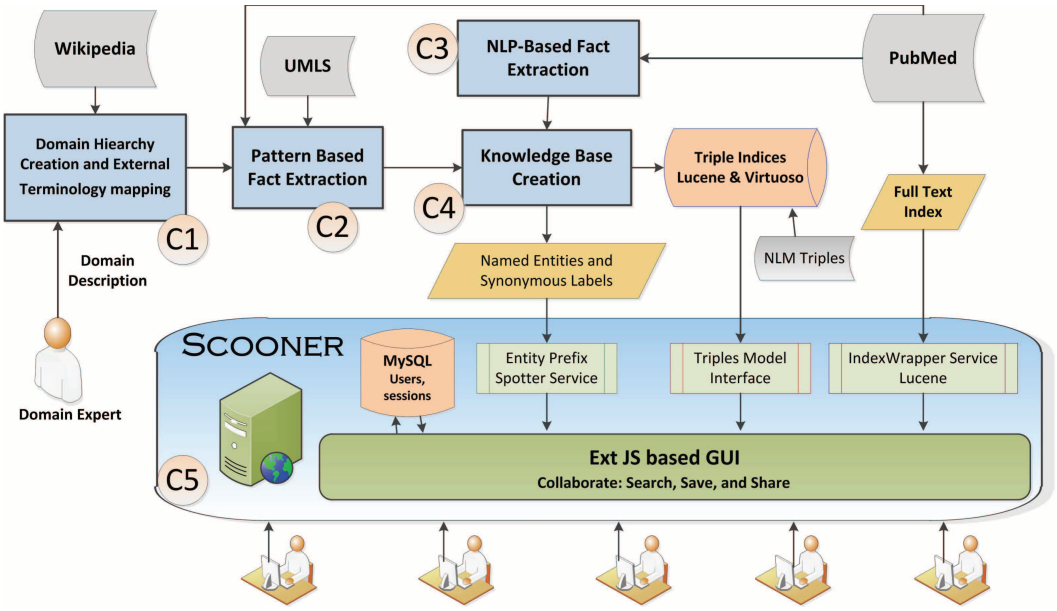

Figure 1: Up-to-Date Knowledge-Based Search and Exploration Framework

next two components, two complementary approaches are used to add new knowledge related to the domain represented by the hierarchy created here.

C2 Here the concepts in hierarchy obtained in C1 are connected with any possible relationships that exist between them. This is done using a supervised information extraction technique based on lexical patterns that represent relationships in natural language. A training set of triples is used to extract a set of candidate patterns that represent the manifestation of certain relationships in free text. Then the pattern frequencies are used to infer most probable relationships (from the training set) between unseen concepts, here, the concepts from the hierarchy created in C1.

C3 Independent to the domain-specific components C1 and C2, natural language processing (NLP) based techniques are used to facilitate enhanced information extraction. New PubMed abstracts are parsed periodically using the Stanford parser [4] and triples are extracted from the parse trees using specific heuristics based on dependency graphs output by the parser.

C4 Triples extracted from NLP and pattern based techniques in C2 and C3 are mapped to the hierarchy built in C1. This is done using straightforward stemming and string-matching based concept-inclusion heuristics. This results in the final KB for the user specified domain. With C2, C3, and C4, we incorporate triples involving more interesting non-hierarchical relationships between entities. We also keep the KB up-to-date by the periodic extraction of triples from recently released PubMed abstracts.

C5 While the first four components are the foundation, this component embeds the KB created in the fourth step into a knowledge-based search system called Scooner[2], that provides search over PubMed abstracts and allows exploration of the literature via the KB triples. The associated entities of the triples found in the abstracts resemble hyperlinks and essentially let the users browse free text as if they were browsing hypertext. The search process and collaborative extensions are elaborated and preliminary user studies are discussed in Section 5.

All the components of our framework can be summarized as shown in Figure 1. Next we elaborate on each of these components with details of techniques, rationale for selection, evaluation approaches, and examples.

4. KNOWLEDGE BASE CREATION STEPS

When discussing each component, we point out other related efforts. As can be observed, each component is an independent research problem and we build upon our earlier efforts to construct the focused KBs. For brevity and to allow for space constraints, we only give overviews of our efforts with citations to detailed manuscripts.

C1 Domain Hierarchy Creation: The Wikipedia corpus contains a vast category graph on top of its articles and a study [12] shows that Wikipedia is the most sought online resource for *basic* medical information. It is, however, determined [3] to be only a good starting point and that is exactly how we use Wikipedia in our framework. Note that for purely medical purposes recent community efforts like MedPedia[3] can also be used.

Although the articles' content might not be accurate, they, nevertheless, capture the important concepts in a domain of interest by mentions in hyperlinked text. Our original

[2]SCOONER – Semantically Connected Named Entities and Relationships. For a screencast of Scooner, please visit: `http://archive.knoesis.org/library/demos/scooner-demo/`

[3]MedPedia started in 2007 and is growing. It still has many stubs instead of full articles for several topics. Furthermore, the text in MedPedia is not hyperlinked as in Wikipedia.

work [20] on this subtopic takes advantage of the hyperlinked structure and the underlying category hierarchy of Wikipedia articles to build a hierarchy of concepts given an input set of keywords. Besides the set of keywords that describe the domain, we also take as input a smaller set of keywords that describe the broader focus domain for the hierarchy. For example, Dopamine is both a neurotransmitter and the name of a 2003 film. We do not want to consider neighbors of the film article but only from the neurotransmitter article. Similarly, the biological warfare pages also link to various officials involved in warfare issues and facility locations in the US, which might not come under the broader focus domain of harmful bacteria, viruses, fungi, and toxins. We build the hierarchy by using an "expand and reduce" paradigm that allows us to first explore and exploit the concept space before reducing the concepts that were initially deemed interesting to those that are closest to the actual domain of interest.

Expansion: In this phase, first a few keywords describing the focus domain are used to query Wikipedia. The seed set of a predefined number of top ranked articles returned from this search are input into the second step. The seed set of articles is then expanded to a larger set of articles by including "similar" neighboring articles that link to or are linked from the seed set of articles. We determine the similarity between two articles based on a weighted sum of similarity between their shared in-neighbors and out-neighbors. The weighting scheme we used in similarity computation is empirically determined based on the Wikipedia link types such as regular, "common category", or "see also". This link-based similarity measurement is based on the intuition that articles that share a significant number of similar neighbors through various types of hyperlinks are more tightly linked compared to those linked via isolated hyperlinks. This notion of similarity is described in [13] and extends the SimRank similarity measure [11] by assigning different weights to different types of links. The names of the articles in the expanded set represent the concept space. Finally, a category hierarchy is imposed that initially copies the original Wikipedia categories starting at every term in the set to the root node.

Reduction: The set of concept terms from the expansion phase is taken as input for this step. For each term in the list of extracted terms, the conditional probabilities $p(Term|Domain)$ and $p(Domain|Term)$ that describe the importance of each term for the domain and in the domain, respectively, are computed. Knowing both measures is important for the subsequent use of the created domain model in probabilistic document classification tasks and for threshold-based pruning of the concept set in the hierarchy. After pruning, leaf categories that are empty are deleted recursively. All categories outside the broader focus domain are also deleted.

We mine the Wikipedia article link graph and compute inter-article similarities for shared neighbors of each article to be able to find similarity between any two given articles. Some manual pruning at the end might be required as was observed in the case of our HPC hierarchy. The top few classes of the hierarchy are shown in Figure 2. The maximum depth of the hierarchy is 16 levels. There are a total of

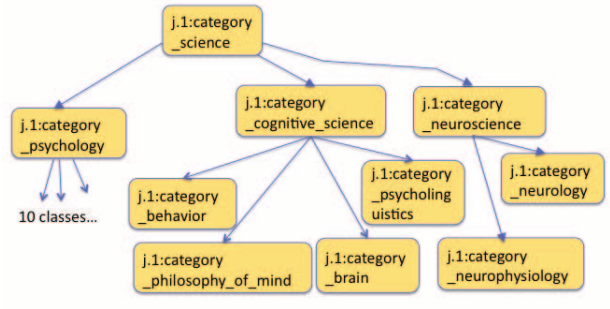

Figure 2: Top Categories in HPC Hierarchy

905 named classes and the average number of children for a class is 15 while the average number of parents per class is 3. In Wikipedia, anchor texts of hyperlinks are often indicators of synonymous labels for the original concept the links point to. Using relative frequencies of these anchor labels among all articles they link to, we also aggregate lists of synonymous labels for each concept. To further increase recall, we choose expert-curated ontologies that are related to the domain and map concepts in the hierarchy to those in the selected ontologies using exact string matching. For the HPC domain, we used the Yale University SenseLab Neuroscience ontologies (`http://neuroweb.med.yale.edu/senselab/`).

C2 Pattern-Based Fact Extraction: Once the hierarchy is available from C1, we connect its concepts with any non-hierarchical (or associative) relationships that might hold between them. Extracting meaningful triples from free text has been a significant challenge in biomedical informatics. We explored two options to populate the hierarchy with non-hierarchical relationships. The first of these is a supervised statistical pattern mining algorithm that is based on the intuition that relationships manifest in free text by occurring with certain lexical patterns. For instance, relationships involving the predicate *is_a* often manifest with the pattern "x such as y". In initial attempts along these lines, a set of hand coded patterns were used to identify *is_a* relationships [9]. Later pattern based approaches for general types of relationships were developed [1, 6, 23]. Most of these systems target news articles or Web pages for relations such as "Company *located_at* Headquarters", "Player *plays_for* Team", or "Person *born_in* City". In the biomedical area, pattern based approaches have been mostly used to extract very specific types of relationships such as protein-protein interactions [15]. Our current work [21] is inspired by Turney's [22] vector space methods and is based on frequency distributions of automatically extracted patterns. We used the UMLS Metathesaurus for the seed set of around 4100 input triples and 84 predicates for relationship types. Concepts pairs in the training triples are used to obtain lexical patterns that are further generalized using wild-card tokens to make a relationship-pattern matrix analogous to term-document matrices commonly used in information retrieval applications. Then the relationship between a concept pair is determined by a Bayesian network that takes into account the conditions: 1. how well do the terms found in text indicate the concepts in the pair? 2. how well is a relationship indicated by the patterns found between the terms? and 3. how likely is it that this type of

relationship occurs with the concepts? After subjecting to a threshold, the relationship type with the maximum similarity value is chosen as the predicate for the new concept pair. Employing random 3:2 splits of our fact corpus for training and testing, we obtain a precision $\approx 80\%$. We note that as long as there are enough number of facts (≥ 25 in our experiments) for any relationship type, new triples that participate in that relationship can be found given access to a representative text corpus.

C3 NLP-Based Fact Extraction: While the statistical pattern-based approach helps determine relationships between two given concepts, it does not cover biomedical entities and relationships of a more complex nature that might not be in the hierarchy created in C1. For example consider the triple "haloperidol *accelerates* dopamine synthesis and metabolism". The object of the triple " dopamine synthesis and metabolism" is what we call a complex entity, usually a noun phrase that represents a biomedical concept. While the hierarchy contains 'dopamine', 'metabolism', as concepts, it does not have 'synthesis' and in particular this complex entity that characterizes metabolism and synthesis of dopamine. Since C2 only connects concepts already present in the hierarchy, interesting triples such as the one discussed here might not be covered through C1–2. Also the set of nearly 600 predicates in the UMLS Metathesaurus (2010 release) does not include interesting predicates such as *increase, decrease, modulate, stimulate*, and also *accelerate* that is used in the example triple here. From a manual inspection of the 600 predicates, other than 'regulates' and 'affects', we do not see any that come close to what these predicates mean. That is, a number of interesting predicates expressed in the text do not have interesting training facts for C2. Hence, supervised extraction does not suffice to cover complex entities and interesting predicates. So in this component, we use an NLP-based "open extraction" approach that will help us increase the recall by capturing more interesting triples. In the open extraction approach, both the predicates and concepts do not come from predetermined sets, they rather emerge as they are encountered when NLP techniques are applied to free text. However, once identified, these entities and the associated triples can be later filtered to curate a subset that is interesting to a domain of interest.

We used the Stanford parser [4], a probabilistic CFG-based parser, for parsing all sentences of PubMed abstracts available as of August, 2008. Our work [17] uses heuristics based on long range dependencies, among different terms in sentences, that are obtained from the dependency graphs [4] output by the Stanford parser for each sentence. Dependencies are grammatical relations expressed as triples between words that are not necessarily adjacent in the sentence parsed. Consider the example sentence: "These results demonstrate that in the septum NMDA receptors tonically activate GABAergic neurons which in turn inhibit the cholinergic septohippocampal neurons." From the parser output we see dependency relations such as *amod* (neurons, GABAergic), *advmod* (activate, tonically), *nn* (receptors, NMDA), and *nsubj* (activate, receptors); here *amod* is the adjectival modifier stating that the adjective GABAergic modifies the head noun 'neurons'. Similarly *advmod* stands for the adverbial modifier and *nn* is for the noun compound modifier. Note that the nominal subject dependency *nsubj*(activate, receptors) connects non-adjacent words. The various types of dependencies form a hierarchy and the dependency triples for any given sentence also form a directed graph. Thus dependencies are a preliminary mechanism to capture long range dependencies. We then use heuristics involving the modifier, complement, and preposition classes of the dependency hierarchy to extract compound entities and triples from the parsed output. We discard entities that are longer than five words to avoid noun phrases that are too complicated. On a set of five most frequent predicates, we report an accuracy of 68% in triple extraction and 82% in complex entity recognition (measured by manual inspection) with a sample size of nearly 2000 triples. Before we proceed, we emphasize that pattern-based extraction (C2) is also important because relationships captured based on lexical patterns (multi word expressions and other phrases that do no involve a verb form) are not covered using the deep parsing based technique we employ here.

C4 Knowledge Base Creation: The triples extracted in C3 using NLP-techniques are anchored to the output of C1 and C2: the focused hierarchy and associated non-hierarchical triples. This anchoring is done by mapping the complex entities of those triples from C3 whose subject and object labels match at least one concept in the domain hierarchy to the matching concepts through a *related_entity* predicate. This way only those triples that are related to concepts in the domain of focus are included in the KB.

Note that since the original open extraction using NLP techniques does not consider plurality (for entities and predicates) and tense variants, we used Wordnet to normalize predicates and entity labels. However, predicates that are expressed in passive voice are retained as separate from those in the active voice to differentiate the switch between the subject-object roles of the entities involved. For example, predicates indicated by verbs used in active voice such as 'contribute', 'contributing', 'contribute', are all normalized to 'contribute' while those used in passive voice such as 'was contributed' are treated differently and are not subjected to normalization. This appears to be in agreement with some of the passive forms in the Metathesaurus relationship types that include pairs like *uses* and *used_by*.

5. KNOWLEDGE-BASED SEARCH

The final KB at output from C4 in Section 3 is captured as an Web Ontology Language (OWL) file. However, it does not semantically satisfy all the requirements of a strict ontology as several named entities which would be considered synonymous by human experts are treated as different instances in the ontology. We, however, have synonyms based on anchor texts and redirect labels present in Wikipedia for concepts in the domain hierarchy. For example, the Wikipedia page for 'prolactostatin' redirects to 'dopamine', and hence, a relationship for prolactostatin is also considered to hold for dopamine. The domain hierarchy from C1 is also not a strict taxonomy owing to Wikipedia category hierarchy not being a strict *part_of* or *instance_of* taxonomy. However, for the application component of the project, this is observed to be sufficient. For our use case of the HPC KB, we have about 2 million entities and 3 million triples in the final KB pending more efficient normalization of similar entities. Next we discuss the application component C5 that deals with how the KB in C4 is used in a search system.

C5 Scooner: Scooner is the search and exploration application over PubMed abstracts that is based on the focused KB output from C4 in the previous section. The idea of an up-to-date KB-enhanced search system arises from our belief that the search process can benefit from recently published results that are not well known in the research community and also by incorporating relationship types that go beyond the taxonomic ones. This is part of our approach to address the shortcomings mentioned in the abstract. The key aspect of Scooner is *the domain-specific KB that guides the search process is extracted from the universe of literature that is being searched.*

5.1 Scooner's Search & Exploration Process

The starting point of Scooner is a simple keyword based search interface. But the search process is modeled as an interactive process where, after retrieving ranked results, the points of interaction are based on the triples, there by simultaneously encouraging users to explore relationships. To elaborate, raw text results (based on conventional search engine ranking procedures) are input to a spotter module that annotates them with entities from the focused KB. When an annotated entity (shown akin to a hyperlink) is clicked on, a relations window pops up and displays all triples where the entity participates as a subject or object. Clicking on the corresponding object (resp. subject) would then bring up abstracts that contain that triple and also those that mention the subject and object of the triple. Users can also do further searches within the list of abstracts corresponding to a triple in the relations window. Once these abstracts are studied, to know more about the triples, users can import some interesting abstracts to a workbench and continue exploring relationships for entities spotted in these abstracts. This way triples can be browsed in the context of the abstracts from which they were extracted.

From our experiments using the 2006 Text REtreival Conference (TREC) Genomics dataset questions [10], this approach of using KB-based browsing resulted in an average 83% coverage of answer documents over all the 26 questions that had answers in the dataset. Besides performing regular searches like in any search engine, Scooner can be used to explore the background KB in the context of the corresponding abstracts and can lead to new knowledge (See next section, 5.2.1).

We encourage the readers to watch the video screencast of Scooner made available at `http://tinyurl.com/6g7ntrr`. Although, it does not appear in the screencast yet, we also recently implemented hierarchical filtering of search results based on MeSH headings and qualifiers associated with each abstract available on PubMed. With this, users can also limit search results to specific topics (particular diseases, organisms, experimental techniques) before using the focused KB enhanced search.

5.2 Example Search & Exploration Tasks

In this section we show a few examples using Scooner's knowledge-based search process.

5.2.1 Literature-Based Hypothesis Generation

Scooner's exploration process also enables users to follow a trail of triples that surface any implicit knowledge in the literature (e.g., DR Swanson's popular magnesium deficiency – migraine connection [19]) and in many cases gives insights into new interesting hypotheses to be experimentally validated. An interesting example we found involves concepts: Vasoactive Intestinal Peptide (VIP) and fear conditioning. If we search for VIP and click on one of the annotations for the VIP peptide in an abstract, we see several relations, but the following sequence of relations make up for an interesting hypothesis. We notice the triples

1. VIP peptide *increases* Catecholamine biosynthesis
2. Catecholamines *induce* Beta-adrenergic receptor activity
3. Beta-adrenergic receptors *are_involved* contextual fear conditioning

Note that these triples are not randomly selected; when exploring the first triple in the list, Scooner actually exposes the user to articles where Catecholamine is annotated since the object of the first triple is the biosynthesis of catecholamine. In fact, the first article that is displayed when the users clicks on the object "catecholamine biosynthesis" contains the sentence: "Vasoactive intestinal peptide (VIP) increased catecholamine biosynthesis in bovine adrenal chromaffin cells by 50-200%." This is actually the sentence from which the triple was extracted and the user can observe the context of the triple; in this particular example the user would know the information conveyed by the triple is observed in cattle and the degree of catecholamine increase is up to 4 times. Next, this particular abstract contains the word catecholamine elsewhere and is annotated so the user can explore the relationships of "catecholamine". From there, users can continue browsing more related articles and explore relationships for annotated entities. Coming back to the sequence of the three triples mentioned earlier, from the predicates *increases*, *induce*, and *are_involved*, one can hypothesize the new triple: VIP $affects$ fear conditioning. A caveat here is that the first triple in the list is observed in cattle, the second one in mice, and the third one in humans. Hence, we term these new triples as hypotheses as opposed to treating them as factual knowledge. That these triples come from different organisms should be apparent to the users since they are browsing the triples in the context of the corresponding abstracts.

5.2.2 Searching for Answers to Specific Questions

Here we discuss how Scooner can be used to quickly find answers to specific questions in the bioscience areas. In the 2006 TREC Genomics challenge [10], the questions dealt with associations between genes and diseases, some of which are neurodegenerative diseases such as Alzheimer's, Huntington's, and Parkinson's, topics related to the human performance and cognition domain for which we built the KB. Here we consider the question:
"How do Presenilin-1 gene mutations affect Alzheimer's disease (AD)?"

Note that in bioscience domains, most questions like this do not necessarily have simple answers that can be found in a single document. More often, the full answer spans multiple documents and ongoing research generally adds to the understanding of how genes affect diseases. To find answers to this question, we set out by entering "Presenilin-1 mutations" in Scooner's search box. Several entities related to Presenilin-1 from our background KB are spotted in the abstracts on the first two pages of results. Among those

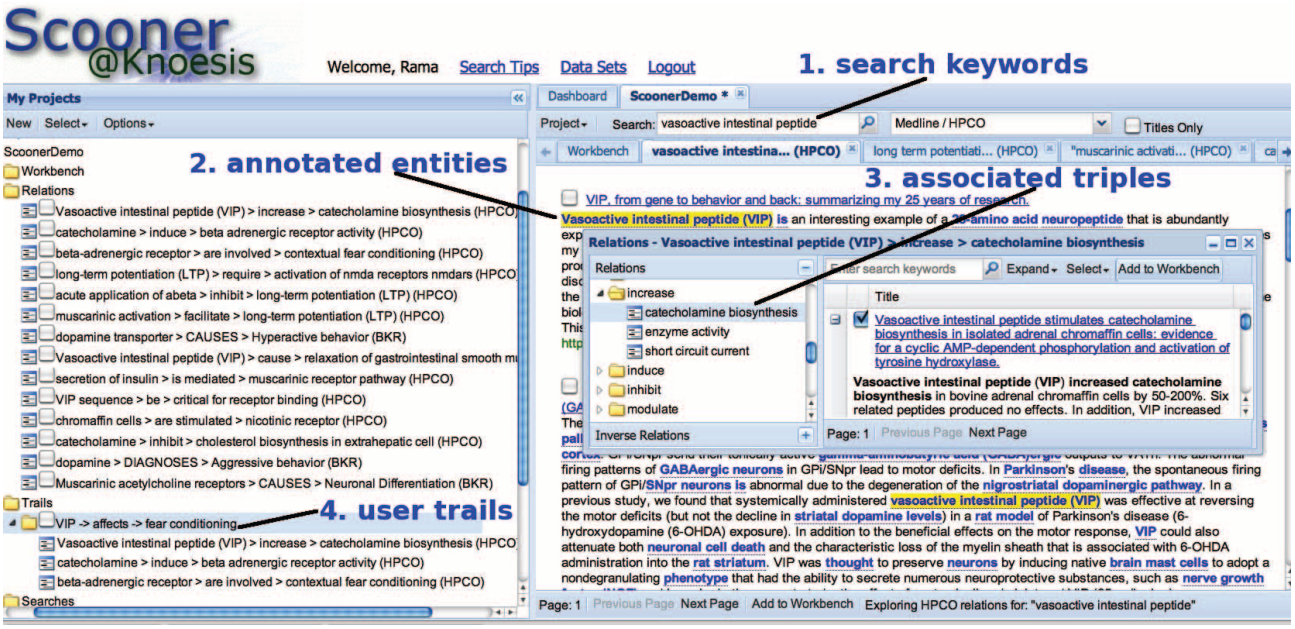

Figure 3: Screenshot of Scooner's Interface and Features

entities "presenilin-1 mutation" and "PS1 mutation" appear best matches. Clicking on these entities and exploring triples gave us significant information on different mutations of the gene that affect AD. The triples we found included

1. PS1 mutation *causes* familial Alzheimer's disease
2. PS1 mutation *associated* classic Alzheimer's disease
3. presenilin-1 mutations *are involved* Alzheimer's disease pathology

Once these triples are shown in the relations window, clicking on the objects (the AD related entities) surfaced abstracts that contain the triple and also those that talk about specific Presenilin-1 mutations and their role in AD.

Although browsing through the search results list in a regular search engine might also give insights into the mutations, the triples for the entities discussed above also provide specific information related to the "how" part of the question. For example, "PS1 mutation" participates in these triples:

1. PS1 mutation *alters* pepstatin binding site
2. PS1 mutation *increases* Gadd153 protein translation

When clicked on the objects of these triples, the first abstract for each triple contains the sentence from which we extracted the triple. Furthermore, the abstracts also discuss how specific mutations use pepstatin binding and Gadd153 protein translation in affecting AD. Thus new results extracted from abstracts can be used to quickly provide the user specific information that answers the question. Again, the actual abstracts that contain the triple will inform the user how the triple came about (experimental parameters, techniques, organisms involved) and lets the user validate the triple. During this experiment, when browsing abstracts in the relations window for some of the triples above, we also found some abstracts corresponding to the original TREC corpus of the gold standard full text documents that discuss Presenilin-1's effects on AD. Since the TREC corpus had only $\approx$ 162,000 documents (we indexed 18 million abstracts) and most of them were published before 2006, the TREC answer document abstracts ranked lower than (appeared after) several recent documents in Scooner. This is also because of the boosting we give to recent abstracts.

5.3 Collaborative Features and Implementation Details:

Scooner combines the ideas of conventional and triple-based search and exploration with persistent search sessions. Users can create search projects and store their search history including the abstracts they felt important and triples they found useful. Users can also create new meaningful trails by combining individual triples they explore. The workbench tab in Scooner facilitates a central aggregation of important abstracts imported for further review by the user. The work bench can be filtered to only show only those abstracts that pertain to a selected set of triples or trails. Additionally, collaborative features were incorporated using which users can write comments on abstracts they find relevant and then share their (sub) projects with other users on a public dashboard visible to other users. A screen shot of Scooner is shown in Figure 3. Details of different implementation components of Scooner (see C5 in Figure 1) are given here.

- **Full text index wrapper service**: Our full text index consists of abstracts released by PubMed until Oct 2010. Our next release will automatically index new updates. We support two types of queries. The first is a traditional query with fields including *title, abstract, author, year, and pmid*. The second query is a phrasal range query which looks for a co-located presence of a subject, predicate, and object labels in the abstract field to retrieve abstracts that contain (information relevant to) the triple. Indexing is done using Lucene API where an En-

glish stemmer is used for the abstract and the title fields, while the remaining fields are analyzed to support exact matching. Also, boosting on the year of publication is performed at the query time so that relevant articles that are more recent are ranked higher.

- **Triples model interface**: The triples in the original HPC KB extracted by us are serialized using conventional search engine indexing techniques (through Lucene) for efficient programmatic access through Java. We also incorporated a second set of triples biased to the HPC domain from the National Library of Medicine's (NLM) Biomedical Knowledge Repository (BKR), which consists of triples from PubMed abstracts involving UMLS concepts, extracted based on shallow parsing and rule based approaches. These are hosted on a Virtuoso RDF triple store[4] and are accessed programmatically using a Java interface to the SPARQL end point that provides triple access.

- **Spotter service**: Named entities in either of the data sets are spotted using an in-memory prefix tree data structure populated with the entity labels when the server is set up to host Scooner. We avoided on-the-fly parsing and part-of-speech tagging and other NLP-based approaches to provide faster response with annotated abstracts. The prefix tree algorithm spots the longest available label in the set of the entities of the KB. For example when it encounters, "long term memory formation", even if both "long term memory formation" and "memory formation" are entities in the KB, it spots the longest label, that is, long term memory formation. But the next time, it encounters the same phrase, "memory formation" is spotted instead because we chose to spot each entity only once in each abstract.

- **Ext JS GUI front end**: An Ext JS based Javascript framework is used to generate the user interface that captures and responds to users' interactions with the system. Search sessions can be broken down by users into various projects depending on their needs. The triples browsed, trails created, searches made, and important abstracts added to the workbench of each project, all persist in a MySQL database to be retrieved by users after subsequent logins. The Ext framework provides maintainability to the GUI components of the system and also assists in addressing cross browser compatibility; Scooner has been tested in Firefox, IE, and Chrome.

5.4 Preliminary User Study

Note that we presented evaluations of techniques used in C1–3 in Section 4 when discussing the components used in the creation of the KB. The search project management and collaborative features are not present in PubMed and hence those aspects of Scooner are not used in our evaluations. Here we present a user study based on day-to-day information seeking efforts undertaken by researchers at the AFRL. Qualitative evaluations of the first version of Scooner were conducted by five researchers from the human effectiveness directorate of the AFRL. They reported that Scooner provided an useful way to navigate between documents through the relationships and that its organization of various searches performed and triples browsed made it convenient when pursuing a focused search task. Two researchers reported that they were able to save significant time relative to their experience with PubMed. The head of the team reported that it saved him a lot of efforts in easily delegating tasks to his team members by sharing his sessions with his comments and notes on the various abstracts in the workbench.

After these qualitative observations, two researchers (different from those mentioned earlier, say, evaluators E1 and E2), who have been conducting research for approximately the same number of years in the area, participated in evaluating Scooner by conducting a study on two different topics in the HPC domain: 1. Nootropics (pharmaceutical cognitive enhancers) and 2. Neurotrophic factors (proteins responsible for growth and survival of neurons). In both topics, the specific subtopics searched include most efficacious examples, adverse effects, molecular mechanisms of agents, effects on normal young people, and synergistic effects of combination treatment. E1 used Scooner for Nootropics and PubMed for Neurotrophic factors and E2 swapped the tools used by the E1. Both evaluators were also asked to spend the same amount of time to search for information and write an organized report. There reports were later consolidated into a final evaluation by Victor Chan, the group leader and one of the authors of this paper.

Due to different personal preferences in conducting search, evaluator E1 used 25% of the time in searching and 75% in writing up the findings, while E2 used 75% in searching and 25% for report writing. By considering the time spent on searching (instead of searching and report generation), Scooner appeared to have more leverage based on the focus facilitated by the background KB. However, at this point it is not completely clear whether Scooner has definitive advantages over PubMed in terms of the *amount of information* gained in a fixed amount of time spent on search alone. As far as relevance of information obtained is concerned, Scooner encouraged users to get to more specific and in-depth information, while PubMed let them peruse and pick from a list of articles. Evaluator E1 studied few articles that provided in-depth analysis on some subtopics, while E2 studied significantly more articles and provided better coverage of the topics. Scooner's functionality that lets users discover new implicit knowledge (Section 5.2.1) or find connections between given entities has not been evaluated here. Both evaluators reported the following observations.

1. Advantages: The evaluators noted that Scooner helped them stay focused in the cognitive research area, which is one of the original goals of the framework. They felt that the narrow focus helped them to perform in-depth exploration of specific topics without significant perusing of PubMed results. They also found the persistent projects and collaborative features made it easier to organize their search tasks.

2. Improvements suggested: The evaluators felt that using Scooner needs some getting used to before they can reap the benefits of KB-enhanced search. They thought the number of relationships for some entities was at times overwhelming, especially for users new to the domain, and felt that additional filtering and contextual ranking would be beneficial.

We plan to conduct a more rigorous evaluation using more specific questions with known answers involving metrics such

[4] `http://virtuoso.openlinksw.com/`

as number of articles read, time spent on each article, triples browsed compared to useful triples found, and the proportion of relevant information found via the usage of the KB component.

A more important challenge is to develop use-cases that will help evaluate the literature-based knowledge discovery abilities of Scooner. These initial evaluations have helped us design yet to be completed more detailed evaluations that can measure 1. accrued benefits once a user is trained; 2. benefits to a less trained user due to the use of background knowledge; and 3. benefits of use of KB for collaboration and training.

6. CONCLUDING REMARKS

We presented an up-to-date knowledge-based search and exploration framework that addresses some shortcomings in the state-of-the-art biomedical search systems. Our framework first carves a focused concept hierarchy, which is then populated with associative relationships using both pattern-based and deep parsing based approaches. We demonstrated the utility of the framework by building the KB for the domain of human performance and cognition; the KB has 2 million entities and 3 million non-trivial triples, nearly all of which are extracted computationally. The KB is finally used to annotate and facilitate triple-based search and exploration of PubMed abstracts. Initial evaluations of the resulting search system, Scooner, show that this framework improves upon conventional search and is an important complementary contribution to the state-of-the-art by addressing focus, up-to-dateness, and recall.

It is well known that evaluation of knowledge-based systems is a very hard problem especially considering many components involved in constructing the KB. In spite of a three year research and development effort involved in what is reported here, each of our components C1–5 in Figure 1 can be improved and we are currently exploring the following opportunities for improvement.

- We aggregate synonymous labels from Wikipedia anchor labels, redirects, and also from other related controlled vocabularies for concepts that have exact matching labels. We are exploring ways to handle partial or approximate matches which are hard to map to particular concepts in standard terminologies. For example, the entity "catecholamine biosynthesis" does not have an exact match in Metathesaurus, but has a partial match with "catecholamine biosynthetic process", which appears to be a synonym. Determining such mapping for complex entities, would give us access to the relationships that are well known in the area. One immediate way of accomplishing this is to use the Norm program from the Lexical Tools offered by the NLM.

- Normalizing predicates is also an important task we plan to work on. The open extraction approach, while surfacing many interesting relationships, poses the problem of creating too many predicates. Like mentioned in the C3 component, predicates like *increase*, *accelerate*, and *stimulate* appear similar but do not appear in the UMLS Metathesaurus (600+ predicates) or the UMLS Semantic Network (54 predicates). Even if mapping is not possible, imposing hierarchies and identifying domain/range semantic type restrictions for these predicates would assist in auditing the quality of the significant number of new triples that arise out of NLP-based techniques.

- We plan to extend Scooner using synonyms-based query expansion (recently incorporated in PubMed) that would further reduce the efforts on part of the user.

- Currently, Scooner uses dictionary ordering on the predicates when displaying triples for an entity. Several well known and thoroughly studied concepts will have significantly large number of triples with them as subject/object. Displaying all those triples for such entities creates a data overload problem (as reported by the evaluators) when there is a predefined search objective. We see two different ways of addressing this. One is to use the MeSH hierarchy and the associated provenance information of the triples (that is, the ids of abstracts they came from) to let users browse triples based on hierarchical filtering. The association between MeSH terms and the triples needed for this filtering is established based on the association between the abstracts that contain the triples and the MeSH terms assigned as metadata for the abstracts by the NLM. We have already implemented this and found it beneficial, for example, to find triples that only appear in abstracts that report research on a specific disease or employ a certain experimental technique. A second way of addressing the overwhelming number of triples we plan to pursue is to take users interaction with the system in the active search session as implicit feedback to automatically rank the triples to be displayed for subsequent user sessions.

- The quality of a KB extracted using our framework is hard to control because the ultimate correctness of the triples extracted depends on several, possibly confounding, factors. First, computational techniques dealing with natural language cannot guarantee 100% precision. So the associated error rates of the techniques involved affect the quality of the database. Second, the quality of the journals that report results also affects the extracted KB. One way to address this is to use journal impact factors and confidence scores from algorithms to associate a quality measure to triples extracted. This measure can act as an additional parameter in ranking the triples displayed in Scooner.

With the improvements outlined above, our framework aims to reduce the cognitive load on the users by presenting interesting and latest facts relevant to a specific domain of interest.

Acknowledgments

Many thanks to Paul Fultz for his contributions in implementing Scooner. Thanks to Cartic Ramakrishnan, Pablo Mendes, and Delroy Cameron for their inputs to the HPC Ontology project and their efforts on previous versions of Scooner. We also thank the Human Effectiveness Directorate, Applied Biotechnology Branch of the AFRL for supporting this work.

7. ADDITIONAL AUTHORS

Armando Soto and Amy Walters (Air Force Research Lab, email: {Armando.Soto, Amy.Walters}@wpafb.af.mil).

8. REFERENCES

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *5th ACM conf. on Digital libraries*, pages 85–94, 2000.

[2] O. Bodenreider. Biomedical Ontologies in Action: Role in Knowledge Management, Data Integration and Decision Support. *Yearbook of medical informatics*, page 67, 2008.

[3] K. Clauson, H. Polen, M. Boulos, and J. Dzenowagis. Scope, completeness, and accuracy of drug information in Wikipedia. *The Annals of pharmacotherapy*, 42(12):1814, 2008.

[4] M. de Marneffe, B. MacCartney, and C. Manning. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of LREC 2006*.

[5] 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*, pages 385–399, 2009.

[6] O. Etzioni, M. J. Cafarella, D. Downey, S. Kok, A. M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in knowitall: (preliminary results). In *Proceedigns of WWW '04*, pages 100–110. ACM, 2004.

[7] M. Gillam, C. Feie, J. Handler, E. Moody, B. Shneiderman, C. Plaisant, M. Smith, and J. Dickason. *The healthcare singularity and the age of semantic medicine*, pages 57–63. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, 2009.

[8] M. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, et al. The Gene Ontology (GO) database and informatics resource. *Nucleic acids research*, 32(Database issue):D258, 2004.

[9] M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *14th conf. on Computational linguistics-Volume 2*, pages 539–545, 1992.

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

[11] G. Jeh and J. Widom. SimRank: A measure of structural-context similarity. In *ACM SIGKDD*, pages 538–543, 2002.

[12] M. Laurent and T. Vickers. Seeking health information online: does Wikipedia matter? *Journal of the American Medical Informatics Association*, 16(4):471–479, 2009.

[13] D. Lizorkin, P. Velikhov, M. Grinev, and D. Turdakov. Accuracy estimate and optimization techniques for simrank computation. *The VLDB Journal*, 19(1):45–66, 2010.

[14] Z. Lu. PubMed and beyond: a survey of web tools for searching biomedical literature. *Database: the journal of biological databases and curation*, 2011, 2011.

[15] Q. Nguyen, D. Tikk, and U. Leser. Simple tricks for improving pattern-based information extraction from the biomedical literature. *Journal of Biomedical Semantics*, 1(1):9, 2010.

[16] C. Perez-Iratxeta, P. Bork, and M. Andrade. XplorMed: a tool for exploring MEDLINE abstracts. *Trends in biochemical sciences*, 26(9):573–575, 2001.

[17] C. Ramakrishnan, P. Mendes, R. Gama, G. Ferreira, and A. Sheth. Joint Extraction of Compound Entities and Relationships from Biomedical Literature. In *IEEE Intl. Conf. on Web Intelligence and Intelligent Agent Technology*, pages 398–401, 2008.

[18] A. Ruttenberg, T. Clark, W. Bug, M. Samwald, O. Bodenreider, H. Chen, D. Doherty, K. Forsberg, Y. Gao, V. Kashyap, et al. Advancing translational research with the Semantic Web. *BMC bioinformatics*, 8(Suppl 3):S2, 2007.

[19] D. Swanson. Migraine and magnesium: eleven neglected connections. *Perspectives in biology and medicine*, 31(4):526–557, 1988.

[20] C. Thomas, P. Mehra, R. Brooks, and A. Sheth. Growing Fields of Interest-Using an Expand and Reduce Strategy for Domain Model Extraction. In *Intl. Conf. on Web Intelligence and Intelligent Agent Technology*, pages 496–502, 2008.

[21] C. J. Thomas, P. Mehra, A. P. Sheth, W. Wang, and G. Weikum. Automatic Domain Model Creation from Structured and Unstructured Sources. In *submitted to ISWC 2011*, 2011.

[22] P. Turney. Expressing implicit semantic relations without supervision. In *Proceedings of ACL 2006*, pages 313–320, 2010.

[23] F. Wu and D. S. Weld. Open Information Extraction using Wikipedia. In *ACL-2010*, 2010.

[24] Y. Yamamoto and T. Takagi. Biomedical knowledge navigation by literature clustering. *Journal of Biomedical Informatics*, 40(2):114–130, 2007.