Extracting, Representing and Mining Semantic Metadata from Text: Facilitating Knowledge Discovery in Biomedicine

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

By

CARTIC RAMAKRISHNAN
B.E., Maharashtra Institute of Technology, 1999

2008
Wright State University

________________________________________

Amit P. Sheth, Ph.D.
Dissertation Director

________________________________________

Thomas Sudkamp, Ph.D.
Director, Computer Science Ph.D. Program

________________________________________

Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies

Committee on Final Examination

________________________________________

Amit P. Sheth, Ph.D.

________________________________________

Michael Raymer, Ph.D.

________________________________________

Vasant Honavar, Ph.D.

________________________________________

Thaddeaus Tarpey, Ph.D.

________________________________________

Guozhu Dong, Ph.D.

________________________________________

Shaojun Wang, Ph.D.
ABSTRACT

Ramakrishnan, Cartic  Ph.D., Department of Computer Science & Engineering, Wright State University,  2008.  Extracting, Representing and Mining Semantic Metadata from Text: Facilitating Knowledge Discovery in Biomedicine.

The information access paradigm offered by most contemporary text information systems is a search-and-sift paradigm where users have to manually glean and aggregate relevant information from the large number of documents that are typically returned in response to keyword queries. Expecting the users to glean and aggregate information has lead to several inadequacies in these information systems. Owing to the size of many text databases, search-and-sift is a very tedious often requiring repeated keyword searches refining or generalizing queries terms. A more serious limitation arises from the lack of automated mechanisms to aggregate content across different documents to discover new knowledge. This dissertation focuses on processing text to assign semantic interpretations to its content (extracting Semantic metadata) and the design of algorithms and heuristics to utilize the extracted semantic metadata to support knowledge discovery operations over text content. Contributions in extracting semantic metadata in this dissertation cover the extraction of compound entities and complex relationships connecting entities. Extraction results are represented using a standard Semantic Web representation language (RDF) and are manually evaluated for accuracy. Knowledge discovery algorithms presented herein operate on RDF data. To further improve access mechanisms to text content, applications supporting semantic browsing and semantic search of text are presented.
## Contents

1 Introduction .................................................. 1
   1.1 Motivations ........................................... 1
   1.2 Knowledge Discovery from Text ....................... 4
      1.2.1 The Rolex analogy .................................. 4
      1.2.2 Harry potter & Leonardo DaVinci ................. 6
      1.2.3 Knowledge Discovery from Biological Text .......... 7
         1.2.3.1 Discovering a systems representation of hearing and deafness ........................................... 7
      1.2.4 Indexing and Browsing Text Intuitively ............. 9
   1.3 Summary .................................................. 10

2 Background & Motivation .................................... 11
   2.1 Knowledge Discovery - An AI perspective .......... 11
      2.1.1 Scale and the need for automation ................. 11
      2.1.2 Large Scale Knowledge Discovery ................. 12
   2.2 Knowledge Discovery from Text ....................... 14
      2.2.1 A changing perspective ............................ 14
         2.2.1.1 The Robot Scientist .......................... 15
      2.2.2 Generalizing Knowledge Discovery from Text ...... 16
      2.2.3 Text Mining ........................................ 18
   2.3 Semantic Web ........................................... 21
      2.3.1 Ontologies and their Representation ............... 22
         2.3.1.1 RDF ........................................... 23
         2.3.1.2 OWL .......................................... 26
      2.3.2 Semantic Analytics .................................. 26

3 Contributions ................................................. 28
   3.1 Research focus & my contributions .................. 28
      3.1.1 Research Vision .................................... 29
      3.1.2 Research outline .................................... 30
# 4 Ontology-enabled information extraction

4.1 Relationship extraction from text ........................................ 34
   4.1.1 Methodology ......................................................... 36
   4.1.2 Part-of-Speech tagging and parsing .............................. 36
   4.1.3 Rule-based post-processing ...................................... 38
   4.1.4 Experiments to evaluate generated RDF ......................... 41
   4.1.5 Limitations of this approach ..................................... 45

# 5 Unsupervised information extraction

5.1 Unsupervised joint extraction of compound entities and relationships .... 47
   5.1.1 Related Work ....................................................... 48
   5.1.2 Rule based sentence segmentation for compound entity discovery .... 50
   5.1.3 rules ................................................................. 53
   5.1.4 Predicting constituents of Compound entities via corpus statistics .. 56
      5.1.4.1 Implementation details ....................................... 57
      5.1.4.2 Mutual Information as a measure of entity viability ........... 57
      5.1.4.3 Preliminary Results ......................................... 58

# 6 Evaluating Extraction Results

6.1 Extracting Triples from OMIM ........................................ 63

# 7 Applications

7.1 A Semantic Browser for Biomedical Literature .......................... 69
   7.1.1 Related Work ....................................................... 69
   7.1.2 Using Semantic Metadata to Browse Text .......................... 70
      7.1.2.1 Design of the Semantic Browser ............................. 71
   7.2 Semantic search ..................................................... 76
      7.2.1 Related Work .................................................... 76
      7.2.2 Using Semantic Metadata to Search for Text ................... 77

# 8 Semantic Metadata Guided Knowledge Explorations and Discovery

8.1 Discovering Informative Connection Subgraphs in Multi-relational Graphs .. 79
   8.1.1 Related Work ....................................................... 80
   8.1.2 Algorithms .......................................................... 82
      8.1.2.1 Candidate graph Generation Algorithm ........................ 82
      8.1.2.2 Display graph Generation Algorithm .......................... 83
      8.1.2.3 Heuristics ....................................................... 84
   8.1.3 Dataset and preliminary results ................................... 88
   8.1.4 Results and Evaluations ......................................... 91
      8.1.4.1 Evaluation using Path Ranks .................................. 91
      8.1.4.2 Qualitative Evaluation ....................................... 95

# 9 Conclusions & Future Work

9.1 Conclusion ............................................................. 98
9.2 Future Work ........................................................... 99
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.2.1</td>
<td>Expert-stipulated information extraction</td>
<td>99</td>
</tr>
<tr>
<td>9.2.2</td>
<td>Discovering informative entity-relationship subgraphs</td>
<td>100</td>
</tr>
<tr>
<td>9.2.3</td>
<td>Strength of a triple</td>
<td>100</td>
</tr>
<tr>
<td>9.2.4</td>
<td>Associations over time</td>
<td>101</td>
</tr>
<tr>
<td>9.2.5</td>
<td>Hypothesis-based composed-passage retrieval</td>
<td>102</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>103</td>
</tr>
<tr>
<td>10</td>
<td>APPENDIX A</td>
<td>112</td>
</tr>
<tr>
<td>10.1</td>
<td>RDF generated for sample sentences</td>
<td>112</td>
</tr>
<tr>
<td>10.1.1</td>
<td>Sentences</td>
<td>112</td>
</tr>
<tr>
<td>10.1.2</td>
<td>Generated RDF</td>
<td>113</td>
</tr>
<tr>
<td>11</td>
<td>APPENDIX B</td>
<td>130</td>
</tr>
<tr>
<td>11.1</td>
<td>Screenshots showing flow of evaluation</td>
<td>130</td>
</tr>
</tbody>
</table>
# List of Figures

1.1 Example surprising connections between entities in text .......................... 6
1.2 Example of genes affecting disease mechanisms at various levels of granularity ................................................................. 8

2.1 Information Extraction vs. Text Mining .......................................................... 18
2.2 Text Mining in terms of Information Extraction - Segmentation and Classification for Entity Identification followed by Association for Relationship Extraction (This picture has been reproduced from [23] with permission from an author.) ................................................................. 20
2.3 Ontology schema & Instances-An Example in the biology domain .............. 22
2.4 Example RDF graph: a simple schema modeling politicians is shown in the upper part of the figure, and example instance data is shown in the lower part of the figure. Resources are represented with ovals and literals are represented with rectangles ................................................................. 24

3.1 Semantic Association between Migraine and Magnesium ............................ 28

4.1 Sample sentence from a PubMed abstract showing compound and modified entities ................................................................. 34
4.2 Schema (UMLS Semantic Network), Instances (MeSH) and documents (PubMed) ................................................................. 35
4.3 Sample sentences from abstract of PMID-1254239 for illustration (Numbers in the figure indicate PubMed ID-Sentence Number) .............. 37
4.4 (a) Parsing a sentence (b) Rule-based Post Processing of the Parse tree ... 42
4.5 Complex Knowledge Structures & Resulting RDF ......................................... 42
4.6 Paths found using bi-directional BFS ............................................................ 44
4.7 An example of an interesting path found .................................................... 45

5.1 Sample sentence from a PubMed abstract showing compound and modified entities ................................................................. 47
5.2 (A) Sample sentence (B) dependency parse tree (C) Sentence Annotated with types of compound entities discovered ................................................................. 51
5.3 Dependencies triggering rules for tree in Figure 5.2 (B) ............................... 52
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>Compound entities identified (Note: prefix subscripted text indicates UMLS type if applicable)</td>
</tr>
<tr>
<td>5.5</td>
<td>Sentence segmented into relationships and compound entities</td>
</tr>
<tr>
<td>5.6</td>
<td>A compound entities that is a candidate for splitting</td>
</tr>
<tr>
<td>5.7</td>
<td>Incorrect parse</td>
</tr>
<tr>
<td>5.8</td>
<td>The impact of lexical guidance in parsing accuracy</td>
</tr>
<tr>
<td>6.1</td>
<td>Juxtaposition of a sentence from OMIM and the corresponding RDF</td>
</tr>
<tr>
<td>6.2</td>
<td>Triple extraction-percent correct triples - measured across relationship types compared across evaluators</td>
</tr>
<tr>
<td>6.3</td>
<td>Entity extraction-percent correct triples - measured across relationship types compared across evaluators</td>
</tr>
<tr>
<td>7.1</td>
<td>Sample RDF used to build Lucene index in the Semantic Browser</td>
</tr>
<tr>
<td>7.2</td>
<td>Text and Semantic Metadata combined in the Semantic Browser</td>
</tr>
<tr>
<td>7.3</td>
<td>Design of the Semantic Browser</td>
</tr>
<tr>
<td>7.4</td>
<td>Navigation in the Semantic Browser - Hover on term to get relationships to other terms</td>
</tr>
<tr>
<td>7.5</td>
<td>Navigation in the Semantic Browser - Choose target term to get documents containing target term</td>
</tr>
<tr>
<td>7.6</td>
<td>Navigation in the Semantic Browser - Choose document containing target term - Note trail forming</td>
</tr>
<tr>
<td>7.7</td>
<td>Searching for text using a subject</td>
</tr>
<tr>
<td>7.8</td>
<td>Search for text using a triple</td>
</tr>
<tr>
<td>7.9</td>
<td>Browse the result in the Semantic Browser</td>
</tr>
<tr>
<td>8.1</td>
<td>Illustrative example for Instance Participation Selectivity</td>
</tr>
<tr>
<td>8.2</td>
<td>Illustrative example for the SPAN heuristic</td>
</tr>
<tr>
<td>8.3</td>
<td>The business ontology</td>
</tr>
<tr>
<td>8.4</td>
<td>The sports ontology</td>
</tr>
<tr>
<td>8.5</td>
<td>The entertainment ontology</td>
</tr>
<tr>
<td>8.6</td>
<td>Example snippet of a subgraph returned for the subgraph query between (Actor_5567, Captain_8262) on my synthetic dataset</td>
</tr>
<tr>
<td>8.7</td>
<td>Quality of the Candidate graph as percentage of maximum score for averaged over 30 queries</td>
</tr>
<tr>
<td>8.8</td>
<td>Quality of the Display graph Note that all weighting heuristics turned off results in poor graph quality in contrast with all heuristics turned on</td>
</tr>
<tr>
<td>8.9</td>
<td>Current Flow in 5 Successive Display graphs</td>
</tr>
<tr>
<td>8.10</td>
<td>Quality of 5 Successive Displays relative to the best</td>
</tr>
<tr>
<td>8.11</td>
<td>Display graph with budget as 12 WITHOUT using edge weighting scheme</td>
</tr>
<tr>
<td>8.12</td>
<td>Display graph with budget as 12 using edge weighting scheme</td>
</tr>
<tr>
<td>11.1</td>
<td>Entry screen</td>
</tr>
<tr>
<td>11.2</td>
<td>Pick a predicate to evaluate</td>
</tr>
<tr>
<td>11.3</td>
<td>Selecting the predicate “inhibited”</td>
</tr>
<tr>
<td>11.4</td>
<td>Starting evaluation of all instances of the “inhibited” relationship</td>
</tr>
</tbody>
</table>
11.5 Subject Predicate and Object are correct .......................... 132
11.6 Click next, Subject Predicate and Object are correct for the next one too . 132
11.7 Click Next on previous screen, Subject and Object are correct but Triple is not ................................................................. 133
11.8 Click Next on previous screen, Subject, Predicate and Object are correct & triple is correct ....................................................... 133
11.9 To save partial evaluation results copy the evaluation id (show highlighted) . 134
11.10 Point your browser to start URL and paste the number in text box . . . . 134
11.11 Takes you back to evaluation for “inhibited” to 4/198 where you left off . 134
List of Tables

2.1 RDFS Entailment Rules. The first column shows the rule name. The second column shows a set of RDF statements for a given RDF graph G, and the third column shows what statements should be added to G. 25
Acknowledgement

I owe a debt of gratitude to a number of my friends and family members who have supported me through the year in this endeavor. I would like to thank Pablo N. Mendes, Christopher Thomas, Meenakshi Nagarajan, Karthik Gomadam, Ajith Ranabahu for their friendship and support during the difficult transition that we all made together. Each of them has made invaluable contributions to my dissertation through stimulating discussions. They have also kept me sane through some of the most trying times in my career so far. I am especially thankful to my friends Christopher Thomas, Dr. Frederick Maier, Meenakshi Nagarajan, Dr. Brianna Peterson, Pablo N. Mendes for their belief in my abilities and their unconditional friendship.

I thank my advisor Dr. Amit Sheth for a well rounded education, the opportunity to choose my own path through the Ph.D. and the great learning atmosphere he provided via Kno.e.sis. I owe special thanks to Dr. Shaojun Wang for helping me through some of the technical issues and for help with proof-reading my papers. I would also like to thank Dr. Michael Raymer and Dr. Guozhu Dong for their insightful comments and suggestions.

I am thankful to my parents for providing me with the best education possible and sending me on this journey toward my Ph.D.

Perhaps the one person who has unwittingly been the single largest steadying and motivating factor in my life is my wife Drashti R. Dave. I thank her for her unconditional love, patience and for keeping me true to myself.
This dissertation is dedicated to my parents Mrs. Geeta Ramakrishnan & Mr. V.S. Ramakrishnan and the loving memory of my grandparents Mrs. Swarnam Anantaraman and Mr. N.K. Anantaraman.
Introduction

1.1 Motivations

The information access paradigm offered by most contemporary Web information systems is a search-and-sift paradigm where users have to manually glean relevant information from the large number of Web page results that is typically returned in response to a keyword query. In many cases, users will need to conduct multiple keyword searches and integrate relevant results across searches in order to satisfy their information need. A consequence of this paradigm on the design of such systems is that there is an emphasis on the use of effective techniques for determining the importance of a Web page relative to a keyword query i.e. ranking. Quite often, the technique for ranking results considers the distribution of keywords in documents as well as the importance of Web pages conferred upon them by the hyperlink structure of the Web [1]. For the vast collections of unstructured text whose content is devoid of hyperlinks, topic hierarchies and structural metadata standards (Dublin CORE\(^1\) and MEDLINE [3]) are some of the prevalent methods for assigning relevance. Databases of scientific publications are a good example.

Scientific literature is an indispensable investigative tool in any science. The advent of the Web has greatly improved the accessibility of such literature. Consequently, web-accessible databases of papers (such as the ACM portal) indexed by keywords that describe

\(^{1}\text{http://dublincore.org/}\)
their content are commonplace in many domains. A great example in the biomedical domain is the PubMed [3] database which contains over 16 million manually-classified abstracts of scientific publications. Using the Medical Subject Headings (MeSH)[4] topic hierarchy, PubMed allows users to effectively retrieve abstracts of publications in the biomedical domain. However, users still have to sift through the abstracts to find the ones that may be relevant. Such literature databases are growing at a phenomenal rate. An estimate of the number of abstracts in the PubMed database in 1986 was approximately 8 million [5] which has grown to over 16 million today. With the increased diversification and specialization of research in biomedicine the number of journals and conferences in the field are steadily increasing. This will only serve to speed up the growth of such databases. I argue that there is an imminent information overload problem, especially with biomedical literature, which cannot be addressed by the traditional search-and-sift paradigm. From our experience, the following pieces of evidence are symptomatic of the severe inadequacies of search-and-sift in different but related uses of biomedical literature. To begin researching a specific aspect of a gene or a protein scientists often have to painstakingly do keyword searches against PubMed only to find hundreds or thousands of “relevant” articles. This is followed by several months of reading to find areas where they could make contributions.

Perhaps a more serious drawback was pointed out by Dr. D.R. Swanson in 1986. By searching biomedical literature manually, he discovered previously unknown connections between Fish Oils and Raynaud’s Syndrome [6] which were implicit in the literature. He followed this up with several more examples such as the association between Magnesium and Migraine[7]. In fact, the paper revealed eleven neglected, potentially useful associations that Magnesium might have with Migraine. These discovered connections have since been validated by clinical trials and experiments. Such hidden, valuable associations have been termed Undiscovered Public Knowledge [6]. Expecting the user to unearth Undiscovered Public Knowledge can lead to important pieces of knowledge being missed. I therefore argue that there is a need for information systems that support hypothesis discov-
1.1. MOTIVATIONS

ery and validation style operations over text. Going beyond the search-and-sift paradigm clearly has promising benefits in the context of retrieval and knowledge discovery over scientific literature.

From the perspective of browsing, a third limitation of the search-and-sift paradigm becomes apparent. Contemporary Web Information systems rely on pre-specified hyperlinks to support user navigation. Such links capture a content developer/editor’s view of related information, and thus are necessarily limited by the link specifier’s context and knowledge. As a result the sifting operation is constrained by the hyperlinks, leading to a certain amount of rigidity in the sifting process. This rigidity is clear when a search operation followed by subsequent sifting fails to retrieve results that are relevant to the user. At this point the only recourse is to change the keywords and conduct another search. Although there is some flexibility in browsing when it comes to database-driven websites the user is not able to customize browse paths based on the content of Web resources.

The design of the search-and-sift paradigm assumes that the user, upon not finding the desired information among the documents in the result, has two options:

- Option 1 - Browse the results using hyperlinks (if they exist) to find the information sought;

- Option 2 - Augment input keywords and refine or generalize the query.

These options are symptomatic of some fundamental limitations of conventional access methods to content of scientific literature. Some of the limitations I observe are:

- Rigidity of Browsing - In case of Option 1, the user is limited to using the hyperlinks that were created by the creator of the page in the result. The hyperlinks therein therefore reflect the intention of the page creator and are constrained by their interpretation of the source and target page’s content.
1.2. KNOWLEDGE DISCOVERY FROM TEXT

- Tedious Manual Analysis - The user has to read each abstract to assess its relevance to the query and repeat this process over several results to piece together the sought-after information. In the event that the returned results do not satisfy the information need, the query is either refined or generalized.

- Need for Repeated Searching - Domain specific metadata (e.g. MeSH [4] hierarchy) is often used to aid in the query refinement/generalization process. However, the need to refine and repeat searches can be quite tedious and frustrating at times.

The main aim of this dissertation is to make a contribution towards changing the way information contained within text is accessed and used. In the following sections I present analogies and examples that help clarify the meaning of knowledge discovery in the context of this dissertation.

### 1.2 Knowledge Discovery from Text

#### 1.2.1 The Rolex analogy

In almost any field of scientific research, today’s great ideas were part of science fiction in the past. One such idea was posited in the 2002 science fiction thriller “Minority Report”. A report in the California Computer News (October 20, 2004) edition read², ”In the recent science-fiction thriller ‘Minority Report,’ Tom Cruise plays a detective who solves future crimes by being immersed in a ‘data cave’, where he rapidly accesses all the relevant information about the identity, location and associates of the potential victim. A team at Purdue University currently is developing a similar ‘data-rich’ environment for scientific discovery that uses high-performance computing and artificial intelligence software to display information and interact with researchers in the language of their specific disciplines.

²Transcribed from http://www.aaai.org/aitopics/pmwiki/pmwiki.php/AITopics/ScientificDiscovery#good
“If you were a chemist, you could walk right up to this display and move molecules and atoms around to see how the changes would affect a formulation or a material’s properties,” said James Caruthers, a professor of chemical engineering at Purdue. The method represents a fundamental shift from more conventional techniques in computer-aided scientific discovery. “Most current approaches to computer-aided discovery center on mining data in a process that assumes there is a nugget of gold that needs to be found in a sea of irrelevant information,” Caruthers said. “This data-mining approach is appropriate for some scientific discovery problems, but scientific understanding often proceeds through a different method, a ‘knowledge discovery’ approach. Instead of mining for a nugget of gold, knowledge discovery is more like sifting through a warehouse filled with small gears, levers, etc., none of which is particularly valuable by itself. After appropriate assembly, however, a Rolex watch emerges from the disparate parts. Discovery informatics depends on a two-part repeating cycle made up of a ‘forward model’ and an ‘inverse process’ and two types of artificial intelligence software: hybrid neural networks and genetic algorithms.”

Caruthers’s distinction between data mining and knowledge discovery is an important one. He likens the knowledge discovery process to component assembly to produce a whole that is has greater value than the sum of the parts (i.e. the Rolex). I believe that this sort of emergent behavior is exactly what is needed to facilitate serendipitous discovery of new knowledge from textual source.

In Caruthers’s Rolex analogy the components are assumed to exist in a warehouse. However, when dealing with unstructured text data, the warehouse (the corpus) contains raw metal and plastic (words) instead of usable components (entities). It therefore seems necessary to mine the text to build components that can then be assembled on demand to reveal complex knowledge artifacts.
1.2. KNOWLEDGE DISCOVERY FROM TEXT

1.2.2 Harry Potter & Leonardo DaVinci

Other examples of serendipitous discovery are seen in public text data sources such as Wikipedia. The Da Vinci Code was a controversial mystery/detective novel by US author Dan Brown, published in 2003 by Doubleday. When public interest in this story was at its peak, browsing Wikipedia entries about the pseudo-history behind this book revealed some interesting and unexpected connections. These connections are shown in the figure below.

The unexpected connections involve a character by the name Nicolas Flamel mentioned in the second novel of the Harry Potter series by J.K. Rowling, titled Harry Potter and the Chamber of Secrets. Flamel is mentioned as Claude Frollo’s scientific inspiration in Victor Hugo’s The Hunchback of Notre Dame\(^3\). Frollo seems to be obsessed with Flamel’s work with the Philosopher’s Stone. Flamel, Victor Hugo and Leonardo Davinci are listed

\(^3\)http://en.wikipedia.org/wiki/Nicolas_Flamel
as alleged grand masters of the Priory of Scion, a secret organization central to the Davinci code story.

All of this information used to construct such connected graphs is available in Wikipedia. Yet the revelation of such connections evokes surprise among many.

1.2.3 Knowledge Discovery from Biological Text

Swanson’s discoveries described earlier, found associations between migraine and magnesium in PubMed in 1986. PubMed which contained 8 million record then, contains 18 million today. The amount of text data on the Web and in domain specific settings is growing at a phenomenal rate. According to Neumann [8], the estimated total number of words ever spoken by humans is 12 exabytes, while the rate of data capture in biology is 1 exabyte per year. Arguably, a signification portion of this growth is in text data. There is therefore likely to be a commensurate growth in the amount of Undiscovered Public Knowledge. This accentuates the need for knowledge discovery operations over text.

In addition to the need for such serendipitous discovery other complex operations in biological investigations can be automated. Here I consider an example involving the creation of disease mechanistic models from text.

1.2.3.1 Discovering a systems representation of hearing and deafness

Mechano-sensory sound conduction accomplishes the conversion of sound into neural activity. The inner ear contains a rich framework of interacting and specialized cells and proteins that accomplish various aspects of hearing. Genetic mutation, developmental defects, infection, trauma, and adverse environmental and drug reactions can all lead to its disruption or functional destruction via the disruption of discrete molecular, cellular, or macro-anatomic elements. As an example let us consider “Usher Syndrome”, a genetic disorder
that is a leading cause of deaf-blindness. The table in Figure 1.2 shows the three types of the Usher syndrome that are caused by molecular level defects in two different parts of the hair cells (stereocilia) which are located inside the cochlea. Tip links (Figure 1.2 d) refer to the spring-like molecules at the connecting the tip of one stereocilium to another. These spring proteins serve to open and close the auditory-signal-transduction channels when the cilia are deflected by ripples in the cochlear fluid. The tip links are formed by the adhesion of the proteins CHD23 (OctOCadherin) and PCDH15 (protocadherin). This adhesion takes place at under very specific conditions (i.e. "CDH23 and PCDH15 interact at 0.1 mM Ca\(^{2+}\)"[9]. This scenario shows mechanistic models of diseases and developmental mechanisms of the affect anatomical regions. Each fact in such a model is distilled from one or more scientific publications. In almost all such investigations scientists build partial models of diseases based on knowledge in related literature and data from databases containing experimental data. These incomplete models then serve as a source for the generation of hypotheses that are then investigated using well-established investigative techniques. As and when new assertions gain substantial consensus in the scientific community the model of the said disease is enriched and this process repeats. Each new assertion of fact dis-
covered is corroborated by textual as well as numerical data from either prior publications or current research. This process of initial partial model creation followed by subsequent enrichment is largely a manual process that requires access to relevant published literature and relevant experimental data. Furthermore, in order to get a comprehensive systems biology view of the mechanism and processes that together cause diseases like Usher syndrome it is clear that fine-grained extraction of entities, relationships and processes from text will be required.

1.2.4 Indexing and Browsing Text Intuitively

Dr. Vannevar Bush, in 1945 [10], referring to the human brain said, “It operates by association. 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 vision may seem anachronistic given that topic hierarchies are used extensively today to index and retrieve documents (nonhyperlinked) in many domains. However, this vision emphasizing relationships and associations continues to be highly relevant, and can indeed drive the next generation of search and analysis capabilities. Emphasizing a stream-of-consciousness paradigm for access to text, the following example given by Dr. Bush, clearly highlights the merit of his idea. In his description of a device named MEMEX, he wrote -“The owner of the memex, let us say, is interested in the origin and properties of the bow and arrow. Specifically he is studying why the short Turkish bow was apparently superior to the English long bow in the skirmishes of the Crusades. He has dozens of possibly pertinent books and articles in his memex. First he runs through an encyclopedia, finds an interesting but sketchy article, leaves it projected. Next, in a history, he finds another pertinent item, and ties the two together. Thus he goes, building a trail of many items. Occasionally he inserts a comment of his own, either linking it into the main trail or joining it by a side trail to a particular item. When it becomes evident that the elastic
properties of available materials had a great deal to do with the bow, he branches off on a side trail which takes him through textbooks on elasticity and tables of physical constants. He inserts a page of longhand analysis of his own. Thus he builds a trail of his interest through the maze of materials available to him.”

This notion of building trails on demand and connecting documents together is a very powerful metaphor. I believe that contrary to conventional hierarchy-based indexing and retrieval of documents, this paradigm is far more intuitive. Furthermore, building such a networked space of documents in an on-demand fashion while taking into consideration user preference, will allow different users to create their own personal knowledge spaces over the same set of documents. Users browsing such a knowledge space will no longer be restricted to browsing text using hyperlinks provided by the document creator.

1.3 Summary

So far I have discussed some drawbacks of contemporary information systems when it comes to retrieving and utilizing textual content. I have also seen some examples of knowledge discovery from text data both in the biology domain and outside of it. I have also shown the need for the extraction of complex knowledge artifacts from literature.

In the next section I will study knowledge discovery as a research discipline in Artificial Intelligence. Further, taking knowledge discovery in the context of textual data, I will study it’s relationship to the areas of text mining, information extraction and semantic web.
Background & Motivation

We will first review background concepts. The next two sections focus on two key topics that motivate investigations herein. The first is Knowledge Discovery as studied in philosophy of science, artificial intelligence and other related disciplines. The second topic focuses specifically on approaches aimed at improving access to textual content. This includes an overview of text mining, information extraction, semantic web, a review of ontologies and representation languages and an introduction to semantic analytics.

2.1 Knowledge Discovery - An AI perspective

In this section I present a discussion of the Artificial Intelligence perspective on Knowledge Discovery. I specifically address issues pertaining to the biomedical domain. Our discussions are focused on understanding the issues of scale and novel information discovery.

2.1.1 Scale and the need for automation

The rate of data generation within specific fields of human scientific endeavour, such as biology, is estimated at 1 exabyte per year. Comparing this to the estimated total number of words ever spoken by humans, approximately 12 exabytes [8], gives some indication of the scale of data over which scientific investigation will be conducted in the future. Muggleton [11] made an interesting observation that stresses the need for automation to deal with data
on such a scale. “During the twenty-first century, it is clear that computers will continue to play an increasingly central role in supporting the testing, and even formulation, of scientific hypotheses. This traditionally human activity has already become unsustainable in many sciences without the aid of computers. This is not only because of the scale of the data involved but also because scientists are unable to conceptualize the breadth and depth of the relationships between relevant databases without computational support. The potential benefits to science of such computerization are high – knowledge derived from large-scale scientific data could well pave the way to new technologies, ranging from personalized medicines to methods for dealing with and avoiding climate change.” The reference to earth’s climate change is a reinforcement of Microsoft’s Science 2020 vision [12]. In this 15 year plan ranging from 2005 to 2020, Microsoft has identified issues, computational platforms needed, computer science techniques required and scientific challenges towards goals such as building models of the earth’s ecosystem. Focusing on the computational platforms in the first 10 years\(^1\) Data Semantics and Semantic Web seem to play a critical role. Semantics here is seen as the foundation for the mechanism by which computer processable interpretations of raw data are used to support automated data processing on a large scale.

### 2.1.2 Large Scale Knowledge Discovery

In 2004 Buchanan & Livingston [13] wrote, “The end point of scientific discovery is a concept or hypothesis that is interesting and new (Buchanan 1966). Insofar as there is a distinction at all between discovery and hypothesis formation, discovery is often described as more opportunistic search in a less well-defined space, leading to a psychological element of surprise.” In order to support knowledge discovery over the massive volumes of data being produced, I need to build automated systems that support hypothesis formula-\(^1\)http://research.microsoft.com/towards2020science/downloads/T2020S_Roadmap.png
2.1. KNOWLEDGE DISCOVERY - AN AI PERSPECTIVE
September 21, 2008

tion and validation. As discussed earlier, this requires the assignment of interpretation to
data using computer processable constructs. Buchanan draws a distinction between knowl-
edge discovery and hypothesis formulation by associating the element of surprise with the
results of the former operation. Computational methods supporting knowledge discovery
will need to use some heuristics to guide the discovery process towards more interesting
concepts or hypotheses.

According to Buchanan & Livingston, “The earliest demonstration of self-directed,
opportunistic discovery was Doug Lenat’s program, AM (Lenat 1982). It was a success-
ful demonstration of AI methods for discovery in a formal domain characterized by ax-
ioms (set theory) or rules (games). AM used an agenda-based framework and heuristics
to evaluate existing concepts and then create new concepts from the existing concepts. It
continued creating and examining concepts until the ‘interestingness’ of operating on new
or existing concepts (determined using some of AM’S heuristics) dropped below a thresh-
old. Although some generalization and follow-up research with AM was performed (Lenat
1983), this research was limited to discovery in axiomatic domains (Haase 1990; Shen
1990; Sims 1987).” Such self-directed, opportunistic discovery is exactly what is needed to
support knowledge discovery over massive volumes of multi-modal content in various do-
 mains. Focussing on text data in the biomedical domain, scientific literature is one sources
of knowledge that is diverse and voluminous.

Scientific knowledge contained in literature is the distilled form of countless experi-
ments, trains of thought and the result of years of research precedent. There is therefore
the notion of data corroborating the text content which itself is often argumentative in na-
ture as opposed to factual. This is in sharp contrast with encyclopedic knowledge that is
factual. Scientific text containing the results of an investigation are often made up of intu-
itions expressed as interesting directions of research originating in related work leading to
the generation of hypotheses and eventually to the design of an experiment. Experimental
results corroborate or refute hypotheses leading to a more crystallized factual conclusion.
These factual conclusions may lead to contradictions with past knowledge. Some notion of confidence in these facts based on quality and reputation of the authors of the content, venue of publication and the reader personal bias often drives such conflict resolution.

In order to facilitate the discovery of surprising knowledge in biomedicine, these automated hypothesis formulation and validation operations will have to leverage a variety of data sources covering multiple related aspects of the investigation domain. Such well organized, aspect-wise diversity and richness of data is a characteristic of many scientific domains, especially biomedicine. Supporting such hypothesis validation over biomedical data therefore presents significant data integration challenges.

It is therefore clear that knowledge contained in scientific literature does not form the kind of relatively straightforward axiomatic system on which the design of past knowledge discovery systems was predicated.

2.2 Knowledge Discovery from Text

2.2.1 A changing perspective

In this section I draw a connection between the AI perspective on knowledge discovery operations discussed in past work and the field of text mining. In the process I relate the fields of information Extraction, text Mining and knowledge discovery.

Buchanan & Livingston [13] further said, “Our long-range goal is to develop an autonomous discovery system for discovery in empirical domains, namely, a program that peruses large collections of data to find hypotheses that are interesting enough to warrant the expenditure of laboratory resources and subsequent publication. Even longer range, I envision a scientific discovery system to be the generator of plausible hypotheses for a completely automated science laboratory in which the hypotheses can be verified ex-
2.2. KNOWLEDGE DISCOVERY FROM TEXT

Experimentally by a robot that plans and executes new experiments, interprets their results, and maintains careful laboratory records with the new data.” Knowledge discovery from biomedical literature is therefore a very complex operation. The long-term goal of an automated scientist robot planning experiments, executing them and interpreting their results remains the AI perspective on knowledge discovery. Whether such truly “intelligent” behavior of machines is possible, is a debatable question. Such debates often boil down to issues pertaining to the definition of intelligence and ethical issues pertaining to the creation of such intelligent machines. These are two issues that have drawn criticism on the AI perspective to knowledge discovery from the scientific community as well as outside of it. However there have been some recent success stories in building the robot scientist.

2.2.1.1 The Robot Scientist

In an article titled “Mark of time” published in The Engineer Online (September 18, 2006), a description of the robot scientist project read: “A pioneering study at Manchester University is using a ‘robot scientist’ to examine blood samples for biological markers that may diagnose Alzheimer’s disease.” It went on to say, “The robot scientist combines the automatic operation of a blood analysis technique called GC-MS with artificial intelligence to determine which experiment to carry out next. … The robot uses Inductive Logic Programming, a machine learning process. The scientists give it the background knowledge about the experiment, called the domain. It then decides which hypothesis to follow using the available data.”

From this example it is clear that the focus here is on the automation of the reasoning within a very narrow domain in biology. Examining blood samples for specific markers indicative of diseases is just one of the operations that a scientist might perform in a long chain of investigative techniques. The problem of scale emerges again, this time in the

\[\text{http://www.theengineer.co.uk/Articles/Article.aspx?liArticleID=296124}\]
2.2. KNOWLEDGE DISCOVERY FROM TEXT

September 21, 2008

guise of the breadth of knowledge required for generalized scientific investigation. In this example the background knowledge is given to the robot scientist. This knowledge is meant only for the specific case of Alzheimer’s diagnosis and pertains only to blood tests.

2.2.2 Generalizing Knowledge Discovery from Text

It would therefore seem that in order to achieve automation the problem and the domain needs to have some structure which should be represented in some computer processable form. However structure seems to be the very thing missing in real situations where intuition guides manual exploration often leading to serendipitous knowledge discovery.

This very thought was put forth very eloquently by Dr. Herbert A. Simon. “Understanding the processes scientists use to discover new laws and to test hypotheses has been an active domain of cognitive research and AI modeling for several decades. Scientific discovery is an interesting and important task domain because it involves highly ill-structured problems that call on the whole range of human cognitive resources, and thereby provides deep insights into complex and creative human thinking. . . . Thus, research on scientific discovery requires one to address fundamental problems in cognitive psychology (the processes of discovery), in the philosophy of science (the relation between the discovery and validation, or disconfirmation, of hypotheses), and in computer science (languages for discovery, heuristic search in discovery environments).” Dr. Simon’s perspective sheds some light on just how complex knowledge discovery in an empirical environment can be. He points out that the ill-structured nature of knowledge discovery means that the automation of the discovery process will require automation systems to address issues pertaining to cognitive psychology, philosophy of science in addition to computer science. The purpose in studying cognitive psychology in this context is to understand the human processes of knowledge discovery and encode their basic operations into the design of automated dis-

3http://www.psy.cmu.edu/psy/faculty/hsimon/sci-dis.html
2.2. KNOWLEDGE DISCOVERY FROM TEXT

covery algorithms. Philosophy of Science aspects will allow for a better understanding of the notions of correctness or incorrectness of a hypothesis with respect to given data, allowing automated systems to address issues pertaining to correctness and consistency of hypotheses. Computer science aspects are relevant to the design of query languages for discovery operations and heuristics engineered from observations about human discovery processes. The one piece that seems to have been left out in this puzzle is the formalism that is used to represent the data and the knowledge.

A basic tenet of the symbolic school of thought in artificial intelligence states that sufficient richness in representation of knowledge and data can simplify the algorithms that are employed to support reasoning operations. An editorial by Simon, Valdes-Perez & Sleeman [14] read, “Complexity of programs or of their outputs is not a measure of their ‘intelligence’. Given very complex tasks, complex algorithms may be a necessity, but they are clearly not a virtue. A critical lesson of artificial intelligence, and of computing in general, is that if a task domain has strong structure and if sufficient domain information can be obtained, either \textit{a priori} or in the course of computation, then rather simple programs may suffice.” Taken together this seems to indicate that expressive knowledge representation formalisms are key to enabling complex discovery operations over large sets of data.

In the context of knowledge discovery over biological data knowledge contained in structured databases have already been assigned computer processable interpretations. However, the unstructured content in biomedical literature is raw text that has not been assigned a semantic representation. This raw representation is human-readable and is most expressive and unusable for automated reasoning. Approaches to assigning interpretations to text content and representing these interpretations in a formalism amenable to algorithmic processing is required.
2.2. KNOWLEDGE DISCOVERY FROM TEXT

2.2.3 Text Mining

“Text Mining is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. A key element is the linking together of the extracted information to form new facts or new hypotheses to be explored further by more conventional means of experimentation.” - Marti Hearst

If the extraction of information from text is either explicitly or implicitly guided by some representation of the expected outcome, it is said to be a supervised approach to information extraction. This supervision can be in the form of instance exemplars that allow the algorithm to “learn” distinctive patterns of the sought-after information. It can also take the form of generic templates which are not instance exemplars but higher level concepts. A majority of the techniques used in text mining literature are either supervised or semi-supervised. The need for explicit representation of expected outcome (training data) means that text mining is not knowledge discovery. Knowledge discovery over text and text mining can be seen as two complementary operations that can be applied iteratively to understand and utilize text content automatically. The first sentence in the quote above gives some indication of some of the basic operations that constitute text mining. It indicates that text mining is a utilization of the basic information extraction primitives. The figure 2.1 shows these basic primitives as described by Cohen.

According to Cohen the basic operations of Information Extraction are

- Segmentation - The task of breaking text into meaningful fragments.

---

4 http://people.ischool.berkeley.edu/~hearst/text-mining.html
5 http://www.cs.cmu.edu/~wcohen/ie-survey.ppt
2.2. KNOWLEDGE DISCOVERY FROM TEXT

- Classification - The task of associating interpretations with these fragments by classifying them into known categories.

- Association - The task of inferring implicit or explicit connections or relationships between these fragments.

- Clustering - The task of grouping fragments into cohesive groups.

These basic operations can be applied in different combinations to suit different problems. When applied with the aim of extracting named entities, segmentation and classification are the two operations combined in sequence. Subsequent extraction of relationships between them requires the association primitive. These two text mining operations, Entity Identification and Relationship extraction utilizing the information extraction primitives are shown in the figure 2.2.

Applying these operations to text corpora belonging to different but related domains can divulge relationships between entities bearing that element of surprise referred to by Buchannan (Section 2.1.2). When dealing with domains that can span large corpora (e.g. biomedicine) a prohibitively large number of entities and relationships could result from text mining over these corpora. In order to facilitate serendipitous discovery of new knowledge contained within these entities, relationships and their complex interconnections two things seem necessary. First, a representation mechanism for these entities and relationships that is amenable for use by discovery algorithms. Second, discovery algorithms (operating over these representations) that are guided by heuristics based on human intuitions used in manual discovery.

The idea of creating formal computer processable representations of multi-modal Web content has evolved into the hugely popular “Semantic Web” vision. In the Section 2.3

---

6http://lsdis.cs.uga.edu/lib/download/HSK02-SEE.png

7Parts of this section are reproduced from the dissertation titled “A Framework to Support Spatial, Temporal and Thematic Analytics over Semantic Web Data” by Dr. Matthew J. Perry with due permission from the author.
2.2. KNOWLEDGE DISCOVERY FROM TEXT

Figure 2.2: Text Mining in terms of Information Extraction - Segmentation and Classification for Entity Identification followed by Association for Relationship Extraction (This picture has been reproduced from [23] with permission from an author.)
I discuss the Semantic Web vision, associated representation formalisms and query languages supporting discovery operations.

2.3 Semantic Web

In 2001, Berners-Lee et al. [15], posited the vision of the Semantic Web. This vision promises an extension of the current web in which all data is accompanied with machine-understandable metadata allowing a much higher degree of automation and more intelligent applications. To make this idea more concrete, consider the statement “Cartic Ramakrishnan is a Ph.D. student at Knoesis which is located at Wright State University in Dayton Ohio.” The meaning of this statement is clear to a human with knowledge of colleges and universities and the geography of the United States. In addition, upon seeing this statement, other related information comes to mind such as professors who work at the University. The goal of the Semantic Web is to make the semantics of such data on the web equally clear to computer programs and also to exploit available background knowledge of related information. On the Semantic Web this statement would be accompanied with semantic metadata identifying an instance of the concept “University” with the name “Wright State University”. Similarly, the instance of City and State, “Dayton, Ohio,” would unambiguously describe the university’s geographic location. Note the distinction between semantic metadata describing high-level concepts and relationships and syntactic and structural metadata describing low level properties like file size and format. The Semantic Web vision further describes future applications that will leverage this sort of semantic metadata to automate things like appointment scheduling for users, taking into account a variety of constraints, all of which are expressed using similar semantic metadata.

One major milestone in the evolution of the Semantic Web is the development of knowledge representation formalisms that allow for representation of semantic metadata at
2.3. ONTOLOGIES AND THEIR REPRESENTATION

Ontologies are central to realizing the Semantic Web, as they formally specify concepts and their relationships and provide the means to create semantic metadata for objects (documents, data files, databases, etc.). Ontology is defined as “a specification of a conceptualization” [16]. In database terms, I can divide an ontology into two parts: a schema and instance data. The schema models a domain by defining class types (e.g., University, City) and relationship types (e.g., located_in). The schema is populated with instances of classes and relationships (e.g., Wright State University located_in Dayton) to create facts representing knowledge of the domain. The Figure 2.3 shows an example of an ontology schema, instances and the relationships that might hold between them. The Semantic Web requires a standard, machine-processable representation of ontologies. The W3C has defined standard models and languages for this purpose. Here, I discuss the standard lan-
2.3. SEMANTIC WEB

The use of ontologies represented with these languages is becoming very popular. Many such ontologies are available on the Web. These include domain specific ontologies, for example Gene Ontology [19], NCI Cancer Ontology [20].

2.3.1.1 RDF

RDF has been adopted by the W3C as a standard for representing metadata on the Web. Resources in RDF are identified by Uniform Resource Identifiers (URIs) that provide globally-unique and resolvable identifiers for entities on the Web. These resources are described through participation in relationships. Relationships in RDF are called Properties and are binary relationships connecting resources to other resources or resources to Literals, that is, literal values such as Strings or Numbers. These binary relationships are encoded as triples of the form (Subject, Property, Object), which denotes that a resource i.e. the Subject has a Property whose value is the Object. These triples are referred to as Statements. RDF also allows for anonymous nodes called Blank Nodes which can be used as the Subject or Object of a statement. I call a set of triples an RDF graph, as RDF data can be represented as a directed, labeled graph with typed edges and nodes. In this model, a directed edge labeled with the Property name connects the Subject to the Object. An example RDF graph showing both schema statements and instance statements is shown in Figure 2.4.

RDF Schema (RDFS) [21] provides a standard vocabulary for describing the classes and relationships used in RDF graphs and consequently provides the capability to define ontologies. Ontologies serve to formally specify the semantics of RDF data so that a common interpretation of the data can be shared across multiple applications. Classes represent logical groups of resources, and a member of a class is said to be an instance of the class.
The RDFS vocabulary offers a set of built-in classes and properties. Two of the most relevant classes are `rdfs:Class` and `rdf:Property`, and some of the most relevant properties are `rdf:type`, `rdfs:domain`, `rdfs:range`, `rdfs:subClassOf` and `rdfs:subPropertyOf`. The `rdf:type` property is used to define class and property types (e.g., the triple `(S, rdf:type, rdfs:Class)` asserts that `S` is a class). `rdf:type` is also used to denote instances of classes (e.g., `(s, rdf:type, S) asserts that s is an instance of S). `rdfs:domain` and `rdfs:range` allow us to define the domain and range for a given property, and `rdfs:subClassOf` and `rdfs:subPropertyOf` allow us to create class and property hierarchies.

A set of entailment rules are also defined for RDF and RDFS [22]. Conceptually, these rules specify that an additional triple can be added to an RDF graph if the graph contains triples of a specific pattern. Such rules describe, for example, the transitivity of the `rdfs:subClassOf` property. The set of entailment rules defined by [22] are shown in Table 2.1.

With respect to other data models, the unique aspects of the RDF model are (1) rela-
Table 2.1: RDFS Entailment Rules. The first column shows the rule name. The second column shows a set of RDF statements for a given RDF graph G, and the third column shows what statements should be added to G.

<table>
<thead>
<tr>
<th>Rule Name</th>
<th>If G Contains:</th>
<th>Then Add:</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs1</td>
<td><code>uuu aaa lll .</code> where <code>lll</code> is a literal</td>
<td><code>_nnn rdf:type rdfs:Literal .</code> where <code>_nnn</code> identifies a blank node allocated to <code>lll</code></td>
</tr>
<tr>
<td>rdfs2</td>
<td><code>aaa rdfs:domain xxx . uuu aaa yyy .</code></td>
<td><code>uuu rdf:type xxx .</code></td>
</tr>
<tr>
<td>rdfs3</td>
<td><code>aaa rdfs:range xxx . uuu aaa vvv .</code></td>
<td><code>vvv rdf:type xxx .</code></td>
</tr>
<tr>
<td>rdfs4a</td>
<td><code>uuu aaa xxx .</code></td>
<td><code>uuu rdf:type rdfs:Resource .</code></td>
</tr>
<tr>
<td>rdfs4b</td>
<td><code>uuu aaa vvv .</code></td>
<td><code>vvv rdf:type rdfs:Resource .</code></td>
</tr>
<tr>
<td>rdfs5</td>
<td><code>uuu rdfs:subPropertyOf vvv . vvv rdfs:subPropertyOf xxx .</code></td>
<td><code>uuu rdfs:subPropertyOf xxx .</code></td>
</tr>
<tr>
<td>rdfs6</td>
<td><code>uuu rdf:type rdf:Property .</code></td>
<td><code>uuu rdfs:subPropertyOf uuu .</code></td>
</tr>
<tr>
<td>rdfs7</td>
<td><code>aaa rdfs:subPropertyOf bbb . uuu aaa yyy .</code></td>
<td><code>uuu bbb yyy .</code></td>
</tr>
<tr>
<td>rdfs8</td>
<td><code>uuu rdf:type rdfs:Class .</code></td>
<td><code>uuu rdfs:subClassOf rdfs:Resource .</code></td>
</tr>
<tr>
<td>rdfs9</td>
<td><code>uuu rdfs:subClassOf xxx . vvv rdf:type uuu .</code></td>
<td><code>vvv rdf:type xxx .</code></td>
</tr>
<tr>
<td>rdfs10</td>
<td><code>uuu rdf:type rdfs:Class .</code></td>
<td><code>uuu rdfs:subClassOf uuu .</code></td>
</tr>
<tr>
<td>rdfs11</td>
<td><code>uuu rdfs:subClassOf vvv . vvv rdfs:subClassOf xxx .</code></td>
<td><code>uuu rdfs:subClassOf xxx .</code></td>
</tr>
<tr>
<td>rdfs12</td>
<td><code>uuu rdf:type rdfs:ContainerMembershipProperty .</code></td>
<td><code>uuu rdfs:subPropertyOf rdfs:member .</code></td>
</tr>
<tr>
<td>rdfs13</td>
<td><code>uuu rdf:type rdfs:Datatype .</code></td>
<td><code>uuu rdfs:subClassOf rdfs:Literal .</code></td>
</tr>
</tbody>
</table>
2.3. SEMANTIC WEB

Relationships are represented as first class objects rather than represented implicitly with, for example, foreign key constraints in the Relational model and (2) a formal semantics is specified according to the defined entailment rules for RDF and RDFS.

2.3.1.2 OWL

OWL is designed to facilitate greater machine interpretability of data (i.e., more logical reasoning) than what is capable with RDF(S). OWL is based heavily on Description Logics and extends the fact-stating abilities of RDF and the class and property defining abilities of RDFS with additional vocabulary [18]. OWL allows the definition of classes as logical combinations (e.g., intersection, union, complement) and allows additional assertions about property types (e.g., I can state that a property is transitive, symmetric, functional, or the inverse of another property). Another important capability of OWL is the ability to define restrictions on the behavior of a property with respect to a given class. For example, I can define the class of Graduate Student as all individuals who are enrolled in at least one course of type Graduate Course. OWL provides three increasingly expressive sublanguages: OWL-Lite, OWL-DL and OWL-Full. OWL-DL is a subset of OWL that allows maximum expressiveness while guaranteeing computational completeness and decidability. OWL-Lite consists of a carefully chosen subset of OWL-DL that eliminates some computational complexity problems that may occur during the inferencing process. OWL-Full provides maximum expressiveness with no computational guarantees.

This dissertation focuses on RDF(S) rather than OWL because I am most interested in exploiting the rich web of named relationships in RDF graphs.

2.3.2 Semantic Analytics

The fundamental premise behind research in Semantic Analytics is that relationships are at the heart of semantics. Sheth et.al. [24] observed the changing focus from documents
to entities and on to relationships, and have investigated a broad variety of issues related to modeling, validating, discovering and exploiting various types of relationships between entities in content. These ideas led to the concept of Metadata Reference Links (MREFs) which proposed associating semantic metadata with hypertext links [25], the development of the InfoQuilt system [26] that investigated support for hypothesis validation style operations, and the OBSERVER system that focused on inter-ontological relationships and multi-ontology query processing [27].

More recently, and in step with the emergence of the Semantic Web, research on complex relationships led to the definition of Semantic Associations [28]. Semantic associations are based on intuitive notions such as connectivity and semantic similarity (see Figure 3.1). In [28] a formalization of semantic associations is given using the RDF data model, and a set of $\rho$-operators for querying semantic associations are defined. The most fundamental of these operators is the $\rho$-path operator. The $\rho$-path operator asks the following question: “How is resource $X$ related to resource $Y$?” over an RDF graph and returns a set of paths connecting $X$ to $Y$. Researchers have also investigated the challenging issue of ranking these paths [29, 30]. Ranking of Semantic Associations was necessitated by the sheer number of such associations even on moderate-size RDF graphs. Even a ranked list of associations could be a daunting task for a user to interpret and may in some cases cause a severe cognitive overload. In a related effort aimed at reducing such a cognitive overload, subgraph discovery techniques have been adapted to discover relatively small but informative subgraphs connecting the entities in the result of a given execution of the $\rho$-path operator [31].

This dissertation extends the work on Semantic Associations to support the extraction of interesting subgraphs from RDF graphs. Subgraph extraction is the first of many operations that will eventually form a suite of investigative tools over semantic metadata.
Contributions

3.1 Research focus & my contributions

A great example of a source of textual data in the biomedical domain is the PubMed database [3] which contains over 16 million abstracts of scientific publications, manually classified into the MeSH topic hierarchy. In this domain, it is rare that the information sought by the user is completely contained in one document. The nature of biomedical research is such that information gathered from multiple publications serve to corroborate or refute a fact. Let us assume for the sake of argument that some publication asserts that “stress can lead to loss of magnesium in the human body”. Another publication might present evidence of the fact that “Migraine Patients seem to be experiencing stress”. It is therefore implicitly expected that a user of PubMed will piece together the partial informa-

![Figure 3.1: Semantic Association between Migraine and Magnesium](image)

28
3.1. RESEARCH FOCUS & MY CONTRIBUTIONS

One critical goal of scientific investigations in biology is that of building multi-dimensional mechanistic models of diseases. The largely manual process of building such models involves the derivation of a set of linked concepts and relationships for etiological factors, molecular systems compositions, signaling and regulatory relationships, pathological processes and consequences, modifying factors, therapeutics, and outcome. Both textual and experimental data are used in this process. While access to experimental data is improving there remains a paucity of systematic literature processing tools. In almost any field of scientific investigation there are vast collections of knowledge in textual form. Large portions of this knowledge are lost to most investigators either due to the sheer volume of text data or due to the limitations in expressivity of current keyword search mechanisms.

This work focuses on the extraction and synthesis of knowledge contained in textual scientific literature. Current access mechanisms to scientific literature over the Web are
still driven by keyword searches. This is a \textit{search-and-sift} paradigm where users have to manually glean relevant information from the large number of results that is typically returned in response to a keyword query. In many cases, users will need to conduct multiple keyword searches and integrate relevant results across searches in order to satisfy their information need.

My aim in this work is to transform the access to scientific literature from the tedious \textit{search-and-sift} paradigm to one involving hypothesis-formulation, testing, extrapolation leading to knowledge discovery.

### 3.1.2 Research outline

To achieve the vision described above my work takes a three-step approach with each step extending the state-of-the-art in their respective computer science disciplines to provide transformative capability of information access to biologists. The steps are:

- **Ontology enabled information extraction**
  - **Extraction and Identification of compound entities**: State-of-the-art entity identification and extraction mechanisms have focused on specific types of entities such as gene names or protein names. I draw a distinction between the identification of entity mentions in sentences and the identification knowledge about these entities expressed in a sentence. This distinction results in my unsupervised approach to the identification of compound entities which express the knowledge in the sentence [32]. Results of this unsupervised identification have been manually curated.
  
  - **Extraction of named relationships between entities**: State-of-the-art relationship extraction focuses on attribute and implicit relationship extraction. Some approaches that extract named relationships are supervised or partially
supervised approaches requiring training data. Training data for relationship extraction in biology is rather difficult to obtain, consequently restricting investigations in relationship extraction to a specific set of relationship types. My initial approach to the relationship extraction problem used constituency parses of sentences and deployed a bottom-up agglomeration approach to generate RDF representations of compound entities and the relationships between them [33]. Although precision was high recall of the extraction suffered considerably. Extraction quality was further hampered by inaccuracies in the identification of entities resulting from the prevalence of compound entities in biomedical text, and inaccuracies in the parsing of sentences. The next approach to I investigate for relationship extraction utilizes a feature rich dependency parse to overcome these problems.

- Joint Extraction of Entities and Relationships: The earlier observation led me to an interesting duality. Entities occurring in a sentence affect the relationships that occur and vice-versa. I therefore investigate this putative, mutually reinforcing behavior of entities and relationships in a sentence. Using limited supervision from a domain ontology I develop an unsupervised approach that uses dependency parses of sentences and employ a few rules to segment biomedical text sentences into entities and relationships. Using these segmentations as initial rough guesses I develop a mutual information-based approach to predict the most likely sub-sequences of the initial guesses that form entities and relationships [34]. The extracted relationships and entities are represented as RDF to facilitate ease of use in analytical application that use the extracted information.

- Semantic Metadata Guided Knowledge Explorations and Discovery

  - A Semantic browser for biomedical literature: In a vision paper that was
perhaps far ahead of its time, Dr. Vannevar Bush [10] suggested that Memex, a personal information store, would help a user “stitch” related documents together into structures that he referred to as trails. He presented a few examples of the use of such trails, some of which are listed below. “The physician, puzzled by her patient’s reactions, strikes the trail established in studying an earlier similar case, and runs rapidly through analogous case histories, with side references to the classics for the pertinent anatomy and histology. The chemist, struggling with the synthesis of an organic compound, has all the chemical literature before him in his laboratory, with trails following the analogies of compounds, and side trails to their physical and chemical behavior. Having extracted RDF from PubMed text my next natural step is to superimpose the extracted RDF back onto the original text and annotate biomedical abstracts with entities and relationships between them. Based on this and inspired by Dr. Bush’s vision I have developed a Semantic Browsing paradigm in which the user of such a Semantic Browser [35] is able to traverse a space of documents based on named relationships between entities of interest.

– Semantic Analytics In the previous section I described analytical operations over text that focused on the discovery of new and interesting connections between entities of interest. In order to support such operations over text data I develop:

* Relevant Connection Subgraph Discovery: Discovering patterns in graphs has long been an area of interest. In most approaches to such pattern discovery either quantitative anomalies, frequency of substructure, or maximum flow is used to measure the interestingness of a pattern. In my approach I introduce heuristics that guide a subgraph discovery algorithm away from banal paths towards more “informative” ones. Given an RDF graph a user might pose a question of the form: “What are the most rele-
vant ways in which entity X is related to entity Y?” the response to which is a subgraph connecting X to Y. My heuristics, based on weighting mechanisms derived from edge semantics suggested by the RDF schema, guide the algorithm towards more informative subgraphs. [31]
Ontology-enabled information extraction

4.1 Relationship extraction from text

In my preliminary investigations I used constituency parses over sentences in biomedical text. Following a bottom-up agglomerative strategy over parse trees I generate RDF from text. The objective of this initial investigation was to show that the extraction of entities and relationships could be used as a foundation to automate knowledge discovery from text.

As discussed earlier MeSH terms (simple entities) may be combined with other simple entities to form composite entities or may occur as modified entities. They may be related to each other by complex relationships. My aim was to identify and extract these three types of entities and the relationships between them occurring in biomedical text. Figure 4.1: Sample sentence from a PubMed abstract showing compound and modified entities.

Figure 4.1: Sample sentence from a PubMed abstract showing compound and modified entities.
Figure 4.2: Schema (UMLS Semantic Network), Instances (MeSH) and documents (PubMed)

5.1 shows an example of these forms of entities. In [33] I proposed a rule-based method for (1) extraction of such complex entities, (2) relationships between them and (3) the conversion of such relationships into RDF. I outline this method here and present an in-depth evaluation of the resulting RDF to ensure that the semantic metadata generated is of a high quality. I present substantial insight into the usefulness of said RDF in the context of knowledge discovery by locating paths between entities. Figure 4.2 provides an overview of the schema, instances and documents used in the process of relationship extraction in [33]. The objective therefore is to extract instance-level relationships between MeSH terms.

MeSH contains 22,807 named entities which include 316 pharmacological names. The UMLS Semantic Network has 136 classes, which are related to each other by one or more of 49 named relationships. Each named entity in MeSH has been manually asserted as an instance of one or more classes in UMLS. Furthermore, MeSH contains synonyms of entities. For instance, “Neoplasms” has the synonym “Tumors”. This obviates the need for entity disambiguation [39]. Further, UMLS also contains synonyms of the 49 relationships. These synonyms have been created by domain experts and used in biomedical abstracts indexed by PubMed. I use this information to spot named relationships occurring in PubMed abstracts. I split biomedical abstracts into sentences and generate RDF on a per-sentence
4.1. RELATIONSHIP EXTRACTION FROM TEXT

basis. Therefore, in my preliminary work I do not address the problem of coreference resolution [40] or pronominal anaphora resolution [41]. We treat MeSH terms as simple entities. These entities may be mentioned in several different contexts in PubMed abstracts. MeSH terms (simple entities) may be combined with other simple entities to form composite entities or may occur as modified entities. They may be related to each other by complex relationships. My aim in relationship extraction [33] was to identify and extract these three types of entities and the relationship between them from biomedical text. The process of converting text from biomedical abstracts into RDF Figure 4.5 consists of the following:

1. An off-the-shelf part-of-speech tagger [42] and a chunk parser [43] used to produce parse trees of sentences in biomedical abstracts.

2. A rule-based post-processing technique used to enrich the generated parse trees. The rules serve to identify complex entities and known relationships between them.

3. These processed trees are then converted into the corresponding RDF structures.

4.1.1 Methodology

Throughout this section, I will use a sample abstract from PubMed to illustrate the steps of my methodology. I chose this abstract at random. The only criterion was that it should contain known entities (MeSH terms) and known relationships (from UMLS) so as to allow me to illustrate all structure types that I extract. The sentence listing of this abstract is shown below.

4.1.2 Part-of-Speech tagging and parsing

Given a sentence, my first step is to tag parts-of-speech in the sentence and parse it to generate a parse tree. I use the SS-Tagger [42] to tag sentences, which claims to offer fast
4.1. RELATIONSHIP EXTRACTION FROM TEXT

[1254239-1] An excessive endogenous or exogenous stimulation by estrogen induces adenomatous hyperplasia of the endometrium.
[1254239-2] The age of the patient and the origin of the estrogenic stimulus however influence the morphology of the hyperplasia.
[1254239-3] Those resulting from exogenous estrogen rapidly regress after the estrogen is discontinued.
[1254239-4] To cure hyperplasias brought on by endogenous estrogen, however, therapy with high doses of gestagen is required.

Figure 4.3: Sample sentences from abstract of PMID-1254239 for illustration (Numbers in the figure indicate PubMed ID-Sentence Number)

tagging (2400 tokens/sec) with state-of-the-art accuracy (97.10% on the Wall Street Journal corpus). This tagger uses an extension of Maximum Entropy Markov Models (MEMM), in which tags are determined in the easiest-first manner. To parse the result of this tagger and produce a parse tree I use the SS-parser [43]. According to the authors, this CFG parser offers a reasonable performance (an f-score of 85%) with high-speed parsing (71 sentences/sec). Although there are possibly more accurate parsers available [44][45][46], the speed of this parser makes it a better choice for us. A comparison of my results obtained by using each of these parsers is something I plan to investigate in the future. I also plan to consider domain specific taggers [47]. The output of the SS-Parser is converted into a main-memory tree representation. The figure below shows such a tree for the sentence 1254239-1. As is shown in Figure 5.1, known entities (MeSH terms) and relationships (from UMLS) are identified in the parse tree. In this example, estrogen (D004967), hyperplasia (D006965) and endometrium (D004717) are the simple entities spotted. The verb “induces” turns out to be a synonym of the relationship “causes” (UMLS ID-T147). Besides recording the known entities and relationships occurring in each node, pointers are maintained to their siblings. For ease of discussion, I group the nodes in the tree into terminal nodes (referred to as _T henceforth) and non-terminal nodes (referred to as _NT henceforth). The text corresponding to a _T node is a single word and that for a _NT node is the phrase formed by its children. This text for each node will be referred to as the token.
4.1. RELATIONSHIP EXTRACTION FROM TEXT

4.1.3 Rule-based post-processing

Entities that occur in biomedical text (or in any text for that matter) seldom occur in their simple unmodified form. They typically occur in a sentence, combined with other entities to form a composite entity or are combined with some modifier to form a modified entity. Consequently, relationships in such sentences may connect two entities which may be either composite entities, modified entities or just simple entities. In the following subsections, I define the three types of entities. We present the rules for identifying them in a sentence along with an algorithm for applying these rules. Finally, I present an algorithm for extracting relationships between the identified entities in the sentence.

4.1.3.0.1 Entity Types  We define simple entities as MeSH terms. Modifiers are siblings of any entity type which are not entities themselves and have one of the following linguistic types:

- determiners (except the words “the”, “an” or “a”)
- noun/noun-phrases
- adjectives/adjective-phrases
- prepositions/prepositional-phrases

Determiners are included in the definition of modifiers to account for negative modifiers such as the words no, not, etc. which identify negative facts. Modified Entities are Simple Entities or other Modified Entities that have a sibling which is a Modifier. Composite Entities are those that are composed of one or more Simple or Modified Entities. The definitions above form a rather simple model that can be used to describe the patterns
that trigger the extraction of entities and relationships from text. In some ways, my model is very similar to the one in [48] which the author uses to learn linguistic structures from text. In [48], the model described treats certain linguistic types (noun phrases, personal pronouns, etc.) occurring in parse trees as nuclei to which adjuncts (adjectival phrases) may be attached. Furthermore, linkers are defined as either conjunctions or punctuation. The purpose of this model is the induction of rules that capture linguistic structure. However, it does not account for named relationships connecting entities.

4.1.3.0.2 Rules for Entity Identification We use the following rules to identify the defined entity types in sentences.

- Rule 1: Modifiers attach themselves to Simple Entities in sentences forming Modified Entities. Therefore, if a Modifier M is a sibling of a Simple Entity SE a Modified Entity is produced.

- Rule 2: Modifiers can attach themselves to other Modified Entities to form other modified entities. Therefore, if a Modifier M is a sibling of a Modified Entity ME another Modified Entity is produced.

- Rule 3: Any number of modified or simple entities can form a composite. Therefore, if one or more Modified Entities ME and Simple Entities SE are siblings then a Composite Entity CE comprising of all these siblings is produced.

4.1.3.0.3 Algorithm for Modified and Composite Entity Identification In this section I describe the algorithm for systematic application of the rules discussed above. The algorithm (Identify_Entities) makes two passes over the parse tree in a bottom-up manner.

- Pass 1
4.1. RELATIONSHIP EXTRACTION FROM TEXT

- **Step 1:** The first pass of Identify Entities begins with Simple Entities found in terminal nodes. It propagates this information about identified simple entities up the parse tree recording this information in all \_NT nodes till a sentence node is reached. This information will later be useful when identifying modified non-terminal entities. Instances of relationships found in \_T nodes are also propagated up in a similar manner. This information will later be useful when identifying the subject and object of a relationship in that sentence.

- **Step 2:** The next step in the first pass is to look at siblings of all \_T nodes carrying simple entities to identify modifiers. For every identified modifier Rule 1 is triggered and the parent node is marked as containing a modified entity.

- **Pass 2**

  - **Step 1:** Next, the set of non-terminal (\_NT) nodes which were marked as carrying entities in Pass 1 is considered. For each node in this set which is not a Verb Phrase (VP) or an Adverb Phrase (ADVP), its siblings are checked.
    * Case 1: If modifiers are found in the siblings Rule 2 is triggered and the parent of the current node is marked as containing a Modified Entity.
    * Case 2: If Simple entities or other Modified entities are found Rule 3 is triggered and the parent node is marked as a Composite Entity.

**4.1.3.0.4 Algorithm for Relationship Identification** After Identify Entities has processed a parse tree, the children of the node marked S (Sentence) contain the information necessary to produce a relationship between the entities involved. To identify this relationship, I use the following algorithm.

If the children of the node marked S contain an entity followed by a relationship and another entity then such a pattern suggests the existence of a relationship between those entities. To guarantee that this relationship R is indeed valid, I use the information from the
4.1. RELATIONSHIP EXTRACTION FROM TEXT

UMLS schema. Note that a candidate subject (Subject) and object (Object) of the suggested relationships could be composite or modified entities as per my definitions. Note that RDFS allows a property to have multiple domains and ranges. Let the domain and the range of R be the sets $\text{domain}(R) = \{C_1, C_2, \ldots, C_n\}$ and $\text{range}(R) = \{C_1, C_2, \ldots, C_m\}$. If $\exists C_i, C_j$ for $1 \leq i \leq n, 1 \leq j \leq m$ such that $\text{Subject} \in C_i$ and $\text{Object} \in C_j$ then I say that the Subject and Object are related by the relationship R. Figure 5.1 shows the relationship “induces” between the modified entity “An excessive endogenous or exogenous stimulation by estrogen” and “adenomatous hyperplasia of the endometrium”.

4.1.4 Experiments to evaluate generated RDF

In my experiments, I tested my methodology for relationship extraction on two datasets. Both datasets are subsets of PubMed. The first is the set of abstracts obtained by querying PubMed with the keyword “Neoplasms”. Unless otherwise specified, PubMed returns all abstracts annotated with a MeSH term as well as its descendants defined in MeSH. As of today, such a query returns over 500,000 abstracts. This forms the dataset which I refer to as ALLNEOPLASMS in this section.

Our objective in extracting triples from the ALLNEOPLASMS set at this point is to test the scalability of my system. In the future, I plan to sample the generated triples to evaluate my methodology in terms of precision and recall. Processing approximately 1.6 million candidate sentences from the ALLNEOPLASMS set resulted in over 200,000 triples. The second dataset is a more focused, smaller set containing abstracts of papers that describe the various roles of Magnesium in alleviating Migraine. Among the eleven neglected connections described in [7], I focus my attention on four connections. These involve the intermediate entities Stress, Calcium Channel Blockers, Platelet Aggregation and Cortical Spreading Depression. To retrieve documents pertaining to these intermediate entities and either Migraine or Magnesium, I searched PubMed with pair-wise combinations of each
4.1. RELATIONSHIP EXTRACTION FROM TEXT

Figure 4.4: (a) Parsing a sentence (b) Rule-based Post Processing of the Parse tree

Figure 4.5: Complex Knowledge Structures & Resulting RDF
intermediate entity with both Migraine and Magnesium. This resulted in a set of approximately 800 abstracts. I call this set MIGRAINEMAGNESIUM. In this case, my objective was to investigate two aspects of my results. They can be characterized by the following questions.

**Question 1**: How effective are my rules in extracting relationships and the entities involved from text? To answer this question, I identify candidate sentences for relationship extraction as those that contain at least two instances of MeSH terms and at least one instance of a named relationship (or its synonym). In the MIGRAINEMAGNESIUM set, I identified 798 such candidate sentences. These sentences are therefore the ones which I expect to generate instances of triples. In my results, these relationships never relate simple entities but always seem to relate modified or composite entities. The number of entities of each type and the relationship instances extracted for the MIGRAINEMAGNESIUM set are as follows: Simple Entities (752), Modifiers (2522), Modified Entities (4762), Composite Entities (377) and Relationships (122). I found that 122 relationship instances were extracted from the 798 candidate sentences. To measure recall accurately, a domain expert would have to read each of the 798 sentences manually to see if they should generate a triple. I plan to conduct just such an experiment in the future. This is however infeasible for larger datasets. I analyzed those candidate sentences that did not produce relationship instances. In my approach to relationship extraction I used the fairly simple rule which expected the subject and the object entity in the same sentence. Close to 90% of the candidate sentences that failed to generate relationships were of a more complex form where the subject is an entity and the object is a sentence itself. Such a structure is an ideal candidate for a reified statement in RDF. I plan to increase the recall of my system by adding a rule to generate such a structure. Of the 122 relationships, 5 were incorrect extractions resulting in 95% precision. Precision directly affects the usefulness of the extracted relationships.

**Question 2**: How useful is the extracted RDF data? To answer this second question, I study the usefulness of the extracted relationships in the context of the Undiscov-
4.1. RELATIONSHIP EXTRACTION FROM TEXT

Figure 4.6: Paths found using bi-directional BFS

<table>
<thead>
<tr>
<th>Path length</th>
<th>Total Number of paths found</th>
<th># of interesting paths</th>
<th>Max. # of named relationships per path</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>260</td>
<td>54</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>4103</td>
<td>1864</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>106450</td>
<td>33403</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 4.6: Paths found using bi-directional BFS

In the RDF produced, every modified entity is "connected" to its constituent modifiers by the umls:hasModifier relationship and to its constituent simple or modified entities by the umls:hasPart relationship. In the case of a composite entity, each of its constituents are "connected" to it by the umls:hasPart relationships. Besides these "connections" there are named relationships connecting entities (SE, ME and CE). As described earlier, the entities Stress, Platelet Aggregation, Spreading Cortical Depression and Calcium Channel Blockers are some of the intermediate entities that serve to describe the beneficial affect that Magnesium has in alleviating Migraine. The usefulness of the RDF extracted from the MIGRAINEMAGNESIUM could therefore be demonstrated if the abovementioned intermediate entities occur in paths connecting Migraine and Magnesium in the RDF. To test for this, I run a simple bidirectional length-limited breadth first search for paths connecting Migraine and Magnesium. I limit the path length since I expect the number of paths to be prohibitively large and long paths are seldom of interest. Only the paths that contain named relationships (besides umls:hasPart and umls:hasModifier) are considered interesting to us. The results of these length-limited searches on the MIGRAINEMAGNESIUM RDF data are shown below.

To see the utility of these paths, I examined some of the paths among those of length 6. I focused my attention on the ones that had 2-3 named relationships. Figure 4.7 shows an example of such a path. This path indicates that migraine is caused by abnormality of platelet behavior (PMID 2701286, sentence number 1), collagen stimulates platelets (PMID 8933990, sentence number 9) and Magnesium has a stimulating effect on collagen induced platelet aggregation (PMID 10357321, sentence number 7). I have included here
4.1. RELATIONSHIP EXTRACTION FROM TEXT

Figure 4.7: An example of an interesting path found

the pointers to the specific sentences in each abstract that corroborates each of the 3 facts above to form the said path. This example clearly demonstrates that my extraction process was successful in extracting relationship instances from PubMed abstracts. It further demonstrates that by virtue of the umls:hasPart and umls:hasModifier these relationship instances extracted from different documents can be chained together to form paths.

4.1.5 Limitations of this approach

Constituency parse trees represent the phrasal nesting structure of a given sentence. This nesting structure is used by my algorithm to extract compound entities and relationships. A consequence of this is that compound entities extracted using my approach are always made up of contiguous token sequences. Pyysalo et al. [50] and Alex et al. [51] recently described discontinuous entities, wherein the tokens forming the entity do not form a contiguous sequence. Since my method utilizes the nested phrasal containment in constituency parses my preliminary work does not identify nested entities.

Non-terminal nodes in constituency parse trees represent groupings of terminal nodes (tokens). The root of a constituency parse tree therefore represents the sentence. When dealing with compound entities of the types discussed, I would like to be able to identify the principal verb and the principal component of each compound entity thereby allowing me to extract the main assertion expressed along with it’s arguments. Constituency parses
do not have the structural properties or the annotation information necessary for the identification of these principal components. Furthermore, sentences in biomedical text often contain a very large number of tokens. In my experience some sentences contained over 250 tokens. A constituency parse of such sentences often results in a very complex parse tree where long range associations between words are often missed. In order to capture these long range dependencies between words in a sentence, I use dependency parse trees. Dependency parse trees have the added advantage that they provide syntactic role specific connections between words, in contrast with mere phrasal containment. In the next section I discuss the use of these role-specific features in the identification of compound entities and relationships.
Unsupervised information extraction

5.1 Unsupervised joint extraction of compound entities and relationships

Relationships in text seldom manifest themselves between simple entities. Compound entities are entities that contain one or more known entities and modifiers, as shown in Figure 5.1. The presence of these modifiers alters the semantics of these compound entities, necessitating the identification of their constituent entities and their types. I therefore focus on identifying compound entities rather than mentions of simple entities. To illustrate the point that entity mentions may differ from the concepts formed by their combinations, I use “hyperplasia”, “endometrium” and “estrogen” as search terms to obtain an exemplary set of PubMed abstracts. One sentence from this set is shown below.

In this sentence estrogen occurs in a modified form as “an excessive endogenous or ex-
ogenous stimulation by estrogen” while hyperplasia and endometrium occur in a composite form as “hyperplasia of the endometrium” further modified by the term adenomatous. This example also shows an example of entities that are composed of non-adjacent tokens (discontinuous entities [51]). Here “an excessive endogenous” and “stimulation by estrogen” together form an entity. The example in Figure 5.1 shows that variants of the entities in MeSH are often found in sentences.

5.1.1 Related Work

Supervised approaches to entity identification, or named entity recognition (NER), typically utilize training data in the form of manually labeled corpora, with tags marking entity mentions (e.g. [49] and [50]). Corpora such as [49] and [50] contain labeled entity mentions (e.g. estrogen, hyperplasia etc. in Figure 5.1). Such tagged corpora are used to collect orthographical [52], contextual [53] and lexical features [54] among others. These features have been shown to perform very well in sequential labeling approaches [54] for identifying specific types of entities such as gene names, protein names etc. [52] & [55]. In these cases the types of entities sought were known and consequently a limited number of atomic observations encoded as features sufficed to identify these entities. However, a quick look at sentences in these corpora shows that token sequences marked as entities are often contained within larger logical entities that are themselves unmarked.

Recently, nested and discontinuous entities [51][56] have received attention. The authors of [51] compare three approaches to identifying such entities through compositions of simple sequential labeling approaches viz. layering, cascading and joint labeling. They acknowledge that their approach is likely to result in a prohibitively large label set when dealing with many entity types. In the biomedical domain it seems possible to find arbitrarily complex nestings of simple entities, making this approach unsuitable for my purposes. For example, in Figure 5.1, a specific process (i.e. stimulation), is the subject of the as-
5.1. **UNSUPERVISED JOINT EXTRACTION OF COMPOUND ENTITIES AND RELATIONSHIPS**

Moreover exogenous and endogenous are modifiers of this subject. These convey additional knowledge about the role of estrogen in the induction of hyperplasia. I therefore draw a distinction between the identification of entity mentions versus the meaning of the compound entities expressed in the sentence. For example, in Figure 5.1, “estrogen” means a biologically active substance, whereas “an excessive endogenous or exogenous stimulation by estrogen” means a biological process initiated by estrogen. It is the identification of the latter that is key in hypothesis validation.

Our approach makes it possible to perform such identification through the use of domain independent linguistic rules. I obviate supervision by using corpus-based information theoretic measures to analyze the structure of compound entities. Here I investigate mutual information as the measure of choice. Theoretically, other measures could be used as well. A side effect of my compound entity identification process is the extraction of possible connections between entities.

Supervised and unsupervised approaches to relationship extraction have been attempted in the past. Machine learning approaches to the extraction of relationships between diseases and their treatments [57] have met with considerable success. Craven and Kumlien [58] present an extraction mechanism for five binary relations involving proteins. Although these approaches yield good precision and recall they still need sets of relations to be defined first. Furthermore, supervised approaches require the expensive human effort to annotate or label corpora to produce training data. In domain-specific settings like biomedicine, owing to the rapid creation of scientific knowledge, the number of new entities and their complex combinations are resulting in rapid proliferation of new concepts in text. I therefore focus my efforts on unsupervised approaches.

Unsupervised approaches to relationship extraction have received considerable attention due to the training data bottleneck plaguing supervised approaches. Based on the interaction with a domain expert, Rinaldi et al. [59] identify a set of relations along with their
5.1. UNSUPERVISED JOINT EXTRACTION OF COMPOUND ENTITIES AND RELATIONSHIPS

morphological variants (bind, regulate, signal etc.) that are of particular interest in the biology domain. Using the dependency parse of GENIA sentences they developed a number of axioms over the dependency patterns that capture the relations that are of interest in this domain. Axiom formulation was, however, a manual process involving a domain expert. Other approaches have relied on hand-coded domain specific rules that encode extraction patterns used to extract molecular pathways [60] and protein interactions [61].

Ciaramita et al.[62] use the entity annotations of the GENIA [50] corpus to learn semantic relationships. In this work the authors extracted patterns indicating relationships from parse trees. The patterns themselves do not encode domain information. However they do assume prior knowledge of the entities in sentences (manual annotation from GENIA). This work provides the main motivation for my work. Because I do not assume that entity annotations are given a priori, my problem is significantly harder.

Our method is based on rules over dependency parse trees. These rules are thus agnostic to domain knowledge and my method does not require prior knowledge of entities. Thus I can apply this method to all biological text without having to reengineer the rules.

5.1.2 Rule based sentence segmentation for compound entity discovery

The main idea behind my approach is to segment dependency trees to facilitate further extraction of (subject, predicate, object) triples. I use rules over dependency relations to determine token sequences that together compose compound entities. In doing so I aim to identify and connect the appropriate entities with relationships. Using the Stanford parser [63] I collect dependencies between tokens in each input sentence. Iterating over the dependencies, I mark words as dominant terms (also referred to as entity/relationship “heads”), or entity/relationship modifiers. Following this step I then establish connections between
5.1. UNSUPERVISED JOINT EXTRACTION OF COMPOUND ENTITIES AND RELATIONSHIPS

Anti-Ro(SSA) autoantibodies are associated with T cell receptor beta genes in systemic lupus erythematosus patients.

![Figure 5.2](image)

Figure 5.2: (A) Sample sentence (B) dependency parse tree (C) Sentence Annotated with types of compound entities discovered.

I use the example in Figure 5.2 to describe my rules. The figure shows a sentence from the GENIA\(^1\). This sentence shows a simple case in which the GENIA annotations mark compound entities correctly. Subsequent examples will deal with the case when entities identified by my method are different from those in corpora such as BioInfer and GENIA. I process dependency trees to determine cut points. Consider the parse tree in Figure 5.2. The dependency types that trigger rules for this tree are shown in Figure 5.3. The \textit{nsubjpass} results in the classification of autoantibodies as a compound entity head.

\(^1\)This sentence is in the Genia corpus version 3.02. This sentence is the title of the abstract 90110496 in GENIA.
and associated as a predicate head. Therefore the link between “autoantibodies” and “associated” indicates that a compound entity governed by “autoantibodies” play the subject role of the predicate “associated”. Similarly with auxpass, part-of-speech tests on the two words in this dependency trigger an association that the word “are” is a modifier of the relationship “associated”. The dependencies prep_with and prep_in describe relational roles associated_with and associated_in, between the relationship associated and their dependents (“genes” and “patients”).

The words “genes” and “patients” are recorded as the syntactic heads of candidate compound entities playing the object role in this sentence. Having recorded these role specific connections between relationships and their subject/object, I recursively expand the heads of candidate compound entities collecting modifiers to compose the token sequence that makes up each compound entity. Since dependency parses are not guaranteed to be acyclic I terminate the recursive expansion when I detect cycles. The recursive expansion procedure results in the entities “T cell receptor beta genes” and “systemic lupus erythematosus patients”. The information recorded in this way is used by the second phase of my algorithm. In this phase the words in a compound entity are used to assign a semantic type to the compound entity. Work addressing the semantics of noun compounds [65] has aimed at inferring semantic types for two word biomedical noun compounds using the MeSH hierarchy. Typing arbitrarily large noun compounds presents a significant research challenge [51]. I use the type of the compound entity head as an indicator of the possible type of the entity. Using the heads of compound entities I match them to single-word MeSH terms. Using UMLS class that this entity belongs to, I assign that class to the compound entity.
This is a simple approach to get good initial guesses of entity types. However, this may not yield correct results in all cases and further investigation extending the work in [65] is warranted. A recent approach to unsupervised extraction described in [62] relied on a sentence simplification strategy where entities (multi-word entities) were replaced with their semantic types. This resulted in a simplified parse tree and allowed for fewer rules to guide the extraction process. My method is similar to this approach applied in the context of un-annotated corpora using dependency trees. The Stanford parsers dependency hierarchy allows for a more principled approach to reducing the number of rules.

### 5.1.3 Rules used for sentence segmentation

In order to minimize the number of rules encoded I use the hierarchy of dependencies provided within the Stanford parser. Dependency types are organized in a hierarchy based on similarity in their grammatical roles. I consider a dependency $d$ to belong to a dependency type $C$ if $d$ is located under $C$ in the dependency hierarchy. This affords me the generalization capability needed to reduce the rule space. I iterate over all edges of a dependency parse and observe the following rules:

1. If a dependency $d(w_1, w_2)$ is within the dependency class SUBJECT, I mark $w_2$ as a head of a subject and $w_1$ as a head of a predicate.

2. If a dependency $d(w_1, w_2)$ is within the dependency class COMPLEMENT, I mark $w_1$ as a head of a predicate and $w_2$ as a head of a object. e.g. $\text{dobj}(w_1 = \text{induces}, w_2 = \text{hyperplasia})$.

3. If a dependency $d(w_1, w_2)$ is within the dependency class PREPOSITION, and $w_1$ is a verb, I mark $w_2$ as the head of a object, $w_1$ as a head of a predicate and combine it with the preposition (e.g. $\text{prep.with}(\text{associated, genes})$ results in “associated with” and “genes”). If $w_1$ is not a verb, I combine $w_1$ and $w_2$ as a modified entity.
e.g. prep_of($w_1 = \text{hyperplasia}, w_2 = \text{endometrium}$) results in “hyperplasia of endometrium”.

One effect of using very few rules is that my algorithm is opportunistic by design. This often results in large compound entities composed of many words. In many cases words that are marked as a part of a compound entity do not belong inside an entity. There are two situations in particular that I commonly observed:

1. Verbs mistakenly identified as nouns leading to their inclusion in compound entities. This mistake can be attributed to the parser.

2. Words that are mere connectors between entities (e.g. punctuations being included in entities). Furthermore, compound entities composed using simple ones connected by prepositions are very prevalent in biomedical text. There are therefore two reasons to identify points where compound entities should be split. One reason is to reduce false positives in compound entity identification and the second reason is to identify the component of a compound entity and the ways in which they are combined.

Our initial experiment testing the utility of my rules was run on the BioInfer corpus [50]. This corpus contains 1100 sentences that have been either manually or automatically annotated. These annotations mark nested and discontinuous entities as well as relationships. My algorithm produced 5614 entity guesses from these 1100 sentences without using the existing annotation information. The figures below show compound entities discovered by my system and segmentations of my example sentences into relationships and compound entities.

The results above are obtained by using the 4 rules described in my algorithm. A close inspection of the entities in Figure 5.4 shows that all entities except the last are correct. The last entity listed in Figure 5.4 seems to be mistakenly tagged as an instance of UMLS class umls:Social_Behavior. This is because I use the head of the compound entity as a simple
5.1. UNSUPERVISED JOINT EXTRACTION OF COMPOUND ENTITIES AND RELATIONSHIPS

Figure 5.4: Compound entities identified (Note: prefix subscripted text indicates UMLS type if applicable)

Figure 5.5: Sentence segmented into relationships and compound entities

[anti-Ro(SSA) autoantibodies UMLS:Immunologic_Factor] are associated with [T cell receptor beta genes UMLS:Gene_or_Genome] in [systemic lupus erythematosus patients UMLS:Patient_or_Disabled_Group].

[An excessive internal or external stimulation by estrogenstimulation] induces [adenomatous hyperplasia of the endometrium UMLS:Pathologic_Function].
5.1. UNSUPERVISED JOINT EXTRACTION OF COMPOUND ENTITIES AND RELATIONSHIPS

the simultaneous quantification of myosin heavy chain, myosin light chain, phosphorylatable myosin light chain

Figure 5.6: A compound entities that is a candidate for splitting

mechanism to assign a type to compound entities. Using the word “role” as the head, this entity’s type is assigned. Future work will investigate this in further detail, considering issues pertaining to the semantics of noun compounds [65].

A significant proportion of the compound entities that I found were other compound entities put together using punctuation such as the entity shown in Figure 5.6. This entity does indeed form the subject of an assertion in a BioInfer sentence\(^2\). However, the correct interpretation of this entity is as follows: all three types of myosin are modified by the words “simultaneous quantification”. As per my previous observations, directly comparing the compound entities predicted using only the rules will show poor performance. This is due to a preponderance of entity predictions like the one in Figure 5.6. To address this issue I developed an entity prediction strategy that leverages corpus statistics.

5.1.4 Predicting constituents of Compound entities via corpus statistics

The main intuition behind my entity prediction strategy is as follows. Subsequences of tokens belonging to a predicted compound entity, which co-occur across a large corpus, dependent on each other in the same manner, are likely to themselves form sub-entities. Here I do not use the typical definition of co-occurrence (i.e. adjacent terms). Instead, I use a role-specific definition of co-occurrence which treats two terms as co-occurring if they are connected by a dependency of any type in a dependency parse tree. Furthermore, for this (or any form) of co-occurrence statistic to be effective, a large corpus is required.

\(^2\)See sentence with id 1610 in the BioInfer corpus @ http://mars.cs.utu.fi/BioInfer/
5.1. UNSUPERVISED JOINT EXTRACTION OF COMPOUND ENTITIES AND RELATIONSHIPS

5.1.4.1 Implementation details

In order to compute corpus-wide statistics I “expanded” the BioInfer corpus by increasing the number of sentences pertaining to each entity in BioInfer. Using the known entities in BioInfer as seed queries, I queried the PubMed database obtaining 100 abstracts corresponding to each entity. This resulted in a set of approximately 77,000 abstracts. Splitting these abstracts into sentences yielded approximately 850,000 sentences. Using the Stanford dependency parser I parsed these 850,000 sentences. I then built a Lucene index that indexes each dependency of the form rel(gov,dep) using the name of the dependency (i.e. rel) the governor term and the dependent term (i.e. gov and dep respectively) as fields.

5.1.4.2 Mutual Information as a measure of entity viability

Mutual information [66] was introduced as a measure for discovering interesting word collocations. Intuitively, mutual information measures the information that two random variables share: it measures how much knowing one of these variables reduces my uncertainty about the other. If two variables are independent, the mutual information is zero. Mutual information increases when two words occur together often. Consider a pair of words $w_i$ & $w_j$. The pointwise mutual information between $w_i$ & $w_j$, $I(w_i, w_j)$ is computed as follows:

$$I(w_i, w_j) = p(w_i, w_j) \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}, \quad \text{where} \quad p(w_i, w_j) = p(w_i)p(w_j|w_i) \quad (5.1.1)$$

The maximum likelihood estimate for $p(w_i)$ or $p(w_j)$ is the ratio of the frequency of occurrence of the corresponding word with the total number of words in the corpus, and the maximum likelihood estimate for the conditional probability $p(w_j|w_i)$ is the ratio of the frequency of the co-occurrence of $w_i$ & $w_j$ with the frequency of $w_i$. 

57
5.1. UNSUPERVISED JOINT EXTRACTION OF COMPOUND ENTITIES AND RELATIONSHIPS

September 21, 2008

\[ p(w_j|w_i) = \frac{\text{count}(w_i, w_j)}{\text{count}(w_i)} \]  

(5.1.2)

The definition above is based on co-occurrence of words in a sentence. In my case, however the definitions of the co-occurrence counts are based on the number of dependencies connecting the two words. The idea being that, non-adjacent tokens can, in some cases, be combined to form entities. Therefore:

\[ p_d(w_j|w_i) = \frac{\text{count}_d(w_i = \text{dep} \land w_j = \text{gov}) + \text{count}_d(w_j = \text{dep} \land w_i = \text{gov})}{\text{count}_d(w_i = \text{dep} \lor w_i = \text{gov})} \]  

(5.1.3)

and \( p_d(w_i) = \frac{\text{count}(w_i)}{N} \) where \( N \) is the total number of dependencies across the entire corpus. \( \text{count}_d(w_j = \text{dep} \land w_i = \text{gov}) \) represents the number of dependencies that have \( w_i \) \& \( w_j \) as governor AND dependent respectively while \( \text{count}_d(w_i = \text{dep} \lor w_i = \text{gov}) \) represents the number of dependencies in which \( w_i \) is either the governor OR the dependent.

5.1.4.3 Preliminary Results

Using this dependency-based mutual information as a guide I predict token subsequences of compound entities that are most likely to form entities themselves. Entities predicted by my algorithm are shown in Figure 5.4, Figure 5.5 and Figure 5.6. Using these compound entities as a starting point my sub-entity predication mechanism groups tokens to form sub-entities. This results in a segmentation of compound entities into its constituents. Some results of this process are shown below.
### 5.1. UNSUPERVISED JOINT EXTRACTION OF COMPOUND ENTITIES AND RELATIONSHIPS

<table>
<thead>
<tr>
<th>Compound Entity</th>
<th>Constituent Entities Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cdc42-induced nucleation of actin filaments</td>
<td>nucleation of actin filaments, actin filaments</td>
</tr>
<tr>
<td>affinity of yeast profilin for rabbit actin</td>
<td>affinity of yeast profiling, affinity of yeast profilin for rabbit actin</td>
</tr>
<tr>
<td>main inhibitory action of p27, with the cyclin E/cyclin-dependent kinase 2 (Cdk2)</td>
<td>main inhibitory action, main inhibitory action of p27, with the cyclin E/cyclin-dependent kinase 2 (Cdk2)</td>
</tr>
<tr>
<td>tumor necrosis factor receptor</td>
<td>tumor necrosis, tumor necrosis factor receptor</td>
</tr>
<tr>
<td>actin-binding proteins of low molecular weight</td>
<td>actin-binding proteins, actin-binding proteins of low molecular weight</td>
</tr>
<tr>
<td>Inactivation of the Rb pathway cell lung carcinoma</td>
<td>cell lung carcinoma, Inactivation of the Rb pathway cell lung carcinoma</td>
</tr>
<tr>
<td>Three components of Drosophila adherens junctions</td>
<td>components of Drosophila adherens junctions, Drosophila adherens junctions</td>
</tr>
</tbody>
</table>

A comprehensive list of entities predicted by my system is available at [http://knoesis.wright.edu/research/semweb/projects/textMining/jbi2008](http://knoesis.wright.edu/research/semweb/projects/textMining/jbi2008).

These entities were predicted using the 1100 sentences that are in BioInfer. Corpus statistics gathered from 850,000 sentences were used to obtain entity predictions based on mutual information as discussed. In addition to entities my system predicts possible relationship triples that might hold between the entities in a sentence. The table below shows some
of these relationships\(^3\).

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Sentence Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>increased</td>
<td>A pre-treatment of cells with SGE from partially fed ticks in amounts salivary glands → increased → the level of both viral nucleocapsid ( N ) protein phosphoprotein ( P ) in a dose-dependent manner</td>
</tr>
<tr>
<td>inhibits</td>
<td>alpha-catenin → inhibits → beta-catenin signaling</td>
</tr>
<tr>
<td>inhibits</td>
<td>MgCl2 → inhibits → these effects of profilin, most likely</td>
</tr>
<tr>
<td>causes</td>
<td>The cardiac myosin heavy chain Arg-403 Gln mutation → causes → hypertrophic cardiomyopathy</td>
</tr>
<tr>
<td>causes</td>
<td>Moreover, addition of profilin to steady-state actin filaments → causes → slow depolymerization</td>
</tr>
<tr>
<td>causes</td>
<td>(11-22 microM) into infected PtK2 cells → causes → a marked slowing of actin tail elongation and bacterial migration</td>
</tr>
<tr>
<td>binds</td>
<td>the cytoplasmic domain of E-cadherin → binds → either beta-catenin or plakoglobin</td>
</tr>
<tr>
<td>binds</td>
<td>a constituent → binds → RBC alpha-spectrin antibody plus the presence of significant quantities of actin</td>
</tr>
</tbody>
</table>

This table clearly show the benefit of my approach. Relatively large entities with several possible sub-entities are identified by my system. Evaluation of compound entities and relationships extracted requires human subject evaluation. This becomes clear from the occurrence of entities like “An excessive endogenous or exogenous stimulation by estrogen”. Such a complex token sequence is seldom marked as an entity in corpora. A simple lookup in a biomedical lexicon will identify estrogen as an entity. Whereas the principal

\(^3\)The complete set of extracted relationships is available at http://knoesis.wright.edu/research/semweb/projects/textMining/jbi2008.
part of the compound entity conveying meaning in this sentence is the word “stimulation”. Furthermore this term is “double” modified by the terms exogenous and endogenous. The semantically correct interpretation of the compound entity is both exogenous AND endogenous types of stimulation by estrogen. A method is therefore required to:

1. separate the terms endogenous and exogenous and independently attach them as modifiers to the term stimulation, and

2. connect stimulation to estrogen and generate two distinct entities.

Figure 5.7 shows the parse tree for the same sentence as the one in Figure 5.1. I replaced the two words exogenous and endogenous with semantically similar words external and internal. The corresponding parse is seen in Figure 5.8.

The shaded sequence of dependencies was parsed correctly. The words internal and external are connected by the dependency conj_or and the word stimulation forms the subject head for the relationship induces. Comparing this parse to the one in Figure 5.7 it is clear that the erroneous parse in Figure 5.7 was due to the fact that the parser model I used in the Stanford parser was the one trained on news text from the Wall Street Journal. I believe that training the parser on biology-specific text would reduce these kinds of errors, thereby improving extraction accuracy.
5.1. UNSUPERVISED JOINT EXTRACTION OF COMPOUND ENTITIES AND RELATIONSHIPS

Figure 5.8: The impact of lexical guidance in parsing accuracy

An excessive internal or external stimulation by estrogen induces adenomatous hyperplasia of the endometrium.

[A] An excessive internal or external stimulation by estrogen induces adenomatous hyperplasia of the endometrium.

[B] [An excessive internal or external stimulation by estrogen stimulates] induces [adenomatous hyperplasia of the endometrium (UMLS:Pathologic_Function)].
Evaluating Extraction Results

In order to evaluate the quality of my extraction results I sought the help of domain experts at the Cincinnati Children’s Hospital Medical Center Research Foundation. A research group here at the Division of Biomedical Informatics, lead by Dr. Bruce Aronow was interested in studying diseases related to the kidney. I therefore used the keyword “renal” against a database of text known as OMIM. OMIM (Online Mendelian Inheritance in Man) is a repository of phenotype information for genes in humans and other model organisms. I therefore extracted triples from OMIM text.

6.1 Extracting Triples from OMIM

Using the query term “renal” against OMIM[^1], I collected the 1248 records pertaining to disease phenotypes returned by this query. Splitting these text records into sentences resulted in 90,000 sentences. I use this set for my experiments in this section. Running my extraction algorithm on these sentences resulted in 328 Megabytes of RDF, containing approximately 153K triples. An ideal evaluation for the relationships and compound entity extraction proposed here would involve comparison with respect to existing manually annotated corpora. My work draws a distinction between the identification of entity mentions and their relationships versus compound entities and their relationships expressed in the sentence. This makes it difficult to evaluate my results against existing manually annotated

corpora that have narrow objectives and scope. However, a quantitative evaluation of relationship and compound entities is necessary. To this end I have built an evaluation tool that allows the user to perform a per-predicate evaluation. The generated RDF is loaded into a Jena\textsuperscript{2} model. The ARQ\textsuperscript{3} query language extension for SPARQL\textsuperscript{4} is used to formulate queries over the model and generate a list of relationship names sorted on their frequency of occurrence in the RDF. From this sorted list of relationships I picked the following relationships at random from among the more frequent ones:

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Number of triples per relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>encodes</td>
<td>397</td>
</tr>
<tr>
<td>is expressed</td>
<td>356</td>
</tr>
<tr>
<td>induced</td>
<td>305</td>
</tr>
<tr>
<td>produced</td>
<td>221</td>
</tr>
<tr>
<td>inhibited</td>
<td>181</td>
</tr>
<tr>
<td>derived</td>
<td>172</td>
</tr>
<tr>
<td>affect</td>
<td>166</td>
</tr>
<tr>
<td>binds</td>
<td>140</td>
</tr>
</tbody>
</table>

My evaluation tool (Figure 6.1) allows the user to iterate over the triples involving each relationship selected from the list above and juxtaposes the original sentence from which the triple was extracted, with the triple. The user is therefore able to see whether the entities and the triple involving them are indeed correct. The user rates each subject, object and the triple on binary a rating system (correct/incorrect). These ratings are recorded in a Lucene index which is later read to measure the precision of the extraction for each relationship type and their corresponding subject-object pairs. Since my evaluation tool is web-based multiple users are able to evaluate different predicates whose ratings are stored

\textsuperscript{2}http://jena.sourceforge.net/
\textsuperscript{3}http://jena.sourceforge.net/ARQ/
\textsuperscript{4}http://www.w3.org/TR/rdf-sparql-query/
6.1. EXTRACTING TRIPLES FROM OMIM

![Diagram](image1.png)

Processing 0/198
Writing to 128557431993

(a)

(b)

Figure 6.1: Juxtaposition of a sentence from OMIM and the corresponding RDF
in unique indexes for later retrieval. In my experiment each user therefore evaluates a total of 1938 instances of triples and their corresponding subject-object pairs corresponding to the 8 relationship types shown in the table above. The process of deciding whether an entity/triple is correct is based on the readers interpretation of the sentence but follows some generic rules. A valid entity should be treated as correct if it does not need any other words from the sentence to describe it correctly and does not have any unnecessary words that do not refer to the head of the entity. A valid triple should be treated as correct if it has the correct and full subject/object, or some word that represents it (such as a preposition). I employ this relaxation since my system does not currently address pronominal reference resolution. Another objective of my experiment is to see if generality or specificity of a relationship affects the accuracy with which a triple containing that relationship is extracted. While the sparsity of rules used by my system affects this accuracy in general, evidence presented by Carroll et. al. [20] suggests that my rules do cover a majority of the dependency types. It therefore seems likely that I will be able to show the effect of relationship specificity on extraction quality. My long term goal in studying these differences is to try and map them onto linguistic patterns that are indicative of certain relationships and entity types, thereby enabling the creation of type-specific transducers for entity and relationship extraction. The results in Figure 6.3 and Figure 6.2 show precision comparison of triples and the entities respectively across the 8 relationship types for each evaluator. A close look at the results of my experiments in Figure 6.2 shows that the relationships “encodes”, “is expressed” and “induced” are extracted with much higher precision than “affect”, “binds” and “produced”. The results in Figure 6.3 show that the accurate identification of entities is closely tied to the relationship type. In other words, when domain specific relationships such as “encodes”, “is expressed” and “induced” occur in a sentence, my rules are able to identify entities more accurately than in the cases where general relationships such as “affect” occur. In my experiment I discovered 46,490 distinct predicates. Many of these are variants of other predicates. Normalizing these variations might allow me to get stronger
patterns indicating entity prediction accuracy being affected by relationship type. However this variant normalization of predicates is not a trivial task and beyond the scope of this paper.
Figure 6.3: Entity extraction-percent correct triples - measured across relationship types compared across evaluators
Applications

7.1 A Semantic Browser for Biomedical Literature

7.1.1 Related Work

Typically the term “Semantic Browser” has been used in literature to describe browsers for semantic metadata. One of the earliest such efforts was the CS AKTiveSpace application [38]. This application provided a way to explore the UK Computer Science Research domain across multiple dimensions for multiple stakeholders ranging from funding agencies to individual researchers. Semantic metadata about these agencies, researchers, etc. represented in RDF was used to drive this application. Longwell\(^1\) and Tabluator\(^2\) are recent examples of faceted browsing tools for RDF. These tools provide an application-independent browsing perspective for RDF data, allowing the user to select facets of interest to customize the browsing process. I see these tools as browsers for semantic metadata. My definition of semantic browsing can be seen as utilizing Semantic metadata (represented as RDF) to browse data (in our case text). This requires the extraction of entities and relationships from text and the superimposition of this metadata onto text. I believe that this superimposition is crucial to support future knowledge discovery applications over text where hypotheses are corroborated via evidence from text. In the following sub-section I

---

\(^1\)http://simile.mit.edu/wiki/Longwell
\(^2\)http://www.w3.org/2005/ajar/tab
describe the construction of our Semantic Browser for Biomedical Literature.

### 7.1.2 Using Semantic Metadata to Browse Text

My unsupervised method for sentence segmentation described earlier produces compound entities and relationships that are represented in RDF. Many of these compound entities form logical subjects and objects of assertions in sentences. However they are often made up of simpler entities connected by punctuation and connectives such as prepositions or conjunctions. I presented a method based on dependency-based mutual information to predict the subentities of compound entities. My results have shown that these predicted subentities are more likely to be keywords that users of PubMed are familiar with. Use of these intuitive simple entities as keywords to search for documents is a major merit of PubMed making it very simple to use.

In designing advanced methods to support analytics over literature, it is desirable preserve this simplicity of use. However, I would like to leverage semantic interpretations of the text while simultaneously linking inter-document content using these interpretations. In this section I describe the construction of our Semantic Browser that uses sub-entities, compound entities and relationships between compound entities to support browsing operations over literature.

Figure 7.2 shows the flow of text and extracted semantic metadata into our semantic browser. My RDF extraction mechanism creates a main-memory model using Jena\(^3\). This model is queried using SPARQL to retrieve all extracted facts involving named relationships (e.g. causes, inhibited etc.). The 6.1 shows a sample sentence with the RDF extracted from it. Note that the extracted RDF maintains a pointer to the location where the original sentence occurred in the original corpus (OMIM). The number 191170-48-1 indicates that this sentence is the first sentence in the 48th text section of the OMIM record with

\(^3\)http://jena.sourceforge.net/
7.1. A SEMANTIC BROWSER FOR BIOMEDICAL LITERATURE

September 21, 2008

Figure 7.1: Sample RDF used to build Lucene index in the Semantic Browser

ID 191170. Each compound entity and triple that it participates in has hasSource property with the value uniquely identifying the sentence from which it was extracted. Every sentence from the corpus is therefore indexed on the subject, predicate and object of the facts contained within that sentence.

7.1.2.1 Design of the Semantic Browser

7.1.2.1.1 Implementation details The browser’s user interface and interaction with the Lucene index are built using Ajax technology. Ajax (asynchronous JavaScript and XML),
7.1. A SEMANTIC BROWSER FOR BIOMEDICAL LITERATURE

Figure 7.2: Text and Semantic Metadata combined in the Semantic Browser

Figure 7.3: Design of the Semantic Browser
7.1. A SEMANTIC BROWSER FOR BIOMEDICAL LITERATURE

or AJAX, is a group of interrelated web development techniques used for creating interactive web applications or rich Internet applications. With Ajax, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. Data is retrieved using the XMLHttpRequest object or through the use of Remote Scripting in browsers that do not support it. Despite the name, the use of JavaScript, XML, or asynchrony is not required.  

7.1.2.1.2 Browsing mechanism  The idea behind the semantic browser is to allow the user to navigate the space of documents using semantic connections between entities or pseudo-semantic connections between words in text. I use the latter in this case to demonstrate the usefulness of this browsing paradigm. Using the stanford dependency parser I parsed a 90,000 sentence subset of literature from PubMed. I then built a Lucene index on the dependencies produced by the parser. In other words, a dependency represented as \( \text{rel}(\text{dep}, \text{gov}) \) generated by a parse of a sentence \( S \) is used to index the sentence \( S \). Here \( \text{dep} \) & \( \text{gov} \) represent words that are related by a dependency \( \text{rel} \). As shown in Figure 7.4 hovering on a term shows the list of dependencies that relate it to other terms across the corpus. Choosing one of these dependencies lists a set of terms to which that term is connected via the chosen dependency (Figure 7.5). Choosing one of the target terms result in a list of documents that contains the said target term (Figure 7.6). Selecting one of these documents displays it and records the one hop connection between the terms as a part of a trail shown in Figure 7.6. I have used dependencies here to demonstrate the browsing process. However the semantic browser will utilize named relationships like “causes”, “inhibits” etc. to connect entities identified by our extraction mechanisms discussed in previous chapters.

\( \text{http://en.wikipedia.org/wiki/AJAX} \)
7.1. A SEMANTIC BROWSER FOR BIOMEDICAL LITERATURE

Figure 7.4: Navigation in the Semantic Browser - Hover on term to get relationships to other terms

Figure 7.5: Navigation in the Semantic Browser - Choose target term to get documents containing target term
7.1. A SEMANTIC BROWSER FOR BIOMEDICAL LITERATURE

Figure 7.6: Navigation in the Semantic Browser - Choose document containing target term
- Note trail forming
7.2 Semantic search

7.2.1 Related Work

Beyond keywords search, the idea of using semantics to improve search became popular with the advent of the Semantic Web vision. This idea was first discussed by Sheth et. al. [2] where sample search applications using semantics were described. Hammond & Sheth et. al.[23] posited the idea of using domain knowledge to augment search. The idea here was to enrich text content with semantics by matching terms in the text with entities in a knowledge base, and using the knowledge base to enrich the users understanding of the text. This can be seen as a top-down approach to Semantic Search. Guha et. al. [37] introduced the notion of a “Research Search” as a type of Semantic Search. The idea here was to discover information about an entity by aggregating information from different documents. A more specific form of this search has been implemented in the MedIE\(^5\) system. This system uses Head-Driven Phrase Structure Grammar (HPSG) to parse sentences in PubMed obtaining predicate-argument structures that identify the arguments of each verb in the sentence. Using this information the creators of this system have implemented a search engine over approximately 15 million PubMed abstracts. The search interface allows users to search for abstracts using a subject, a verb and an object. Subjects and objects are matched to sentence fragments using a region algebra.\(^6\) Consequently, MEDIE does not extract entities or relationships from text. It merely uses the arguments of a verb identified by the HPSG parser to index documents on the subject, verb and object triple. Query term matches in the index are extended using a region algebra. Consider the example sentence “Whereas transient activation of JNK upon TNF treatment is associated with cellular survival, prolonged JNK activation contributes to cell death”. The MEDIE system treats the word “is” as a relationship and attaches the word “associated” with the object when in re-

\(^5\)http://www-tsujii.is.s.u-tokyo.ac.jp/medie/

\(^6\)Region algebra represents document structures by specifying containment and ordering relations of regions, such as a fragment of text from a beginning tag to the corresponding end tag.
7.2. SEMANTIC SEARCH

Figure 7.7: Searching for text using a subject

ability the correct relationship is “is associated with”. Their approach will not work for our long term goal of knowledge discovery from literature. In contrast with the MEDIE system I seek to extract compound entities, identify their constituent entities, ground the entities by asserting them as instances of a schema and extract simple (e.g. causes) and complex relationships (e.g. is rarely caused by) between them.

7.2.2 Using Semantic Metadata to Search for Text

The Semantic Browser described in the previous section was driven by keyword search. Instead of using any random document returned by these keyword searches as a starting point for the browsing process, I provide a mechanism to obtain a more focused and specific starting document. I achieve this specificity by providing a triple based search wherein the user can search using either the subject, predicate or object of a triple extracted by our extraction mechanism. The user could also use all three in conjunction to constrain the search. Figure 7.7 shows the use of the subject “neoplasia” as a search term. This returned four sentences that were indexed by the word “neoplasia” in the subject field of the index. A further constrained version of the search shown in Figure 7.8 which shows the addition
7.2. SEMANTIC SEARCH

Kaneda and Feinberg reviewed the study of Sakatani et al., showing that a mouse model of LOI of the Igf2 gene, which shows aberrant activation of the normally silent maternal allele, modifies the risk of intestinal neoplasia caused by mutations of the adenomatous polyposis coli (Apc) gene. 

Figure 7.8: Search for text using a triple

Figure 7.9: Browse the result in the Semantic Browser

of the predicate and object, reducing the results to one specific sentence that contain the triple “neoplasia caused by mutations”. The preposition “by” indicates the directionality of the relationship “caused”. My extraction mechanism does pick up this nuance. This version of semantic search is a preliminary version. Future work will extend this to a version that allows the user to map keywords to fragments of RDF and thereby compose SPARQL queries against the generated RDF to retrieve documents containing matching RDF fragments.
8.1 Discovering Informative Connection Subgraphs in Multi-relational Graphs

Given an RDF graph and any two entities X and Y there could be a myriad of relatively short chains (i.e. six degrees) of relationships linking the two. Hence the need for some way of semantically constraining and discovering the possible ways in which X and Y could be related. Faloutsos et.al. [70] address this issue by developing an algorithm to extract relatively small connection subgraphs. They define the Connection Subgraph Problem as follows:

**Definition 8.1.1.** Given: an edge-weighted undirected graph G, vertices s and t from G and an integer budget b.
Find: a connected subgraph H containing s and t and at most b other vertices that maximizes a “goodness” function g(H).

Faloutsos et.al. [70] applied their techniques to a graph where nodes represented famous people and the edges between these nodes represented strength of acquaintance between them. These connection strengths were derived from name co-occurrences in Web.
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN MULTI-RELATIONAL GRAPHS

September 21, 2008

All edges in their dataset therefore have exactly the same interpretation. Clearly this weighting scheme will not work for finding relevant subgraphs in RDF graphs. Also, naively using a uniform weight on each edge is insufficient, as the semantics of each property type (edge) in RDF is different. Therefore a systematic way of weighting edges based on the semantics conveyed by the ontology represented using RDF schema [21] is needed. To adapt the approach in [70] to the more general case of an RDF graph:

- I propose heuristics for edge weighting that depend indirectly on the semantics of entity and property types in the ontology and on characteristics of instance data. More specifically, I define class and property specificity, Instance Participation Selectivity and a Span Heuristic.

- I evaluate the generated subgraphs using path ranking schemes suggested in [29], [72] and [73].

- I present empirical evidence that my weighting schemes do indeed help identify “informative” patterns in the output subgraphs.

- I present results that support the electricity based [70] model of path “importance” for RDF graph relevance.

8.1.1 Related Work

Reasoning and knowledge discovery over graph data models has been studied in the Graph mining community and more recently in the context of the Semantic Web. The remainder of this section highlights work which is most relevant to ours. The work most directly related to graph-based knowledge discovery and reasoning for the Semantic Web is that of Semantic Associations which were first introduced in [67]. Semantic Associations (termed $\rho$-operators) represent meaningful directed paths in an RDF meta-base. Anyanwu and Sheth define the $\rho$-path operator among others. Two entities X and Y are said to be
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN MULTI-RELATIONAL GRAPHS

ρ-path associated if there exists a sequence of properties (relationships) starting at X connecting intermediate entities and ending at Y. The nature of web data [68] often leads to an overwhelming number of associations between two entities. To combat this problem, [29] and [72] propose to rank Semantic Associations. As an alternative approach, the method in [73] filters the search space before computing associations. They adapt the HITS algorithm [74] to compute importance of Semantic Web resources and then only consider nodes with importance greater than some threshold when computing Semantic Associations. Their preprocessing step based on importance thresholds is likely to discount those paths that contain even a single unimportant node. My approach to this problem is fundamentally different from these two. I try to find the “best” set of associations which contain a visually comprehensible number of resources. There has been a considerable amount of work done in the field of graph mining to detect patterns in graphs. Patterns discovered are characterized either by their anomalous nature or frequent occurrence, among other things. Efficient algorithms have been developed for many variations of the frequent subgraph discovery problem [75][76][77]. Community and group detection is another well-studied graph mining problem which attempts to discover communities and groups based on link analysis. The problem has been studied on both the web graph [78][79] and other data sets [80]. These graph mining problems focus on graphs with single node types and single edge types, however. For the Semantic Web and Link Mining I need algorithms which take into account the semantics of different node and edge types. Community detection and mining in multi-relational networks has recently received a lot of attention [81]. Novel Link Discovery was introduced in [72] and involves finding novel paths between entities, novel loops, and significantly connected nodes. The methodology used in this work considers different node and edge types but differs from ours in that importance is determined purely from rarity. Also the paths examined are considerably shorter than the ones I examine.
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN MULTI-RELATIONAL GRAPHS

8.1.2 Algorithms

My method for finding a connection subgraph between two RDF resources is based on the algorithms from [70]. The authors present an algorithm for extracting a so-called candidate graph from an input graph. They also propose an algorithm based on electrical circuits to extract a display graph from the candidate for a given budget b. For my purposes I refer to these as Candidate $\rho-$graph and Display $\rho-$graph. I assume that the properties (edges) in the RDF graph are bidirectional (i.e. every relationship has a corresponding inverse relationship). This assumption is necessary because two resources may not be connected by a directed path but by a path which contains inverse relations. To ignore this path could exclude vital information about the connections between the entities. My approach modified the algorithm presented in [70] using hueristics to adjust the edge weight. This adjustment ensures that the most informative paths are added during the construction of the display graph.

8.1.2.1 Candidate graph Generation Algorithm

The candidate $\rho-$graph generation algorithm is used to prune the search space in very large graphs. It is based on a notion of distance between two nodes. The algorithm grows a set S around the source node s and a set T around the sink node t (s and t are referred to as the roots of their respective sets) until a certain threshold is met: a maximum number of total nodes or maximum number of cut edges between S and T. At each iteration, a pending list is maintained for each of these sets which consists of those nodes $n \notin S$ and $n \notin T$ and adjacent to some node $k \in S$ or $k' \in S$. The sets S and T are expanded by choosing from the pending list the node with shortest distance to either s or t. Let $u'$ be the predecessor of u (the node adjacent to u on the shortest path to its root). For an edge (u, v) the distance between u and v is given by:
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN
MULTI-RELATIONAL GRAPHS

\[ \text{distance}(u, v) = \log \left( \frac{(\text{deg}(u) + \text{deg}(v))^2}{w(u, v) \times \beta_{u' \rightarrow u \rightarrow v}} \right) \]  

(8.1.1)

The calculation of \( \beta_{u' \rightarrow u \rightarrow v} \) and \( w(u, v) \) is explained in the following section. The length of a path is the sum of the distances between its edges. The aim of my initial experiments is to determine the quality of the Candidate \( \rho \)-graph in terms of its ability to capture the best paths between the query endpoints.

8.1.2.2 Display graph Generation Algorithm

The display generation algorithm extracts a small connection subgraph from the input graph. In [70] the authors present a rather elegant solution to this by modeling the graph as an electrical circuit where the edge weights represent the conductance values in the circuit. They use the fact that current flows from high voltage to low voltage to impose direction on an otherwise undirected graph. Using Ohm’s law and Kirchoff’s law, a system of linear equations is created with voltages at each node as a variable in these equations. Solving this system of equations gives voltages at each node. This is step takes time, which motivates the need for the Candidate graph generation. The greedy display generation algorithm attempts to find a display graph of at most \( b \) nodes (set to a maximum of 100 in my experiments) which maximizes the amount of total current delivered from the start node to the end node. Starting with an empty subgraph, this algorithm iteratively adds paths until meeting the budget \( b \). At each of the iterations, a dynamic programming algorithm is used to find the path which has the maximum ratio of delivered current to number of new nodes added to the subgraph. This choice may not be globally optimal, hence the greedy nature of the algorithm. In my experiments I test this model based on current flow used to compute these display \( \rho \)-graphs.
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN
MULTI-RELATIONAL GRAPHS

8.1.2.3 Heuristics

RDFS vocabulary allows users to represent classes and relationships (properties) connecting them thereby indirectly imposing meaning on resources that are instances of these classes. Hence I define three quantities (Class and Property Specificity, Instance Participation Selectivity, and Span) indirectly based on semantics and RDF statement types and frequencies. My aim in doing this is to use semantics suggested by the schema to systematically convert an arbitrary unweighted RDF graph into an edge-weighted graph appropriate as input to the algorithms described previously. I define a schema $S$ as the union of the set of classes ($C$) and property types ($P$). Further, I define an RDF data store $R = (\Pi, I)$ where $\Pi = \bigcup S$ and $I$ is the set of class and property instances corresponding to the schemas. A single entity could be an instance of multiple classes belonging to different schemas. I assume that such an entity instance is uniquely identified by one URI. In other words, no data integration operation is required.

8.1.2.3.1 Class and Property Specificity (CS and PS) Intuitively, more specific resources and properties convey more information than general ones. For instance, it is more informative if one knows that Michael Jordan was a basketball player as opposed to knowing that he is a person. Similarly, knowing that Rudy Giuliani was an employee of New York City is less informative than the fact that he was mayor of New York City.

As a result of the rdfs:subClassOf and rdfs:subPropertyOf properties provided by RDF schema it is possible to impose a partial ordering of properties and classes in the schema resulting in a well formed hierarchy of classes and properties. For a given property $p$, let $H(p)$ be the length of the longest path in the hierarchy tree that contains $p$, and for a given class $c$, let $H'(c)$ be the length of the longest path in the hierarchy tree from the root to $c$. Properties and classes at the root of their respective hierarchy trees in the schema are considered most general while those at the leaves of these trees are considered most
specific. Therefore a measure of specificity can be associated with each class or property commensurate with its position in its hierarchy. Let the depth of an arbitrary property in its property hierarchy be $d(p_i)$ and the depth of an arbitrary class in its class hierarchy be $d(c_j)$. Therefore, the specificity of property $p_i$ and class $c_j$ are given by $\mu(p_i) = \frac{d(p_i)}{H(p)}$ and $\mu(c_j) = \frac{d(c_j)}{H(c)}$. Every resource that is an instance of the class $c_j$ is assigned this weight. If a resource $r$ is an instance of $k$ distinct classes it is assigned the value $\mu(r) = \max_{1 \leq m \leq k} \mu(c_m)$, since I want the most specific nodes and properties to be in the output subgraph. To convert this node weight into an edge weight, the value is equally distributed among all edges incident on the resource $r$. This weighting scheme favors nodes with lower degree since the node specificity is divided equally among its incident edges, therefore edges incident on nodes with high degree will get a lower weight.

### 8.1.2.3.2 Instance Participation Selectivity (IPS)

Another guideline I use is that rarer facts are typically more informative than frequently occurring ones [72]. Consider the example shown in Figure 8.1. The example shows two relationships lives_in and council_member_of defined on the classes Person and City. The instances $p_1, p_2, \ldots, p_m$ of the class Person are members of the council of City $c_1$, hence the relationship council_member_of between each $p_1, p_2, \ldots, p_m$ to $c_1$. Instances of class Person $p_{m+1}, p_{m+2}, \ldots, p_{k-2}, p_{k-1}, p_k$ represent people who live in City $c_1$ and therefore are related to $c_1$ by the relationship lives_in. From the perspective of the node $c_1$, following an edge labeled lives_in will lead to one node among $k - m$ possible nodes. In contrast, following an edge labeled council_member_of will lead to one node among $m$ nodes. Given that rarer paths are considered more informative, the amount of information gained by choosing to traverse the council_member_of relationship to a node in the set $\{p_1, p_2, \ldots, p_m\}$ is more than the gain achieved by choosing to traverse the lives_in relationship to a node in the set $\{p_{m+1}, p_{m+2}, \ldots, p_{k-2}, p_{k-1}, p_k\}$.

This is akin to choosing the hop with maximum information gain. To define this
heuristic formally, I introduce the notion of the type of an RDF statement. The type of an RDF statement \( \langle s, p, o \rangle \) is defined as the triple \( \pi = \langle C_i, p, C_j \rangle \) where \( \text{typeOf}(s) = C_i \) and \( \text{typeOf}(o) = C_j \). Further, \( |\pi| \) is thus the number of statements of type \( \pi \) in a given RDF instance base. I therefore define Instance Participation Selectivity for each RDF statement as \( \sigma_\pi = \frac{1}{|\pi|} \). Going back to Figure 8.1, let \( \pi = \langle \text{Person}, \text{lives_in}, \text{City} \rangle \) and \( \pi' = \langle \text{Person}, \text{council_member_of}, \text{City} \rangle \). According to this example, \( \sigma_\pi = \frac{1}{k-m} \) and \( \sigma_{\pi'} = \frac{1}{m} \) and if \( k > m \) then \( \sigma_{\pi'} > \sigma_\pi \).

### 8.1.2.3.3 The SPAN heuristic

I consider resources that are instances of classes belonging to different schemas as being indicative of anomalous paths between the given entities, since they tie different domains together. What makes such paths anomalous and therefore interesting is the fact that these paths represent a deviation from the expected paths suggested by the schemas. In Figure 8.2 the path from \( e_1 \) to \( e_8 \) is an example of a path favored by this heuristic. It contains three nodes (colored black) that are instances of more classes belonging to more than one schema. I use the number of different schemas that an instance of a class belongs to as an indicator of its anomalous nature. This number is used to bias search during the creation of the display graph.
Figure 8.2: Illustrative example for the SPAN heuristic
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN
MULTI-RELATIONAL GRAPHS

8.1.3 Dataset and preliminary results

I used a synthetic dataset for my experiments since I needed control over characteristics
of the data. This helps me ensure that my results are not unduly affected by unknown
aspects such as connectivity, relative instance distribution etc. of the dataset. Collection of
real world data follows an almost opportunistic approach since availability often dictates
design. As a result there is room for skew in instance data population. This skew may
not always reflect real-world distributions, as was observed in my experience with SWETO
[30] a non-synthetic RDF dataset covering the DBLP\(^1\) data. To circumvent this I built
a utility [83] that takes as input a set of schemas and a properties file specifying relative
distributions of instances of classes and properties that would be expected in the real world.
For example, consider two classes in a Business ontology: Trustee and Employee. It would
be reasonable to assume that if there are 5,000 instances of the class Employee then there
are unlikely to be 1,000 instances of the class Trustee. Instances of the class Trustee are
more likely to be approximately 10. These numbers are domain specific. The result of
running this utility is an RDF graph that contains nodes and edges that are instances of
classes and property types belonging to any or all of the classes in the given schemas. The
graph for my experiments contains 30,000 nodes and 45,000 edges.

Before evaluating the results of this algorithm I present a scenario to describe the use
of subgraph discovery in a realistic setting. A fraud investigator with the Securities and
Exchange Commission (SEC) receives the following piece of information about a week
after the stock prices for EntertainmentCompany\_9982 plummet. Actor\_5567 sold 70% of
his shares of EntertainmentCompany\_9982 one week after Capt\_8262 sold all of his shares
in the same company. Both transactions took place two weeks before the prices plum-
meted. The example subgraph shown in Figure 8.6, might help an investigator visualize
the connections between the resources: Actor\_5567 and Captain\_8262.

\(^1\)http://www.informatik.uni-trier.de/~ley/db/
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN
MULTI-RELATIONAL GRAPHS

Figure 8.3: The business ontology

Figure 8.4: The sports ontology
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN MULTI-RELATIONAL GRAPHS

Figure 8.5: The entertainment ontology

Figure 8.6: Example snippet of a subgraph returned for the subgraph query between (Actor\_5567, Captain\_8262) on my synthetic dataset
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN MULTI-RELATIONAL GRAPHS

8.1.4 Results and Evaluations

I recognize the fact that the notion “best” subgraph is very subjective and dependent on the user’s perspective. It is however desirable to have an objective measure that could be used to quantify the quality of a generated subgraph. The issue of judging relevance of paths (i.e. path ranking) has been addressed in [29] and [30]. In [72] the authors use rarity of the path as a measure of its interestingness. I therefore use these path ranking mechanisms to evaluate the quality of both the Candidate $\rho-$graph and the Display $\rho-$graph. In my experiments the Candidate $\rho-$graphs generated contained 3000 nodes and the Display $\rho-$graphs were restricted to a maximum of 100 nodes making them easy to visualize.

8.1.4.1 Evaluation using Path Ranks

In my data set there are over 60 million paths of length 13 between the two endpoints used. Paths of this length are unlikely to be of much interest to the user. To evaluate my subgraphs, I run an exhaustive k-hop limited Depth-First Search (DFS) on the input graph between the two entities. I use a depth limit of 9 hops for my experiments for feasibility of path enumeration for ranking. Note that both the Candidate $\rho-$graph and Display $\rho-$graph generated do contain arbitrary length paths, but I only consider paths of length at most 9 for fairness of comparison. I represent the paths returned by the k-hop DFS as the set $FGPaths_9$ (paths of up to 9 hops in the full graph). There are therefore 30 distinct $FGPaths_9$ sets, one for each query in my experiments. I rank the paths in each of the $FGPaths_9$ sets using the ranking mechanisms proposed in [29] and [30] in addition to what I call Rarity Rank based on the method suggested in [72]. The rank of a path $p$ based on the Rarity Rank scheme is given by the inverse of the number of paths that share the same type as path $p$. Each of the ranking mechanisms applied to the set $FGPaths_9$ results in a list of ranked paths. Let me assume that this leads to ranking from 1 to $M$ where $M$ is the rank of the least relevant path. Let this set of ranked paths be represented...
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN MULTI-RELATIONAL GRAPHS

as FGRankedPaths\(_9\). I therefore have three distinct scales (FGRankedPaths\(_9\) sets) against which the quality of both the Candidate \(\rho\)–graph and the Display \(\rho\)–graph can be measured. In all of the graphs, a table shown below the x-axis represents the 16 possible combinations of the four heuristics I use viz. class and property specificity (CS and PS), Instance Participation Selectivity (IPS) and The Span Heuristic (SPAN).

8.1.4.1.1 Measuring Candidate graph quality

To measure the quality of a given Candidate \(\rho\)–graph I compare the best paths in the entire graph to those in the Candidate \(\rho\)–graph. Let \(CGPaths\(_9\)\) represent the set of paths in the Candidate \(\rho\)–graph with maximum length 9. For each path \(p_{\text{candidate}} \in CGPaths\(_9\)\) I count the number of paths \(p \in FGRankedPaths\(_9\)\) such that \(\text{rank}(p) > \text{rank}(p_{\text{candidate}})\). This gives me the rank of each path in the Candidate \(\rho\)–graph with respect to all paths in the set FGRankedPaths\(_9\). The score of a path is given by:

\[
\text{score}(p_{\text{candidate}}) = |FGRankedPaths\(_9\)| - \text{rank}(p_{\text{candidate}}) \quad (8.1.2)
\]

Quality of the Candidate \(\rho\)–graph is therefore given by:

\[
Q(CGPaths\(_9\)) = \frac{\sum_{p_{\text{candidate}} \in CGPaths\(_9\)} \text{score}(p_{\text{candidate}})}{\sum_{r=1}^{[CGPaths\(_9\)]} |FGRankedPaths\(_9\)|} \quad (8.1.3)
\]

Figure 8.7 shows that the Candidate \(\rho\)–graph containing k paths obtained using my edge weighting schemes achieves between 80-90% of the score that can be achieved by choosing the top-k ranked paths from the full graph (entire dataset of 30,000 nodes and 45,000 edges). The Candidate \(\rho\)–graph in my results typically contain 30-40% of the paths in the entire graph between the endpoints yet are 80-90% as “good” as the top paths in the entire graph between the two endpoints. Measuring Display \(\rho\)–graph quality

Similar to Candidate \(\rho\)–graph quality, I compare the paths in the Display \(\rho\)–graph to the best paths in the entire graph. Let the set D\(G\)Paths represent the paths in the Display \(\rho\)–graph. The rank of
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN
MULTI-RELATIONAL GRAPHS

a path in the Display $\rho-$graph is computed exactly the same way the rank of a path in the
Candidate $\rho-$graph is computed, as is the score.

\[
\text{score}(p_{\text{display}}) = |FGRankedPaths_9| - \text{rank}(p_{\text{display}})
\] (8.1.4)

The quality of a display graph is computed by comparing its cumulative score to the best
possible display that could be obtained from the ranked set of paths in the full graph. I refer
to this best possible display as Pseudo-Display. In my experiments I use a budget of 100
nodes for my Display $\rho-$graphs. Starting with an empty Pseudo-Display graph and the path
with rank 1 in the set FGRankedPaths9 I add paths to the Pseudo-Display until 100 nodes
have been added. The cumulative score of the Pseudo-Display is then computed as the sum
of the scores of the paths. The quality of a Display $\rho-$graph is therefore given by:

\[
Q(DGPaths) = \frac{\sum_{p_{\text{display}} \in DGPaths} \text{score}(p_{\text{display}})}{\sum_{p_{\text{pseudo}} \in Pseudo-display} \text{score}(p_{\text{pseudo}})}
\] (8.1.5)

Figure 8.8 shows that starting with the Candidate $\rho-$graphs with 80-90% quality the $\rho-$graph
computed captures a maximum of 84% of the score that can be obtained by taking the best
paths in the full graph. My results show the quality of Display $\rho-$graph with respect to
SemRank [29] to be surprisingly low - 43%. Further investigation of the methods used
revealed that the difference between the ranking scheme in [30] and that in [29] is that in
the former instance node degrees affect the rank of a path (nodes of lower degree being
favored) whereas in the latter rank of path is determined purely by properties in the path.
My heuristics favor lower degree nodes and hence the observed trend. A personal commu-
nication with the authors of [29] revealed that extending SemRank to include the effect of
nodes is an intended follow up to this work.

8.1.4.1.2 Display graph quality  With the intention of validating the current flow model
for subgraph relevance [70] I conducted the following experiment. I computed what I term
Figure 8.7: Quality of the Candidate graph as percentage of maximum score for averaged over 30 queries

Figure 8.8: Quality of the Display graph Note that all weighting heuristics turned off results in poor graph quality in contrast with all heuristics turned on
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN
MULTI-RELATIONAL GRAPHS

Successive Display $\rho-$graph. To construct these displays I successively run the Display $\rho-$graph generation algorithm on the candidate graph. At each successive run I discount the paths used in previous displays. This results in the next best Display $\rho-$graph at every successive run. This process is repeated five times in my experiments to obtain five Display $\rho-$graph. The current flow in each of these Display $\rho-$graph is plotted relative to the current flow in the first Display $\rho-$graphs on a log scale in Figure 8.9. The quality of these Display $\rho-$graphs is plotted relative to the quality of the first Display $\rho-$graph in Figure 8.10. There is a large difference both in the current flow and the display quality between the first display and the next display. This confirms that there is a correspondence between current flow in the Display $\rho-$graphs and their quality. This in turn supports the electricity based model for RDF graph relevance. Note that the plots below are averages of the relative differences of successive graph displays over all ranking schemes.

8.1.4.2 Qualitative Evaluation

I conducted this experiment on factual data extracted from public access government websites. For this particular experiment I picked two persons of interest at random. Figures 8.11 and 8.12 compare Display graphs generated for end points: Arnold Schwarzenegger and Bill Clinton. In both cases the budget is set to 12 nodes. Figure 8.12 contains
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN MULTI-RELATIONAL GRAPHS

September 21, 2008

Figure 8.10: Quality of 5 Successive Displays relative to the best

a very interesting path that connects Arnold Schwarzenegger to Bill Clinton via Maria Shriver followed by Edward Kennedy. Edward Kennedy and Bill Clinton both spoke at consecutive Democratic National Conventions. This indicated a strong similarity in their political ideologies. A strong familial relationship between Edward Kennedy and Arnold Schwarzenegger via Maria Shriver is also seen in the Figure 8.12. In comparison to these paths, the paths in Figure 8.11 are rather uninteresting. This provides empirical evidence that my weighting schemes are useful. I believe that further investigation of semantic methods for subgraph discovery is warranted.
8.1. DISCOVERING INFORMATIVE CONNECTION SUBGRAPHS IN
MULTI-RELATIONAL GRAPHS

Figure 8.12: Display graph with budget as 12 using edge weighting scheme
Conclusions & Future Work

9.1 Conclusion

In this dissertation I have investigated mechanisms to extract, represent and utilize biomedical textual knowledge effectively. My extraction mechanisms presented herein are unsupervised and therefore require no time investments for training on the part of the domain expert using my system. My extraction mechanism is domain knowledge agnostic and therefore is able to extract metadata from any kind of text. Beyond extraction I have presented a mechanism to associate semantic interpretations to extracted content by grounding entities and relationships in domain schema. In this dissertation I develop a tool for use by domain experts to evaluate the quality of extraction. Use of this evaluation mechanism shows that entity identification accuracy is 82%, averaged over the set of relationships I evaluated. Accuracy of both entity and triple extraction is heavily dependent on the domain specificity of the relationship being evaluated. I also present several mechanisms that show effective utilizations of my extracted semantic metadata. Applications shown herein help support effective retrieval, browsing. I have also presented a technique for knowledge discovery over RDF graphs.

I see all of these as preliminary efforts in supporting knowledge discovery over scientific literature. Beyond these steps several more enhancements are required to achieve this goal.
9.2 Future Work

9.2.1 Expert-stipulated information extraction

The extraction algorithms that I have proposed in this dissertation are preemptive, in the sense that they extract all the relationships that they can possibly extract. Every verb in a sentence is a potential relationship. My recent conversations with domain experts indicate that while the results of my preemptive extraction approach look good, the prohibitively large amount of metadata produced makes it difficult for the domain expert to focus on the metadata relevant to them. One alternative is to query the generated metadata using the SPARQL query language. Such a query however requires precise knowledge of the content being sought. In most real situations however the only thing available to an expert is a vague idea of the subdomain and some keywords of interest. The keyword-in-documents-out paradigm provided by PubMed ensures ease of use but does not automate the extraction and interpretation of the document contents. In order to preserve this ease-of-use I plan to investigate an expert-driven knowledge elicitation paradigm where the user provides a set of keywords that act as seeds for the extraction mechanism. The objective would be to iteratively instantiate mechanistic models of diseases.

In addition to extraction of compound entities and relationships, descriptive mechanistic parameters for relationships can also be extracted. These parameters describe the conditions under which interactions occur and are a part of the mechanistic model described by one or more relationships. Beyond the 49 different coarse-grained relationships in UMLS, I plan to develop transducers that are capable of extracting parameters that are associated with processes or relationships. “CDH23 and PCDH15 interact at 0.1 mM $Ca^{2+}$”[9] shows an example of such a parameter. Here the token sequence “0.1 mM $Ca^{2+}$” describes the concentration of a substance at which the relationship ”interacts” holds. These parameters, once extracted, will be associated with entities and relationships. The combination of
9.2. **FUTURE WORK**

Parameter extraction, knowledge-enhanced semi-supervised learning and complex entity transducers will lead to high-resolution capture of knowledge leading to the population of mechanistic models. I plan to experiment with representation mechanisms like CellML (http://www.cellml.org/) for use in representing the extracted mechanistic models.

Such fine-grained extraction will result in the initial population of a mechanistic-model describing the disease and gene related term in the user input keywords. I plan to develop algorithms to semi-automatically iteratively refine these initial models based on corroborating evidence from literature and scientific databases.

### 9.2.2 Discovering informative entity-relationship subgraphs

A network of inter-entity relationships extracted from text can be used as a dataset to support knowledge discovery mechanisms. I plan to develop algorithms for transforming the extracted relationship network into an edge weighted graph based on user input and corpus statistics. Using the edge weights to compute heuristics, I plan to develop subgraph discovery algorithms to support knowledge discovery operations. As a follow up to this work I plan to apply my techniques to develop tools for finding correlations between genes and diseases. I further plan to develop algorithms to support queries involving k endpoints for RDF graphs. Another interesting direction involves formalizing the notion of Context and investigating Context-Aware Subgraph Discovery algorithms.

### 9.2.3 Strength of a triple

Using an index of syntactic entity variants I discovered in my unsupervised extraction, it is possible to get frequency counts for each component of such a fact relative to the subject, predicate and object. These frequencies can be used to compute corpus wide probabilities of occurrence of each component of a triple. I then propose to use Pointwise Mutual Infor-
9.2. FUTURE WORK

PMI as one measure of a “significance” function to compute the collocations of the subject, predicate and object. Using this metric, it is possible to determine a measure of ”strength” for each triple of a hypothesis as dictated by the corpus. There may be other means of computing the strength of a triple which I intend to investigate in future. Co-citation information extracted from PubMed is another factor that influences the strength of a fact. I argue that if two different papers strengthen two different triples of a hypothesis without citing each other, these two papers should be better candidates for the user to consider since these are indicative of potential missing connections. This captures the notion of complementary non-interacting literature proposed by [5]. Another factor that can affect the consideration of a document in support of a triple is its date of publication. Another factor that influences the strength of a triple is confidence in an assertion based on degree words such as “severely”, “mildly” etc. I plan to use interpretation assigned to these qualifiers to influence strength of assertions and therefore those of triples.

9.2.4 Associations over time

Each abstract in PubMed has a date of publication timestamp associated with it. The timestamps of documents in PubMed range from 1960 to the present day. Over this time period the MeSH hierarchy has undergone changes. Certain topics have been merged and others have been split and further specialized. Semantic connection subgraphs (SCS) represent the associations between the entities in question (input). In domains such as biomedicine previously accepted theories are sometimes disproved or often modified in light of new evidence. It is possible that researchers may wrongly believe that two diseases are very similar and hence ignore subtle differences between them. To be able to discover such differences it might be useful to compare the content of PubMed abstracts across different time intervals.
9.2.5 Hypothesis-based composed-passage retrieval

The composed passage retrieval method combines both concept subgraph based retrieval and full-text based passage retrieval. Concept subgraph based retrieval tends to be more accurate while full-text passage retrieval tends to be more comprehensive. I plan to develop methods for combining the evidence of subgraph based retrieval and full-text passage retrieval with an intuitive evidence aggregating formula and a probabilistic evidence combination model.
Bibliography


[18] http://www.w3.org/TR/owl-features/


[22] http://www.w3.org/TR/rdf-mt/


[32] Cartic Ramakrishnan, Pablo N. Mendes, Shaojun Wang and Amit P. Sheth “Unsupervised Discovery of Compound Entities for Relationship Extraction” EKAW 2008 - 16th International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns


[75] Xifeng Yan, Jiawei Han. CloseGraph “Mining Closed Frequent Graph Patterns.” In Proceedings of the 2003 Conference on Knowledge Discovery and Data Mining, 2003.


[81] Deng Cai, Zheng Shao, Xiaofei He, Xifeng Yan, Jiawei Han, “Mining Hidden Community in Heterogeneous Social Networks”, Proc. 2005 ACM-SIGKDD Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD’05), Chicago, IL, Aug. 2005.


APPENDIX A

10.1 RDF generated for sample sentences

The following is a list of sentences that I used as test sentences in the development of these rules.

10.1.1 Sentences

- Sentence ID=10000 Sentence= “p53 gene product is a transcription factor that regulates the expression of a number of DNA-damage and cell-cycle regulatory genes and genes regulating apoptosis.”

- Sentence ID=102520-1-1 Sentence= “Scheuerman et al. stated that previously described anomalies in acrorenal syndrome included unilateral renal agenesis, duplication of the collecting system, renal hypoplasia leading to renal insufficiency, and vesicoureteral reflux with hydropnephrosis.”

- Sentence ID=102520-1-2 Sentence= “They described 2 patients with what they believed to be a new variant of acrorenal field defect.”

- Sentence ID=102520-1-3 Sentence= “Both patients had horseshoe kidney, which the authors stated had not previously been described with acrorenal field defect.”
10.1. RDF GENERATED FOR SAMPLE SENTENCES

- Sentence ID=102520-1-4 Sentence= “One of the patients had preaxial polydactyly of the right hand.”

- Sentence ID=102520-1-5 Sentence= “The other had left hand ectrodactyly.”

- Sentence ID=90110496-1 Sentence= “Anti-Ro(SSA) autoantibodies are associated with T cell receptor beta genes in systemic lupus erythematosus patients.”

- Sentence ID=adenomatous Sentence= “An excessive internal or external stimulation by estrogen induces adenomatous hyperplasia of the endometrium.”

- Sentence ID=omim Sentence= “This MEK dependency was observed in BRAF mutant cells regardless of tissue lineage, and correlated with both downregulation of cyclin D1 protein expression and the induction of G1 arrest.”

10.1.2 Generated RDF

The RDF resulting from these sentences is listed below.

```xml
<?xml version="1.0"?><rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns="http://kno.e.sis.org#"
     xmlns:umls="http://lsdis.cs.uga.edu/semdis/umls#"
     xml:base="http://lsdis.cs.uga.edu/semdis/umls#">
  <rdf:Description rdf:about="http://kno.e.sis.org#D005796">
    <rdfs:label xml:lang="en">genes</rdfs:label>
    <rdf:type rdf:resource="http://kno.e.sis.org#T028"/>
    <umls:hasSource>10000</umls:hasSource>
  </rdf:Description>
  <rdf:Description rdf:about="http://kno.e.sis.org#m_1">
    <rdfs:label xml:lang="en">cell-cycle</rdfs:label>
    <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
    <umls:hasSource>10000</umls:hasSource>
  </rdf:Description>
  <rdf:Description rdf:about="http://kno.e.sis.org#m_2">
    <rdfs:label xml:lang="en">regulatory</rdfs:label>
    <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
    <umls:hasSource>10000</umls:hasSource>
  </rdf:Description>
  <rdf:Description rdf:about="http://kno.e.sis.org#m_3">
    <rdfs:label xml:lang="en">and</rdfs:label>
    <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
    <umls:hasSource>10000</umls:hasSource>
  </rdf:Description>
</rdf:RDF>
```
10.1. RDF GENERATED FOR SAMPLE SENTENCES

<rdf:Description rdf:about="http://kno.e.sis.org#ce_1">
  <rdfs:label xml:lang="en">cell-cycle regulatory genes and genes</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
  <umls:hasSource>10000</umls:hasSource>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_3"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_2"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_1"/>
  <umls:hasPart rdf:resource="http://kno.e.sis.org#D005796"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#D017209">
  <rdfs:label xml:lang="en">apoptosis</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#T043"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_4">
  <rdfs:label xml:lang="en">factor</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_5">
  <rdfs:label xml:lang="en">a</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_6">
  <rdfs:label xml:lang="en">transcription</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_7">
  <rdfs:label xml:lang="en">expression</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_8">
  <rdfs:label xml:lang="en">the</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_9">
  <rdfs:label xml:lang="en">of</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_10">
  <rdfs:label xml:lang="en">number</rdfs:label>
  <umls:hasSource>114</umls:hasSource>
</rdf:Description>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>10000</umls:hasSource>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#m_11">
  <rdfs:label xml:lang="en">dna-damage</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#ce_3">
  <rdfs:label xml:lang="en">the expression of a number of dna-damage and cell-cycle regulatory genes and genes</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
  <umls:hasPart rdf:resource="http://kno.e.sis.org#D005796"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_3"/>
  <umls:hasPart rdf:resource="http://kno.e.sis.org#D005796"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_11"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_9"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_7"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#m_12">
  <rdfs:label xml:lang="en">product</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#m_13">
  <rdfs:label xml:lang="en">p53</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#D005796">
  <rdfs:label xml:lang="en">gene</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#T028"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#ce_4">
  <rdfs:label xml:lang="en">p53 gene product</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
  <umls:hasPart rdf:resource="http://kno.e.sis.org#D005796"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_13"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_12"/>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#triple_1">
  <rdfs:label xml:lang="en">ce_4--&gt;is--&gt;ce_2</rdfs:label>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#triple_2">
  <rdfs:label xml:lang="en">ce_4--&gt;is--&gt;ce_2</rdfs:label>
  <umls:hasSource>10000</umls:hasSource>
</rdf:Description>
an excessive internal or external stimulation by estrogen
10.1. RDF GENERATED FOR SAMPLE SENTENCES

<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D004967"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_20"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_19"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_18"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_17"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_16"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_15"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_14"/>
<umls:hasSource adnomatous</umls:hasSource/>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#D006965">
<rdfs:label xml:lang="en">hyperplasia</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#T046"/>
<umls:hasSource adnomatous</umls:hasSource/>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_21">
<rdfs:label xml:lang="en">adenomatous</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource adnomatous</umls:hasSource/>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#D004717">
<rdfs:label xml:lang="en">endometrium</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#T023"/>
<umls:hasSource adnomatous</umls:hasSource/>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#ce_6">
<rdfs:label xml:lang="en">adenomatous hyperplasia of the endometrium</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D004717"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_9"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_21"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D006965"/>
<umls:hasSource adnomatous</umls:hasSource/>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#triple_4">
<rdfs:label xml:lang="en">ce_5--&gt;induces--&gt;ce_6</rdfs:label>
<rdf:subject rdf:resource="http://kno.e.sis.org#ce_5"/>
<rdf:predicate rdf:resource="http://kno.e.sis.org#induces"/>
<rdf:object rdf:resource="http://kno.e.sis.org#ce_6"/>
<umls:hasSource adnomatous</umls:hasSource/>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#D001323">
<rdfs:label xml:lang="en">autoantibodies</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#T129"/>
<umls:hasSource 90110496-1</umls:hasSource/>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_22">
<rdfs:label xml:lang="en">anti-ro(ssa)</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource 90110496-1</umls:hasSource/>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#ce_7">
<rdfs:label xml:lang="en">anti-ro(ssa) autoantibodies</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
</rdf:Description>
10.1. RDF GENERATED FOR SAMPLE SENTENCES

<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_22"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D001323"/>
<umls:hasSource>90110496-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#D010361">
<rdfs:label xml:lang="en">patients</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#T101"/>
<umls:hasSource>90110496-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_23">
<rdfs:label xml:lang="en">systemic</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>90110496-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#D008177">
<rdfs:label xml:lang="en">lupus</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#T047"/>
<umls:hasSource>90110496-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_24">
<rdfs:label xml:lang="en">erythematosus</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>90110496-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#ce_8">
<rdfs:label xml:lang="en">systemic lupus erythematosus patients</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_24"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D008177"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_23"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D010361"/>
<umls:hasSource>90110496-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_25">
<rdfs:label xml:lang="en">t</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>90110496-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#D002477">
<rdfs:label xml:lang="en">cell</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#T025"/>
<umls:hasSource>90110496-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_26">
<rdfs:label xml:lang="en">receptor</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>90110496-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_27">
<rdfs:label xml:lang="en">beta</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>90110496-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#ce_9">
<rdfs:label xml:lang="en">t cell receptor beta genes</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
</rdf:Description>
10.1. RDF GENERATED FOR SAMPLE SENTENCES

September 21, 2008

```xml
<rdf:Description rdf:about="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasSource>102520-1-3</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_33">
<rdfs:label xml:lang="en">which</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>102520-1-3</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_34">
<rdfs:label xml:lang="en">authors</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>102520-1-3</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#ce_12">
<rdfs:label xml:lang="en">the authors</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_34"/>
<umls:hasSource>102520-1-3</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_35">
<rdfs:label xml:lang="en">both</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>102520-1-3</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#ce_13">
<rdfs:label xml:lang="en">both patients</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_35"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D010361"/>
<umls:hasSource>102520-1-3</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#triple_7">
<rdfs:label xml:lang="en">ce_13--&gt;had--&gt;ce_11</rdfs:label>
<rdf:subject rdf:resource="http://kno.e.sis.org#ce_13"/>
<rdf:predicate rdf:resource="http://kno.e.sis.org#had"/>
<rdf:object rdf:resource="http://kno.e.sis.org#ce_11"/>
<umls:hasSource>102520-1-3</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#triple_8">
<rdfs:label xml:lang="en">ce_11--&gt;stated had been described--&gt;m_33</rdfs:label>
<rdf:subject rdf:resource="http://kno.e.sis.org#ce_11"/>
<rdf:predicate rdf:resource="http://kno.e.sis.org#stated+had+been+described"/>
<rdf:object rdf:resource="http://kno.e.sis.org#m_33"/>
<umls:hasSource>102520-1-3</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#triple_9">
<rdfs:label xml:lang="en">ce_12--&gt;stated had been described--&gt;m_33</rdfs:label>
<rdf:subject rdf:resource="http://kno.e.sis.org#ce_12"/>
<rdf:predicate rdf:resource="http://kno.e.sis.org#stated+had+been+described"/>
<rdf:object rdf:resource="http://kno.e.sis.org#m_33"/>
<umls:hasSource>102520-1-3</umls:hasSource>
</rdf:Description>
```

<rdfs:label xml:lang="en"></rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">one</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">one of the patients</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D010361"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_9"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_37"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_36"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">polydactyly</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#T019"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">preaxial</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">right</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">hand</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#T023"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">preaxial polydactyly of the right hand</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D006225"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_9"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_38"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D017689"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">preaxial polydactyly of the right hand</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D006225"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_9"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_38"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D017689"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">preaxial polydactyly of the right hand</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D006225"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_9"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_38"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D017689"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">preaxial polydactyly of the right hand</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D006225"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_9"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_38"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D017689"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">preaxial polydactyly of the right hand</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D006225"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_9"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_38"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D017689"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">preaxial polydactyly of the right hand</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D006225"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_9"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_38"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D017689"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>

<rdfs:label xml:lang="en">preaxial polydactyly of the right hand</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D006225"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_8"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_9"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_38"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#D017689"/>
<umls:hasSource>102520-1-4</umls:hasSource>
</rdf:Description>
10.1. RDF GENERATED FOR SAMPLE SENTENCES

<sparql rdf:about="http://kno.e.sis.org#m_56">
  <rdfs:label xml:lang="en">scheuerman</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>102520-1-1</umls:hasSource>
</sparql>

<sparql rdf:about="http://kno.e.sis.org#m_57">
  <rdfs:label xml:lang="en">et</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>102520-1-1</umls:hasSource>
</sparql>

<sparql rdf:about="http://kno.e.sis.org#m_58">
  <rdfs:label xml:lang="en">al.</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>102520-1-1</umls:hasSource>
</sparql>

<sparql rdf:about="http://kno.e.sis.org#ce_18"> 
  <rdfs:label xml:lang="en">scheuerman et al.</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_58"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_57"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_56"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_36"/>
  <umls:hasSource>102520-1-1</umls:hasSource>
</sparql>

<sparql rdf:about="http://kno.e.sis.org#triple_11"> 
  <rdfs:label xml:lang="en">ce_16-\rightarrow\text{included}-\rightarrow\text{ce_17}</rdfs:label>
  <rdf:subject rdf:resource="http://kno.e.sis.org#ce_16"/>
  <rdf:predicate rdf:resource="http://kno.e.sis.org#included"/>
  <rdf:object rdf:resource="http://kno.e.sis.org#ce_17"/>
  <umls:hasSource>102520-1-1</umls:hasSource>
</sparql>

<sparql rdf:about="http://kno.e.sis.org#m_59"> 
  <rdfs:label xml:lang="en">variant</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>102520-1-2</umls:hasSource>
</sparql>

<sparql rdf:about="http://kno.e.sis.org#m_60"> 
  <rdfs:label xml:lang="en">new</rdfs:label>
  <umls:hasSource>102520-1-1</umls:hasSource>
</sparql>
10.1. RDF GENERATED FOR SAMPLE SENTENCES

<rdf:Description rdf:about="http://kno.e.sis.org#m_65">
  <rdfs:label xml:lang="en">both</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>omim</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#ce_20">
  <rdfs:label xml:lang="en">ce_20--&gt;believed--&gt;ce_21</rdfs:label>
  <rdf:subject rdf:resource="http://kno.e.sis.org#ce_20"/>
  <rdf:predicate rdf:resource="http://kno.e.sis.org#believed"/>
  <rdf:object rdf:resource="http://kno.e.sis.org#ce_21"/>
  <umls:hasSource>102520-1-2</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#ce_19">
  <rdfs:label xml:lang="en">ce_20--&gt;believed--&gt;ce_19</rdfs:label>
  <rdf:subject rdf:resource="http://kno.e.sis.org#ce_20"/>
  <rdf:predicate rdf:resource="http://kno.e.sis.org#believed"/>
  <rdf:object rdf:resource="http://kno.e.sis.org#ce_19"/>
  <umls:hasSource>102520-1-2</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_64">
  <rdfs:label xml:lang="en">m_64--&gt;believed--&gt;ce_21</rdfs:label>
  <rdf:subject rdf:resource="http://kno.e.sis.org#m_64"/>
  <rdf:predicate rdf:resource="http://kno.e.sis.org#believed"/>
  <rdf:object rdf:resource="http://kno.e.sis.org#ce_21"/>
  <umls:hasSource>102520-1-2</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_64">
  <rdfs:label xml:lang="en">m_64--&gt;believed--&gt;ce_19</rdfs:label>
  <rdf:subject rdf:resource="http://kno.e.sis.org#m_64"/>
  <rdf:predicate rdf:resource="http://kno.e.sis.org#believed"/>
  <rdf:object rdf:resource="http://kno.e.sis.org#ce_19"/>
  <umls:hasSource>102520-1-2</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_51">
  <rdfs:label xml:lang="en">m_51--&gt;be--&gt;ce_19</rdfs:label>
  <rdf:subject rdf:resource="http://kno.e.sis.org#m_51"/>
  <rdf:predicate rdf:resource="http://kno.e.sis.org#be"/>
  <rdf:object rdf:resource="http://kno.e.sis.org#ce_19"/>
  <umls:hasSource>102520-1-2</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#D015536">
  <rdfs:label xml:lang="en">downregulation</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#T043"/>
  <umls:hasSource>omim</umls:hasSource>
</rdf:Description>
both downregulation of cyclin d1 protein expression and the induction of g1 arrest.

dependency

g1 arrest.

this
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>omim</umls:hasSource></rdf:Description>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#m_73">
<rdfs:label xml:lang="en">mek</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>omim</umls:hasSource></rdf:Description>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#ce_23">
<rdfs:label xml:lang="en">this mek dependency</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_73"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_72"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_71"/>
<umls:hasSource>omim</umls:hasSource></rdf:Description>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#m_74">
<rdfs:label xml:lang="en">braf</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>omim</umls:hasSource></rdf:Description>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#m_75">
<rdfs:label xml:lang="en">mutant</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>omim</umls:hasSource></rdf:Description>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#D002477">
<rdfs:label xml:lang="en">cells</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#T025"/>
<umls:hasSource>omim</umls:hasSource></rdf:Description>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#m_76">
<rdfs:label xml:lang="en">regardless</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>omim</umls:hasSource></rdf:Description>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#D014024">
<rdfs:label xml:lang="en">tissue</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#T024"/>
<umls:hasSource>omim</umls:hasSource></rdf:Description>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#m_77">
<rdfs:label xml:lang="en">lineage,</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
<umls:hasSource>omim</umls:hasSource></rdf:Description>
</rdf:Description>
<rdf:Description rdf:about="http://kno.e.sis.org#ce_24">
<rdfs:label xml:lang="en">braf mutant cells regardless of tissue lineage,</rdfs:label>
<rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_77"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#m_77"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#m_9"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#m_76"/>
<umls:hasPart rdf:resource="http://kno.e.sis.org#m_74"/>
<umls:hasModifier rdf:resource="http://kno.e.sis.org#m_74"/>
10.1. RDF GENERATED FOR SAMPLE SENTENCES

<rdf:Description rdf:about="http://kno.e.sis.org#triple_19">
  <rdfs:label xml:lang="en">ce_23--&gt;was observed--&gt;ce_24</rdfs:label>
  <rdf:subject rdf:resource="http://kno.e.sis.org#ce_23"/>
  <rdf:predicate rdf:resource="http://kno.e.sis.org#was+observed"/>
  <rdf:object rdf:resource="http://kno.e.sis.org#ce_24"/>
  <umls:hasSource>omim</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#triple_20">
  <rdfs:label xml:lang="en">ce_25--&gt;had left--&gt;D006225</rdfs:label>
  <rdf:subject rdf:resource="http://kno.e.sis.org#ce_25"/>
  <rdf:predicate rdf:resource="http://kno.e.sis.org#had+left"/>
  <rdf:object rdf:resource="http://kno.e.sis.org#D006225"/>
  <umls:hasSource>102520-1-5</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_78">
  <rdfs:label xml:lang="en">other</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>102520-1-5</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#m_79">
  <rdfs:label xml:lang="en">the</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#ModifierClass"/>
  <umls:hasSource>102520-1-5</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="http://kno.e.sis.org#ce_25">
  <rdfs:label xml:lang="en">the other</rdfs:label>
  <rdf:type rdf:resource="http://kno.e.sis.org#CompositeEntitiesClass"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_79"/>
  <umls:hasModifier rdf:resource="http://kno.e.sis.org#m_78"/>
  <umls:hasSource>102520-1-5</umls:hasSource>
</rdf:Description>

</rdf:RDF>
APPENDIX B

11.1 Screenshots showing flow of evaluation

If the user is starting a new evaluation clicking the submit button results in the ranked list of predicates show in the figure 11.2. These predicates are ranked in order of their frequency of occurrence in the RDF generated for the dataset at hand (in this case the OMIM set).

Selecting a relationship from the dropdown list as shown in 11.3, followed by clicking on the button “Pick predicate” starts the evaluation for that predicate (e.g. “inhibited”) 11.4.

The figures 11.5, 11.6, 11.7 and 11.8 show a typical evaluation process.

Figure 11.1: Entry screen
11.1. SCREENSHOTS SHOWING FLOW OF EVALUATION

Figure 11.2: Pick a predicate to evaluate

Figure 11.3: Selecting the predicate “inhibited”

Figure 11.4: Starting evaluation of all instances of the “inhibited” relationship
11.1. SCREENSHOTS SHOWING FLOW OF EVALUATION

September 21, 2008

Figure 11.5: Subject Predicate and Object are correct

Figure 11.6: Click next, Subject Predicate and Object are correct for the next one too
11.1. SCREENSHOTS SHOWING FLOW OF EVALUATION

Figure 11.7: Click Next on previous screen, Subject and Object are correct but Triple is not.

Figure 11.8: Click Next on previous screen, Subject, Predicate and Object are correct & triple is correct.
11.1. SCREENSHOTS SHOWING FLOW OF EVALUATION

September 21, 2008

Figure 11.9: To save partial evaluation results copy the evaluation id (show highlighted)

Figure 11.10: Point your browser to start URL and paste the number in text box

Figure 11.11: Takes you back to evaluation for “inhibited” to 4/198 where you left off