A Framework for Schema-Driven Relationship Discovery from Unstructured text

Cartic Ramakrishnan, Krys J. Kochut and Amit P. Sheth
LSDIS Lab, Dept. of Computer Science, University of Georgia, Athens, GA
{cartic, kochut, amit}@cs.uga.edu

Abstract. We address the issue of extracting implicit and explicit relationships between entities in biomedical text. We argue that entities seldom occur in text in their simple form and that relationships in text relate the modified, complex forms of entities with each other. We present a rule-based method for (1) extraction of such complex entities and (2) relationships between them and (3) the conversion of such relationships into RDF. Furthermore, we present results that clearly demonstrate the utility of the generated RDF in discovering knowledge from text corpora by means of locating paths composed of the extracted relationships.

Keywords: Relationship Extraction, Knowledge-Driven Text mining

1 Introduction

Dr. Vannevar Bush, in 1945 [1], 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 (non-hyperlinked) in many domains. But as we demonstrate in this paper, this vision emphasizing relationships and associations continues to be highly relevant, and can indeed drive the next generation of search and analysis capabilities.

A good quality hierarchical organization of topics can serve as a very effective method to index and search for documents. A great example in the biomedical domain is the PubMed [2] database which contains over 16 million *manually classified* abstracts of scientific publications. 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 each scientific publication in this domain serves 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 the user of PubMed will piece together the *partial information* from *relevant documents* returned by PubMed searches to conclude that, for instance, "*Migraine could lead to cause a loss of Magnesium*".

One major drawback of this expectation 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 [3], which were implicit in the literature. He followed this up with several more examples such as the association between Magnesium and Migraine [4]. In fact, the paper revealed eleven neglected, potentially beneficial effects that Magnesium might have in alleviating Migraine. These discovered connections have since been validated by clinical trials and experiments. Such hidden, valuable relationships have been termed *Undiscovered Public Knowledge.* However, there is practically no support in contemporary information systems for users to unearth such undiscovered knowledge from public text in an automated manner.

2   Background and Motivation

It is clear that there are large bodies of knowledge in textual form that need to be utilized effectively (*e.g.* PubMed [2]). The creation of MeSH and UMLS are steps aimed at making such textual knowledge more accessible. PubMed, however, has been growing at a phenomenal rate. Consequently, the amount of *Undiscovered Public Knowledge* is also likely to increase at a comparable rate. Meanwhile, in the Semantic Web community analytical operators over semi-structured data have been receiving increased attention. Notable among these are *Semantic Association* [5] and *Relevant sub-graph Discovery* [6]. Both are aimed at discovering named relationships between entities in RDF data. Guha et. al. [7] introduced the notion of a "*Research Search*" as a type of Semantic Search. Users start with a search phrase which refers to an entity. The *"Research Search"* then helps users to gather pieces of information from multiple documents which collectively satisfy their information need.

It is critical to support such search, query and analytics paradigms over text data. Currently, these paradigms assume the existence of a rich variety of named relationships connecting entities in an instance base. Our aim, and indeed one of the aims of the Semantic Web community, is to apply these search and analytics paradigms to text data. It is clear that to enable this, we need to bridge the gap between unstructured data (free text) and semi-structured data (such as that represented in RDF, a W3C standard). As a step towards bridging this gap, in this paper, we address the challenge of **extracting implicit and explicit relationships between known entities in text.**

Recently, relationship extraction from biomedical text has received a lot of attention among several research communities. A comprehensive survey of current approaches to biomedical text mining is presented in [8]. Particular attention has been paid to surveying Named Entity Recognition. Most of the attention in this sub-area has focused on identifying gene names. One very effective method is AbGene [9]. This method uses training data in the form of hand-tagged sentences that contain known gene and protein names and is combined with the Brill Tagger [10] to extract names of genes and proteins. According to the authors in [8], most approaches to the relationship extraction consider very specific entities (such as genes), while relationships vary from general (e.g., any biochemical relationship) to specific (e.g.,

regulatory relationships). This becomes clear when we look at the approaches to relationship extraction surveyed in [8]. These include pattern based approaches [11] where patterns such as "also known as" are used to identify synonymy in protein and gene names. Template based approaches have also been investigated in the PASTA system [12]. Natural Language Processing (NLP) methods have been used in [13] and [14]. In [13] the authors focus their attention on cellular pathways and extract structured information from biomedical literature. Since they focus on cellular pathways their GENESIS system processes the entire article as opposed to just the abstract. Their system considers 125 fine-grained verbs that are classified into 14 broad semantic classes. The critical difference between GENESIS and our system is that our system uses empirical rules as opposed to grammatical rules to extract relationships between entities. In [14], the author uses NLP techniques to generate underspecified parses of sentences in biomedical text. Semantics from UMLS are then used to extract assertions from these parses. Our technique is most similar to this approach. The difference, however, is that our approach extracts modified and composite entities and relationships between them. This allows us to extract variants of known entities and assertions involving these variants.

From our perspective, all relationships of interest in these approaches are very specific. One obvious reason for this is that there is a dire need for such specific relationships to be extracted. In this paper, our approach focuses on more general relationships that are defined in UMLS and is not dependent on any specific type of relationship. The reasons for this are two-fold. First, our long-term goal is to support semantic browsing, searching and analysis of biomedical abstracts. The intended users of such a system could range from a layperson to domain experts. The second reason is that once instances of genes, proteins, etc. and relationships among them are extracted (by approaches discussed above) these could be integrated with clinical trials data which is arguably at the same level of specificity. Such integration would only be possible if the more general entities and the relationships between them were known.

The main difference between our work in this paper and all previous work aimed at relationship extraction is, that our extraction mechanism, in contrast with most past work, can easily be applied to any domain where a well defined ontology schema and set of know entity instances is available. For this project, we choose the biomedical domain since it has all the characteristics that are required to demonstrate the usefulness of the structured data we extract.

3. Our approach

The general problem of relationship extraction from text is very hard. Our approach recognizes and takes the advantage of special circumstances associated with the biomedical domain. More specifically, we leverage the availability of a controlled vocabulary called the Medical Subject Headings (MeSH) [15] and domain knowledge in the form of the Unified Medical Language System (UMLS) [16]. We combine this domain knowledge with some of the established NLP techniques for relationship extraction. The use of domain knowledge eliminates the need for two key constituent,

but challenging steps, namely Named Entity Identification and Named Entity Disambiguation/Reference Reconciliation, both of which are required before relationships can be extracted.

MeSH is a controlled vocabulary organized as a taxonomy, which is currently used to index and retrieve biomedical abstracts from the PubMed database. We treat MeSH terms as 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*. Our aim in this paper is to identify and extract these three types of entities and relationship between them occurring in biomedical text. In this paper:

1. We use an off-the-shelf part-of-speech tagger [17] and a chunk parser [18] to produce parse trees of sentences in biomedical abstracts. This is described briefly in Section 4.2.1.
2. We present a rule-based post-processing technique to enrich the generated parse trees. The rules serve to identify complex entities and known relationships between them. This is described in detail in Section 4.2.2.
3. The conversion of these processed trees to the corresponding RDF structures is described in Section 4.3. Sample sentences from PubMed abstracts are used to illustrate the effectiveness of our methodology.
4. An evaluation of the effectiveness of our post-processing rules in terms of precision and recall is presented in Section 5. The dataset which provides the framework for this study is also discussed in Section 5.
5. Finally, we demonstrate the usefulness of our results in the context of Semantic Analytics, presented in Section 5.

4   Relationship Discovery

In this section we describe the features of our dataset used in our research. We then detail the methodology for relationship extraction.

4.1 Dataset

As mentioned earlier, PubMed contains over 16 million abstracts of biomedical publications. Each abstract is uniquely identified by a PubMed ID (PMID). These abstracts are manually classified by domain experts and annotated as pertaining to one or more entities in the MeSH hierarchy. MeSH contains 22,807 named entities which include 316 pharmacological names. UMLS contains a Semantic Network containing 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 Named Entity Identification and Disambiguation for the purposes of this paper. 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. We use this information to spot named relationships occurring in PubMed abstracts. We split biomedical abstracts into sentences and generate RDF on a per-sentence basis. Therefore, in this paper we do not address the problem of Co-Reference Resolution or Pronominal Anaphora Resolution.

4.2 Methodology

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

[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.

Fig. 1. Sample sentences from abstract of PMID-1254239 for illustration (Numbers in the figure indicate PubMed ID-Sentence Number)

4.2.1 Part-of-speech Tagging and Parsing

Given a sentence, our first step is to tag parts-of-speech in the sentence and parse it to generate a parse tree. We use the SS-Tagger [17] to tag sentences, which claims to offer fast 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 we use the SS-parser [18]. 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 [19-21], the speed of this parser makes it a better choice for us. A comparison of our results obtained by using each of these parsers is something we plan to investigate in the future. We also plan to consider domain specific parsers [22].

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 Fig. 2, 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, we 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* of that node throughout this paper.

4.2.2 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 sub-sections, we 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, we present an algorithm for extracting relationships between the identified entities in the sentence.
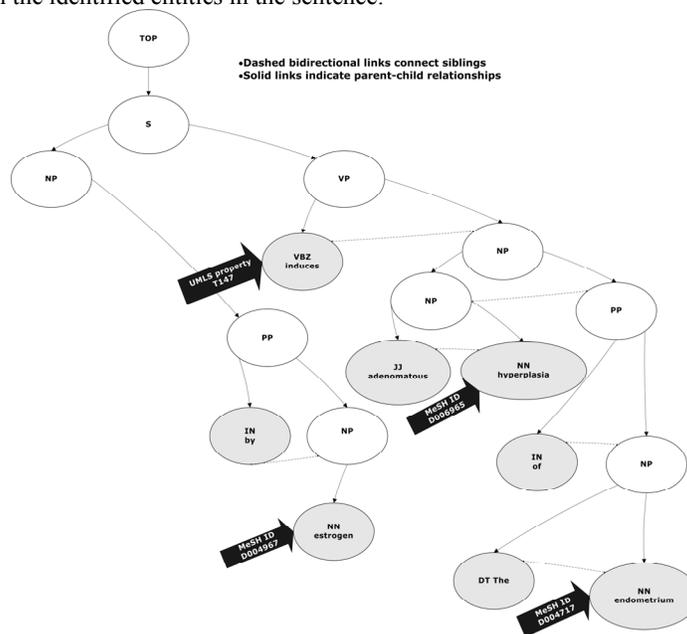

Fig. 2 Fragment of the parse Tree (Shaded nodes are terminals (_T) and clear nodes are non-terminals (_NT))

4.2.2.1 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*.

Table 1 Symbols used and their definitions

Symbols	Definitions
SE	*Simple Entity*
M	*Modifier*
ME	*Modified Entity*
CE	*Composite Entity*
R	*Relationship*
T	*Terminal node in parse tree*
NT	*Non-Terminal node in parse tree*

The definitions discussed 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, our model is very similar to the one in [23] which the author uses to learn linguistic structures from text. In [23], 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 punctuations. The purpose of this model is the induction of rules that capture linguistic structure. However, it does not account for named relationships connecting entities. Therefore, although some of our ideas are similar to the ones in [23], the overall purpose is very different.

4.2.2.2 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.

4.2.3 Algorithm for Modified and Composite Entity Identification

In this section we 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:

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.

4.2.5 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, we use the following algorithm.
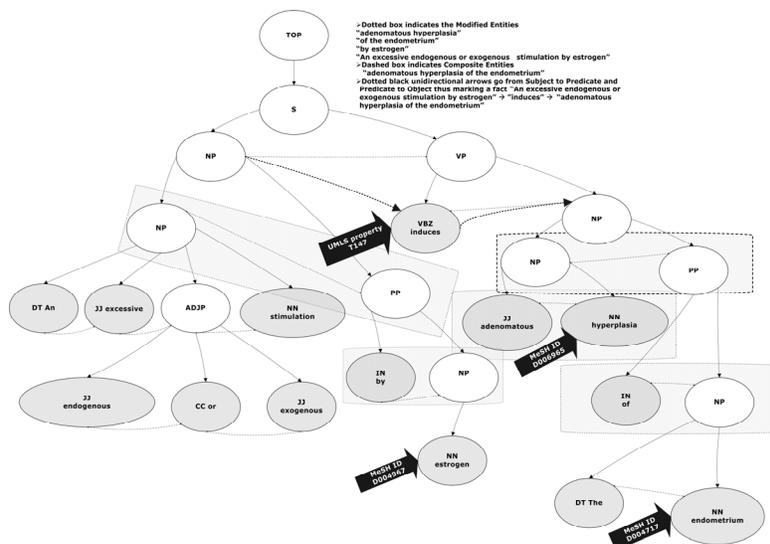
Figure 3 Processed tree showing modified entities, composite entities and a relationship "induces"

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, we use the information from the UMLS schema. Note that a candidate subject (Subject) and object (Object) of the suggested relationships could be composite or modified entities as per our definitions. Further, note that RDFS allows a property to have multiple domains and ranges. Let the domain and the range of R be the sets $domain(R) = \{C_1, C_2... ,C_n\}$ and $range(R) = \{C_1, C_2... ,C_m\}$. If $\exists C_i, C_j$ for $1 \le i \le n$ and $1 \le j \le m$ such that $C_i \in Subject$ and $C_j \in Object$ then we say that the Subject and Object are related by the relationship R. Fig.3. shows the relationship *"induces"* between the modified entity *"An excessive endogenous or exogenous stimulation by estrogen"* and *"adenomatous hyperplasia of the endometrium"*.

4.3 Serializing Identified Structures in RDF

In this section we use the running example of sentence 1254239-1 to describe the RDF resources generated by our method.

4.3.1 Simple Entities in RDF

Fig. 4. shows the RDF generated for simple entities. Note that the MeSH term identifiers are used here as URIs for the resources corresponding to each simple entity.

```
<rdf:Description rdf:about="#D004967">
    <rdfs:label xml:lang="en">estrogen</rdfs:label>
    <rdf:type rdf:resource="#Organic_Chemical"/>
    <rdf:type rdf:resource="#Pharmacologic_Substance"/>
    <rdf:type rdf:resource="#Hormone"/>
    <umls:hasSource>1254239-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="#D006965">
    <rdfs:label xml:lang="en">hyperplasia</rdfs:label>
    <rdf:type rdf:resource="#Pathologic_Function"/>
    <umls:hasSource>1254239-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="#D004717">
    <rdfs:label xml:lang="en">endometrium</rdfs:label>
    <rdf:type rdf:resource="#Body_Part_Organ_or_Organ_Component"/>
    <umls:hasSource>1254239-1</umls:hasSource>
</rdf:Description>
```

Fig. 4 RDF serialization of Simple Entities

4.3.2 Modified Entities in RDF

To generate RDF for the modified entities we need to create a resource corresponding to each modifier. Therefore, we have augmented the UMLS schema with a generic class which we call *umls:ModifierClass*. In addition, we have created a special property *umls:hasModifier*. This property has domain *rdf:resource* and range *umls:ModifierClass*. Using this property. instances of *umls:ModifierClass* are attached to instances of *rdf:resource* that are entities. Fig. 5(a). shows the RDF resources generated for the modified entities in sentence 1254239-1.

4.3.3 Composite Entities in RDF

By definition, composite entities are made up of one or more simple of modified entities. To create such composites, we had to further augment the UMLS schema to include a new class *umls:CompositeEntityClass* and a new property *umls:hasPart*. The new property has as its domain and range *rdf:resource* and therefore serves to connect the parts of a composite to the resource that represents the composite entity. Fig. 5(b) shows the composite extracted from sentence 1254239-1.

4.3.4 Property instances in RDF

Each of the 49 relationship in UMLS has been defined with its appropriate domain and range in the UMLS schema. For instance, the verb *induces* is a synonym of the property *umls:causes*. This property has several domains and ranges. One pair of classes that this property relates is *umls:Pharmacologic_Substance* and *umls:Pathologic_Function*. Since *estrogen* is an instance of *umls:Pharmacologic_Substance* (Fig. 5(a)) and "hyperplasia" is an instance of class *umls:Pathologic_Function*, we generate the RDF shown in Fig. 5(c).

```
<rdf:Description rdf:about="#m_1">
    <rdfs:label xml:lang="en">adenomatous</rdfs:label>
    <rdf:type rdf.resource="#ModifierClass"/>
    <umls:hasSource>1254239-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="#me_1">
    <rdfs:label xml:lang="en">adenomatous_hyperplasia</rdfs:label>
    <umls:hasModifier rdf:resource="#m_1"/>
    <umls:hasPart rdf:resource="#D006965"/>
    <umls:hasSource>1254239-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="#m_2">
    <rdfs:label xml:lang="en">of</rdfs:label>
    <rdf:type rdf:resource="#ModifierClass"/>
    <umls:hasSource>1254239-1</umls:hasSource>
</rdf:Description>

<rdf:Description rdf:about="#me_2">
    <rdfs:label xml:lang="en">of_the_endometrium</rdfs:label>
    <umls:hasModifier rdf:resource="#m_2"/>
    <umls:hasPart rdf:resource="#D004717"/>
    <umls:hasSource>1254239-1</umls:hasSource>
</rdf:Description>
```

(a)

```
<rdf:Description rdf:about="#ce_1">
    <rdfs:label xml:lang="en">me_1|me_2</rdfs:label>
    <rdf:type rdf:resource="#CompositeEntitiesClass"/>
    <umls:hasPart rdf:resource="#me_1"/>
    <umls:hasPart rdf:resource="#me_2"/>
    <umls:hasSource>1254239-1</umls:hasSource>
</rdf:Description>
```

(b)

```
<rdf:Statement rdf:about="#triple_1">
    <rdfs:label xml:lang="en">me_4|induces|ce_1</rdfs:label>
    <rdf:subject rdf:resource="#me_4"/>
    <rdf:predicate rdf:resource="#induces"/>
    <rdf:object rdf:resource="#ce_1"/>
    <umls:hasSource>1254239-1</umls:hasSource>
</rdf:Statement>
```

(c)

Fig. 5. RDF serialization of (a) Modifiers and Modified entities (b) Composite Entities and (c) Instance of a relationship between entities

5 Discussion of Results

In our experiments, we tested our 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 we refer to as ALLNEOPLASMS in this paper. 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 [4], we focus our 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 we searched PubMed with pair-wise combinations of each intermediate entity with both Migraine and Magnesium, respectively. This resulted in a set of approximately 800 abstracts. We call this set MAGNESIUMMIGRAINE. Our objective in extracting triples from the ALLNEOPLASM set at this point is to test the scalability of our system. In the future, we plan to sample the generated triples to evaluate our methodology in terms of precision and recall. Processing approximately 1.6 million candidate sentences from the ALLNEOPLASM set resulted in over 200,000 triples. In the case of the MIGRAINEMAGNESIUM test our objective was to investigate two aspects of our results. They can be characterized by the following questions.

Question 1: How effective are our rules in extracting relationships and the entities involved from text?

Questions 2: How useful is the extracted RDF data?

We 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, we identified 798 candidate sentences. These sentences are therefore the ones which we expect to generate instances of relationships. In our 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). We 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 relationship. We plan to conduct just such an experiment in the future. This is however infeasible for larger datasets. We analyzed those candidate sentences that did not produce relationship instances. In our approach to relationship extraction we 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. We plan to increase the recall of our 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. We therefore study the usefulness of the extracted relationships in the context of the Undiscovered Public Knowledge.

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, we run a simple bidirectional length-limited breadth first search for paths connecting Migraine and Magnesium. We decided to limit the path length since we expected the number of paths to be prohibitively large, and since very long paths are seldom of interest. As expected, there are a very large number of paths and this number increases exponentially with path length. 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.

Table 2 Paths between Migraine and Magnesium

Paths between Migraine and Magnesium			
Path length	Total Number of paths found	# of interesting paths	Max. # of named relationships in any path
6	260	54	4

| 8 | 4103 | 1864 | 5 |
| 10 | 106450 | 33403 | 5 |

To see the value of these paths, we examined some of the paths among those of length 6. We focused our attention on the ones that had 2-3 named relationships. Fig. 6 below 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 an inhibitory effect on collagen induced platelet aggregation (PMID 10357321, sentence number 7). We have included here 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 our 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.
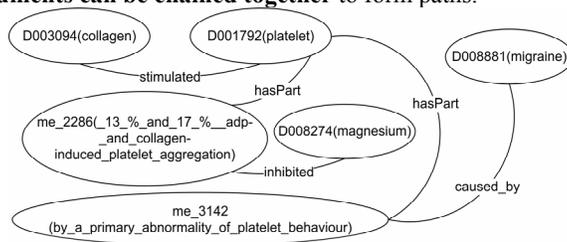
Fig. 6 Example path between Magnesium and Migraine

The edges in the figure are left undirected although the relationships are directed in the generated RDF. Directionality of these relationships can be deduced from the schema. The generated RDF can serve as the foundation for applying analytical operators such as those in [5] and [6] to provide support for discovering Undiscovered Public Knowledge. All the generated data from our experiments in this paper is available at http://lsdis.cs.uga.edu/projects/semdis/relationExt/.

6 Applications and Future work

In order to thoroughly evaluate the accuracy of our extracted relationships and consequently that of the resulting paths, we plan to enlist the help of a domain expert. We plan to do this for the MIGRAINEMAGNSIUM dataset. We also plan to test this on the Fish Oils and Raynaud's disease associations. We plan to investigate the following potential applications resulting from our work:

"Semantic" Browsing - Our next natural step is to superimpose the extracted RDF back onto the original text and annotate biomedical abstracts with entities and relationships between them. We envision a Semantic Browsing paradigm in which the user of such a Semantic Browser will be able to traverse a space of documents based

on named relationships between entities of interest. This vision is in line with the *"trailblazing"* idea posited by Dr. Vannevar Bush [1].

Knowledge-Driven ("Semantic") Document Retrieval - Paths between entities in our generated RDF instance base can be used as a query for documents. A simple example of such a query can be seen in the association between Migraine and Magnesium, where intermediate entities like Stress or Calcium Channel Blockers would serve to constrain the returned documents to only that set which corroborates the said associations.

Semantic Analytics over Literature - The operators described in [5] return paths between entities in the query. The sub-graph discovery operator described in [6] takes as input two entities in an RDF instance base and returns a set of paths between them that are not vertex-disjoint (i.e. forming a sub-graph). Applying these queries to RDF generated by mining biomedical literature will allow us to quantify the relevance of the returned paths. This gives rise to a very powerful mechanism for exploratory analysis of large document sets.

7 Conclusions

Our experiments have demonstrated the utility of extracting relationships from biomedical text to support analytical queries. The effectiveness of our method augmented with rules to extract more complex structures remains to be investigated. It is however clear that domain knowledge can be effectively combined with NLP techniques to good effect. We intend to continue this work and investigate the use of other vocabularies in addition to MeSH to aid in relationship extraction. The relationship-centric view of document organization, in our opinion, will mark the next generation of search and analytics over document corpora. This work is funded by NSF-ITR-IDM Award#0325464 (SemDIS: Discovering Complex Relationships in the Semantic Web).

8 References

1. Bush, V., *As We May Think.* The Atlantic Monthly, 1945. **176**(1): p. 101-108.
2. NLM, *PubMed*, The National Library Of Medicine, Bethesda MD.
3. Swanson, D.R., *Fish Oil, Raynaud's Syndrome, and Undiscovered Public Knowledge.* Perspectives in Biology and Medicine, 1986. **30**(1): p. 7-18.
4. Swanson, D.R., *Migraine and Magnesium: Eleven Neglected Connections.* Perspectives in Biology and Medicine, 1988. **31**(4): p. 526-557.
5. Anyanwu, K. and A. Sheth, *ρ-Queries: enabling querying for semantic associations on the semantic web*, in *Proceedings WWW*. 2003, ACM Press: Budapest, Hungary.
6. Ramakrishnan, C., et al., *Discovering informative connection subgraphs in multi-relational graphs.* SIGKDD Explor. Newsl., 2005. **7**(2): p. 56-63.
7. Guha, R., R. McCool, and E. Miller, *Semantic search*, in *WWW '03* p. 700-709.
8. Cohen, A.M. and W.R. Hersh, *A survey of current work in biomedical text mining.* Brief Bioinform, 2005. **6**(1): p. 57-71.

9.   Tanabe, L. and W.J. Wilbur, *Tagging gene and protein names in biomedical text.* Bioinformatics, 2002. **18**(8): p. 1124-1132.
10.   Brill, E., *Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging.* Comput. Linguist., 1995. **21**(4): p. 543-565.
11.   Yu, H., et al., *Automatically identifying gene/protein terms in MEDLINE abstracts.* J. of Biomedical Informatics, 2002. **35**(5/6): p. 322-330.
12.   Gaizauskas, R., et al., *Protein structures and information extraction from biological texts: the PASTA system.* Bioinformatics, 2003. **19**(1): p. 135-143.
13.   Friedman, C., et al., *GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles.* Bioinformatics, 2001. **17 Suppl 1**: p. 1367-4803.
14.   Rindflesch, T.C., et al., *EDGAR: extraction of drugs, genes and relations from the biomedical literature.* Pac Symp Biocomput, 2000: p. 517-528.
15.   NLM, *Medical Subject Heading (MeSH)*, The National Library Of Medicine, Bethesda, MD.
16.   NLM, *Unified Medical Language System (UMLS)*, The National Library Of Medicine, Bethesda, MD.
17.   Tsuruoka, Y. and J.i. Tsujii, *Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data*, in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. 2005, Association. p. 467-474.
18.   Tsuruoka, Y. and J.i. Tsujii, *Chunk Parsing Revisited*, in *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT 2005)*. 2005. p. 133-140.
19.   Charniak, E., *A maximum-entropy-inspired parser*, in *Proceedings of the first conference on North American chapter of the ACL*. 2000, Morgan. p. 132-139.
20.   Collins, M., *Head-driven statistical models for natural language parsing.* 1999.
21.   Collins, M. and N. Duffy, *New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron*, in *ACL '02* p. 263-270.
22.   Tsuruoka, Y., et al., *Developing a Robust Part-of-Speech Tagger for Biomedical Text*. Lecture Notes in Computer Science. 2005. 382-392.
23.   Déjean, H., *Learning rules and their exceptions.* J. Mach. Learn. Res., 2002. **2**: p. 669-693.