Ontology-Based Information Extraction on PubMed abstracts using the OMIT ontology to discover inconsistencies

Nisansa de Silva
Department of Computer and Information Science
University of Oregon
Email: nisansa@cs.uoregon.edu

***Abstract*—Scientific progress, at its core, is about constant change. Progress is brought about by introducing new findings where there were none, altering previously established knowledge, and in some cases, deconstructing prior knowledge to replace it with new findings. This process happens, generally, through scientific publications; in journals, at conferences, at workshops, in articles. Thus it is possible to analyze these documents and see how knowledge would change with time. This study accomplishes said task in the domain of MicroRNA which is a subdomain of Medical Science. Research paper abstracts are taken from an online medical research database and are analyzed to notice any changes in paradigms, ideas or claims.**

I. Introduction

Recognizing how knowledge alters over the course of time is important for various analytical tasks. It is also important to note these changes because as the Latin metaphor *"nanos gigantum humeris insidentes"* says, the discovery of truth is done by building on previous discoveries [1]. Thus, any changes in the foundation upon which some later research is done, and conclusions drawn thereupon, will lead to a need for re-evaluation of that later research.

While important, research about extracting information from the abstracts of biomedical papers is limited to a very narrow area of topics. An example is the seminal work by Kulick, et al. [2] that extracted information on drug development and cancer genomics.

II. Background

Information extraction is a process to acquire knowledge by looking for occurrences of a particular class of objects and looking for relationships among objects in a given domain. The objective of information extraction is to find and retrieve certain types of information from text. However, it does not attempt to comprehend natural language. Comprehending natural language is handled by the research area; *natural language understanding*. *Natural language understanding* is what chat bot AIs or personal assistant AIs attempt to do. Information extraction is also different from *information retrieval*, which retrieves documents or parts of documents related to a user query from a large collection of documents. *Information retrieval* is what search engines do. The main difference between *information retrieval* and *Information extraction* is that the latter goes one step further by providing the required information itself, instead of a pointer to a document.

In an information extraction task, the input is either unstructured text or slightly structured such as HTML or XML. Usually the output is a template set filled in with various information that the system was supposed to find. Thus, the information extraction process is a matter of analyzing document(s) and filling template slots with values extracted from document(s).

There are two main methods of information extraction: (a) attribute-based extraction; and (b) relation extraction. In attribute-based extraction, the system assumes the entire text to be referring to a single object. Thus, the task is to extract attributes of said object. This is typically done using regular expressions. Relation extraction, on the other hand, extracts multiple objects, and relationships thereof from a document. One famously efficient way to do this is the FASTUS method by Hobbs et. al [3].

A. Information Extraction techniques

There are many techniques used for information extraction. Some are widely used and some others are not as much. The most widely used information extraction technique is *linguistic rules represented by regular expressions*. In this method, a domain expert explicitly writes down linguistic rules of how various concepts are mentioned in the domain of the relevant concept. Then those rules are converted to rules based on regular expressions. Part-of-speech (PoS) taggers and noun phrase chunkers are needed to prepare the text to be given to these rules. The set of regular expressions are often implemented using finite-state transducers, which are made up of a series of finite-state automata. An example of a simple rule of this manner is $(watched|seen)\ <NP>$; where $<NP>$ stands for a *Noun Phrase*. This rule can capture names of movies, plays, and tv-shows from sentences such as "I have watched Amadeus" or "He has seen Hamilton". While most of these systems have manually constructed rules, some do attempt to automatically mine extraction rules from text. However, attempts at automatically mining extraction rules from text are always riddled by the presence of conceptual drift. Thus, it is preferred to get a domain expert to write the rules, as explained above. However, this has the obvious problem of being a highly tedious, human labour intensive task.

Another famous information extraction technique is the usage of gazetteer lists. The idea comes from gazetteers, which were geographical dictionaries or directories used in conjunction with a map or atlas [4]. Thus, a gazetteer list is a list of potential items that pertain to a given concept. The extraction task is, in this case, checking the text to see any instance of any of the list items and reporting back whether there were any, along with the items that were found. Similar to the previous method, this too, uses finite-state automata. However in this case, the system recognizes individual words or phrases, instead of patterns that span sentences.

Machine learning now has a place in every subfield of computing. Thus, it is not surprising to find information extraction techniques based on machine learning. In most machine learning information extraction tasks, the process is designed as a set of binary classification tasks. The models are trained using tagged instances, and subsequent test documents are classified using the learnt model.

An information extraction technique specific to HTML/XML documents is analyzing tags. For example, in most cases, the first row of a table denotes attributes. The following rows indicate the attribute values for individual records or instances. This is used to extract attribute information for the said records or instances with labels. XML tends to be easier for this type of information extraction, given the fact that XML allows users to define their own tags with custom schemas.

Another method, albeit one used less often is the web-based search approach. In this, a given document is annotated by searching the proper nouns found on that document on the web. This method is a meta-method given that the retrieved webpages have to be subjected to some other form of information extraction so that the tags for annotation can be extracted from those documents to tag the original document.

B. Medical background sources

Background knowledge of the following medical domain sources is helpful to contextualize the present study.

1) Micro RNA: MicroRNA (miRNA) is a small non-coding RNA molecule found in plants, animals and some viruses, that functions in RNA silencing and post-transcriptional regulation of gene expression. They contain about 22 nucleotides. Majority of miRNAs are located within cells. However some do exist in the extracellular environment.

Gene regulation seems to be the function of miRNAs. For this, miRNAs play a complementary role to mRNAs (messenger RNAs) [5].

2) PubMed: PubMed [6] is a free search engine accessing primarily the MEDLINE database of references and abstracts on life sciences and biomedical topics. The United States National Library of Medicine (NLM) at the National Institutes of Health (NIH) maintains the database as part of the Entrez system of information retrieval.

Many PubMed records contain links to full text articles, some of which are freely available, often in PubMed Central. As of 8 February 2015, PubMed has over 24.6 million records going back to 1966, selectively to the year 1865, and very selectively to 1809; about 500,000 new records are added each year. As of the same date, 13.1 million of PubMed's records are listed with their abstracts, and 14.2 million articles have links to full-text (of which 3.8 million articles are available full-text for free for any user).

3) Medical Subject Headings (MeSH): Medical Subject Headings (MeSH) [7] is a comprehensive controlled vocabulary for the purpose of indexing journal articles and books in the life sciences; it serves as a thesaurus that facilitates searching. Created and updated by the United States National Library of Medicine (NLM), it is used by the MEDLINE/PubMed article database and by NLM's catalog of book holdings. MeSH is also used by ClinicalTrials.gov registry to classify which diseases are studied by trials registered in ClinicalTrials.gov [8].

MeSH was introduced in 1960, with the NLM's own index catalogue and the subject headings of the Quarterly Cumulative Index Medicus (1940 edition) as precursors. The yearly printed version of MeSH was discontinued in 2007, and MeSH is now available online only. It can be browsed and downloaded free of charge through PubMed. Originally in English, MeSH has been translated into numerous other languages and allows retrieval of documents from different languages.

C. Ontology

An ontology is defined as "formal, explicit specification of a shared conceptualisation" [9] in information science. The term is borrowed from philosophy, where an Ontology is a systematic account of Existence. An ontology consists of a set of concepts from a selected domain and relationships among the concepts. In addition to this, an ontology has data type properties and object properties. Conversely, the primary task of an ontology is to provide the vocabulary to model the domain that it represents by providing the types of concepts that exist in the domain along with their properties and interrelations [10]. This set of objects (concepts), and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge.

Ontologies are used to organize information in many areas as a form of knowledge representation. These areas include; artificial intelligence, the Semantic Web, biomedical informatics, library science, enterprise bookmarking, and information architecture. In each of these use cases the ontology may model either the world or a part of it as seen by the said area's viewpoint.

At the ground level of an ontology are "Individuals" (instances). Depending on the nature of the Ontology in question, this may include concrete objects or abstract objects. Concrete objects may be people, animals, planets, etc. Abstract objects maybe numbers or words. An ontology may carry property values of these instances as well.

Individuals are grouped into structures called "classes". A class in an ontology can be referred to as a concept, type, category, or a kind, depending on the domain on which the ontology is based. Yet again, depending on the convention used with a particular ontology, the definition of a class and the role thereof can either be analogous or distinct from that of a collection of individuals. One important ability of classes in an ontology for this research is the ability to subsume, or be subsumed by, another class. A class that has been subsumed by another class is called a "subclass", while a class that has subsumed one or more classes is called a "superclass".

By this act of subsuming, a hierarchy of classes is formed. The critical importance of the subsumption relation is the phenomenon called "inheritance" whereby the properties of the subsuming class are inherited by the subsumed class. Traditionally the subsuming class is called the "parent" and the subsumed classes are called "children". Thus two or more classes that are sharing a parent would refer to each-other as "sibling". This naming convention is used throughout this paper. This child-parent relationship, while being the most basic, is not the only relationship type seen in a well-defined ontology. Other constraint rules that are specific to the domain also come with the ontology. These constraints might sometimes be of the following types: *antonym* relationship where it defines instances in one class to be the antonyms of the instances in some other class; or a *part of* relationship in which instances in one class are parts or components of the instances in some other class. There are also *measurement* relationships, *substance* relationships, and more complex relationship rules (axioms). An example of a complex relationship rule is if an instance is added to this particular class and some other particular class, a relevant third instance has to be added to yet another class. A good specific example for this situation is a genealogy ontology, where adding a $MarriageEvent$ to the relevant class would mean there will be additions to $family$ class as well as to the *spouse* class. These relationships are crucial for the ontology-based information extraction process as shown in section II-D1.

1) Ontology for MIcroRNA Target (OMIT): The Ontology for MIcroRNA Target (OMIT) [11], [12] was created with the purpose of establishing data exchange standards and common data elements in the microRNA (miR) domain. Biologists and bioinformaticians can make use of OMIT to leverage emerging semantic technologies in knowledge acquisition and discovery for more effective identification of important roles performed by miRs (through their respective target genes) in humans' various diseases and biological processes. The OMIT has reused and extended a set of well-established concepts from existing bio-ontologies; e.g., Gene Ontology [13], Sequence Ontology [14], PRotein Ontology [15], and Non-Coding RNA Ontology (NCRO) [16], [17]. It has the metrics shown in table I.

Table I
METRICS OF OMIT

Number of classes:	2226
Number of individuals:	1158
Number of properties:	126
Maximum depth:	35
Maximum number of children:	316
Average number of children:	14
Classes with a single child:	280
Classes with more than 25 children:	104
Classes with no definition:	2226

As it is explained in section II-D1, one most important component of an ontology for an OBIE system is the set of relationships present in the ontology. They are the ones that can be used to build extraction rules for the information extraction system. This is exactly the problem with OMIT. Even though it has a very extensive hierarchy of concepts and instances, it has little to no relationships between the said entities. Thus some of the most powerful conventional OBIE methods discussed in section II-D1 cannot be used alongside OMIT. Later sections discuss how this study overcame this challenge.

2) Wordnet: WordNet [18] is well-known large lexical ontological [10], [9] database developed by the Cognitive Science Laboratory of the Princeton University, United States by Miller et al. [19]. It is a representation of the semantic relationships between words, which are grouped together into sets of synonyms called synsets. The database contains more than 150,000 different words. In addition to this, each word is coupled with a short description so that it can be used as a Dictionary as well as a thesaurus. The Database and the accompanying software tools are released under a BSD type license.

Out of the various semantic mappings present in WordNet, this study utilizes the Hyponym - Hypernym mapping and Antonym mapping. Fig. 1 shows an example of an extract of the Hyponym – Hypernym tree present in WordNet.

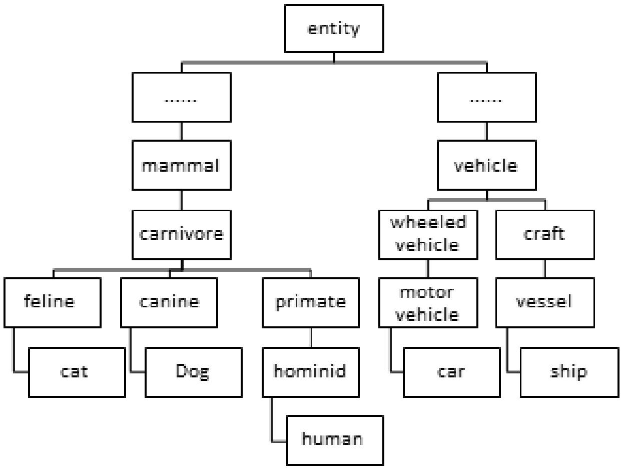


Figure 1. Hyponym – Hypernym graph [20]

As will be mentioned in Section II-D1, most OBIE systems use WordNet as their lexicon. Even though it is adequate for general-purpose information extraction, it does not contribute well for specific domains. Thus, in this study, while we use WordNet as the primary lexicon, we also build a supporting, complimentary lexicon in Section III-D, using the medical abstract text corpus we already have, to compensate for this shortcoming.

D. Advanced Information Extraction Methods

Given that this study is involved in non-trivial information extraction, it is not possible just to be content with the basic IE techniques discussed in Section II-A. Thus following Information Extraction (IE) and Information Retrieval (IR) methodologies are used.

1) Ontology Based Information Extraction: Ontology-based information extraction (OBIE) is a subfield of information extraction. In this, ontologies are used to make the information extraction process more efficient and effective. In most cases the output is also presented through an ontology. But that is not a requirement. As mentioned in II-C, generally, ontologies are specified for particular domains. Given that information extraction is essentially concerned with the task of retrieving information for a particular domain as mentioned in the first paragraphs of Section II, it is rational to conclude that an ontology that has formally and explicitly specified the concepts in that domain would be helpful in this process.

A more formal definition of OBIE was given by Wimalasuriya and Dou in [21]: “a system that processes unstructured or semi-structured natural language text through a mechanism guided by ontologies to extract certain types of information and presents the output using ontologies.”

Following is a brief introduction as to how an ontology can improve the information extraction process.

> Les Misérables, is a novel by Victor Hugo. The main character is Jean Valjean, a peasant. The story is about his quest for redemption after serving nineteen years in jail. Valjean decides to break his parole and start his life anew after a kindly bishop inspires him by a tremendous act of mercy, but he is relentlessly tracked down by a police inspector named Javert.

Figure 2. Les Misérables description adapted from [22]

Consider the description of the book Les Misérables given in Fig. 2. It can be observed that a named entity recognition process would extract the proper nouns; *Les Misérables*, *Victor Hugo*, *Jean Valjean*, *Javert*. But a general information extraction system would not know what each of these proper nouns are. A human, on the other hand, would know that *Victor Hugo* is an actual person, while *Jean Valjean* is a character in the story. This exactly is the problem solved by OBIE. For the book domain, a simple ontology can be introduced, as shown in Fig 3.

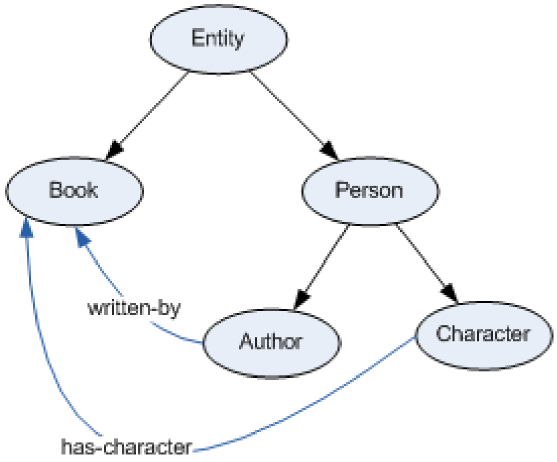


Figure 3. A simple ontology for books

With the help of this ontology, it is possible to determine that since the *written-by* relationship is there, *Victor Hugo* is the *Author* and *Les Misérables* is the book. With the *has-character* relationship, it is possible to extract that *Jean Valjean* is a character in the book *Les Misérables*. OBIE systems that use the GATE architecture rely at least partly on Linguistic rules represented by regular expressions such as this.

Another way that an Ontology can facilitate information extraction is by creating gazetteer lists. The process of creating a gazetteer list from an ontology is rather straight-forward: The tree of the concept hierarchy that is rooted at the desired concept is selected, and all the instances that occur in the said rooted tree are then added to the gazetteer list.

When information extraction is done with machine learning algorithms, it is possible to use ontologies in several ways. Classification algorithms can be used to recognize instances and property values from the ontology. Maximum entropy models can be used to predict attribute values in a sentence. Similarly, Conditional Random Fields (CRF) can be used to identify attribute values in a sentence.

The above-described methodologies make up the *Information Extraction Module* of a typical Ontology-Based Information Extraction system. Figure 4 shows the simplest form of an OBIE system.

Other than the *Ontology* and the above described *Information Extraction Module*, there are two other main components in an OBIE system. The first one is the *Preprocessor*. The text input of an OBIE system first goes through a preprocessor component, which converts the text to a format that can be handled by the IE module. For example, tags from an HTML file can be removed in this component. Thus the *Information Extraction Module* would be receiving pure text content.

A *semantic lexicon* for the language is usually used as a helper for the *Information Extraction Module*. As mentioned in section II-C2, for the general English language based information extraction tasks it is most common to use the WordNet [18] lexical database and toolkit thereof.

2) Open Information Extraction: The requirement of having pre-specified relations of interest is the main drawback

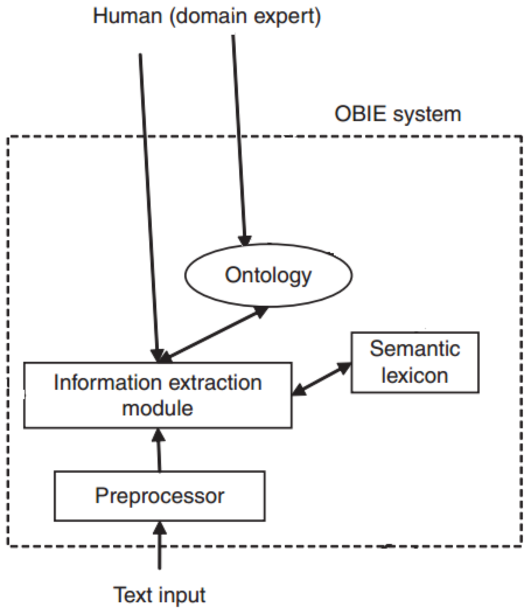


Figure 4. A simple Ontology-Based Information Extraction system

of the traditional information extraction systems. Open Information Extraction systems solve this problem by extracting relational triples from text, by identifying relation phrases and associated arguments in arbitrary sentences without requiring a pre-specified vocabulary. Thus it is possible to discover important relationships that are not pre-specified.

Usually, Open Information Extraction systems automatically identify and extract binary relationships from sentences given the parsed text of the target language. The parsed text provides the dependency relationships between the various phrases of the sentence. The Open Information Extraction system used in this paper, OLLIE [23], is different from others in its genre in that it works on a tree-like representation (a graph with only small cycles) of the dependencies of the sentence, based on the Stanford's compression of the dependencies, while other Open Information Extraction systems operate on flat sequences of tokens. Thus OLLIE is uniquely qualified to capture even long-range relations.

Given that open information extraction does not depend on pre-configured rules, we are using Open Information Extraction as a bridge between OMIT, which is an ontology with little to no relations as described in section II-C1, and the conventional OBIE methods described in II-D1. More information on this is discussed in Section VIII.

3) Semantic Similarity Measure: Semantic similarity of two entities is a measure of the likeness of the semantic content of the said two entities. It is common to define semantic similarity using topological similarity by means of ontologies.

Using WordNet, Wu and Palmer [24] proposed a method to give the similarity between two words in the 0 to 1 range. The approach proposed by Jiang and Conrath measures the semantic similarity between word pairs using corpus statistics and lexical taxonomy [25]. By means of [26], the strengths of these algorithms were evaluated and Wu and Palmer's implementation was selected for the purposes of this paper.

A set of examples of word similarities are shown in Table II. For the similarity with Car, the same word gets the perfect score of 1; $Truck$ gets a higher score than $Ship$, because a $Truck$ too, is a land vehicle, like a Car, however, $Ship$ gets a higher score than $Book$ because a $Ship$ is a vehicle and a $Book$ is not, $Book$ gets a higher score than Air because the $Book$ is solid and Air is not, Air gets a higher score than $Thought$ because Air is a physical entity and a $Thought$ is not.

Table II
WORD SIMILARITIES USING WU AND PALMER METHOD

Word 1	Word 2	Similarity
Car	Car	1.0000
Car	Truck	0.9231
Car	Ship	0.7200
Car	Book	0.5217
Car	Air	0.3158
Car	Thought	0.2105

A useful observation from this is the fact that, no matter how dissimilar two words are, if both of those words exist in the WordNet, this method will return a greater than zero value. Thus, there exists an inherent bias towards declaring that two words have a non-zero similarity, rather than declaring that there exists a difference. Thus, in the sections V-B and IV, we use dissimilar weights named ”yes weight” (W_{yes}) and ”no weight” (W_{no}), where W_{no} is larger than W_{yes}.

4) TF-IDF: TF-IDF (term frequency–inverse document frequency) [27] is a statistic that is used in information retrieval to indicate how important a given word is in a document, which belongs to a certain corpus. The TF-IDF statistic has two measures. The first part is the term frequency (tf) which indicates how important the given word is in the given document. Most commonly, it is used with 0.5 double normalization where $f(T, d)$ is the frequency of term t in document d as follows in Equation 1.

$$tf(t, d) = 0.5 + \frac{0.5 * f(T, d)}{max\{f(w, d) : w \in id\}} \tag{1}$$

Second part of the statistic is the inverse document frequency (idf) where N is the total number of total documents and d is number of documents where t appears. The formula is as follows in Equation 2.

$$idf(t, D) = log\frac{N}{1 + |\{d \in D : t \in d\}|} \tag{2}$$

III. Data Preparation

The methodology is significantly different than that of [28], even though, ideologically, incremented inconsistency finding is common to the two approaches. The main difference is the fact that in [28], the inconsistencies were found by adding the discovered triplets to the existing ontology and running reasoners on it to see if the ontology has become inconsistent.

This study, on the other hand, uses the ontology as a tool in information extraction, as per the concept of OBIE, and does the inconsistency detection outside.

A. Obtaining PubMed Abstracts

The first step was to obtain a list of relevant PubMedIDs. This was done by querying the on-line PubMed site with the header ”mRNA”. The PubMedIDs are then processed to remove duplicates, and they are then separated into easily manageable files with a maximum of 1000 IDs each.

These IDs are then used to extract the abstracts out of the PubMed system. One important thing to note here is the fact that even though PubMed has an option to query its system with an ID to supposedly return the relevant abstract, we found it to be inefficient for this study. The reason for this is the following: More oft than not, the formatting of the free text was done in different ways, as shown in Fig. 5(a) and Fig. 5(b). Thus it proved that extracting the pure abstract out of this output would require some unnecessary effort. Instead, it was decided to use the XML interface provided by PubMed and extract the abstracts locally. This step corresponds to the ”preprocessor” component described in II-D1.

B. Creating Ollie triples

The downloaded free text is then subjected to the open information extraction system introduced in [23](Ollie) that was described in II-D2. This process extracts triples in the form of binary relations from the free text and creates a set of possible triplets as shown in Example 1. From this point onwards, this paper will refer to these triples as ”Ollie triples”.

Code 1. Open Information Extraction Example

```
Nevertheless, we found that miR—31 was particularly up—regulated in HSCs but not in
        hepatocytes during fibrogenesis.
0.689: (miR—31; was particularly; up—regulated)
0.661: (miR—31; was particularly up—regulated in; HSCs)
```

The first line of the example shows the original sentence itself. Then each line has an extracted triple. The number leading the triple is the confidence that the algorithm has of the triple being valid.

The remainder of the triple is of the format $(A; R; B)$ where A is the $subject$ of the relation R, and B is the $object$ of the relation R. Typically, in regular information extraction processes, as explained in the leading paragraphs of I, these relations (R) are fairly simple and would contain one to a few words. Similarly the Subject (A) and Object (B) are set out to be clear cut singular concepts. However, due to the openness of this methodology, which does not depend on any subject context-specific rule but the grammar rules of the language itself, the output of this step does not have those properties. Typically the relation name is just the text linking the subject and the object. Subject and object themselves are more often phrases rather than coherent concepts as expected. This is an issue that we rectify in a later step.

C. Creating Stanford XML files

The same free text obtained in Section III-A are sent through a system to extract other linguistic information. In this case we are using the methodology developed by Manning, et. al. [29]. The objective of this step is to extract the parse tree, get the lemmatized forms of each word, and get each sentence element separated. The parse tree of the same sentence shown in Example 1 is given in Example 2. The result of lemmatization and PoS tagging of three words of the same sentence is given in Example 3. From this point onwards, this paper will refer to these outputs for each abstract as ”Stanford XML”.

Code 2. Parse tree example

```
(ROOT
(S
(ADVP (RB Nevertheless))
(, ,)
(NP (PRP we))
(VP (VBD found)
(SBAR (IN that)
(S
(NP (NN miR—31))
(VP (VBD was)
(ADVP (RB particularly))
(VP (VBN up—regulated)
(PP
(PP (IN in)
(NP (NNS HSCs)))
(CONJP (CC but)
(RB not))
(PP (IN in)
(NP (NNS hepatocytes))))
(PP (IN during)
(NP (NN fibrogenesis))))))))
(. .)))
```

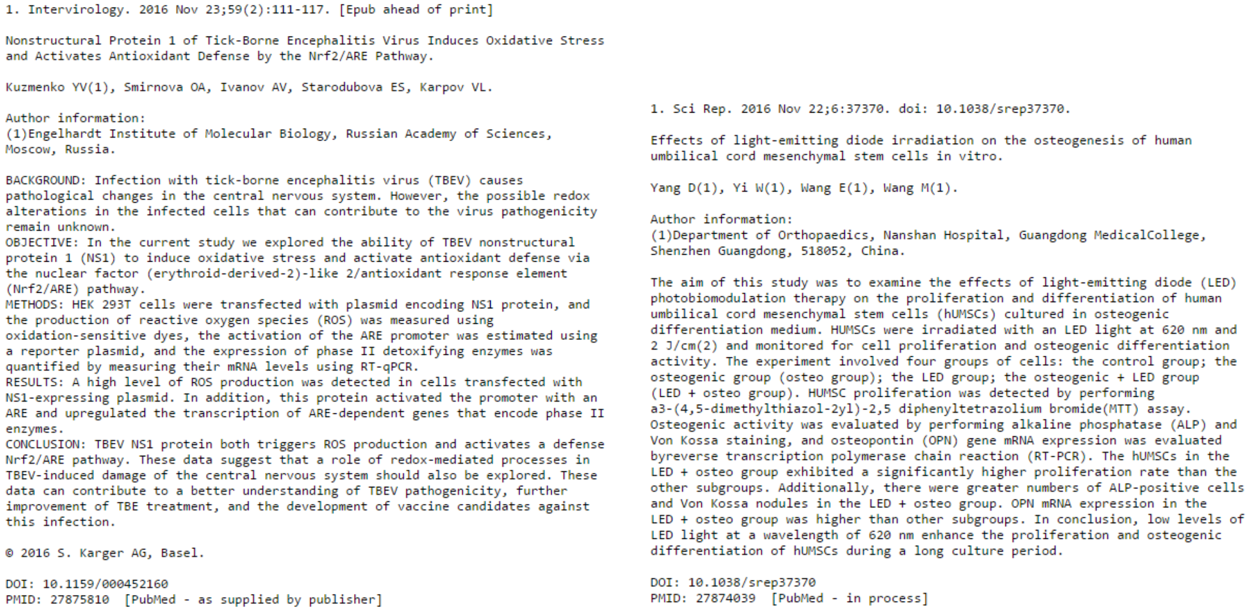
Code 3. Lemmatization example

```
<token id=”4”>
        <word>found</word>
        <lemma>find</lemma>
        <CharacterOffsetBegin>504</CharacterOffsetBegin>
        <CharacterOffsetEnd>509</CharacterOffsetEnd>
        <POS>VBD</POS>
        <NER>O</NER>
        <Speaker>PER0</Speaker>
</token>
<token id=”7”>
        <word>was</word>
        <lemma>be</lemma>
        <CharacterOffsetBegin>522</CharacterOffsetBegin>
        <CharacterOffsetEnd>525</CharacterOffsetEnd>
        <POS>VBD</POS>
        <NER>O</NER>
        <Speaker>PER0</Speaker>
</token>
<token id=”12”>
        <word>but</word>
        <lemma>but</lemma>
        <CharacterOffsetBegin>560</CharacterOffsetBegin>
        <CharacterOffsetEnd>563</CharacterOffsetEnd>
        <POS>CC</POS>
        <NER>O</NER>
        <Speaker>PER0</Speaker>
</token>
```

D. Creating medical term dictionary

Before moving on to the next part of this study, some background data have to be generated pertaining to the abstracts. As described in II-D1, a very important part in an ontology-based information extraction system is the semantic lexicon, and as stated in II-C2, Wordnet is the primary lexicon in this system. But as mentioned in the same section, due to medical domain language being specific, a general lexicon such as Wordnet is not enough to serve as the *Semantic Lexicon* for this system. Thus, a complementary lexicon has to be created with information specific to the medical domain. That is what is done in this step.

A good indication of how important a given term is in a certain domain is the frequency in which it is used within the domain. Therefore, the semantic information of term usage is vital to the following information extraction task and is not something that can be obtained via a generic lexicon such as Wordnet. Given that the semantic information that is to be



```
1. Intervirology. 2016 Nov 23;59(2):111-117. [Epub ahead of print]

Nonstructural Protein 1 of Tick-Borne Encephalitis Virus Induces Oxidative Stress
and Activates Antioxidant Defense by the Nrf2/ARE Pathway.

Kuzmenko YV(1), Smirnova OA, Ivanov AV, Starodubova ES, Karpov VL.

Author information:
(1)Engelhardt Institute of Molecular Biology, Russian Academy of Sciences,
Moscow, Russia.

BACKGROUND: Infection with tick-borne encephalitis virus (TBEV) causes
pathological changes in the central nervous system. However, the possible redox
alterations in the infected cells that can contribute to the virus pathogenicity
remain unknown.
OBJECTIVE: In the current study we explored the ability of TBEV nonstructural
protein 1 (NS1) to induce oxidative stress and activate antioxidant defense via
the nuclear factor (erythroid-derived-2)-like 2/antioxidant response element
(Nrf2/ARE) pathway.
METHODS: HEK 293T cells were transfected with plasmid encoding NS1 protein, and
the production of reactive oxygen species (ROS) was measured using
oxidation-sensitive dyes, the activation of the ARE promoter was estimated using
a reporter plasmid, and the expression of phase II detoxifying enzymes was
quantified by measuring their mRNA levels using RT-qPCR.
RESULTS: A high level of ROS production was detected in cells transfected with
NS1-expressing plasmid. In addition, this protein activated the promoter with an
ARE and upregulated the transcription of ARE-dependent genes that encode phase II
enzymes.
CONCLUSION: TBEV NS1 protein both triggers ROS production and activates a defense
Nrf2/ARE pathway. These data suggest that a role of redox-mediated processes in
TBEV-induced damage of the central nervous system should also be explored. These
data can contribute to a better understanding of TBEV pathogenicity, further
improvement of TBE treatment, and the development of vaccine candidates against
this infection.

© 2016 S. Karger AG, Basel.

DOI: 10.1159/000452160
PMID: 27875810  [PubMed - as supplied by publisher]
```

(a) Sample abstract 1

```
1. Sci Rep. 2016 Nov 22;6:37370. doi: 10.1038/srep37370.

Effects of light-emitting diode irradiation on the osteogenesis of human
umbilical cord mesenchymal stem cells in vitro.

Yang D(1), Yi W(1), Wang E(1), Wang M(1).

Author information:
(1)Department of Orthopaedics, Nanshan Hospital, Guangdong MedicalCollege,
Shenzhen Guangdong, 518052, China.

The aim of this study was to examine the effects of light-emitting diode (LED)
photobiomodulation therapy on the proliferation and differentiation of human
umbilical cord mesenchymal stem cells (hUMSCs) cultured in osteogenic
differentiation medium. HUMSCs were irradiated with an LED light at 620 nm and
2 J/cm(2) and monitored for cell proliferation and osteogenic differentiation
activity. The experiment involved four groups of cells: the control group; the
osteogenic group (osteo group); the LED group; the osteogenic + LED group
(LED + osteo group). HUMSC proliferation was detected by performing
a3-(4,5-dimethylthiazol-2yl)-2,5 diphenyltetrazolium bromide(MTT) assay.
Osteogenic activity was evaluated by performing alkaline phosphatase (ALP) and
Von Kossa staining, and osteopontin (OPN) gene mRNA expression was evaluated
byreverse transcription polymerase chain reaction (RT-PCR). The hUMSCs in the
LED + osteo group exhibited a significantly higher proliferation rate than the
other subgroups. Additionally, there were greater numbers of ALP-positive cells
and Von Kossa nodules in the LED + osteo group. OPN mRNA expression in the
LED + osteo group was higher than other subgroups. In conclusion, low levels of
LED light at a wavelength of 620 nm enhance the proliferation and osteogenic
differentiation of hUMSCs during a long culture period.

DOI: 10.1038/srep37370
PMID: 27874039  [PubMed - in process]
```

(b) Sample abstract 2

Figure 5. Sample PubMed text abstracts

extracted is of the format of term frequencies, it was decided to follow the structure of the famous information retrieval algorithm TF-IDF discussed in II-D4.

Each abstract is considered a separate document, and the term frequency of each term in abstract is calculated. Then the inverse document frequency is calculated across abstracts. These two statistics are combined to calculate a semantic weight for each of the terms. Using the Stanford XML, the lemma of each term is extracted. Next, a triple consisting of the term (word), the lemma of the term, and its semantic weight is created for each term. Finally, the triples for each term (word) are output in to a dictionary file as an intermediate output. Example 4 shows some typical lines from the dictionary file.

Code 4. Dictionary lines example

```
illuminators illuminator 0.045435406
twisting twist 0.0238714
lowering—drugs lowering—drug 0.049106136
mir—362 mir—362 0.03663714
mir—374 mir—374 0.07645514
mir—373 mir—373 0.1492043
mir—372 mir—372 0.13382968
ellas ellas 0.025369484
scavenges scavenge 0.013151284
architectures architecture 0.050796155
```

IV. Creating Final Triples

With the above intermediary outputs ready, we move on to the next step of creating triples. Triples are created on the basis of separate abstracts. Each of the Ollie triple sets for a given abstract is read along side the corresponding Stanford XML. Each triple carries the triple information ($Subject$;$Relationship$;$Object$), confidence value, the relevant original sentence from the text abstract, and the sentence id.

A. Triple building

The first information extraction step is a gazetteer list approach as described in II-D1. In this stage, a gazetteer list of MESH terms is made out of the OMIT ontology by extracting the concept tree rooted at *MESH term* concept and adding all the individuals present in that tree to the gazetteer list. One important thing to note here is the fact that some of the strings in the OMIT ontology are not in the same format that one would use in a text. An example would be *Technology, Pharmaceutical*. Entries such as this were changed to the normalized form; for example *Pharmaceutical Technology*. Next, the subject and the object of the triple are tested for occurrences of an individual now present in the gazetteer list. If any were present, the node list corresponding to the relevant subject or object is updated by appending the returned OMIT concept node to the said list.

Next, REGEX-based information extraction, as described in I, is used. A base REGEX is built on the common usages of mRNA in abstracts and is matched to the counterparts in OMIT as per the descriptions in II-D1. The base REGEX is then expanded to cover all common forms of mentions of mRNA in literature. This is further enhanced by adding other pairings of REGEX and OMIT concepts. All of these REGEXes are then used to find the corresponding OMIT concept nodes for each of the words that exist in the subject or the object of the triple (depending on which one is being examined at the time.) These results, too, are then added to the node list as explained above.

The relationship in the Ollie triple is then analyzed against the corresponding elements in the Stanford XML. In the case of the relationship being a single word, the lemmatized form of the said word is extracted from the Stanford XML, and the relationship is replaced with that lemmatized form. Simplification is not done when the relationship is a phrase. This is for the expectation of future expansions of the system to incorporate sentiment analysis, as described in Section IX.

The above steps are reduction steps, in the sense that out of all the concepts in the English language, only the ones that are directly relevant to the mRNA domain are present in the OMIT ontology. Thus, the subject and/or object of some of the Ollie triples will have empty node lists.

Next a triple is created for every node in the object list for the every node in subject list, using the reduced or pure relationship from the original Ollie triple. (As mentioned above, the relationship is only reduced when it is comprised of a single word.) This is an increment step, given the fact that the resulting number of triples is the multiplication of the number of elements in subject list and the object list of the original Ollie triple. This also means that any OLLIE triple that was reduced to have an empty subject list or an empty object list will produce no triples in this step.

B. Triple simplification

Newly created triples are then sent through two simplification processes. An important point to note here is the fact that these simplifications happen on a sentence-by-sentence basis, here. In this step, triples corresponding to one sentence have no effect on the triples corresponding to a different sentence.

The first simplification step goes through all the given triples and analyses the subject, the object, and the relationship. In the case where all three of them are equal for two given triples, a new merged triple is created with the same subject, object, and relationship along with the average value for the confidence.

The second simplification uses the concept hierarchical information from OMIT. Thus it belongs to the ideas of Ontology-Based Information Extraction discussed in II-D1. Here, the triple list is simplified, on the fact that some triples in the list, are ancestors of other triples in the list as defined in Definition IV.1.

Definition IV.1 (Triple Ancestor)**.** *A triple X is defined as the ancestor of another triple Y if and only if the following two conditions are satisfied: both triples have the same relationship; and the subject node and the object node of X are respectively ancestors of the subject node and object node of Y as defined by Definition IV.2.*

Definition IV.2 (Node Ancestor)**.** *The ancestor relationship for nodes W and Z are defined as follows; a node W is the ancestor of a node Z if and only if, the node W is the same as node Z or the OMIT node of W is an ancestor of OMIT node of Z in the concept hierarchy of the OMIT ontology.*

First the triple list is scanned from left to right to see if any triple would be the ancestor of one that is listed left of it. In the case where an ancestor is found, the ancestor is discarded and the descendant's confidence is set to the average of that of the original confidence value of the descendant and the confidence value of the ancestor. Then the triple list is scanned from right to left to see if any triple would be the ancestor of one that is listed right of it. The same simplification process used in the left to right scan is applied on the ancestors and descendants that are found.

The rationale of this process is the following. In the step in which we created the new triples out of OLLIE triples, we were doing string REGEX matching on the subjects and objects of the Ollie triples and assigning nodes that correspond to a concept in OMIT. There are many cases in OMIT ontology where the name of an ancestor node is a substring of a descendant node. An example is, shown in Fig. 6 where the concept node with the name *"Cells"* has descendants with names such as *"Goblet Cells"* and *"Paneth Cells"*. Thus a sentence that mentioned *"Goblet Cells"* such as *"The goblet cells are found in the intestinal tract"* that is expected to produce the triple *(Goblet Cells ; are found in ; Intestinal Tract)* will also produce the triple *(Cells ; are found in; Intestinal Tract)*. From definition IV.1, it is evident that the latter triple is an ancestor of the former triple. Thus by the simplification process discussed above, the latter triple is removed and the confidence of the former triple is updated using the current confidence values of the former and latter triples. This makes sense because sentences are always relevant to the concept with the smaller granularity as shown in the example.

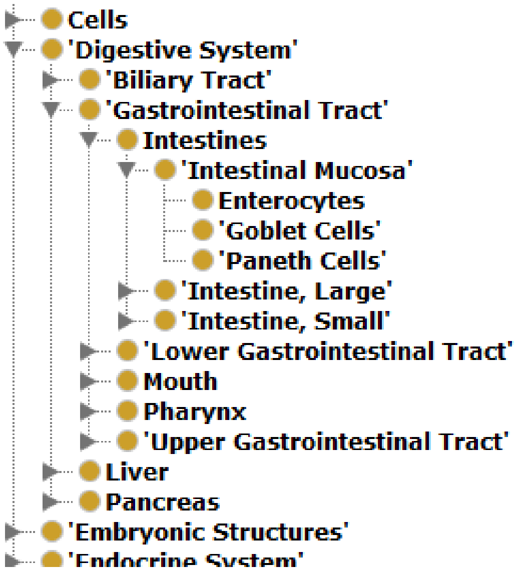


Figure 6. Part of OMIT hierarchy

Once the above process is finished for each sentence, all the resultant triples are added to a single list. Then that list is passed to a simplification process similar to that of the first step but with a slight change. Just like in the per sentence simplification, the process goes through all the given triples and analyses the subject, the object, and the relationship; but this time, it is done over the entire abstract. It should be noted that the second simplification, i.e. ancestor-based simplification is not done here. This is because of the possibility of losing a generalized claim when it exists in an abstract that also makes a specific claim. In the case where all three – subject, object and relationship – are equal for two given triples, a new merged triple is created. But this time, the new triple will carry both sentences (if they are different), and the confidence value is updated to the new value C_{new} according to Equation 3, given: the confidence in triple 1 is given by C_1, the confidence in triple 2 is given by C_2, the sentence count in triple 1 is given by S_1, and the sentence count in triple 2 is given by S_2.

$$C_{new} = \frac{C_1 * S_1 + C_2 * S_2}{S_1 + S_2} \tag{3}$$

The resultant triples of the above process are put in to a list. These are the final triples. The final triples are then written to a set of files as an intermediate output. A separate file is written for each separate abstract. By this point, some abstracts will have empty lists, because none of the Ollie triples of those abstracts have survived the conversion to the final triples form, if the Ollie triples from those abstracts lacked any information relevant to be extracted using the OMIT ontology. These abstracts will have empty files in their name.

V. Finding inconsistencies

From here onwards, we discuss the methodology used to find inconsistencies using the final triples, other resources, and intermediate files created in the previous sections.

A. Preparing to find inconsistencies

First order of business for finding inconsistencies is to load the intermediate files created at IV and III-D for new triples and the dictionary respectively.

Abstracts are read and data are loaded. But instead of storing data with the distinct unit per abstract as we have been doing so far, a new minimum unit is introduced which has a unique entry for each triple. Which means a sentence with multiple candidate triples will be represented in corresponding multiple entries.

All the triple entries are loaded to a list. Each triple entry i is compared with each triple entry j such that i goes from 1 to the length of triple entry list while for each i, j goes from $i+1$ to the length of triple entry list. This way, the triple entries are compared with the triple entries that follow them thus each pair of triple entries only gets compared once.

1) Initial filtering: Before the analysis begin, a couple of filters are applied. First filter makes sure that triple entries of the same abstract are not compared to each other because finding inconsistencies within the same abstract is not the objective of this study. Second filter is applied to handle the case where in some cases, a redacted article is found to have the exact same content as another legitimate article. In this case, one is dropped from the consistency checking. For the purpose of this study, it does not matter which one is dropped for the simple reason that if the legitimate article is dropped and the system end up finding an inconsistency with the redacted article against some third article, it is a simple matter of reconsulting the PubMed database to find the relevant legitimate article. Also, keeping in redacted articles that does not have a matching legitimate article might have potential as a future work as explained in section IX.

2) Cleaning the strings: The relation value of triple entry pairs that pass the filtering process are then put through a cleaning process. Special contractions such as ”can't”, ”won't” are explicitly handled and simple contractions such as ”don't”, ”hadn't” are scripturally handled.

Next the relationship is split to the terms and when there exist a ”not”, it is handled as the negation of the following term. More advanced ways that can be used to do this is discussed in section IX.

Next all the stop words are removed from the list and finally, using the lemmatization results loaded from the dictionary created at section III-D, all words are stemmed to their basic lemma.

B. Calculating oppositeness of relationship strings

The two lists of cleaned strings that were created from the triple relationships are then evaluated against each other word by word. We define the item count of these lists as c_1 and c_2. Before going in to the oppositeness function, some simple comparisons are made to lighten the computing load.

When both the comparing words are exactly the same, the weight of the word is extracted from the dictionary that was created at section III-D and were loaded at the beginning of section V-A. This is raised to the power of two and then multiplied by the constant ”yes weight” (W_{yes}). The resultant value is added to the similarity amount ($simil_T$), the similarity number counter (s_n) is increased by one.

When either of the words is the direct simple negation of the other by the key word ”not”, (i.e.: ”increased”-”not increased”, ”found”-”not found”), again the weight of the non negated word is extracted from the dictionary and raised to the power of two. The resultant value is then multiplied by the constant ”no weight” (W_{no}). This value is added to the difference amount ($diff_T$), the difference number counter (d_n) is increased by one.

1) Oppositeness Function for words: Word pairs that are not handled by either of above situations need specialized work. First, the word pair is checked for similarity by the Wu and Palmer [24] semantic similarity measure (sim) discussed in section II-D3. We show this in equation 4.

$$simil = \frac{sim(w_1, w_2)}{c_1 + c_2} \tag{4}$$

Checking for oppositeness is not as straight forward. Given that the word similarity is between 0 to 1 as mentioned in the section II-D3, it is possible to naïvely assume that just taking the complement of whatever the similarity value would be enough for finding the oppositeness. This, sadly, is not the case. What this means is, semantic difference, is not the same as semantic oppositeness.

We demonstrate this with the following example; assume we have the word *increase* in one hand and the words *expand*, *decrease*, *change*, and *cat* on the other hand to be checked against *increase* to see which one of the said words are the most contradictory in nature to the word *increase*. A human would see these words and see that the word *cat* is irrelevant here. It is neither slimier nor different to $increase$. In fact the meaning is orthogonal to the meaning of $increase$. Next, the human might point out that the word $expand$ is semantically similar to the word $increase$. Both of the words are discussing adding to an amount that already exists. The word $decrease$, the human might say, is the real opposite of the word $increase$. Finally, $change$ should sit somewhere between $increase$ and $decrease$ because it can go either way. However, $change$ is not completely irrelevant to the meaning of $increase$ like cat is. Thus it is possible to use this as the golden standard to order these words in a way that each of these (or at least the opposite words) are easily identifiable.

Now, if one decides to use the naïve approach and take the inverse of calculated the similarities, one would get the result shown in Table III.

Table III
NAÏVE METHOD TO FIND OPPOSITENESS

	$expand$	$decrease$	$change$	cat
Similarity to $increase$	0.80	0.75	0.46	0.25
$1-$**Similarity**	0.20	0.25	0.54	0.75

Now, if the words are sorted in the increasing difference, the word order is $expand$, $decrease$, $change$, and cat. This is not the desired outcome. If this method is used and a threshold is introduce to determine $decrease$ as an opposite of $increase$, automatically $change$ and cat also become opposites of $increase$. Given this issue, instead of the naive approach, we introduce the following method.

First, for each of the pair of words, the lemma is extracted using the dictionary created at section III-D. Let us call them L_1 and L_2. When the word does not exist in the dictionary, the word itself is used as its own lemma. For each lemma, all the synsets relevant for each of the word senses are extracted. Given that a word might have many senses, this is a *one to many* mapping.

For each synset, the list of antonym synsets are collected using WordNet's antonym feature. Given that a word sense can have many antonyms in various contexts, this is yet again a *one to many* mapping. All the retrieved antonym synsets for one original lemma are put into a single list.

Each of the words in each of the synsets in the said list are then taken out to make a word list. Yet again this is a textitone to many mapping given that each synset has one or many words in them.

The resultant word list is then run trough a duplicate remover. This is the first reduction step in the antonym process so far. We name antonym list of L_1 as a_1 and the antonym list of L_2 as a_2. Number of items in a_1 is n while the number of items in a_2 is m.

Next, each antonym of L_1 is checked for similarity against the original L_2 and the maximum difference is extracted as $diff_1$ as shown in equation 5. Similarly each antonym of L_2 is checked for similarity against the original L_1 and the maximum difference is extracted as $diff_2$ as shown in equation 6.

$$diff_1 = max(sim(L_2, a_1(1)), sim(L_2, a_1(2)), ..., sim(L_2, a_1(n))) \tag{5}$$

$$diff_2 = max(sim(L_1, a_2(1)), sim(L_1, a_2(2)), ..., sim(L_1, a_2(m))) \tag{6}$$

Once $diff_1$ and $diff_2$ are calculated, the overall difference, $diff$ is calculated using equation 7.

$$diff = \frac{\frac{diff_1}{c_1} + \frac{diff_2}{c_1}}{2} \tag{7}$$

Table IV shows the results of the $diff$ values for the same example as table III.

Table IV
OPPOSITENESS WITH ONLY $diff$

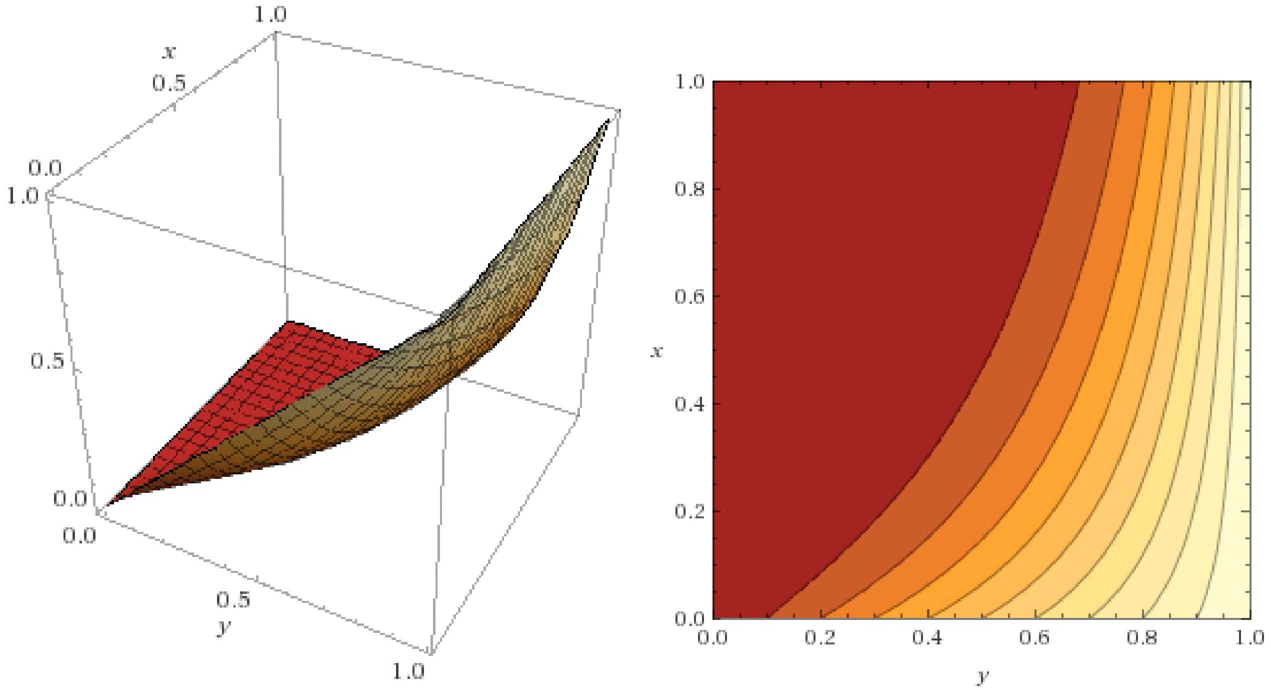
	$expand$	$decrease$	$change$	cat
$diff$ **to** $increase$	0.63	1.0	0.72	0.25

If the words are sorted using $diff$ in the increasing order, they would be cat, $expand$, $change$, $decrease$. We have gotten the expected order where first we have irrelevant word, then the most similar word, next the neutral word and finally the opposite word. However still, the spread of words is not optimum. This can be seen from the gap between each pair of words in the above sorted order. It is; 0.38, 0.09, 0.28 in order. What is needed is a way to magnify the difference value of the opposite word while shrinking the other differences so that the threshold line can be comfortably drawn.

With both the $diff$ and $simil$ values at hand, it is possible to calculate the oppositeness fulfilling the above condition. Before moving on to the equation, it is prudent to look at the example on Table III, once more. The words there are being compared to the word $increase$. As per the above discussion on the golden standard for this, the similarity measure correctly shows that $expand$ and $decrease$ are in the shared context of $increase$. Semantically, this implies that entities that can $increase$ can also $expand$ or $decrease$. They can also $change$, hence the value for $change$ comes next. But it is not as close as the previous two because the word $change$ can apply in a context that is very different from a context that is valid for $increase$. Finally there is the value for cat which is an irrelevant concept. What is observed from this is the fact that, more semantically similar the two words are, the difference value has to be magnified proportional to that closeness. When the two words becomes less similar, the difference value has to be penalized. Thus equation 8 is introduced to calculate oppositeness. Figure 7 shows the plot for the equation. $simil$ is the x variable and $diff$ is the y variable.

$$oppo = diff_T^{(0.5*\frac{W_{no}}{W_{yes}}*simil_T+1)} \tag{8}$$

As evident by Fig 7(a) and Fig 7(b), in higher word similarities ($simil_T$), the difference ($diff_T$) also have to be very



(a) 3D plot

(b) Contour plot

Figure 7. Oppositeness function; (x variable:$simil$, y variable:$diff$)

high for the final *oppo* value to be high. in lower $simil_T$ range, *oppo* becomes closer and closer to being directly proportional to $diff_T$ and achieves it when $simil_T$ becomes zero. This, in this example, effectively pushes $decrease$ farther away from $increase$ than others. Values after this transformation is shown in table V.

Table V
OPPOSITENESS WITH *oppo*

	$expand$	$decrease$	$change$	cat
oppo **to** $increase$	0.05	0.2	0.098	0.022
max scaled to 1	0.25	1	0.49	0.11

Again the word order in increasing oppositeness is; cat, $expand$, $change$, $decrease$. Scaled gap between the words are 0.14, 0.24, 0.51. Now the actual opposite word is placed clearly apart from the rest of the words. The difference between the near synonym $expand$ and neutral word $change$ is more prominent (distance 0.25 and 0.49 from $increase$ compared to 0.63 and 0.72 in previous case). The irrelevant word cat is pushed more downwards.

The final *oppo* value is multiplied by -1 and is returned up as the oppositeness measure of the two words. The returned value is then multiplied by the weights of the two words extracted from the dictionary. If the value is greater than zero, the value is multiplied by the constant ”yes weight” (W_{yes}). The resultant value is added to the similarity amount ($simil_T$), the similarity number counter (s_n) is increased by one.

If it is less than zero, value is then multiplied by the constant ”no weight” (W_{no}) and -1. This value is added to the difference amount ($diff_T$), the difference number counter (d_n) is increased by one. Thus when the value is zero no change happens to any similarity/difference values or counters.

2) Finalizing the oppositeness of relationship strings: Once all the words in the two relationship strings have finished going through the above steps, both $simil_T$ and $diff_T$ are normalized using a small constant ϵ with s_n and d_n as shown in equations 9 and 10.

$$simil_T = \frac{simil_T * (d_n + \epsilon) * W_{yes}}{s_n + d_n + 2 * \epsilon} \tag{9}$$

$$diff_T = \frac{diff_T * (s_n + \epsilon) * W_{no}}{s_n + d_n + 2 * \epsilon} \tag{10}$$

finally, if $simil_T$ is greater than $diff_T$, $simil_T$ is returned as the similarity value of the two relationship strings. Otherwise $diff_T$ multiplied by -1 is returned as the difference value of the two relationship strings.

C. Registering contradictions

The returned value by the above step for a given pair of relationship strings is then multiplied by -1 and put through a threshold test. It it passes the threshold, it is registered as a contradiction that was found.

For each abstract that gets involved in a potential contradiction, PubMed was queried again to obtain the publication date. The reason for doing this at this stage is the fact that only a small portion of all abstracts are relevant for this stage and thus we can do a lesser amount of processing and data storage for the bearable cost trade off of few instances of XML fetching over the Internet.

Each of the contradictions that were found are written to an intermediate result file where a line holds; confidence (the difference value returned), PubMedIds of the contradicting abstracts along with the publication dates, subject and object of the relevant triple, relationship present in the triple in first abstract, relevant sentence id from the first abstract, relationship present in the triple in second abstract, relevant sentence id from the second abstract. An example of some lines from the said intermediate result file is shown at example 5.

Code 5. Intermediate contradiction result example

```
0.8333333;24969691;2014/9/1;27601936;2016/9/7;Cells;Vimentin;increase;3;decrease;7
0.8333333;25435961;2015/1/1;26632856;2015/12/1;DNA;Cells;promote;7;breaks in;12
0.625;25004396;2014/6/15;26257392;2015/11/1;MIR152;Cells;were decreased in;3;be
        Interestingly increased in;10
```

D. Preparing contradiction for analysis

This is the final stage of the methodology. First, the intermediate result file written the previous step is read. Then the Subject and Object of the contradicting triples are checked against OMIT to see if either or both of them are of the type mRNA. The reason we pushed this check to this final step is for the fact that, this way, the intermediate file created before this step can potentially be used for other researches on contradictions in the medical abstracts in domains other than mRNA as well.

If either or both the subject and the object are indeed of the type of mRNA, then for each such contradiction, the relevant Ollie files are read and the contributing actual sentences are extracted using the sentence IDs. Then the information gained from the intermediate result file and the extracted sentences are reformatted to be more readable by humans. Here, finally the original OLLIE confidences are used. The final confidence Con_{fin} is calculated using the contradiction confidence Con_{cont} calculated above, OLLIE confidence of triple 1 Con_1, OLLIE confidence of triple 2 Con_2, and the constant C as shown in Equation 11. C is selected $C > 1$.

$$Con_{fin} = C * Con_{cont} * Con_1 * Con_2 \tag{11}$$

The reformatted contradictions are then written to the final result file to be read and analyzed by human experts. An example of some lines from the said result file is shown in example 6.

VI. Configuration

All steps of the methodology was implemented in Java (jdk1.8) and were run on a computer with Windows 10 Home 64-bit, Intel(R) Core(TM) i7-6700HQ CPU @ 2.60 GHz, 16GB (15.9GB Usable) RAM.

VII. Results

The PubMedID extraction step extracted 39149 relevant abstract ids. From which 36877 were processed and downloaded as text files containing abstracts. Thus 5.8% of relevant PubMed entries did not have an abstract section. One example of when this happens is when it is an entry with some graphs instead of an entire research paper. Some good examples for this are PubMedIds 24324220, 24318653, 24311611, 24303553. Another instance is when there is only a comment about the entry instead of the full entry. PubMedIds 24311611, 24303553 are good examples for this. Finally there are some entries that are empty for everything but the entry name, author names and other meta data. One such example is PubMedID 24313780. However, the point is that each and every abstract from the remaining 94.2% of relevant IDs were downloaded for analysis.

Code 6. Final result file example

```
0.056045435

25738546
2015/5/1
( MIR214 ; was significantly increased in ; Tissues )
4
Our results revealed that miR−214 expression was significantly increased in the BC
        tissues compared with the adjacent benign tissues, and that the upregulation
        of miR−214 was significantly associated with the invasion ability of the BC
        cells.

27109339
2016/6/1
( MIR214 ; were significantly decreased in ; Tissues )
4
Our results revealed that the expression of miR−214 and miR−218 were significantly
        decreased in breast cancer tissues compared with adjacent tissues.
```

All 36877 of the downloaded abstracts were processed to yield Ollie triple files and Stanford XML files. All of these intermediate files were used to create the intermediate result file which detected 67481 unique subject object pairings. Out of which 503 total contradictions were found which involved 224 out of the 36877 downloaded abstracts. This means, it was observed that the percentage of abstracts that contribute to inconsistencies is just 0.61%.

After the reduction steps mentioned in the section V-D was done to keep only the contradictions that involve at least one mRNA entry, we ended up with 102 contradictions involving 95 abstracts. This means, out of the 503 total contradictions only 20.28% were relevant to mRNA. As for abstracts participating in contradictions involving mRNA, it was 0.26% of the total downloaded abstracts and 42.41% of the abstract that were found to have involved in inconsistencies of any kind.

VIII. Conclusion

As statued in the section I, the primary research contribution of this study was to use ontology-based information extraction to observe how contradictions rise in the literature in relation to previously established knowledge in a scientific filed in the form of inconsistencies. This study successfully proposed a method to do that observation and succeeded in finding 503 such contradictions in a corpus of 39149 research paper abstracts. Since these contradictions are rooted in very domain specific medical jargon, they need to be analyzed by medical experts.

Secondly, this study had to face the problem of the ontology that was being used not having the relationship rules that most of the established OBIE systems use. Thus, this study came up with a novel way to solve this problem by involving open information extraction systems to extract the relationships and then using the conventional OBIE systems to do the information extraction. This methodology can be considered as a new way of doing OBIE in addition to the traditional and established methods discussed in section II-D1.

Third contribution is in fact all the relationships that were deemed non-contradicting by the system. While in a usual implementation, these outputs that are not relevant to the primary research objective can be considered not useful, the special circumstances of this study have opened up an avenue for these outputs to be useful. That situation is the fact that OMIT ontology lacking relationships between its concepts as described above. These relationships that were deemed non-contradicting are in fact, candidates to be added to the OMIT ontology. More reinforced a given extracted relationship is, more confidently can we suggest it to be added to the OMIT ontology in the future. Yet again, since these relationships are rooted in very domain specific medical jargon, they need to be analyzed by medical experts.

Fourth contribution is a rather curious one and can only be had once the experts are done with selecting the relevant contradictions form the list of 503 proposed contradictions. The idea is that sometimes the system would return as a contradiction when the same substance act differently on other substances when the background conditions are different. Because of the difference of actions, they have been registered as contradictions. But once the experts claim that both statements can be true under different circumstances, this information too can be added to OMIT as special relationships.

Fifthly, there are the intermediate files generated during the process. This comprise of 39149 medical abstracts in free text; same abstracts PoS tagged, parsed, lemmatized and put in the the Stanford XML format; yet again the same abstracts run through open information extraction and created OLLIE triples. All of these can help make future research on this particular domain faster because future researches do not need to spend computing power or resources to generate these outputs again. Similarly there is the medical jargon lexicon generated in III-D along with the relevant lemmas and the TF-IDF frequency statistic based on the PubMed abstract corpus.

Further, in the natural language processing domain, the oppositeness measure finding method introduced in the section V-B can be quite useful especially for sentiment analysis.

As a whole, there are other usages in other domains that the findings in this study can be used. One such area is analyzing on-line argument/discussion threads. By considering each new post as we have considered a separate abstract in this study, it would be possible to see how the argument progress and how, when and who bring up counter arguments for any of the statements that have been made before. Another domain this methodology can be used is, for fact checking. Once the domain documents and the domain ontology have been used to set up the system, new documents with claims about the domain can be sent through the system to see if any inconsistencies arise.

IX. Future Work

For future work, one most basic thing that can be improved is in the preprocessing stage. It is customary for research papers to use acronyms in their abstracts either after defining them in the first occurrence or not. The latter happens when the said acronym is a commonplace usage in the medical domain. However, this creates slight problems to the current system as the current system sees the acronym and the expanded form as separate entities and also OMIT would only carry the expanded form of the acronym. So the preprocessing step is to replace all the acronyms with the expanded forms before other steps are started. This would need a dictionary of medical acronyms to use when the abstract itself does not define the acronyms in the beginning of itself.

As mentioned in section V-A1, some of the articles in the list are redacted. Some of these are redacted due to duplicate submission and they are already handled in the system discussed in this paper. However there can be articles that were redacted for different reasons. Keeping in these redacted articles that does not have a matching legitimate article might have potential as a future work as to see if the present inconsistencies were the reason for redaction.

In the section V-A2 when cleaning the strings, the system currently uses some simple linguistic rules. Instead, in a future work, it is possible to use the parse tree in the already created Stanford XML to find better rules to clean the strings for the next step.

The said parse tree in the created Stanford XML can also be potentially used when creating the final triples in section IV.

In the section IV-A, when the gazetteer list was created, only the straight forward cases of non conventional text in OMIT were handled. This can be expanded to complex entires, for example *"Musculoskeletal and Neural Physiological Phenomena"*, *"Nevus, Epithelioid and Spindle Cell"*.

One very advanced future work that can be done is argumentation of the relationship matching process with models built for sentiment analysis. That way, it will be possible to better spot relationships that are stating contradicting statements to one another. However, it should be noted that this has to be done before the stop words are removed in section V-A2 and before the word lemmatization is done.

References

[1] B. Keith. *Strategic Sourcing in the New Economy*. Google Books: Palgrave McMillian, 2016.

[2] Seth Kulick, Ann Bies, Mark Liberman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein, Lyle Ungar, Scott Winters, and Pete White. Integrated annotation for biomedical information extraction. In *Proc. of the Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, pages 61–68, 2004.

[3] Jerry R Hobbs, Douglas Appelt, John Bear, David Israel, Megumi Kameyama, and Mabry Tyson. Fastus: A system for extracting information from text. In *Proceedings of the workshop on Human Language Technology*, pages 133–137. Association for Computational Linguistics, 1993.

[4] M. Aurousseau. On lists of words and lists of names. *The Geographical Journal*, 105(1/2):61–67, 1945.

[5] microrna. Online: `https://en.wikipedia.org/wiki/MicroRNA`. Accessed: 2016-12-08.

[6] Pubmed. Online: `https://en.wikipedia.org/wiki/PubMed`. Accessed: 2016-11-22.

[7] U.S. National Library of Medicine. Medical subject headings. Online: `https://www.nlm.nih.gov/mesh/`. Accessed: 2016-11-25.

[8] Medical subject headings (mesh). Online: `https://en.wikipedia.org/wiki/Medical_Subject_Headings_(MeSH)`. Accessed: 2016-11-25.

[9] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993.

[10] F. Arvidsson and A. Flycht-Eriksson. Ontologies i. Online: `http://tomgruber.org/writing/ontolingua-kaj-1993.pdf`. Accessed: 2016-11-27.

[11] Jingshan Huang, Fernando Gutierrez, Harrison J Strachan, Dejing Dou, Weili Huang, Barry Smith, Judith A Blake, Karen Eilbeck, Darren A Natale, Yu Lin, et al. Omnisearch: a semantic search system based on the ontology for microrna target (omit) for microrna-target gene interaction data. *Journal of biomedical semantics*, 7(1):1, 2016.

[12] Jingshan Huang, Jiangbo Dang, Glen M. Borchert, Karen Eilbeck, He Zhang, Min Xiong, Weijian Jiang, Hao Wu, Judith A. Blake, Darren A. Natale, and Ming Tan. Omit: Dynamic, semi-automated ontology development for the microrna domain. *PLOS ONE*, 9(7):1–16, 07 2014.

[13] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, 25(1):25–29, May 2000.

[14] Karen Eilbeck, Suzanna E. Lewis, Christopher J. Mungall, Mark Yandell, Lincoln Stein, Richard Durbin, and Michael Ashburner. The sequence ontology: a tool for the unification of genome annotations. *Genome Biology*, 6(5):R44, 2005.

[15] Darren A. Natale, Cecilia N. Arighi, Winona C. Barker, Judith A. Blake, Carol J. Bult, Michael Caudy, Harold J. Drabkin, Peter D'Eustachio, Alexei V. Evsikov, Hongzhan Huang, Jules Nchoutmboube, Natalia V. Roberts, Barry Smith, Jian Zhang, and Cathy H. Wu. The protein ontology: a structured representation of protein forms and complexes. *Nucleic Acids Research*, 39(suppl 1):D539–D545, 2011.

[16] Jingshan Huang, Karen Eilbeck, Barry Smith, Judith A Blake, Dejing Dou, Weili Huang, Darren A Natale, Alan Ruttenberg, Jun Huan, Michael T Zimmermann, et al. The non-coding rna ontology (ncro): a comprehensive resource for the unification of non-coding rna biology. *Journal of biomedical semantics*, 7(1):1, 2016.

[17] Jingshan Huang, Karen Eilbeck, Barry Smith, Judith A. Blake, Dejing Dou, Weili Huang, Darren A. Natale, Alan Ruttenberg, Jun Huan, Michael T. Zimmermann, Guoqian Jiang, Yu Lin, Bin Wu, Harrison J. Strachan, Nisansa De Silva, Mohan Vamsi Kasukurthi, Vikash Kumar Jha, Yongqun He, Shaojie Zhang, Xiaowei Wang, Zixing Liu, Glen M. Borchert, and Ming Tan. The development of non-coding rna ontology. *Int. J. Data Min. Bioinformatics*, 15(3):214–232, January 2016.

[18] Princeton university, wordnet – a lexical database for english. Online: `http://wordnet.princeton.edu/`. Accessed: 2016-11-25.

[19] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[20] N. H. N. D. de Silva, A. S. Perera, and M. K. D. T. Maldeniya. Semi-supervised algorithm for concept ontology based word set expansion. In *2013 International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 125–131, Dec 2013.

[21] Daya C. Wimalasuriya and Dejing Dou. Ontology-based information extraction: An introduction and a survey of current approaches. *J. Inf. Sci.*, 36(3):306–323, June 2010.

[22] Les misérables (musical). Online: `https://en.wikipedia.org/wiki/Les_Miserables_(musical)`. Accessed: 2016-11-25.

[23] Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 523–534, 2012.

[24] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[25] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc of 10th International Conference on Research in Computational Linguistics, ROCLING'97*, 1997.

[26] H. Shima. Wordnet similarity for java (ws4j). Online: `https://code.google.com/p/ws4j/`. Accessed: 2016-11-23.

[27] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[28] Fernando Gutierrez. *A Hybrid Approach for Ontology-Based Information Extraction*. PhD thesis, Eugene OR, 2015.

[29] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.