Discovering Inconsistencies in PubMed Abstracts through Ontology-Based Information Extraction

Nisansa de Silva
Department of Computer and Information Science
University of Oregon
Eugene, Oregon 97403, USA
nisansa@cs.uoregon.edu

Dejing Dou
Department of Computer and Information Science
University of Oregon
Eugene, Oregon 97403, USA
dou@cs.uoregon.edu

Jingshan Huang
School of Computing
University of South Alabama
Alabama 36688, USA
huang@southalabama.edu

ABSTRACT

Searching for a cure for cancer is one of the most vital pursuits in modern medicine. In that aspect microRNA research plays a key role. Keeping track of the shifts and changes in established knowledge in the microRNA domain is very important. In this paper, we introduce an *Ontology-Based Information Extraction* method to detect occurrences of inconsistencies in microRNA research paper abstracts. We propose a method to first use the Ontology for MIcroRNA Targets (*OMIT*) to extract triples from the abstracts. Then we introduce a new algorithm to calculate the *oppositeness* of these candidate relationships. Finally we present the discovered inconsistencies in an easy to read manner to be used by medical professionals. To our best knowledge, this study is the first ontology-based information extraction model introduced to find shifts in the established knowledge in the medical domain using research paper abstracts. We downloaded 36877 abstracts from the PubMed database. From those, we found 102 inconsistencies relevant to the microRNA domain.

CCS CONCEPTS

•Computing methodologies → Information extraction; Ontology engineering; *Language resources;* **•Applied computing → Bioinformatics;**

KEYWORDS

Ontology; Semantic Oppositeness; Information Extraction; miRNA; PubMed

1 INTRODUCTION

Second only to cardiovascular diseases in the rates of mortality caused by noncommunicable diseases, cancers claim 8.2 million lives worldwide each year [27]. Thus, research that could contribute to preventing or curing cancer is imperative. As the growth of cancer involves abnormal cell division, it is important to look at the agents that get involved in that process. MicroRNA (miRNA) is a small non-coding RNA molecule that plays a complementary role to mRNAs (messenger RNAs) in the gene regulation step of cell division [6]. It is possible to observe the presence of miRNA in plants, animals, and some viruses. Mainly they are involved in RNA silencing and post-transcriptional regulation of gene expression. This vital role played by miRNAs in gene expression is what makes them relevant and interesting for the pursuit of a cure for cancer.

In light of the potential importance of miRNA, an increasing quantity of research is being engaged upon its domain, albeit that not all studies are confirming studies. As such, some of the new studies about miRNA might either alter, or even completely disprove, some of the prior knowledge. Recognizing how this knowledge alters over the course of time is important for various analytical tasks. It is also vital to note these changes so that forthcoming studies would not mistakenly base their assumptions and start conditions on conclusions in a prior body of work that has since been disproved. It should be noted that, 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. This is due to the fact that, given that the aforementioned conclusions on which these new researches were based were found to be not valid anymore.

The objective of this study is to find such alterations in knowledge in the miRNA domain. For this, we need to have a source from which we obtain details of research about miRNA. The best source to get details about scientific research is through research papers which is the common method used in all sciences to publish new findings. In a research paper, the abstract is a fair summarization of the area of importance and the conclusions driven by the research which is being described in the said paper. Given that miRNA falls in the medical domain, an ideal source for searchable medical abstracts is PubMed [8], a free search engine, accessing primarily the MEDLINE database of references and abstracts on life sciences and biomedical topics. It is maintained by the United States National Library of Medicine (NLM) at the National Institutes of Health (NIH) as part of the Entrez system of information retrieval. As of 5 January 2017, PubMed has over 26.8 million records going back to 1966. of those, 13.1 million of PubMed's records are listed with their abstracts.

In addition to keeping records of research papers, PubMed also provides free access to a Medical Subject Headings (MeSH) [25] database. MeSH is a comprehensive controlled vocabulary for the purpose of indexing journal articles and books in the life sciences. Thus it facilitates searching for a particular subject within the medical domain. MeSH is created and updated by the United States National Library of Medicine (NLM).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ACM-BCB'17, August 20-23, 2017, Boston, MA, USA.
© 2017 ACM. 978-1-4503-4722-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3107411.3107452

Based on the above observations and resources, we propose an ontology-based information extraction model to discover inconsistencies in PubMed abstracts. These inconsistencies are found when the knowledge extracted from one abstract disagrees with the knowledge extracted from another abstract. Thus it is an indication of the aforementioned shifts and improvements in the study of miRNAs. Ontology-Based Information Extraction (OBIE) is a subfield of information extraction where an ontology is used to guide the information extraction process [29]. Given that this study is focusing on the miRNA domain, we used the Ontology for MIcroRNA Targets (OMIT) [14] as the guiding ontology for the OBIE process. Because OMIT lacks relationship data, such that traditional OBIE methods were not applicable, we used the Open Language Learning for Information Extraction (OLLIE) [21]. OLLIE is unique in utilizing tree-like representations of the dependencies of the sentence, such that it is able to capture long-range relations. Once relationship information is extracted from the abstracts in the form of triples, we introduce a novel method to calculate the oppositeness between the said relationships on the basis of the semantic similarity measure of Wu and Palmer [30].

The key idea of our methodology is that the information in the PubMed abstracts in the miRNA domain are expressed in terms of concepts and relationships that exist between those concepts. An inconsistency would arise if the relationship that was extracted between given two concepts in a certain abstract is opposite to the relationship that was extracted between the same two concepts in a different abstract. As mentioned above, we use OBIE methods utilizing the OMIT ontology to extract the said concepts from the abstracts. In order to discover the relationships between the extracted concepts, we use OLLIE information extraction system. Our main contributions of are as follows:

- We introduce an ontology-based information extraction model to discover inconsistencies in PubMed abstracts.
- We propose a new methodology to incorporate open information extraction into ontology-based information extraction process in order to compensate for the lack of relationships in the domain ontology.
- We propose a semantic oppositeness measure to be used to calculate the oppositeness between two relationships. We illustrate how this novel semantic oppositeness measure is superior to the antonym method and to the naive similarity inverse method.

The rest of the paper is organized as follows. In Section 2, we first introduce the related works and background in information extraction, Ontologies and semantic similarity. We then introduce the methodology to prepare the data in Section 3. We present our method of creating final triples in Section 4 and the methodology to discover inconsistencies in Section 5. Results and discussion follows that in Section 6. The work is concluded in Section 7.

2 RELATED WORKS AND BACKGROUND

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. [17] that extracted information on drug development and cancer genomics.

Information extraction is a process in artificial intelligence (AI) domain 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 in literature: (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 [12].

2.1 Ontologies and OBIE

An ontology is defined as "formal, explicit specification of a shared conceptualisation" [10] in information science. Ontologies are used to organize information in many areas as a form of knowledge representation. These areas include: artificial intelligence, linguistics [3, 4, 28], biomedical informatics [14], law [15], 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 [4].

2.1.1 Ontology for MIcroRNA Targets (OMIT). The Ontology for MIcroRNA Targets (OMIT) [14] was created with the purpose of establishing data exchange standards and common data elements in the microRNA (miRNA) 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 miRNAs (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 [1], Sequence Ontology [5], PRotein Ontology [24], and Non-Coding RNA Ontology (NCRO) [13].

2.1.2 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 2.1, 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 2, 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 [29]: "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."

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 contains little to no relationships between the said entities. Thus some of the most powerful conventional OBIE methods cannot be used alongside OMIT. Later sections discuss how this study overcame this challenge.

2.2 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 Information Extraction (IE) techniques discussed in the beginning of Section 2. Thus, following advanced IE methodologies are used.

2.2.1 Open Information Extraction. The requirement of having pre-specified relations of interest is the main drawback 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 [21], is different from others in its genre due to the fact 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 2.1.1, and the conventional OBIE methods described in 2.1.2. More information on this, is discussed in Section 7.

2.2.2 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 [22], Wu and Palmer proposed a method to give the similarity between two words in the 0 to 1 range [30]. The approach proposed by Jiang and Conrath measures the semantic similarity between word pairs using corpus statistics and lexical taxonomy [16]. By means of [26], the strengths of these algorithms were evaluated in [3]. According to that, we selected Wu and Palmer's implementation for the purposes of this paper.

A set of examples of word similarities are shown in Table 1. 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 1: 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 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 5.2 and 4, we use dissimilar weights named "yes weight" (W_{yes}) and "no weight" (W_{no}), where W_{no} is larger than W_{yes} by a considerable amount.

2.3 Inconsistency Detection

Inconsistency finding in text is mostly a field researched in the NLP for education domain. This has brought to light a number methods. The first among them is based on the identification of coincident words and n-grams [19]. While this method is adequate for automatic text grading which is based on evaluating characteristics such as the fluency of the text, it is not suitable for the application in this study due to each of the abstracts being independent documents and not descriptions of or summarizations of a source document. The second method is the popular NLP technique Latent Semantic Analysis (LSA) [7, 9]. Here also, the vector representations of the students' documents are matched against that of a gold standard (i.e. correct text). This approach would have been very difficult to scale for this study where all abstracts are compared against each other. The third method is based on Information Extraction (IE) [2, 11, 23] it intends to capture the underlying semantics of the text. Given that the objective of this study matches well with that intention, we move in that direction. Out of the IE studies, the closest one to this study is the one proposed in [11].

But in many ways, our methodology is significantly different than that of [11]. That difference exists despite incremented inconsistency finding being common to the two approaches. The main difference is the fact that in [11], 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.

3 DATA PREPARATION

3.1 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 "miRNA". The PubMedIDs were then processed to remove duplicates, and they were 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 often than not, the formatting of the free text was done in different ways, as shown in Fig. 1(a) and Fig. 1(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 of OBIE [29].

3.2 Creating OLLIE triples

The downloaded free text is then subjected to the open information extraction system introduced in [21], that was described in 2.2.1 by the name OLLIE. This process extracts triples in the form of binary relations from the free text and creates a set of possible triples 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 OLLIE 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, that were explained in the leading paragraphs of Section 2, 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.

3.3 Creating Stanford XML files

The same free text obtained in Section 3.1 are sent through a system to extract other linguistic information. In this case we are using the methodology developed by Manning, et. al. [20]. The objective of this step is to extract the parse tree, get the lemmatized forms of each word, and get each sentence element separated. From this point onwards, this paper will refer to these outputs for each abstract as "Stanford XML".

3.4 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. A very important part in an ontology-based information extraction system is the semantic lexicon [29]. WordNet is the primary lexicon in this system. But 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 extracted is of the format of term frequencies, it was decided to follow the structure of the famous information retrieval algorithm TF-IDF [18].

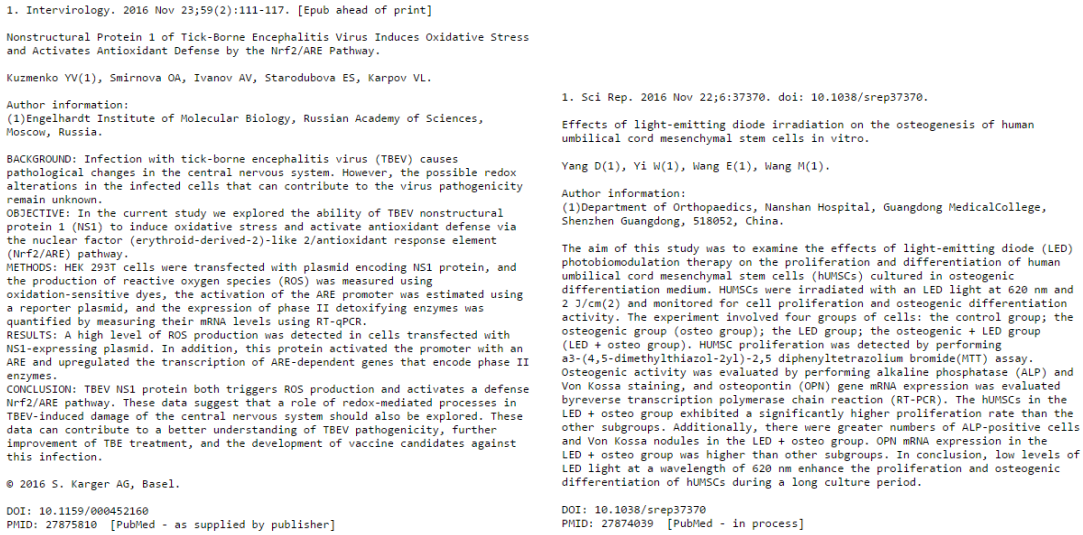
Each abstract is considered a separate document, and the term frequency of each term in abstract is calculated. Then the inverse document frequency is calculated across abstracts. These two statistics are combined to calculate a semantic weight for each of the terms. Using the Stanford XML, the lemma 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.

4 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.

4.1 Triple building

The first information extraction step is a gazetteer list approach as described in [29]. 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



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 1: Sample PubMed text abstracts

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 end of the said list.

Next, Regular Expression (REGEX) based information extraction, is used. A base REGEX is built on the common usages of miRNA in abstracts and is matched to the counterparts in OMIT as per the descriptions in [29]. The base REGEX is then expanded to cover all common forms of mentions of miRNA 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.

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 miRNA 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 each is created using every node in the object list for every node in subject list, utilizing 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. Thus, 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.

4.2 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 2.1.2. 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 4.1.

Definition 4.1 (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 4.2.

Definition 4.2 (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. 2 where the concept node with the name *"Cells"* has descendants with names such as *"Goblet Cells"* and *"Dendritic 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 4.1, 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 above example.

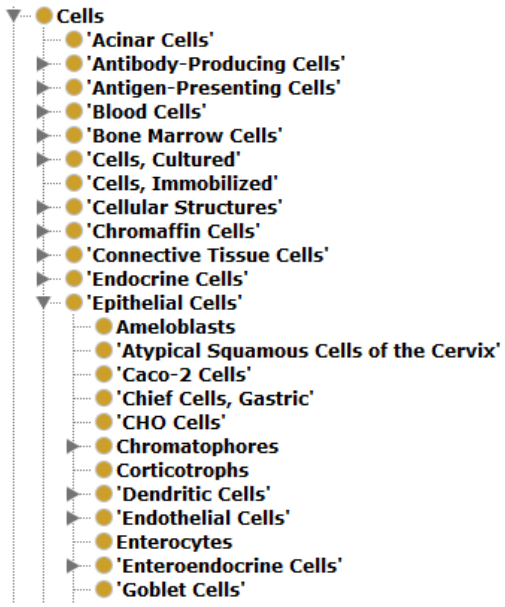


Figure 2: Part of OMIT hierarchy

Once the simplification 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 1, where: the confidence in triple 1 is given by C_1 (Such that $0 < C_1 < 1$), the confidence in triple 2 is given by C_2 (Such that $0 < C_2 < 1$), the sentence count in triple 1 is given by S_1, and the sentence count in triple 2 is given by S_2. Sentence count is never zero.

$$C_{new} = \frac{C_1 * S_1 + C_2 * S_2}{S_1 + S_2} \tag{1}$$

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.

5 DISCOVERING 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.

5.1 Preparing to find inconsistencies

First order of business for finding inconsistencies is to load the intermediate files created at 4 and 3.4 for new triples and the dictionary respectively. Abstracts are read and data are loaded next. 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.

5.1.1 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 by way of the redacted article.

5.1.2 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. Following that, all the stop words are removed from the list and finally, using the lemmatization results loaded from the dictionary created at section 3.4, all words are stemmed to their basic lemma.

5.2 Calculating oppositeness of relationships

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 3.4 and were loaded at the beginning of section 5.1. 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 (dif_T), the difference number counter (d_n) is increased by one.

5.2.1 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 [30] semantic similarity measure (*sim*) discussed in section 2.2.2. We show this in equation 2.

$$simil = \frac{sim(w_1, w_2)}{c_1 + c_2} \tag{2}$$

Checking for oppositeness is not as straight forward. First it should be noted that a simple antonym system is ill-suited for the requirement of this study to be used in lieu of oppositeness. This is because while all relationship words that are antonyms to a given relationship word are in fact indicating an inconsistency, all words that indicate an inconsistency are not antonyms of each other. To overcome this, we need a value on a continuous scale similar to that of the similarity measure discussed above. Given that the word similarity is between 0 to 1 as mentioned in the section 2.2.2, 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 simple antonym system will report *decrease* to be the antonym of *increase*. But it will report all the rest of the words under the umbrella term; not-antonym. Obviously, that is not an adequate result.

In comparison, 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 antonym 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.

If one decides to use the naïve approach and take the inverse of calculated the similarities, one would get the result shown in Table 2.

Table 2: 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

If the words are sorted in the increasing difference according to the above calculated values, 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 naïve approach, we introduce the following method.

First, for each of the pair of words, the lemma is extracted using the dictionary created at section 3.4. 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 *one 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 dif_1 as shown in equation 3. Similarly each antonym of L_2 is checked for similarity against the original L_1 and the maximum difference is extracted as dif_2 as shown in equation 4.

$$dif_1 = max(sim(L_2, a_1(1)), sim(L_2, a_1(2)), ..., sim(L_2, a_1(n))) \quad (3)$$

$$dif_2 = max(sim(L_1, a_2(1)), sim(L_1, a_2(2)), ..., sim(L_1, a_2(m))) \quad (4)$$

Once dif_1 and dif_2 are calculated, the overall difference, dif is calculated using equation 5.

$$dif = \frac{\frac{dif_1}{c_1} + \frac{dif_2}{c_1}}{2} \quad (5)$$

Table 3 shows the results of the dif values for the same example as table 2.

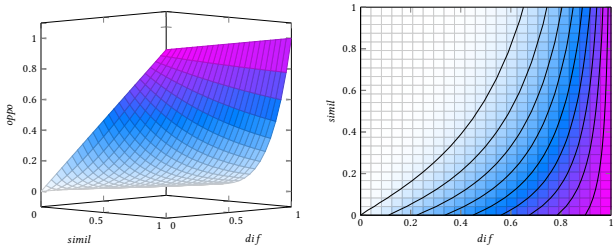
Table 3: OPPOSITENESS WITH ONLY dif

	expand	*decrease*	*change*	*cat*
dif **to** *increase*	0.63	1.0	0.72	0.25

If the words are sorted using dif 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 dif 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 2, 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 6 is introduced to calculate oppositeness. Figure 3 shows the plot for the equation. *simil* is the x variable and dif is the y variable.

$$oppo = dif_T^{(0.5 * \frac{W_{no}}{W_{yes}} * simil_T + 1)} \quad (6)$$



(a) 3D plot (b) Contour plot

Figure 3: Oppositeness function

As evident by Fig 3(a) and Fig 3(b), in higher word similarities ($simil_T$), the difference (dif_T) also have to be very high for the final *oppo* value to be high. in lower $simil_T$ range, *oppo* becomes closer and closer to being directly proportional to dif_T and achieves it when $simil_T$ becomes zero. This quality, in this example, effectively pushes *decrease* farther away from *increase* than others. Values after this transformation is shown in table 4.

Table 4: 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 (dif_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.

5.2.2 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 dif_T are normalized using a small constant ϵ with s_n and d_n as shown in equations 7 and 8.

$$simil_T = \frac{simil_T * (d_n + \epsilon) * W_{yes}}{s_n + d_n + 2 * \epsilon} \quad (7)$$

$$dif_T = \frac{dif_T * (s_n + \epsilon) * W_{no}}{s_n + d_n + 2 * \epsilon} \quad (8)$$

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

5.3 Registering inconsistencies

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. If it passes the threshold, it is registered as an inconsistency.

For each abstract that gets involved in a potential inconsistency, PubMed was queried again to obtain the publication date and other relevant details. 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 inconsistencies 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 2.

Code 2: Intermediate inconsistency 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
```

5.4 Preparing inconsistency 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 inconsistent triples are checked against OMIT to see if either or both of them are of the type miRNA. 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 inconsistencies in the medical abstracts in domains other than miRNA as well.

If either or both the subject and the object are indeed of the type of miRNA, then for each such inconsistency, 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 inconsistency 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 9. C is selected $C > 1$.

$$Con_{fin} = C * Con_{cont} * Con_1 * Con_2 \quad (9)$$

The reformatted inconsistencies 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 3.

Code 3: 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.

.....
```

6 RESULTS AND DISCUSSION

In the PMID-extraction step we obtained 39149 relevant abstract IDs, from which 36877 were processed and downloaded as text files containing abstracts. Around 5.8% of extracted PubMed entries did not have an abstract section, and there were three possible situations. (1) When an entry had some graphs instead of an entire research paper, e.g., PMIDs 24324220, 24318653, 24311611, and 24303553. (2) When there was only a comment about the entry rather than a complete entry, e.g., PMIDs 24311611 and 24303553. (3) When the entry was empty except for the entry name, author names, and other metadata, e.g., PMID 24313780. Other than these three situations, each and every abstract from the remaining 94.2% of relevant IDs were downloaded for analysis.

All 36877 downloaded abstracts were processed to yield OLLIE triple files and Stanford XML files. These intermediate files were used to create the intermediate result file, where a total of 67481 unique subject-object pairs were detected. Then, 503 total inconsistencies were discovered from these subject-object pairs, involving 224 out of 36877 abstracts. This observation indicated that, the percentage of abstracts that contributed to inconsistencies was only 0.61% out of all considered.

After the reduction step (detailed in Section 5.4) was performed to keep only the inconsistencies that involved at least one miRNA entry, we ended up with 102 inconsistencies involving 95 abstracts. This outcome revealed that, out of 503 total inconsistencies, only 20.28% were relevant to miRNA. Abstracts participating in inconsistencies involving miRNA consisted of 0.26% of all downloaded abstracts and 42.41% of those abstracts that were found to be involved in inconsistencies of any kind.

7 CONCLUSION

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

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 [29].

Apart from the above two main contributions, this research also resulted in the creation of the the oppositeness measure introduced in the section 5.2 which would be useful in the natural language processing domain, especially for sentiment analysis.

For future work, one most basic thing that can be improved is in the preprocessing stage to include common medical acronyms that are used but are not defined in the first use. It is also possible to investigate the redacted articles mentioned in section 5.1.1 to see if the redaction was a result of an inconsistency. It is also possible to extend the *cleaning the strings* step and *creation of final triples* step using the already generated Stanford XML.

ACKNOWLEDGMENTS

Funding for this research was provided by the National Cancer Institute (NCI) at the National Institutes of Health (NIH), under the Award Number U01CA180982.

REFERENCES

[1] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, and others. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics* 25, 1 (May 2000), 25–29.

[2] E. Brent, C. Atkisson, and N. Green. 2010. Time-shifted online collaboration: Creating teachable moments through automated grading. In *Proceedings of Monitoring and assessment in online collaborative environments: Emergent computational technologies for e-learning support*. IGI Global, 55–73.

[3] N. H. N. D. de Silva. 2015. SAFS3 algorithm: Frequency statistic and semantic similarity based semantic classification use case. In *Proceedings of Advances in ICT for Emerging Regions (ICTer), 2015 Fifteenth International Conference on*. IEEE, 77–83.

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

[5] K. Eilbeck, S. E. Lewis, C. J. Mungall, M. Yandell, and others. 2005. The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biology* 6, 5 (2005), R44.

[6] Exiqon. 2016. What are microRNAs? (2016). Retrieved December 08, 2016 from http://www.exiqon.com/what-are-microRNAs

[7] P. W. Foltz, D. Laham, and T. K Landauer. 1999. Automated essay scoring: Applications to educational technology. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Vol. 1. 939–944.

[8] National Center for Biotechnology Information. 2017. PubMed Help. (March 2017). Retrieved April 22, 2017 from https://www.ncbi.nlm.nih.gov/books/NBK3827/

[9] M. Franzke and L. A Streeter. 2006. Building student summarization, writing and reading comprehension skills with guided practice and automated feedback. *Highlights from research at the University of Colorado, a white paper from Pearson Knowledge Technologies* (2006).

[10] T. R. Gruber. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5, 2 (June 1993), 199–220.

[11] F. Gutierrez, D. Dou, S. Fickas, D. Wimalasuriya, and H. Zong. 2016. A hybrid ontology-based information extraction system. *Journal of Information Science* 42, 6 (2016), 798–820.

[12] J. R. Hobbs, D. Appelt, J. Bear, D. Israel, and others. 1993. Fastus: A system for extracting information from text. In *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 133–137.

[13] J. Huang, B. Eilbeck, K.and Smith, J. A. Blake, and others. 2016. The Development of Non-coding RNA Ontology. *International journal of data mining and bioinformatics* 15, 3 (Jan. 2016), 214–232.

[14] J. Huang, F. Gutierrez, H. J Strachan, D. Dou, and others. 2016. 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 (2016), 1.

[15] V. Jayawardana, D. Lakmal, N. de Silva, A. S. Perera, K. Sugathadasa, and B. Ayesha. Deriving a Representative Vector for Ontology Classes with Instance Word Vector Embeddings. In *Proceedings of Innovative Computing Technology (INTECH), 2017 Seventh International Conference on*. IEEE, to appear.

[16] J. J. Jiang and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *10th International Conference on Research in Computational Linguistics, ROCLING'97* (1997).

[17] S. Kulick, A. Bies, M. Liberman, M. Mandel, and others. 2004. Integrated annotation for biomedical information extraction. In *Proceedings of the Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*. 61–68.

[18] A. Leskovec, J.and Rajaraman and J. D. Ullman. 2014. *Mining of massive datasets*. Cambridge University Press.

[19] C. Y. Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL-04 workshop: Text summarization branches out*, Vol. 8. Barcelona, Spain.

[20] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, and others. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of Association for Computational Linguistics (ACL) System Demonstrations*. 55–60.

[21] Mausam, M. Schmitz, S. Soderland, R. Bart, and O. Etzioni. 2012. 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)*. 523–534.

[22] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. 1990. Introduction to WordNet: An on-line lexical database. *International journal of lexicography* 3, 4 (1990), 235–244.

[23] T. Mitchell, T. Russell, P. Broomhead, and N. Aldridge. 2002. Towards robust computerised marking of free-text responses. *Proceedings of International Computer Assisted Assessment Conference*.

[24] D. A. Natale, C. N. Arighi, W. C. Barker, J. A. Blake, and others. 2011. The Protein Ontology: a structured representation of protein forms and complexes. *Nucleic Acids Research* 39, suppl 1 (2011), D539–D545.

[25] U.S. National Library of Medicine. 2016. Medical Subject Headings. (2016). Retrieved November 25, 2016 from https://www.nlm.nih.gov/mesh/

[26] H. Shima. 2016. WordNet Similarity for Java (WS4J). (2016). Retrieved November 23, 2005 from https://code.google.com/p/ws4j/

[27] World Health Organization (WHO). 2017. NCD mortality and morbidity. (2017). Retrieved April 22, 2017 from http://www.who.int/gho/ncd/mortality_morbidity/en/

[28] I. Wijesiri, M. Gallage, B. Gunathilaka, M. Lakjeewa, and others. 2014. Building a WordNet for Sinhala. *Volume editors* (2014), 100–109.

[29] D. C. Wimalasuriya and D. Dou. 2010. Ontology-based Information Extraction: An Introduction and a Survey of Current Approaches. *Journal of Information Science* 36, 3 (June 2010), 306–323.

[30] Z. Wu and M. Palmer. 1994. Verbs Semantics and Lexical Selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics (ACL '94)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 133–138.