Ontology Design on Breast Cancer Cause, Symptoms, Diagnosis, and Treatment

Gregory Brown, Sai Sri Narne, Priyanka Gaikwad, and Lakshmi Korrapati

*Department of Computer Science Electrical Engineering*  
*University of Missouri-Kansas City*Kansas City, USA  
gbkhv@ mail.umkc.edu, sn69k@mail.umkc.edu, pvg9gb@mail.umkc.edu, vkvn3@mail.umkc.edu

**Abstract—Ontologies were developed in Artificial Intelligence to facilitate knowledge sharing and reuse. Since the beginning of the nineties ontologies have become a popular research topic investigated by several Artificial Intelligence research communities, including Knowledge Engineering, natural-language processing and knowledge representation. In recent years, there has been explosive growth in the amount of biomedical publications. This paper that deals with the design of an ontology based on causes of breast cancer, Symptoms, identification of Cancer and methodologies used to identify the Cancer in breast cells in order to contribute a knowledge graph in reference to some of the existing ontologies.**

**Keywords — ontology, breast cancer, automated design Spark**

# I. Introduction

Understanding of the cause of disease has improved, as has our capacity to store and process biomedical data. With these advances, the use of complex datasets and applications for processing such datasets present new opportunities to improve our understanding of causes, symptoms, diagnosis, staging, and treatment of disease. We are now able improve the quality of health care decision-making by allowing incorporation of the most recent research results in the decision-making process. Diagnosis and treatment can be guided by development of machine-learning techniques for analysis of the ever-increasing volume of published biomedical research, including unstructured data. Development of an ontological approach to facilitate analysis of this literature would allow more rapid and efficient incorporation of this vast source of data in critical decision-making for patients by physicians and would promote improvements in health care. Specifically, the area of breast cancer causes, symptoms, diagnosis, and treatment, which are the subjects of significant research, would benefit from an ontological approach to facilitate analysis and processing of recently published literature, allowing this literature to be more readily utilized by clinicians to improve patient-oriented outcomes. The focus of this paper is the construction of an ontology using systematic technique for analyzing this literature.

# II. Related Works

Many works have been proposed to create ontologies and their application from a medical corpus. Oshani et al.[4] proposed a prototype system which uses semantic technology approach,which is based on automatic parsing of data from AJCC staging manuals. They have included additional biomarkers which are present in revised 8th edition of AJCC Cancer Staging Manual. The mapping of breast cancer terms to community accepted terms from the National Cancer Institute thesaurus(NCIt) is included as part of ontology creation.

In this work they have considered staging criteria from AJCC 7th edition and AJCC 8th edition to construct Inference agent. The map files which are extracted corresponds to staging criteria specified in AJCC staging manual. For calculation of Inference agent it has not considered the narrower stages specified in AJCC staging manual.

# III. Proposed Work

There are following areas which are focused in our Breast Cancer ontology work.

* Causes
* Symptoms
* Diagnosis
* Treatment

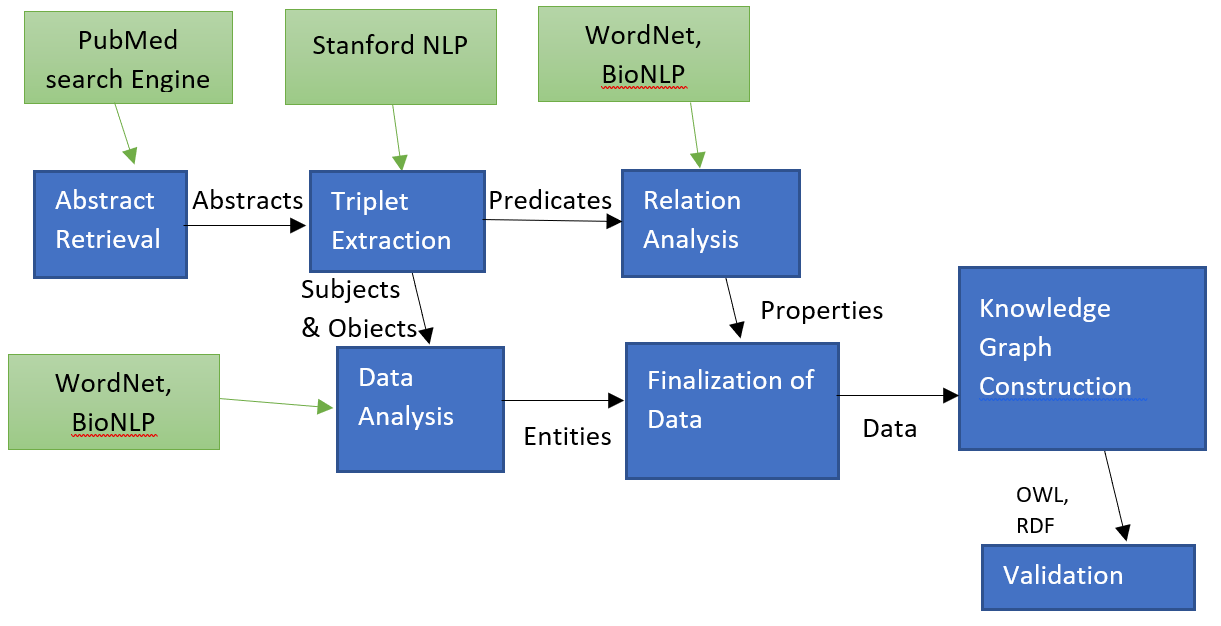


Figure : Step-by-step procedure with tools use

The process of generation of Ontology has following steps which are executed sequentially. The flow of these techniques can be seen in figure 1:

1. Information Extraction:

1. Retrieval of abstracts:

There is lot of research work done on breast Cancer by biomedical Scholars and how to integrating results from biolabs to digitalized form, where each of their is of their work is published in the form of some research paper. Here considering such text from these publications processed data for ontology creation. We use PubMed search engine [8] in the Scala code to retrieve PubMedID accordingly to obtain data set of abstracts. NLP actions are performed on this data.

b. Natural Language Processing (NLP):

This is used to process and analyze large amounts of natural language unstructured data. Following processes are conducted as a part of NLP.

* 1. Tokenization is the process of identifying individual words, numbers, and other constructs.
  2. POS Tagging assigns each word in a sentence its respective part of speech such as noun, verb or adjective.
  3. Named Entity Recognition identifies entities such as persons, locations, and times within a text corpus.

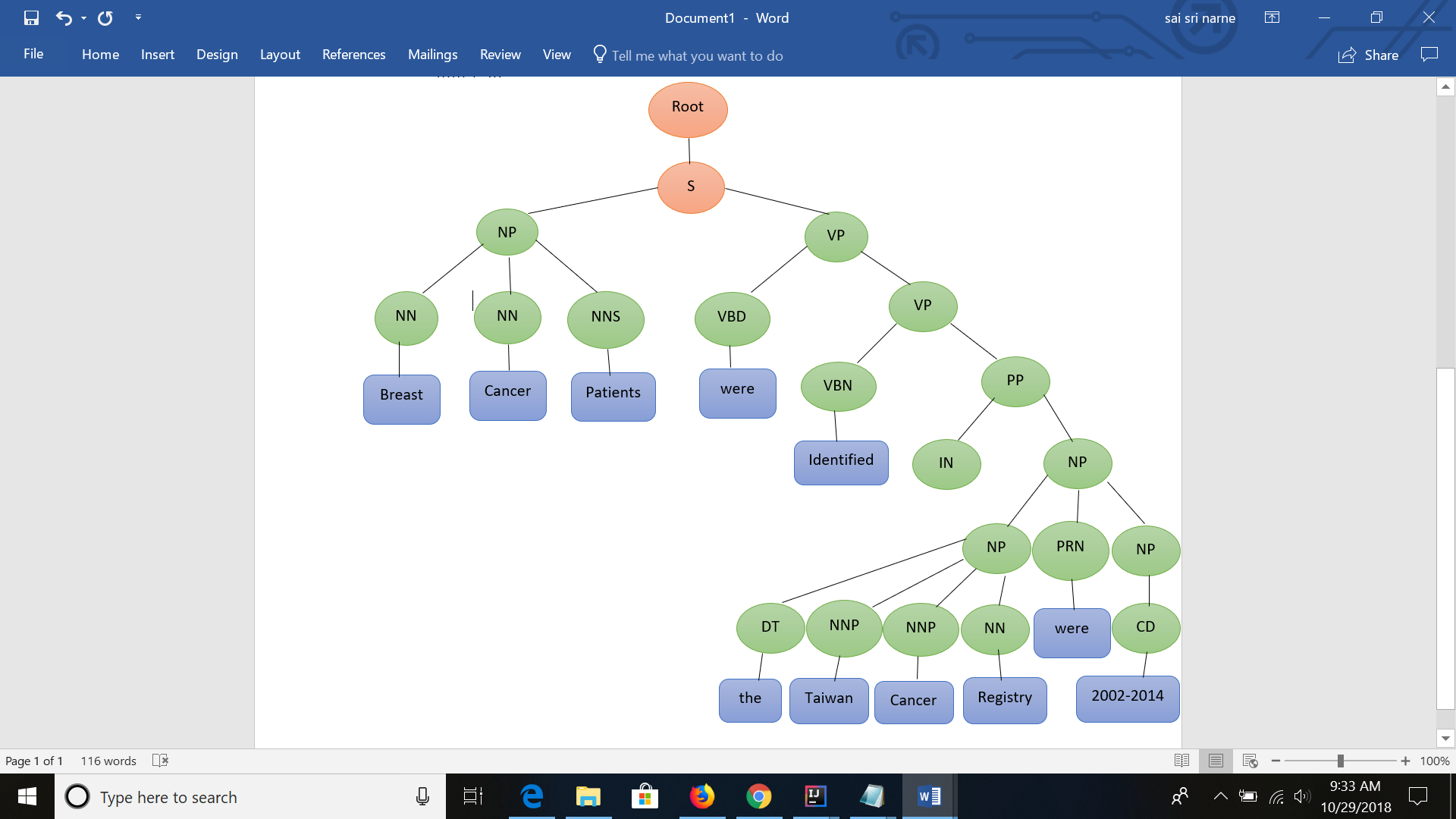
Example sentence: Breast cancer patients were identified from the Taiwan Cancer Registry (2002-2014). Following is the dependencies tree for this sentence. We use NLP for triplet extraction via the Stanford NLP simple document openie function. This triplet extraction is the second step in our ontology construction following retrieval of abstracts. After extracting triplets for each of these sentences we split the triplets into subjects and objects, and predicates.

Figure : CoreNLP Basic Dependencies

2. WordNet:

RiTa WordNet provides methods for returning synonyms. We use scripts to find synonyms for words in our subjects and objects and create classes based on commonly occurring synonyms. This same process can be used to simplify our collection of synonyms, though there is no analog of classes and subclasses as data properties, so we must be careful not to lose too much information.

3. BioNLP

BioNLP can be used to find medical words relevant to documents in our corpus based on the PubMed IDs. Medical words belong to one of 5 categories—Chemical, Disease, Gene, Species, or Mutation—each of which is the basis for a class in our final ontology. For our predicates we use BioNLP categorization to determine appropriate domains and ranges for object properties.

After finding classes and subclasses based on our subjects and objects, and properties based on our predicates we filter to remove duplicates and output these classes, subclasses, and properties along with the triplets and individuals to text files which will be used in ontology construction.

4. Ontology Construction:

The Semantic Web organization provides a library which includes functions for the construction of an ontology. Our ontology construction process incorporates the data from the text files containing classes, subclasses, properties, individuals and triplets and constructs an OWL xml ontology using the functions provided by the Semantic Web organization.

5. (Unused) Information Retrieval:

We have used information retrieval techniques to process the data as well, namely TF-IDF and Word2Vec. The results from these techniques have not yet proven useful and do not contribute to our ontology construction. These techniques do

not refer to any outside sources or known sentence structures, instead analyzing the mathematically.

1. TF-IDF combines term frequency and inverse document frequency of terms in the corpus to determine which terms are relevant.
2. Word2Vec creates vectors, or dynamic arrays, by creating a co-occurrence matrix and performing single value decomposition. The resulting vectors are analyzed using cosine similarity to determine which terms in the corpus are synonymous

**Algorithms and Pseudocode:**

Much of the code for the retrieval of abstracts is done via API calls or use of NLP libraries. In particular for analysis of parts of speech and lemmatization we use the Stanford CoreNLP library and for retrieval of synonyms (prior to word2Vec) we use Rita WordNet library functions. Since Stanford CoreNLP is a Java library, we have used auxiliary scripts in Java in our Scala code.

|  |
| --- |
| **Algorithm1** |
| **Ontology\_Construction**()  {  Abstracts = **RetrieveAbstracts**(PubMedIDs)  CoreNLPResults[] = **StanfordCoreNLP**(Abstracts)  NER = CoreNLPResults[1]  Triplets CoreNLPResults[2]  BioMedical\_Objects = **BioNLP**(Triplets.objects)  BioMedical\_Subjects = **BioNLP**(Triplets.subjects)  Subject\_Synonyms = **WordNet**(Triplets.Subjects)  Object\_Synonyms = **WordnNet**(Triplets.objects)  Subject\_Considered = combine(BioMedical\_Subjects, Subject\_Synonyms)  Object\_Considered = combine(BioMedical\_Objects, Object\_Synonyms)  Predicates\_Considered=compare(Triplets.predicates, NER)  Trips = combine(Subject\_Considered,  Object\_Considered,  Predicates\_Considered)  Triplets\_Considered = compare(Trips, existing\_ontology\_triplets)  OWL = generate\_OWL(Triplets\_Considered)  Visualaize\_Ontology(OWL)  } |

# IV. Software &System Architecture

**Software:**

In this paper we carried our work by Intellij IDEA Community Edition 2018.2.2 environment using Scala SBT version1.0 and Spark 2.12.0 framework. Scala largely provides support for functional programming and static types making lives easier. In order to efficiently perform the operation over large corpuses we used spark framework to divide the entire corpus into small chunks of data and process it in parallel across partitions of our processing power, emulating the Hadoop distributed file system (HDFS). We set the memory and number of partitions to be used in order to utilize more of our machines resources than might otherwise be available. Notably, the pictured Spark initialization is at the head of our script which gets medical word statistics via API call.

**System Architecture:**

In-order to achieve our goal there are many preprocessing steps to be followed as in figure-2. The following are the tools used:

PubMed search engine- In the first step of figure, we used PubMedID’s to retrieve abstract text. Since, PubMed has over 28 million citations of biomedical information from MEDLINE and also a lot of journals, we used this tool to extract data. It is a free resource provides access to United States National Library of Medicine Database [8]. It is also maintained by National Center of Biotechnology Information(NCBI). This search engine makes our process to extract all the literature that contains medical information.

Stanford CoreNLP- As a part of Natural language processing, text annotation, named entity recognition, parts of speech are to be performed on the extracted text or also called to be the linguistic analysis of data, we made use of Stanford CoreNLP [9]. Stanford NLP group has come up with Stanford CoreNLP, a natural language software as their goal to make it very easy for performing linguistic analysis tools to a piece of text using pipeline. The results we got are quite interesting and also accurate mentioned in the table 1

National Center of Biotechnology Information (NCBI)- In general the medical words comprise of various categories like species, gene, chemical etc. NCBI [10], which is a part of United states of National Library of Medicine allows us to search for proteins, minerals, gene, species, chemicals etc. We made sure to find category of each word.

National Center of Biomedical Ontology (NCBO)- There might be some existing list of ontologies on specific topic, NCBO [11] helped us to find the list of ontologies providing BioPortal Rest API call.

WordNet- In order to analysis more about the text data, we used WordNet. WordNet is a large lexical database of English sometimes called as a dictionary that combines nouns, verbs, adverbs, adjectives into a cognitive set of Synonyms (Synsets). We analyzed these results and compared this with Word2Vec and term frequency Id of words, shown in table 2.

Semantic Web- It is an idea of world web inventor, a mesh of data that can be processed by machines easily. Semantic web [12] makes machines work like intelligent humans. We developed W3C web ontology (OWL) to represent rich and complex knowledge about breast cancer cause, treatment, diagnosis using the extracted data.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Words | Unique Words | Nouns | Verbs | Other POS | WordNet Words | Unique Wordnet Words | Medical Terms | Unique Medical Terms |
| 17267 | 3959 | 3960 | 1421 | 4289 | 8559 | 1868 | 679 | 204 |

Table : data statistics

# V. Implementation and Evaluation

We use scripts throughout the analysis of the corpus to automate the process. Currently the steps described in the proposed work section of this paper are executed by collections of scripts, however ultimately, we intend to streamline this. The scripts used are composed in a combination of Java and Scala with Spark. Thus far our corpus consists of 99 abstracts retrieved from PubMed collected automatically by providing query terms, specifically breast cancer causes, symptoms, diagnosis, and treatment. However, we have manually vetted these abstracts to some extent. First, we analyzed the corpus using NLP to get data statistics shown in table 1.

Code used to process and analyze corpus is stored at: <https://github.com/toadSTL/CS5560BreastCancerOntology>

These data statistics are obtained via spark map-reduce functions which map ‘<word>’ to ‘(<word>,1)’ for words in the corpus and then reduce by key. For part-of-Speech Statistics we use Stanford Core NLP pipeline to annotate the data, and then do map-reduce, and similar techniques are used for WordNet word analysis and Medical term analysis.

These data statistics are not used for ontology construction, but are helpful in understanding the data which we continue to process. The major steps of our ontology construction are laid out in section III of this paper. In the remainder of this section we will follow an example of what this process does to a single sentence to show the process. The following the sentence we will use as the example:

“CXCR4 is a chemokine receptor frequently overexpressed in invasive breast cancer that has been shown to play a major role in signaling pathways involved in metastasis.”

The first step is to do triplet extraction on the sentences in the corpus. Our script for this reads a fresh instance of the corpus into a Spark context so that we can first perform triplet extraction. For the example sentence the following triplets are returned:

[(CXCR4, overexpressed In, invasive Breast Cancer), (CXCR4, frequently Overexpressed In, invasive Breast Cancer), (CXCR4, frequently Overexpressed In, breast Cancer), (CXCR4, overexpressed In, breast Cancer)]

Of these triplets we filter so that we get the longest subject and objects as well as the shortest predicate. This filtering leaves us with:

(CXCR4, overexpressed In, invasive Breast Cancer)

While there are other sentences in our corpus which result in predicates including the notion of ‘expression’, overexpression different enough to warrant its own predicate. Thus, we do not do synonym filtering on it.

Thus the next step performed by our scripts is to use RiTa WordNet and BioNLP to categorize the subject and object into classes. BioNLP determines ‘CXCR4’ to be a gene and ‘invasive Breast Cancer’ to be a disease. As a result of these matches we will make ‘CXCR4’ a subclass of the class ‘Gene’ and ‘’ a subclass of the class ‘invasive Breast Cancer’ disease.

At this stage we could perform Information Extraction techniques to help us improve our class-subclass structure. We have performed the Information Extraction techniques TF-IDF and Word2Vec on our corpus and the results are not helpful enough that we can use them to improve our class-subclass structure, or schema. When we do TF-IDF The result of TF-IDF is a matrix corresponding to the terms from the corpus.

We have several scripts to perform TF-IDF on different ‘words’ which we consider from the corpus. Specifically we performs TF-IDF on the words as they are in the corpus, lemmas, bigrams and trigrams. The most important words as determined by TF-IDF are then output to text files to be used by our Word2Vec scripts.

The basic strategy is to initialize the Word2Vec model using the data from the corpus. Afterward the model consists of vectors corresponding to the words from corpus and we check the cosine similarity of these vectors to determine potential synonymous terms. We output the top 3 candidates for synonyms for each of 20 most important words from the corpus as determined by TF-IDF.

The table (below) shows the synonyms found by the W2V process for the terms deemed most important or relevant to the corpus by TF-IDF:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Synonyms for word: PL | Synonyms for lemma: pl | Synonyms for bigram: TX stage | Synonyms for trigram: patients with PA | Synonyms for MedWord: cancer |
| treatment 0.9969 | ' 0.9969 | in patients 0.9912 | leading cause of 0.9658 | recurrent 0.9739 |
| risk 0.9968 | occur 0.9964 | of breast 0.9910 | 0.9585 | due 0.9638 |
| second 0.9967 | cancer-related 0.9963 | breast cancer 0.9909 | cause of death 0.9556 | lines 0.9539 |

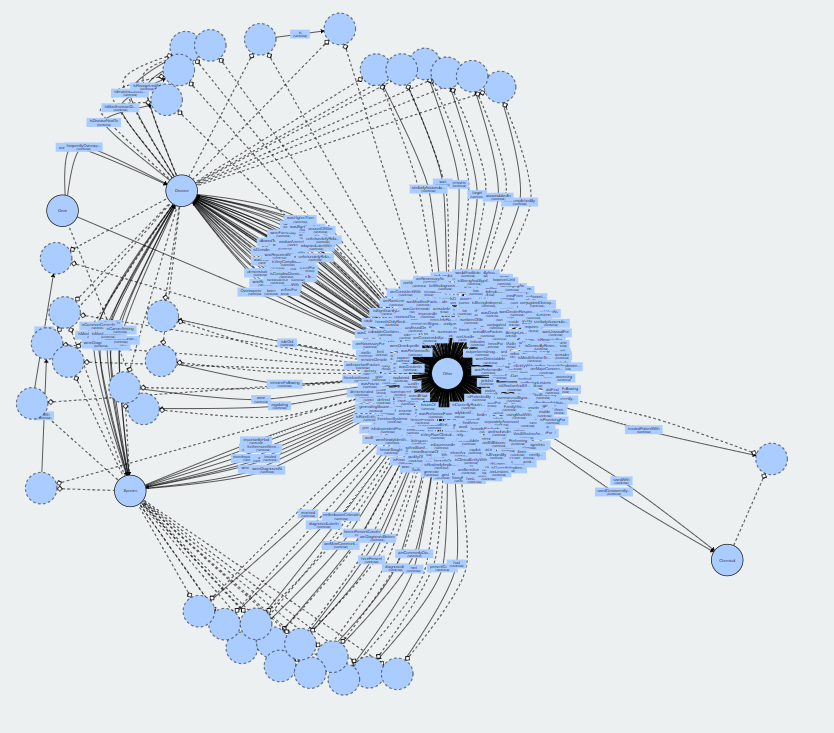


Figure : Ontology Visualization

As the table shows, some of the top TF-IDF terms are appropriate, but largely the synonymous terms found via Word2Vec are not. In particular “‘“ and “ “ should be removed by more thorough pre-processing of the data, but the real terms found as synonyms are perhaps indicative that we should be doing better vetting of the articles which make up our corpus. For now, we simply exclude TF-IDF and Word2Vec from our ontology construction architecture, rather than attempting to use them to improve our class-subclass structure.

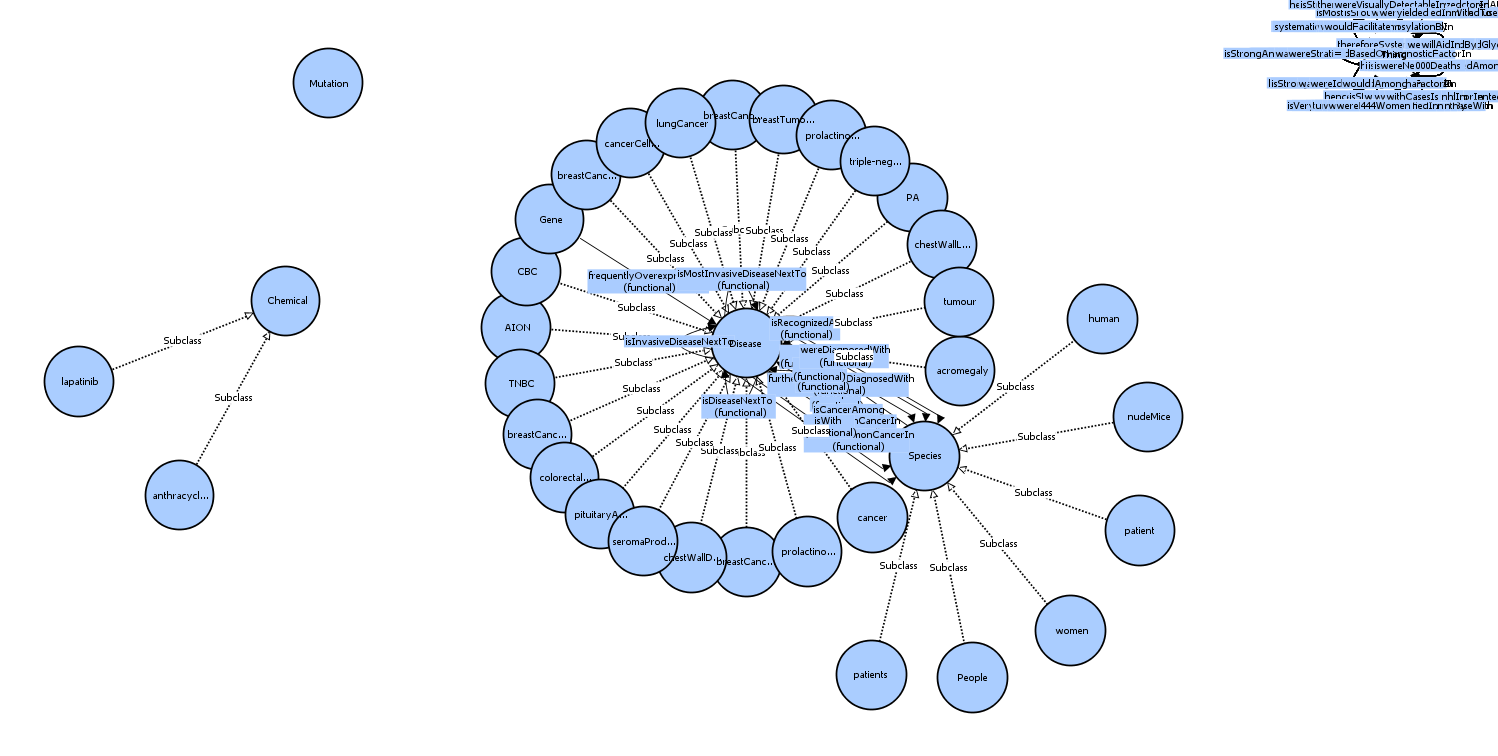
Continuing with our example we prepare the subject, predicate, and object from our sentence for inclusion in our ontology. At this stage we remove stopwords and remove duplicates, but this triplet doesn’t contain any stopwords and without loss of generality we can suppose that this is the version of that triplet which was not removed as a duplicate. Preparing it for inclusion in our ontology involves adding the class and subclasses relevant for this triplet to our lists of classes and subclasses. In this case we add ‘Gene’ and ‘Disease’ as classes and add ‘CXCR4’ as a subclass of ‘Gene’ and ‘invasiveBreastCancer’ as a subclass of ‘Disease’. In addition to adding these items to our subclasses we also currently add them as individals. Please not that for invasive breast cancer we have converted the phrase to camel-case, and removed spaces, we did the same for ‘CXCR4’ but this process does not change that entity’s name. We also add the line ‘overexpressedIn, Gene, Disease, Obj’ to our list of object properties as Gene is the domain for this function and disease is the range. This is determined by comparing the triplet with separate filtered lists based on our classes. Finally we add the triplet to our triplets output document using the following line:

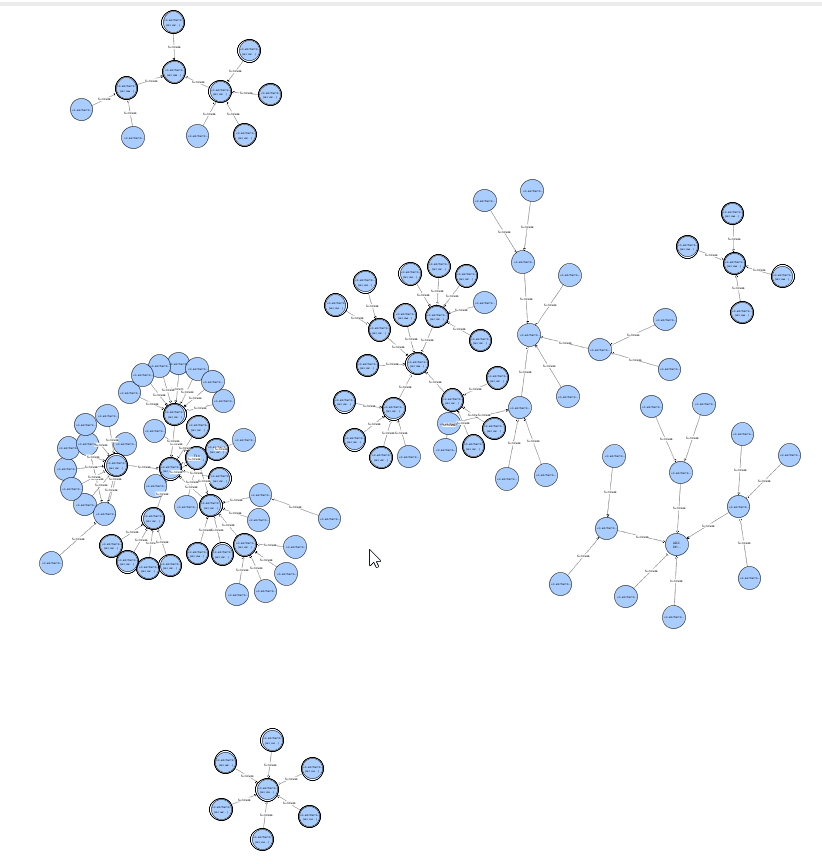
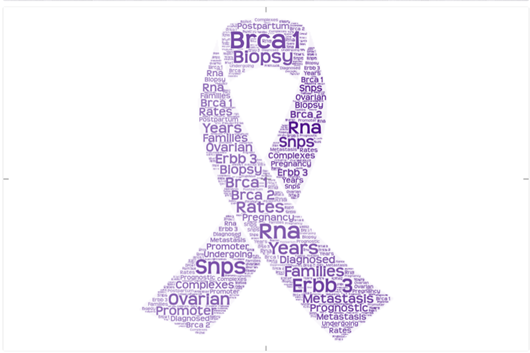
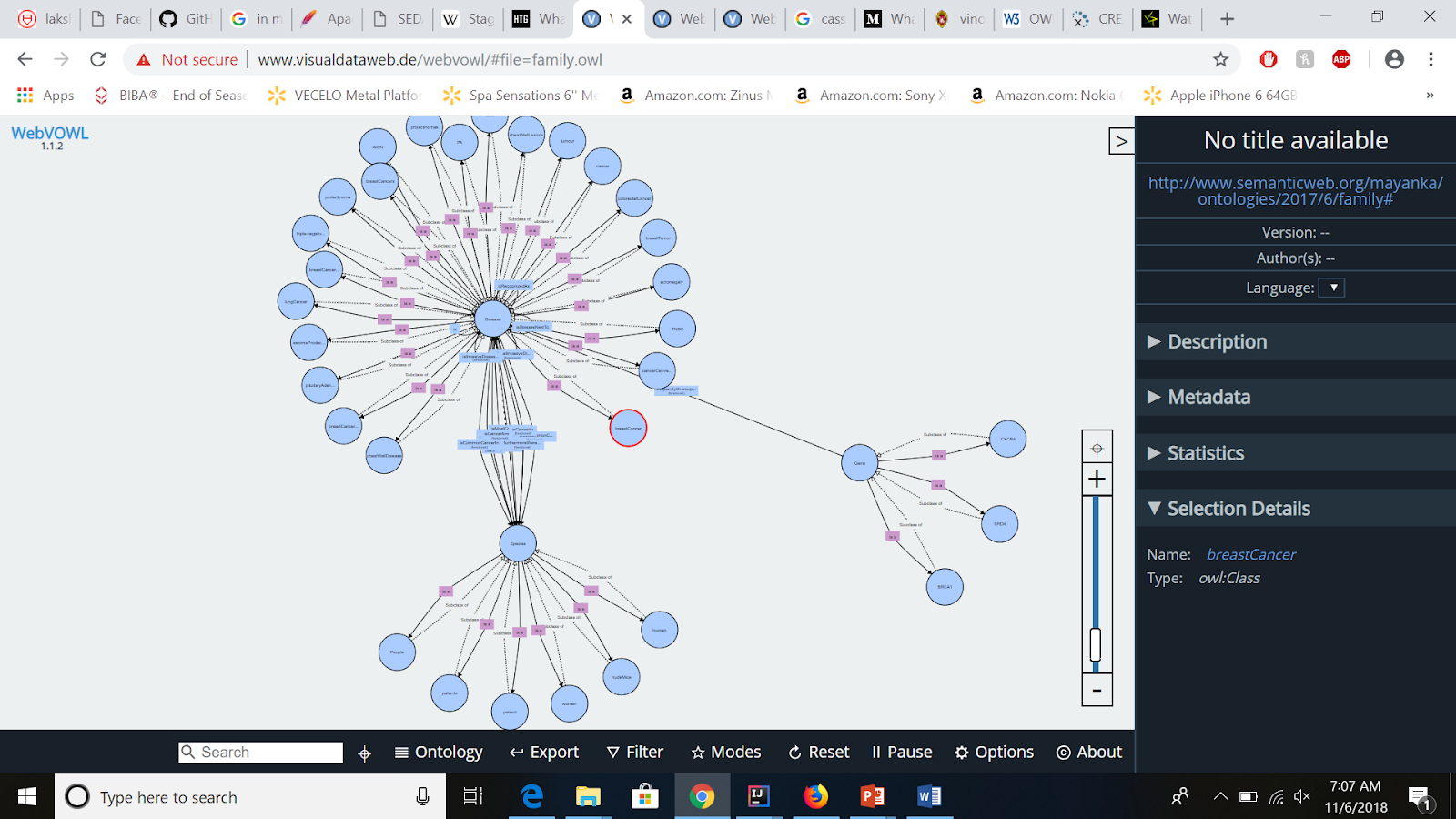
‘CXCR4,overexpressedIn,invasiveBreastCancer,Obj’

Once we have output all of our data to text files we use the Semantic Web organizations function for ontology creation to generate an ontology based on the data. Fig. 3 shows an image of the ontology we have constructed and Table 3 shows statistics about that ontology.

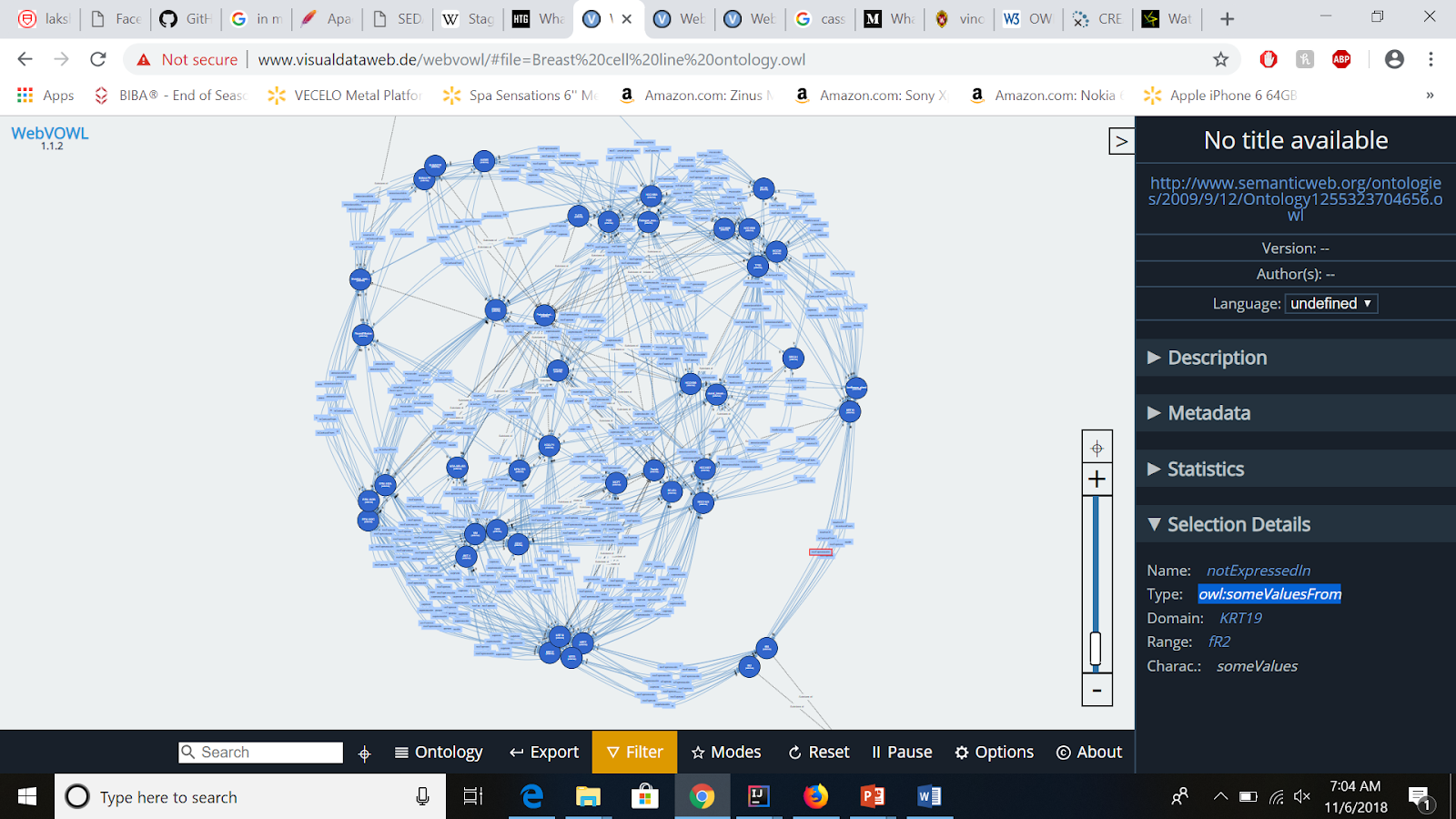
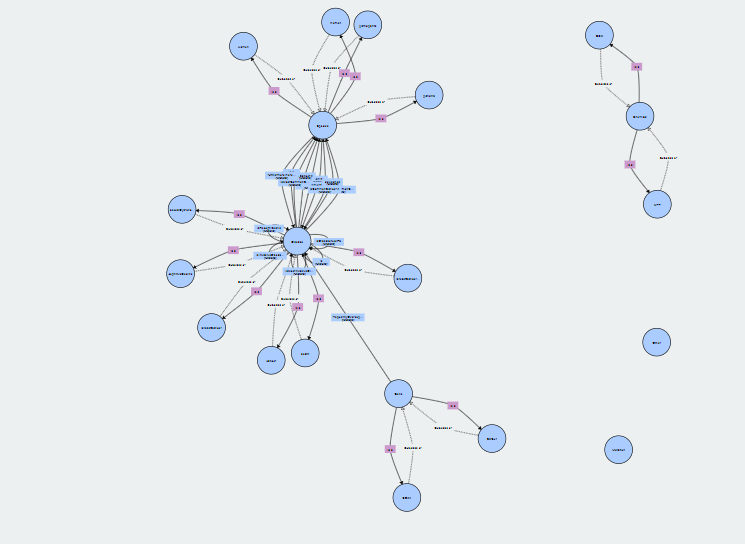
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classes | Object prop. | Datatype prop. | Individuals | Nodes | Edges |
| 1925 | 349 | 0 | 0 | 1926 | 4190 |

# 





# https://lh5.googleusercontent.com/18Hl2kzzNezo8cI2sqdj0B45W6fOtFrrjXt6hrFps1o8AzBxVbZhy4YRP1O704Mwy9dhVUapkAoMf1B3lcVr4MF--BbfwkQdFiW4JOBeYJT1ru8Jz416ykhqcDBaU7rt2fKhNDrTzzA

# V. Conclusion

In this paper, the major work is focused on developing an ontology on breast Cancer using PubMed publications. Ontology consists of breast cancer cause, symptoms, diagnosis and treatment

# References

1. <https://ieeexplore.ieee.org/document/8217715>

1. <http://dline.info/fpaper/jdim/v15i5/jdimv15i5_3.pdf>
2. <https://arxiv.org/pdf/1607.08074.pdf>
3. <https://arxiv.org/pdf/1807.07991.pdf>
4. <https://ieeexplore-ieee-org.proxy.library.umkc.edu/document/4562013/>
5. <https://ieeexplore.ieee.org/document/6834613/>
6. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5174013/>

[8] <https://www.ncbi.nlm.nih.gov/pubmed/>

[9] <https://stanfordnlp.github.io/CoreNLP/>

[10] <https://www.ncbi.nlm.nih.gov/>

[11] <https://bioportal.bioontology.org/>

[12] <https://www.w3.org/standards/semanticweb/>