Dynamic Medical Ontologies

Megha Nagabhushan, Rohithkumar Nagulapati

***Abstract*—In present day medical science, the published literature, in particular medical journals, provide the primary medium of data source for new medical findings and research. This data is raw and cannot be comprehended by people outside medical field. Also, extracting relationship between different concepts in the data and representing it in a format that can be easily queried is required. In our project, we are proposing a model that is going to extract medical data from a reliable source and apply natural language techniques to extract relationships in the form of subject, predicate and object triplet. Similar triplets are being clustered and their topics are assigned to them in order to create ontology. The created Ontology is used to answer questions in our question-answering interface for the dataset.**

I. Introduction

MEDICAL journals contain a lot of useful medical information that cannot be comprehended by common people. Obesity is one of the medical condition addressed in many of the recent medical journals. Obesity can lead to cancer and other severe medical conditions. However, it is difficult to read and understand the causes, treatments and other conditions related to the problem by just reading the journals. Also, these journals contain a lot of medical terms which is very difficult to comprehend.

In our experiment, we are collecting medical data on “obesity” from a reliable source. PubMed in one such source that is maintained by National Center for Biotechnology Information (NCBI) and they comprise of more than 27 million citations and also abstracts for various medical journals. We are using three abstracts that we could retrieve using the PubMed API for our project.

The collected data was processed parallelly through tf-idf (one of the natural language techniques which assigns vectors to the words and helps in determining the top words in the given text) and OpenIE(a technique of retrieving tuples containing subject, predicate and object). The open IE results will also contain some unwanted triplets having stopwords(words like “the”,”a” etc. ,which frequently appear in the document but they do not have any significance). In order to remove these useless triplets, we are matching our triplets with the words generated using tf-idf techniques. Triplets containing the tf-idf words are retained and the other triplets are discarded.

The important triplets are now clustered using the k-means clustering which considers the whole triplet as a string and matches one triplet to another in order to find the similar triplets and cluster them together. The clustered triplets are then processed through a topic discovery technique called LDA which determines the topic in each cluster the authors are talking about.

Using the above methods, we are building our ontology for the obesity data. Our model also provides a question answering interface through which users can query and find answers to their questions on the obesity dataset.

II. Related Work

A. *Towards an obesity-cancer knowledge base*

Towards an obesity-cancer knowledge base, is one of the journal paper published in Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference, where they use three step procedure to construct an ontology for the text corpus.

In the first step, they are performing NER(Name Entity Recognition) to determine the different name and entities present in the text. In the second step, they are doing relation extraction where they are trying to find if two entities are related. As the final step, they are classifying the specific relations.

NER is a very important step in Natural Language processing. In this work, they are using the bio medical entity detection which is very useful to determine the meaning of the various medical terms present in the paper. They are taking the help of domain experts to manually annotate the medical words for the 214 sentences that they retrieved using the PubMed API with the key word search as “obesity” and “cancer”. For a large text corpus, manually annotating the words will be a tedious task. Also, they are using binary classification for the relationship extraction. Binary classification will not perform topic discovery of a particular text corpus rather it just classifies based on if a word belongs to a particular entity relations or not.

To overcome this limitation, in our mode, we are performing k-means classification on our important <Subject, Object, Predicate> triplets from the text corpus. Also, we are using LDA for topic discovery so that we can determine the topic that the authors are discussing about in each cluster of the triplets.

.

B. *FRED*

Fred is a tool that automatically produces RDF/OWL formatted ontologies from natural language sentences. This tool uses multiple natural language processing components

which then formalizes the output to visual knowledge graph. The generated output graph is designed according to Frame semantics where each frame is expressed by verbs or other linguistic constructions formalized as OWL n-ary relations.

Pros:

Fred is domain and task independent tool suitable for task-specific applications

It is available both as rest service and python library

It changes input from discourse representation structures to RDF/OWL n-ary relations

It can represent modality, tense and negation of sentences.

It is capable of handling compositional semantics, taxonomy induction and quality representation

Cons:

Fred is incapable of handling large datasets for generated ontology visualization

Its results are not uniform if both facts and concepts are expressed by natural language text

Fred is not capturing coercion, adjective semantics, polarity, sentiment, frame composition, presuppositions and paraphrases

Tool doesn't extract accurate implicit discourse relations

Frame construction out of real world facts is bad in Fred.

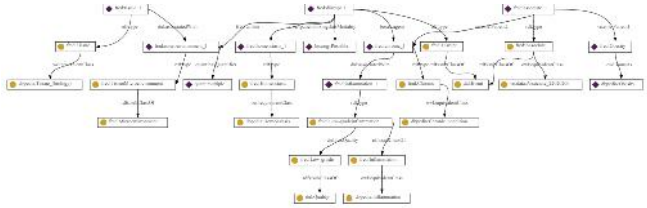

Fig. 1A. FRED Visualization for one sentence of our text corpus.

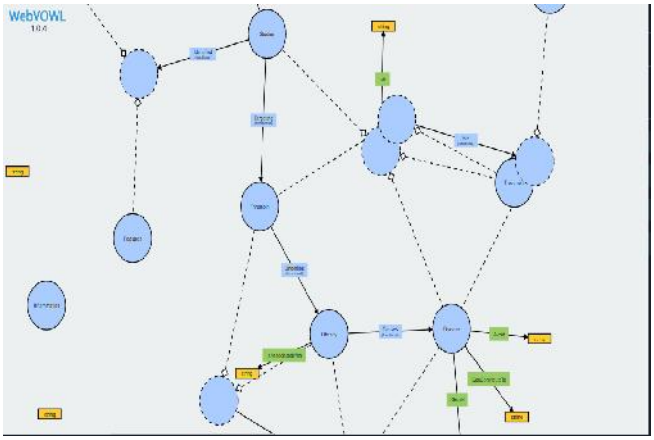

Fig. 1B. WebVowl Visualaization of the generated Ontology

III. IMPLEMENTATION

A. Dataset

We are collecting data from Medical Domain for our Project. We are choosing one medical problem as our dataset. Obesity has become a very serious health problem in the world today. For our model, we have collected data regarding obesity using the PubMed API.

PubMed (pubmed.gov) is a free resource developed and maintained by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine(NLM).PubMed provides free access to MEDLINE, NLM's database of citations and abstracts in the fields of medicine, nursing, dentistry, veterinary medicine, health care systems, and preclinical sciences.PubMed comprises more than 27 million citations for biomedical literature from MEDLINE, life science journals, and online books.

An abstract is a concise summary of a larger project (a thesis, research report, performance, service project, etc.) that concisely describes the content and scope of the project and identifies the project's objective, its methodology and its findings, conclusions, or intended results.

Hence, we retrieved three Abstracts from research papers rendered by PubMed API for search string, "obesity".

B. System Design

The input data is processed parallelly. The work-flow of the system is represented in Figure-2.

As our first step, we are performing Core Natural Language Processing(NLP) tasks on the dataset using the CoreNLP Libraries in Java. The Core NLP tasks that we are performing are:

Tokenization: Tokenization is the process of breaking sentences into tokens which are the smallest constructs of a huge text data.

Lemmatization: Lemmatization is the process of separating words into individual morphemes and identify the class of the morphemes.

Stop word removal: Stop word Removal is the process of removing stop words from the data.For example, the stop words in English be:able, about, above, according, accordingly, across, actually, after, afterwards, again.

Top tf-idf words retrieval: tf–idf is short for term frequency–inverse document frequency, a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. We are using this method on our dataset to extract the top 50 important words in it.

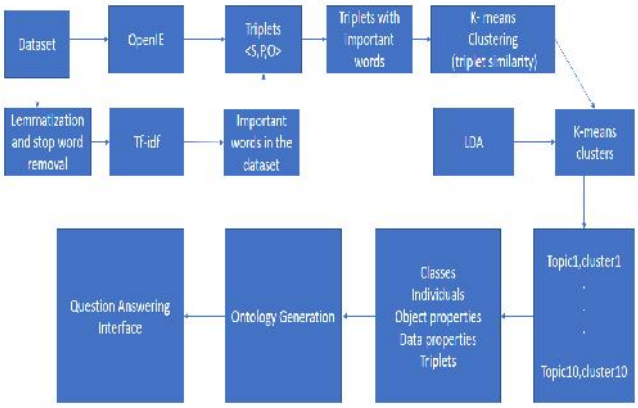

Fig. 2. Workflow of the Dynamic medical ontology question-answering process.

Parallel to the above step, we are performing Open Information Extraction (OpenIE) on our dataset. Open information extraction (open IE) refers to the extraction of relation tuples, typically binary relations, from plain text. The OpenIE delivers its result in the form of Quadruple <Subject,Predicate,Object,Confidence-Score> from which, we are retrieving our own triplet <Subject,Predicate,Object>

We are now going to retain only the important triplets by matching our tf-idf results with the OpenIE triplets. As in, we are going to retain only those triplets whose subject or object string matches one of the top tf-idf words.

The next step is to cluster these triplets into group of similar triplets. For clustering, we are using K-means clustering algorithm. We are string matching each triplet with all other triplets to determine the similar triplets. For our experiment, we are considering K=10.

Now we have 10 clusters from K-means clustering. We have to now discover the topic for each cluster.

For this process, we are using LDA. Latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.

LDA provides the topics along with the score value. We are considering the topic with the top score as the class for each cluster. At the end of this stage, we have 10 topics, each belonging to a cluster. In each cluster, we are considering all other remaining LDA topics as instances of the main topic.

We are going to save our classes in the Classes File. The classes and their instances are saved in the Individuals file. And all the triplets together are saved in the Triplets file.

We are now checking if the objects in the triplet belong to any of the Classes or instances. If they do, they are saved in the ObjectProperties file. If not, they are saved in the DataProperties file.

With all the elements required for the ontology creation, we are constructing our ontology using OWL API. The OWL API is a Java API and reference implementation for creating, manipulating and serializing OWL Ontologies.

We are visualizing the generated ontology using the online WebVOWL tool. WebVOWL is a web application for the interactive visualization of ontologies. It implements the Visual Notation for OWL Ontologies (VOWL) by providing graphical depictions for elements of the Web Ontology Language (OWL) that are combined to a force-directed graph layout representing the ontology.

We are building a question answering interface for our Ontology. Protégé, a free, open source ontology editor and framework for building intelligent systems is being used to query the ontology and retrieve answers for the questions in the interface.

C. *Results and Observations*

Figure-3 explains the steps of the Stage-1 of our experiment. It can be seen that the dataset containing three abstracts have been processed parallelly through tf-idf and OpenIE. The results of the tf-idf are matched with the results of OpenIE and the important triplets are retrieved.

Fig. 3. Dataset is being processed parallelly to generate 50 tf-idf words and 173 triplets and these results are string matched to retrieve the 110 important triplets.

The k-means clustering and the LDA performed on the dataset is shown in Figure-4.

We have considered a value of K=10 and obtained 10 clusters and performed LDA on each cluster and a total of 10 topics, one belonging to each cluster.

Fig. 4. shows the Triplets being clustered using K-means and LDA applied on them to discover the topic.

The different files created for the ontology construction is shown in Figure-5.

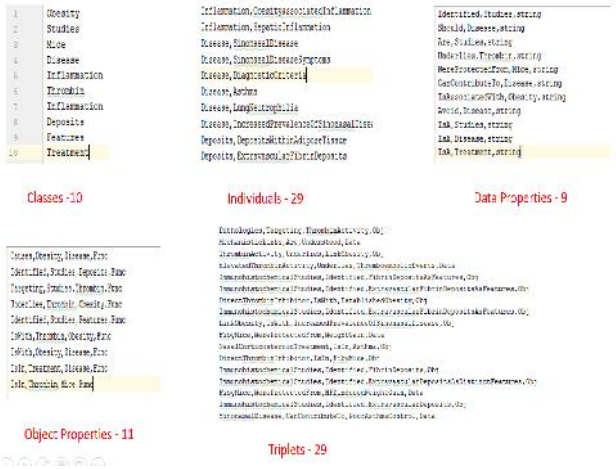

Fig. 5. 10 classes, 29 individuals, 9 data properties, 11 object properties and 29 results are being used for ontology creation.

Figure-6 shows the WebVowl visualization of our ontology constructed using the OWL API.

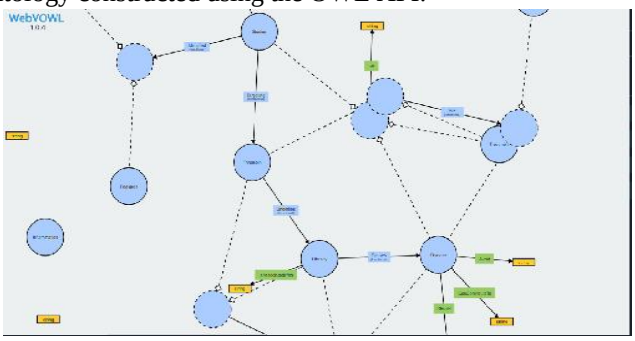

Fig. 6. Ontology Visualization for the obesity data.

We have created a question Answering interface to retrieve the answers for the SparQL queries. The interface was created using AWT.

Figure-7 shows our question answering interface where we are trying to retrieve subject by querying using predicate and object

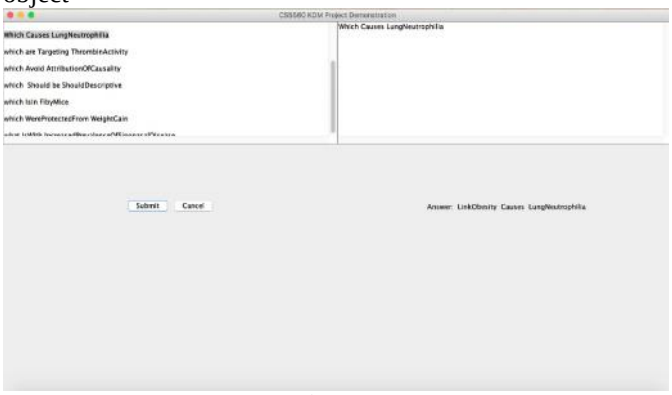

Fig. 7. Question answering interface

IV. Future Work

We are working towards the enhancement of the project by annotating the medical terms used in the abstract to make the ontology more understandable. Also, we are trying to extend this model to multiple ontologies for different papers and comparing their similarity.

References

[1] Juan Antonio Lossio-Ventura, William Hogan, François Modave, "Towards an obesity-cancer knowledge base: Biomedical entity identification and relation detection" in BIBM, 2016 IEEE International Conference .

[2] J. Calic; E. Izuierdo," Efficient key-frame extraction and video analysis"

[3] S. Nussbaum; J.E. Smith, "Statistical simulation of symmetric multiprocessor systems".

[4] A. Cameron, "Optimal tactile sensor placement"

[5] A. Krivoulets, "On coding of sources with two-sided geometric distribution using binary decomposition"

[6] S.T. McMahon; I.D. Scherson," A statistical mechanical approach to a framework for modeling irregular programs on distributed or cluster computers".

[7] C. Cocianu; L. State; V. Panayiotis," On a certain class of algorithms for noise removal in image processing: a comparative study"

[8] On a certain class of algorithms for noise removal in image processing: a comparative study, "Mitigating the impact of weather and climate on railway operations in the UK"

[9] K. Taghva; T. Nartker; J. Borsack; A. Condit, "Determining the usefulness of manually assigned keywords for a vector space system,"

[10] Wei-Ching Tham; S.I. Woolley; S. Cribbs; D. Anderson ," Diagnostically lossless compression of pipeline inspection data'