CPSC 490 Project Proposal: Biological Entity and Relationship Extraction

Robert Tung and Adrian Lin
Advisor: Dragomir Radev
Submitted: September 21, 2017

1 Abstract

Millions of medical research papers have been written in the past century. This explosion in research publications has created an mountain of information that can be difficult to process by a team of researchers much less a single person.

We seek to use natural language processing methodologies to create a tool that can take in one or more biological papers and extract, with high confidence and minimal false positives, biological entities (e.g. genes, mutations, diseases, etc.) and their relationships. Specifically, this project will look at the effect of genes and mutations on diseases.

The tool will be incorporated into a web application and should also be able to extract overall deductions from an aggregate set of papers. If time allows, the strength of extracted relationships should be measured using a confidence metric.

2 Background

There has been previous work done in extracting the relationships between certain biological entities, though as far as we are aware, none are focused on finding the relationship of genes and mutations to diseases. For example, [Saric et al., 2004] use the PubMed corpus in extracting gene to gene relationships.

Prior work has also been done regarding biological named-entity extraction. Perhaps most relevant to our project, [Morgan et al., 2003] used pattern-matching to develop a method of creating training data for extracting genes and proteins from a body of text. Similarly, [Hou and Chen, 2003] use collocation to enhance the extraction of protein names. [Torii et al., 2003] investigate various sources of information used for this classification task, while [Krallinger et al., 2004] assess the correlation between expressions used to characterize the function of genes and proteins and their names. In regards to relationship extraction, [Liu et al., 2007] find protein-organism-location relationships in biomedical papers.
Existing tools that contain named-entity extractors include NLTK, spaCy, the Stanford CoreNLP Library, and the University of Illinois.

As for databases of biological entity names to compare to and train from, BioCreative is one such corpus and OMIM (Online Mendelian Inheritance in Man) is a database of genes and genetic disorders.

Additionally, for any other resources, Professor Radev has a search engine of NLP papers and resources called AAN (All About NLP).

3 Project Outline

Overall, this project will require us to do the following

1. Scrape for Medical Papers - figure out which papers need to be downloaded and build up a corpus
2. Find a PDF to Text Parser and clean up the resulting text.
3. Use named-entity extractors to recognize entities such as proteins and diseases. Find the best named-entity extractor and tune or modify. If necessary, create our own named-entity extractor for this specific area.
4. Find a library that extracts relationships between entities in text. Find the best relationship extractor and tune or modify. If necessary, create our own relationship extractor for this specific area as well. This may be likely since the quality of existing relationship extractors is not as high.
5. Create a web application that takes in a paper and returns the biological entities and relationships.
6. Stretch Goal: Track the source (provenance) of each relationship and update the confidence of each relationship.
7. Stretch Goal: Create a more accurate confidence measure for each relationship extracted. Use this to have aggregate information over a set of papers.
8. Stretch Goal: Update the web application to take a set of papers and extract overall deductions and aggregations. For example, these set of 30 papers found that gene X is correlated with disease A while another 20 claim that gene Y is correlated with disease A.

4 Deliverables (Robert)

To begin, while Adrian is (1) scraping medical papers from sources such as PubMed Open Access, I will be (2) finding a suitable PDF to text parser and cleaning up
any resulting text. One possibility that Professor Radev has brought up is PDFBox, which is available on LILY’s computer cluster.

After this, I will begin (3) looking at named-entity extractors. Possible existing named-entity extractors include that of NLTK, spaCy (an NLP package for Python), the Stanford CoreNLP Library, and the University of Illinois. For each of these, I will implement the named-entity extractor, and use them on a subset of biological articles generated by Adrian. Then I will determine how many of the discovered entities are confirmed biological entities in databases of biological names. For example, BioCreative is one such corpus of biological entities, and OMIM is a database of genes and genetic information. Finally, in addition to testing the above 4 named-entity extractors, I will use Professor Radev’s NLP paper and resource search engine (AAN) to search for other alternatives, in turn implementing and testing the effectiveness of these tools. Finally, I will tune the best tool for genes and diseases specifically and, if necessary, build upon it to create the best possible extractor for our purposes.

While Adrian is working on (4) creating the relationship extractor for genes and diseases, I will then begin work on (5) the front-end of the web application that will ultimately take in a paper or group of papers and return its biological entities and relationships. Adrian will work on the server-side of the application when he finishes the relationship-extractor.

Finally, if time permits, (6) I will additionally track and store the source (e.g. paper) of each relationship found and use this to inform the confidence (provenance) of each relationship. Once Adrian (7) adds other possible aspects of the wording of each relationship to more accurately measure confidence, we will be able to (8) update the web-application to return only those relationships of high confidence across a set of papers. For this, I will again implement the front-end while Adrian will implement the server-side.

As a stretch goal, we can incorporate machine learning to better inform the confidence of the relationships, based on various additional factors to the wording such as the author, verbosity of the paper, length of the paper, etc.

References


