CS410 Project Proposal

Team TR Squirrels

20-Oct-2020

Team TR Squirrels has three members, Captain Ye Xu (Net ID: yex2), member Weidi Ouyang (Net ID: wonyan2) and Raj Datta (Net ID: datta7). We will work on reproducing the listed paper "Generating Semantic Annotations for Frequent Patterns with Context Analysis" using Python as the primary coding language. This proposal addressed all related questions from week1 guideline and topic instruction. Answers to these questions were highlighted.

1. Background of the paper

This paper proposed a novel approach to generate semantic annotation for frequent patterns that can better interpret the in depth and hidden meaning of the pattern. One meaningful application of this algorithm/procedure is in the field of computational biology. With tremendous sequencing data of genes and their transcripts (e.g. mRNA sequence and protein sequence), annotation of functionality is much needed but barely and poorly supported by the lab-work evidence. Most annotations in our biological databases are counted on computational work that link the known or predicted functions to frequent patterns observed in sequence data. One example is to connect the structural motif of a peptide that describe the connectivity of secondary structural element (e.g. "helix-turn-helix" or "Zinc finger") to a potential function of a protein or part of the protein (e.g. DNA binding domain of a transcription regulator). Biologists, either working on "omics" (e.g. genomics, transcriptomics, etc.) or focusing on a particular gene cassette or metabolic pathway, all benefit from such application. A well tagged dataset leads to proficient and precise findings and this is true for applications in areas other than biological science.

However, annotation of patterns is challenging and can be very labor intensive. Taking genome annotation for example, in addition to manual annotation (curation), a large part of annotation work is completed by automatic annotation tools based on different algorithms. The most basic one is the homology search tool "BLAST" though structural and functional annotation are usually needed to identify and tag the biological information to the genomic elements. This whole process often involves both biological experiment (lab evidence) and "*in silico*" bioinformatic analysis. This paper, however, provided a novel approach that completely based on the Text information retrieval and mining to interpret the discovered patterns and tested the procedure on three different datasets including a gene ontology annotation dataset.

2. Resources and Technique to reproduce the paper

We will use the same or similar datasets to reproduce the results.

The first dataset is DBLP dataset that provides bibliographical information about computer science journals and proceedings and is available for download (https://hpi.de/naumann/projects/repeatability/datasets/dblp-dataset.html).

The second dataset is no longer available from the paper link but similar datasets that include the Gene Ontology terms and Motif for Drosophila are accessible from Gene Ontology Website (Annotation database).

The third dataset is provided by BioCreAtIvE Task 1B and is also no longer available. However, we can apply the same approach by crawling abstracts from the MEDLINE database with query keyword “Drosophila” and recreate the dataset.

We will apply the same technique used in the paper which include the two toolkits “FP-Close” and “CloSpan” to generate “Closed Frequent Itemset” and “Krovertz stemmer” to stem the title words. Same or similar clustering algorithms, either Hierarchical Clustering or One-Pass Clustering, will be applied for redundancy reduction. Python libraries and packages, such as “Scipy” and “Scikit-Learn” provide convenient built-in functions to implement these algorithms too.

3. Brief Timeline of the Project

	Course Week
Dataset Acquisition	Week 10
Dataset Processing based on understanding of the pattern context modeling	Week 10 - 12
DataSet Oriented modeling and semantic analysis	Week 13 - 14
Coding and Presentation finalization	Week 15 - 16

The detailed workload and distribution will evolve as the project goes.