GOABL Development Log

Introduction

Part I: Construct Dataset from GEO Sample Data

A species-specified dataset of Shewanella Oneidensis.

A dataset of gene expression in Shewanella Oneidensis is constructed

Collecting Data Sample from GEO

40 gene expression series and 426 samples of Shewanella Oneidensis are collected from Gene Expression Onimbus (GEO) database.

Obtaining Expression Matrix from GEO Sample

Identify Concept from Experiment Condition Protocol

To identify possible GO concepts (terms) related to a certain experiment, we empolyed a sentence embedding model (`sentence transformer`) to natural language description to vectors.

With embedding model, we can convert both the experiment condition protocol and the label (synonym) of GO concepts into vectors, and evaluate their similarity. For a certain experiment protocol, we consider the most similar 10 concepts as its related concepts.

Recognizing Concept from Experiment Condition Protocol (With LLM)

NOTE: We are no longer using this method after Aug.15.

Large Language Models (LLMs) are employed in the preliminary implement. In detail, we used `OpenAI`'s API of `GPT 3.5-Turbo` to understand natural language-based description of `GEO` experiment datasets, and summerize them to corresponding concept IDs in `GO`.

A fundamental risk underlies in the method above: great ambiguity and uninterpretability of LLMs is introduced to the dataset. Therefore, we are exploring better solutions.

Unifying Expression Data with Importance Metric

non-supervised importance

Unifying Expression Data with Differential Expression Tools

A differential expression gene (DEG) analysis is applied to the origin dataset (obtained in last section) to construct a dataset with relative expression level. The DEG was performed with python package PyDESeq2.

Values of first GSM sample in each GSE series is setted as the reference, and relative expression in other GSM samples are measured. In the final dataset, the top 20 most expressed genes (ranked by `p-value`) of each GSM sample is selected as data labels.

Part II: Embedding Method

In a traditional method, the concept name-descripted instances are mapped to ectors to train a learner model.

Ontology Embedding

The concepts of Gene Ontology is mapped to vectors with `owl2vec_star`, which preserves information of knowledge graph structure, word vector and literal meaning of concept names.

Processing Multiple Concepts

For instances with multiple concepts as input, it is mapped to the mean value of the concept embeddings.

Obtaining Learner Model with Concept Embeddings

Part III: Knowledge Graph Preprocessing: Remembering

Knowledge graphs (KGs) like `GO` are often in large scale, containing much information that is irrelevant to specific learning task and a certain degree of noise. To perform tractable reasoning and abductive learning on such large-scale KG, we adopted a method, `ABL-KG` (Huang et al. 2023) to mine logic rules and filter out irrelevant information from large-scale KGs.

Structure of `OWL` Knowledge Graph File

Subgraph Extraction

Rule Mining

Remember Algorithm

Algorithm 1: Remembering

Result: Write here the result
Input : Write here the input
Output: Write here the output

$R_{new} \leftarrow \{r \in \mathcal{KB} \mid \mathrm{Sig}(r) \cap \Sigma' \neq \emptyset\}$;
while $t \leq T$ and $r_{new} \neq \emptyset$ **do**
  $R_{res} \leftarrow \emptyset$;
end

Part IV: Abductive Learning

Introduction to abductive learning (ABL)

TODO

Use `ABL-Kit` as the major implement of reasoner building and bridging

TODO

Handling multi-label prediction

To predict all over/under-expressed genes, we divided the task into 4707 (total size of the Shewanella Oneidensis genome) separated classifiers, with each predicts the expression of one gene.

...

Building Learner

A `random forest classifier` model, which built by `sklearn`, is used as the base model for ABL, for each .

Building the Reasoner and Identifying Subset of Rules

When implement abductive learning with `ABL-Kit`, override the `KB` class and the `logic_forward` method.

The number of violated rules (axioms) is evaluated in the reasoner.

...

To cut off unneccessary rules for each single classifier, we extracted a subset of the whole rule set, according to the GO annotation file of the corresponding gene.

. . .

Bridging Learner and Reasoner

TODO: detailes on **KG2Rules** and **ABL-Kit** construction of **importance**

References