Biomedical Entity Relation Mining and Ranking System

Introduction/Overview

Purpose:
Our overall research problem is to extract and rank the relation candidates based on the search query medical entities.

Challenges:
1. Explore efficient and effective entity relation candidates extraction methods from the evaluated medical professional papers and support online querying.
2. Design reliable and useful ranking function to show most relevant and confidence relation candidates to users based on different query entities and keywords.
3. Efficiently dynamic detect meaningful snippets for future explaining the search result and works as rank evidences as well.

Dataset:
1. Millions of the PubMed medical paper abstracts (~10.47 GB)
2. Five entity types information from PubTator
3. Gene-Mutation relations from CTD

Relation Extraction and Query Syntax(1)

Reverb Extractor:
1. Syntactic Constraint (Verb Oriented)
   - verb + noun + verb
   - <V, V, N> (e.x.)<Jack, walked, his dog>
2. Lexical Constraint
   - A trained logistic regression classifier based on rich features such as relation tuple extracted window size to compute a popularity score and filter relations by it.

Query Syntax:
- [#keywords] #entity1 | [\@relation] | #entity2 | [?relation type]
- # entity1, Relation, entity2

Examples:
- #Headache #Aspirin → find the relations between them
- #normal #Children # ADHD ?positive → find all positive

Relation Sentiment Analysis(2)

Purpose:
Determine the sentiment of the relations was to derive a metric for polarity. Afterwards, polarity could be used as a way of determining the rank of a relation when queried.

Extraction
Use REVERB to extract the relationship phrases from our contextual corpus.

Pruning
Pruned tuple by cross-examining contents with ChemicalPubtator and DiseasePubtator corpora.

Classifying Training Data
Manually classified several hundred tuples to cosine values according to sentiment of the collected phrases with 1 being most negative and 1 being most positive.

Relation Ranking and Snippet Detection(3)

Purpose:
Create a relation evaluation model by ranking a relation phrase with an entity pair.

Input: Entity1, Relation, Entity2
Output: <Ranked Relations>

Results

Features

- Build word2vec polarity dictionary (Corpus Based)
- Build the relation polarity dictionary (Corpus Based)
- The number of verbs inside relation R (the more, the better)
- The popularity score of the rR,l,p,y would receive high score if the confidence
- The distance between x,y to the sentence

References

[2] Simon Cimadevila, Fabio Rinaldi. Ranking relations between diseases, Biomedical Semantics 2012, 3(b)pp 395