IE - Project Phase 1

Johnny Le, Jesus Zarate

February 24, 2018

1 Baseline

The approach that our hypernym discovery system used was a hyponym pattern mining approach [Hearst 1992] in order to find category members. The first thing we did was run through our training data and find a pair of concepts within range of each other in the same sentence. Then the text in between those concepts was extracted as a pattern. By keeping a frequency of encountered patterns we selected top x matched patterns and used those in order to extract new category members. Another approach we took was to use the suggested patterns [Hearst 1992]. We then used the patters to search the corpus for matching patterns and extracted the noun phrase to the left of the pattern and the noun phrase to the right as well.

Then we created a graph with vertices for the candidate hypernyms, and member pair vertices for the hyponym pairs. For this approach we decided to treat the items on the left of the pattern as candidate hypernyms. For each candidate hypernym we created an edge to a hyponym pair (an item on the right side of the pattern), with the frequency as the edge weight, and we treated the inDegree popularity measure to rank the hypernyms.

2 External Resources

The only external resource we used was nltk for the following:

- Tokenization and sentence splitting
- Part-of-speech tagging

3 Evaluation

For our evaluation, we experimented with our baseline system on a portion of the full corpus of data. The reason for this was that the runtime for our system on the full corpus was an estimated 10+ hours which was too inefficient to aid us in refining our system by the end of phase 1. With that in mind, we divided the full corpus for each domain into 324 datasets where we randomly selected a one of the datasets for use.
There were two means of evaluations conducted:

- Unordered

To analyze our results, we compared the hypernyms that were provided in "*2A.medical.test.data.txt*" and "*2A.music.test.data.txt*" to the hypernyms found by our baseline systems (with the hypernyms provided by the training set removed). A value of 1 was given for each hypernym found in both sets and this was divided by the number of hypernyms in the test set. As we used a 1000th of the total corpus, this result was expected to be low.

Medical Unordered Score: 0.016 Projected Medical Unordered Score on Full Corpus: 0.518391
Musical Unordered Score: 0.002 Projected Musical Unordered Score on Full Corpus: 0.085969

- Ordered

To analyze our unordered results, there were ordered hypernyms provided in the ”2*A.medical.test.gold.txt*” and ”2*A.music.test.gold.txt*” to the hypernyms found by our baseline systems. Scoring this was completed by creating two lists.
Each list was a list of hashsets with the first one representing the test hypernyms and the second representing our ordered hypernyms. Each line of hypernyms was given an index based on the order it appeared in a line and that hypernym was placed into the hashset in the respective index. After each list of hashsets was populated, our ordered hashset list was checked at each level if it had any of the hypernyms present on the same level in the test set. Once again, as we used a 1000th of the total corpus, this result was expected to be low.

Medical Rank: 0, Score: 0.015384615384615385 Medical Rank: 1, Score: 0.0040650406504065045
Medical Rank: 13, Score: 0.022727272727272728
Medical Average score over all levels: 0.0013180290238217068

Music Rank: 0, Score: 0.0
Medical Average score over all levels: 0.0

Our experiments were run on corpuses with 10,000 (0.3% of total) lines of text of the:
2A_med_pubmed_tokenized.txt contains 3,239,945 lines of text
2B_music_bioreviews_tokenized.txt contains 4,298,453 lines of text

Conclusions: After reviewing our work, it appears that our poor results were due to a combination of the smaller corpuses as well as being unable to find quality patterns to locate the hypernyms. Future systems will take this into account with revised methods for locating patterns.

4 Contributions

Johnny's Contributions

- Created hypernym ranking methodology, hypernym extraction by pattern matching, catches for circular dependencies, scoring for unordered hypernyms, scoring for ordered hypernyms

Jesus' Contributions

- Extracted patterns using training data, POS tagging and Noun Phrase chunking, Broke down data into smaller file chunks

Both

- Designed Hypernym extraction API, Designed Pattern extraction API