Homework 8 Report

Matheus Schmitz

Task 1:

1. **You need to submit the "inferred-predicates" folder which includes the target results files for each tutorial:**

   Added all inferred-predicates from task1 to the results.zip folder.

Task 2:

1. **Identify another type of entities in your dataset: movies should be the first one. State your choice in the report.** I have opted for directors as the second type of entity to use. Note that because the entries in the original data had multiple directors per movie, I had to manipulate the entries to generate multiple rows for the same movie, each with one of its directors.

2. **Design a PSL model and generate the necessary data files (observations and targets). List and explain the predicates and rules in your report.**

   The predicates I used were:

   1. DirectorName, which maps the names of directors to IDs generated with the intent of facilitating the referring of entities.
   2. DirectorBlock, which contains blocks to reduce the number of pairwise comparisons among directors. The blocks were generated by taking the first letter of each of the director's name.
   3. DirectorOf, a predicate which links the directors (their IDs) to the movies (also movie IDs) they directed.
   4. MovieTitle, which maps movie names to IDs.
   5. MovieBlock, which generated blocks from the first letter of each movie name, ignoring the word "The" and any non-alphanumeric characters.
   6. SimDirector, which contains the pairwise string similarities for all directors in each block. Director name similarities were calculated using the Jaro-Wrinkler similarity metric.
   7. SimMovie, which contains the pairwise string similarities for all movies in each block. Movie name similarities were calculated using the normalized Levenshtein similarity metric.
   8. SameMovie, which contains all pairwise matches of movies in each block, for which predictions are to be generated.
   9. SameDirector, which contains all pairwise matches of movies in each block, for which predictions are to be generated.

   The rules I used were:

   1. String similarity for director names, meaning that there is vote for match proportional to the string similarity among each pair of names.
   2. String similarity for movie names, meaning that there is vote for match proportional to the string similarity among each pair of names.
   3. Pure transitivity for directors, meaning that for 3 candidates in the same block, if A matches B and B matches C, then as long as A isn't C, A will be predicted as being a match for C.

4. Pure transitivity for movies, meaning that for 3 candidates in the same block, if A matches B and B matches C, then as long as A isn't C, A will be predicted as being a match for C.
5. A Collective KG-based ER Rule, which asserts that there is are two predicted directors for two movies (neither being the same ID), then if for both movies one of the directors is predicted to match, then the rule predicts the that second director for each of the movies is also the same person.
6. A Collective KG-Based ER Rules that predicts that if someone is predicted to be the director of two movies, which themselves are predicted to be the same, then the director is predicted to be the same person.
7. A self-reference rule, that states that identical director IDs means it is the same entity.
8. A self-reference rule, that states that identical movie IDs means it is the same entity.
9. A negative priors rule, which states that by default two entities are not the same.

3. Run PSL inference to perform ER. Verify your performance using the labeled file L in your dataset. Include folder "*inferred-predicates*" in your submission. Report your precision, recall and F1-score in your report. Note: There will be two types of entities but you only need to report the performance of the entity type shown in L.

Inside the source folder, under the task 2 folder, there is a jupyter notebook "generate_files.ipynb" where I developed the pipeline to generate all files needed. There is also a notebook "calculate_metrics.ipynb" where the precision, recall and F1 score were calculated after the predictions were made.

The metrics for the positive class in my model are as follows:

Precision = 0.98857
Recall = 0.91534
F1 Score: 0.95055

Added all inferred-predicates from task2 to the results.zip folder.

I was unable to run PSL with the entire dataset, which results in a target file for movies with 300k entries and a target file for directors with 150k entries. As debated with Minh on his office hours, we devised a solution which consist of sampling half of the candidate datasets for comparison, then adding all pairs in the labeled data, to ensure they would receive predictions, then dropping any duplicates which might have be generated. This sampling approach generated 100k movies to be predicted and 60k directors to be predicted – the reduction is not linear as the number of pairs is equals to the square of the number for items in each block. With this approach there are still extra pairs that help to refine the PSL model, while also ensuring that the number of pairs is not so large that a regular notebook cannot handle.

4. You need to make sure that your PSL model has collective ER rules.

Rules 6 and 7 presented on item (2) are collective ER rules, as they are based on DirectorOf, which is a predicate that links both types of entities (Director, Movie) in my Knowledge Graph.