1 README:
Actor Identity through Semantic Relation Extraction

In this project a distantly supervised algorithm is implemented to gather training data that is used to train classifiers that categorize extractions of semantic relations.

1.1 Top Level Directory

Submitted with the program are the following directories used to run the program:

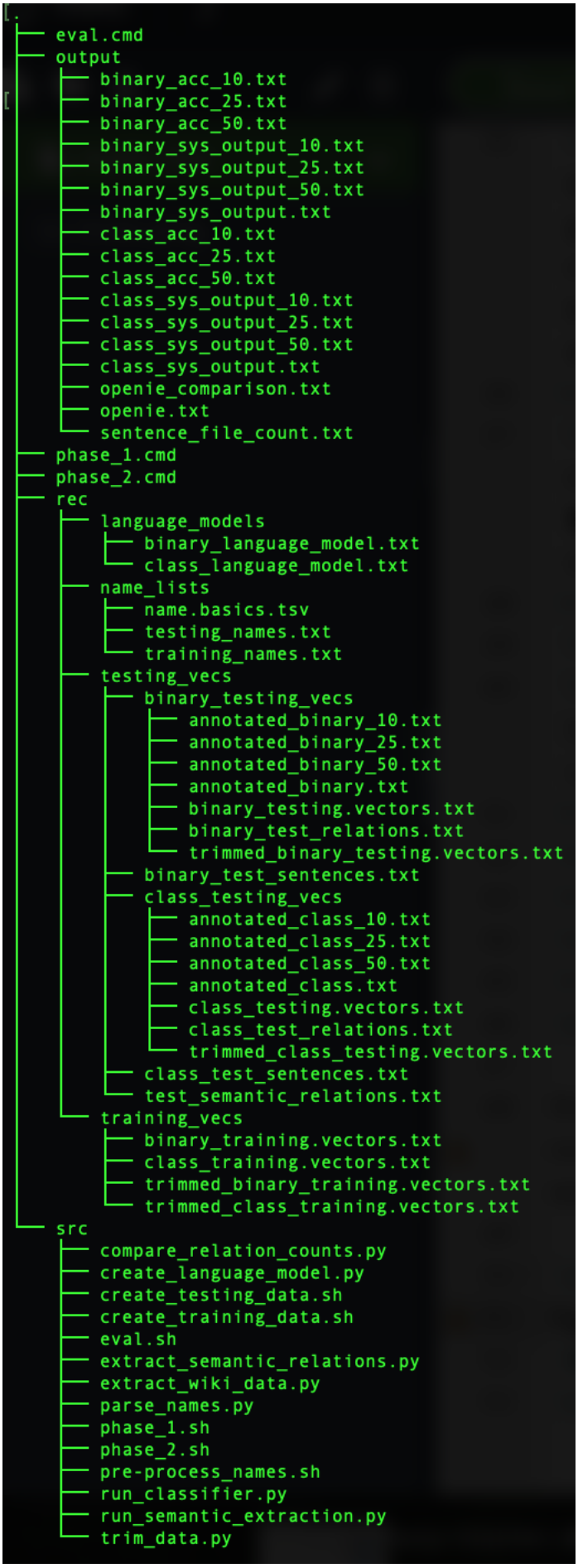
phase_1.cmd	This condor file calls Phase 1 of the program which encompasses several other scripts. This phase pre-processes a list of names (e.g., pre-process_names.sh), uses this list of names to access the html of wikipedia entries and extracts testing and training vectors (create_training_data.sh and create_testing_data.sh), creates language models(create_language_model.py) and extracts semantic relations (extract_semantic_relations.py) .
phase_2.cmd	Phase_2.sh is a wrapper for run_semantic_ extractor.py pipeline. A fixed list of test sentence is sent through a pipeline to extract semantic relations without any labels and currently when called, phase_2.sh will target the unlabeled binary_test_sentences.txt file. However, any file of line separated sentences can be used. Semantic relations are extracted using the get_relation() function in extract_semantic_relations.py file
eval.cmd	For evaluation, this command calls eval.sh which is a wrapper for ./run_classifier.py which is a naive bayes classifier. This is fixed to several sets of annotated data in the resource directory, rec, and then the testing_vectors directory.
rec	This is the resource directory and stores data needed to run the program. That includes language models, name lists, testing vectors, training vectors in both un/annotated and trimmed format.
src	Stores all source code to run the program. This also includes additional libraries like Textacy directory and Cachetools
output	Contains all of accuracy files that contain F1, precision, and recall scores for the binary and multi-class classifiers. Also stores system out files that show classification compared to gold standard by class for all sentences. This directory also contains a sentence_file_count.txt that shows the amount of relations by sentence from an extraction. Finally, there is also a openie_comparison.txt file that shows the Recall from the test extraction compared with the openie.txt file

1.2 Pre-Requisites

Python3 and NLTK must be downloaded in order to use this program.
In addition, Spacy and Textacy.extract are also required. The Spacy language model has an absolute path name to my home directory and would have to be fixed if using off of patas. Textacy is included in the ./src directory

1.3 trimmed_data.py

Trim data takes a file of testing or training vectors and outputs a file in which class counts adhere to a set quota parameter. This script is used to ensure testing and training vectors are even because sometimes wiki extraction may be uneven.


.
├── eval.cmd
├── output
│   ├── binary_acc_10.txt
│   ├── binary_acc_25.txt
│   ├── binary_acc_50.txt
│   ├── binary_sys_output_10.txt
│   ├── binary_sys_output_25.txt
│   ├── binary_sys_output_50.txt
│   ├── binary_sys_output.txt
│   ├── class_acc_10.txt
│   ├── class_acc_25.txt
│   ├── class_acc_50.txt
│   ├── class_sys_output_10.txt
│   ├── class_sys_output_25.txt
│   ├── class_sys_output_50.txt
│   ├── class_sys_output.txt
│   ├── openie_comparison.txt
│   ├── openie.txt
│   └── sentence_file_count.txt
├── phase_1.cmd
├── phase_2.cmd
├── rec
│   ├── language_models
│   │   ├── binary_language_model.txt
│   │   └── class_language_model.txt
│   ├── name_lists
│   │   ├── name.basics.tsv
│   │   ├── testing_names.txt
│   │   └── training_names.txt
│   ├── testing_vecs
│   │   ├── binary_testing_vecs
│   │   │   ├── annotated_binary_10.txt
│   │   │   ├── annotated_binary_25.txt
│   │   │   ├── annotated_binary_50.txt
│   │   │   ├── annotated_binary.txt
│   │   │   ├── binary_testing.vectors.txt
│   │   │   ├── binary_test_relations.txt
│   │   │   └── trimmed_binary_testing.vectors.txt
│   │   ├── binary_test_sentences.txt
│   │   ├── class_testing_vecs
│   │   │   ├── annotated_class_10.txt
│   │   │   ├── annotated_class_25.txt
│   │   │   ├── annotated_class_50.txt
│   │   │   ├── annotated_class.txt
│   │   │   ├── class_testing.vectors.txt
│   │   │   ├── class_test_relations.txt
│   │   │   └── trimmed_class_testing.vectors.txt
│   │   ├── class_test_sentences.txt
│   │   └── test_semantic_relations.txt
│   └── training_vecs
│   ├── binary_training.vectors.txt
│   ├── class_training.vectors.txt
│   ├── trimmed_binary_training.vectors.txt
│   └── trimmed_class_training.vectors.txt
└── src
    ├── compare_relation_counts.py
    ├── create_language_model.py
    ├── create_testing_data.sh
    ├── create_training_data.sh
    ├── eval.sh
    ├── extract_semantic_relations.py
    ├── extract_wiki_data.py
    ├── parse_names.py
    ├── phase_1.sh
    ├── phase_2.sh
    ├── pre-process_names.sh
    ├── run_classifier.py
    ├── run_semantic_extraction.py
    └── trim_data.py