Description

A machine learning program. Its purpose is to learn a classifier that gets 2 words and determines if the first word is a hypernym of the second.

The system is design to run mostly on a local computer, while the rest is done distributed on AWS EMR & S3.

The main idea is to learn patterns suggesting that one word is a hypernym of the other, based on this research paper, with few adjustments.

The input for the system is Google Syntactic N-Grams (the corpus), as well as a list of pairs of words and their label ("True" if the first word is a hypernym of the second, "False" otherwise) – a hypernyms list.

The main steps in short:

1) Stem the corpus (a map-reduce distributed program).
2) Match sentences in the corpus to pairs in the hypernyms list and extract the pattens as explained in the research paper (a map-reduce distributed program).
3) Unite all the reducers outputs from the previous step (a map-reduce distributed program).
4) Make the sample vectors for the classifier from the most common patterns found.
5) Train the classifier. In this project we chose soft-SVM, for linear classifier is well fitted for this type of learning, and very easy and efficient as well. In addition, we used 10-cross validation for the classifier.
6) Show Precision, Recall and F1 measures, and the confusion table.

Manual

The following should be in the directory of HypernymDetection:

1) hypernym.txt
2) nGrams (the corpus, not in the git repository due to its size)
3) Stemmer.jar
4) ExtractPatterns.jar
5) UniteReducersOutputs.jar