CS 481 - NER

Yacin Nadji & Christos Mitillos

October 7, 2008

1 User Manual

See included README file for instructions about how to run our software.

2 Code Overview

We implemented the standard NER algorithm as described in class. Our code is organized in two main portions: nlp.ner.TransformLearner and nlp.ner.Rule. Rule is an abstract representation of a rule that can apply itself to any arbitrary 3-tuple of tokens. TransformLearner represents the bulk of the algorithm. The most important methods are main, runSeedRules and learnNewRules. main runs all the necessary functions to properly parse a POS-tagged document. It initializes everything required by the algorithm, and runs the necessary methods to print out the results.

runSeedRules runs the seed rules we gathered from the beginning of the document, and other common-sense tagging methods. For example, anything followed by "Mr—Dr—Mrs" and is a proper noun, is more than likely a person. With these rules, we define a set of SEED tags to properly tag the remainder of the document.

learnNewRules generates new rules, and applies them to the entire document. In order to determine the relevancy of a rule, the ruleset is applied to a sample of the document that was manually tagged, to determine if the estimated elements are correct. After ranking the rules, the top n are applied according to the method described in the assignment.

3 Experimental Results

1. Our seed rules do a decent job for the neighborhood, which is to be expected. Our rule "Person" of "Organization" caused a lot of false positives, for example: the University of Edinburgh would improperly be tagged with University being labeled as a person, which is obviously incorrect. Naturally, since we were focusing on a small portion of the corpus, we would get problems with other portions of the text.

2. Overall, in relation to the last 1,000 tokens, we improved our overall performance. The added rules caused us to find more entities, however, we also noticed some incorrectly tagged entities.

3. The majority of the errors our system are false negatives, or times our system did not accurately categorize a word. For example, a few names weren't correctly tagged in the remaining portion of the text. "Trish Groves" was incorrectly tagged, as the following word, "captures" wasn't ever introduced as a sample rule during the learning rules phase.

4. Overall, the best improvements could be realized by two main things. First, a larger properly annotated corpus would allow us to have a larger set to compare against for our rules. Secondly, expanding the versatility of our ruleset would also improve our results. For example, we are limited by a one token radius when it comes to comparing tokens. If we were able to compare a range of n tokens, it would allow us to have larger accessibility when it comes to annotating an entire document.