Introduction to Computational Linguistics 2006

Miles Osborne
Informatics
miles@inf.ed.ac.uk

October 24, 2006

1 Introduction

The second assessment deals with *Named Entity Recognition* (NER). NER is about locating items such as *people*, *companies* or *places* in free text. It is similar to chunking, except that labels over words tell us how to recover those entities that we are interested in. More details about NER (and the particular way it is used in this assignment) can be found here:

http://www.cnts.ua.ac.be/conll2003/ner/

2 Tasks

You should complete the following tasks.

- Obtain the data sets and evaluation software. The data can be found here:

  /home/miles/projects/ner/data-eng/
  /home/miles/projects/ner/data-deu/

  And the evaluation script can be found here:

  /home/miles/projects/ner/bin/conlleval.prl

  The first set of data is in English and the second set is in German.

  The data consists of *training* sets (data-eng and deu.train) and two *evaluation* sets for each language pair. You will need to use a training set to build a NER system and then use the evaluation script (which is in *Perl*) to see how well you are doing. Perl programs are run in a very similar way to how Python scripts are run as standalone programs.

  For each language pair, you should use the first evaluation set (eng.testa or deu.testa) when you are developing your system. All results at the end should be in terms of both evaluation sets.

- You are now going to see how well the *TnT* system performs at NER. Information on TnT can be found here:

  http://www.coli.uni-saarland.de/~thorsten/tnt/

  TnT is installed under DICE. First, write a Python program which will convert the various files containing NER material into the format which TnT expects.

- Now, for each language pair, and using the relevant training set, see how well the relevant test set can be labelled. To do this, you will need to convert the newly labelled files back into the original format (but with the newly predicted label at the end of each line). Then, you will have to run the evaluation script to generate performance results.

- What happens when you vary the amount of training material?

- Can you characterise the kinds of errors TnT makes? If you do not speak German then this will be non-trivial.

- Experiment with the various settings that TnT uses. For example, does it matter how unknown words are handled? Does capitalisation help?

- What are the *best* settings for TnT for each language pair? To work this-out, you will need to have some method for systematically exploring various parameters with respect to the first test set and then seeing how well you perform on the second test set.

You need to summarise your findings in a report that is no longer than 10 sides of A4 (code lists do not count to the page length). In particular, you will need to clearly describe how you tackled the tasks (paying attention to assumptions made, problems encountered)

3 Submission

You should use the *submit* command.

Due date: 4pm on 27^{th} November 2006.

If you are going to be late submitting your work, for whatever reason, then you must tell the course secretary or course organiser beforehand. Note: a standard penalty (of the awarded mark) per day operates for all late submissions. Work that is submitted more than one week late will not (usually) be accepted or marked.

Part or all of the late penalty may be waived by the exam committee in certain circumstances, such as illness, or personal problems. If you are ill please make sure that a medical certificate is submitted to the ITO. Also, if other personal crises cause your work to be submitted late, please let the course secretary know.