Conditional Random Fields and sequential tagging

1 Objectives and Material

The goal of this lab session is to implement linear-chain Conditional Random Fields (CRF) for the extraction of named entities. You will first have to study an existing CRF software and and use it for a specific text tagging task, then to implement a part of the CRF inference engine in python. The resources needed can be found in the e-campus page of the course.

1.1 Named Entity Recognition task

"Named Entity Recognition (NER) is a subtask of Information Extraction. The goal is to find the phrases that contain person, location and organization names, times and quantities. Each word is tagged with the type of the name as well as its position in the name phrase (i.e. whether it is the first item of the phrase or not) in order to represent the boundary information[1]."

1.2 Dataset

The dataset to be tagged is a Spanish-language corpus which was provided for the Special Session of CoNLL2002 on NER [Tjong Kim Sang, 2002]. The data is a collection of news wire articles and is labelled for person, organization, location and miscellaneous names. Thus, micro label set consists of 9 labels, that are :

— the beginning and continuation of Person (B-PER, I-PER), Organization (B-ORG, I-ORG), Location (B-LOC, I-LOC), and Miscellaneous names (B-MISC, I-MISC)

— nonname tags (O).

By definition, the continuation of a name type has to be preceded by the beginning or the continuation of the same type. The training data consists of 7230 sentences of average length 36.

2 Getting started python-crfsuite

You will use python-crfsuite, a python binding to crfsuite (originally written in C++). The first steps to take consist in getting familiar with the generation of features under CRF suite by reading the doc available here : http://www.chokkan.org/software/crfsuite/tutorial.html#id485365

3 Training and inference

The remainder of the lab can be found in the notebook file (TpCRF.ipynb) available in e-campus. In order to avoid any issue related to wrong python version, we recommend to use collab. Once the notebook is uploaded, you need to specify "python 2" in the environment parameters (*Modify*-tab and *Notebook parameters*).

The notebook is composed of two parts :

1. **Training a CRF model :** In this part, everything is already implemented. First, you need to run every step once at a time. You need to be sure you understand everything that is done. Second, study the behaviour of the estimated model and the results when considering different types of regularizations (use only L1 and L2, use only L2, etc.) and changing the values of the regularization parameters. Make sure to use l-bfgs to solve the optimisation problem.

1. from *Altun, Yasemin. "Discriminative methods for label sequence learning." (2005).*

2. **Coding your own CRF inference routine :** In this second part, you will implement your own inference routine and compare its results with the `pycrfsuite` library. More specifically, code the `viterbi_decoder()`. If your implementation is correct, the predictions of your implementation should match `pycrfsuite` predictions. In particular, compare the posterior probabilities of the decoded sequences, using your routine and the one defined in `pycrfsuite`.

4 To go further

Try to improve the tagging results by desgining new features.

Create your own features in order to build a chunker on the CoNLL 2000 data, relying on the information provided on http://www.chokkan.org/software/crfsuite/manual.html.

5 Documentation : Python and NLTK - Natural Language processing Toolkit

To start with python :

⋆⋆⋆ http://perso.telecom-paristech.fr/~gramfort/liesse_python/1-Intro-Python.html

⋆⋆⋆ http://perso.telecom-paristech.fr/~gramfort/liesse_python/2-Numpy.html

⋆⋆⋆ http://perso.telecom-paristech.fr/~gramfort/liesse_python/3-Scipy.html

⋆⋆⋆ http://scikit-learn.org/stable/index.html

⋆⋆ http://www.loria.fr/~rougier/teaching/matplotlib/matplotlib.html

⋆⋆ http://jrjohansson.github.io/

NLTK documentation : http://www.nltk.org/

Lab based on https://github.com/scrapinghub/python-crfsuite/blob/master/examples/CoNLL\202002.ipynb