CS 419/519 Document Analysis
Lab and Assignment 2: Information Extraction

October 12, 2015

This lab and assignment involves creating a simple search engine, evaluating it to understand its performance, and making changes to improve performance where you can.

- Maximum marks: 10
- Programing language: Java (only)
- Assignment questions: CS419/519 Canvas Site
- Deadline: Wed Oct 28 @ 2pm (lab attendance mandatory for discussing answers)

Marking scheme and requirements: Full marks will be given for (1) working, readable, reasonably efficient, documented code that achieves the assignment goals and (2) for providing appropriate answers to Q1-Q3 in a file `ANSWERS_lastname.pdf` submitted with your solution.

Please adhere to the collaboration policy on the course website – people you discussed the assignment solution with, or websites with source code you used should be listed in `ANSWERS_lastname.pdf`.

To verify that your code runs end-to-end, you may be asked to make some changes to your search engine during the evaluation lab and then re-evaluate results.

What/how to submit your work: Please submit `ANSWERS_lastname.pdf` and all your source code, zipped into a single file called `Assignment1_lastname.zip` and submit to **Canvas**. *Please do not submit data or other large binary files — Canvas will not accept large submissions.*

Information Extraction: Programming

1 Before the Introductory lab

In the lab you will familiarize your self wityh two pieces of software: CRF++, a classic CRF toolkit for making Name Entity Recognition task, and OpenNLP, a Java library for variety Natural Language Processing tasks. *It's important that you pre-build all of this software before the lab (and install all requisite software to do this)* so that instructors can answer questions relevant to the actual lab during the lab time.

All code should be provided in the github repository that contains this assignment handout.

The code directory includes a README file with instructions for getting started. It is suggested to use an IDE such as Eclipse, IntelliJ, etc. Make sure you can run the classes listed in the README and obtain error-free output.

The CRF++ bundle also contains a README. This software is distributed as C++ source. To build an executable, you'll need "gcc" and "make" installed. You can also download windows version online, which provide executable files. Once you've installed executable, type crf_learn to verify correctly installed. More details of training model and testing can be found in README file of software bundle.

To evaluate test result, you need to use command in terminal:
$perl\ conll - eval.pl\ -d\ '\ \backslash t'\ <\ teste\_result.$

In the lab, you will be required to train, test and evaluate in java. So here is knowledge you have to know to embed terminal command into java:

```
Process proc = rt.exec("/path/crf_learn template eng_train.data model");
proc.getOutputStream().close();
InputStream stdin = proc.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(stdin));
String currentLine = "";
while((currentLine = br.readLine()) != null ){
System.out.println(currentLine);
}
```

2 In the Introductory lab

By the end of the Introductory lab, you should be able to improve Name entity recognition performance through modifying crf++ template, and extract article from news websites. Find sample data from the "DATA" folder. This contains three things:

- conll2002 spanish NER data set, include one training data and two testing data, in the conll2002 directory.
- A sample terminal line command and its output.
- A pl file that you can use to evaluate result.

Note: you need a CRF++ template to train the model, but we don't provide that for you. Please check software instruction in "CRF++-0.58/doc/index.html" to see how to modify template.

After you figured out how to improve NER performance through modifying template, you need to extract a article directly from a news website and then transform it into CRF++ acceptable format. Please read "/code/NER/web/reader/WebPageReader.java"

1. find WebPageReader construct function to understand how a document is read into XHTML and XML DOM
2. find XMLUtils.PrintNode function to understand how how it is printed out
3. find getXPathQueryResults function to understand how XPath queries can be run on it

Once you understand how CRF++ and code work from running examples, your task is to use java code to train a model test it on a web article. Output all name entities line by line.

- You need to extract article from web and transform it into CRF++ acceptable format.
- You may need to modify training data structure that allows you include more features, such as boolean capitalized word detection columns.
- Produce output in the format as shown below:

```
[1]:Barack Obama
[2]:White House
[3]:San Francisco
...
```

3 Main Assignment (Assessed in Evaluation Lab)

In the Introductory lab section, you were asked to get CRF++ to train and test on conll2002 data, run perl to evaluate, and show the evaluation results. You were also asked to modify template file to improve the performance.

In this part of the assignment, you will run java code to extract a web article and extract name entities from it. You will also need to modify the training data structure to get a better trained model and use that model to classify the web article. Note: you may also need to modify the output structure of extracted article.

The data you need should be available on Canvas. The data is Conll2003, which has one four columns training data and two four columns testing data.

Submit your code, the NER model and the feature template. Grading will be based on your implementation (2 pt.), on the performance results of your classifier (2 pt.), on your feature template (1 pt.), the correct output format (1 pt.).

Information Extraction: Written

For this section, simply submit your answers in your `ANSWERS.pdf` file.

Q1 [1 pts]. CRF++ Evaluation
The performance of CRF on name entity recognition depends on what kind of features is included in consideration. In CRF++, template file define the features. You were asked to modify it to improve the over all performance.

- What features, window size did you use to train the model? Why?
- What is impact of bigrams?

Q2 [1 pt]. Web article extraction To extract an article from webpage, you have to understand the XHTML structure. In XHTML, the article is embedded by several layers of HTML tags, which requires recursively search.

- Why should we transform HTML webpage into XML format?
- How can XPath queries be run on XHTML and XML DOM?

Q3 [2 pt]. Name Entity Recognition You were asked to modify the training data structure to get a better trained model and use that model to classify the web article.

- How do you separate sentence in CRF++ acceptable format? Why is that important?
- Is the performance improved after your modification? Please show baseline model result and modified model result.
- What format of training and testing data is more effective to make NER classification.