AFP 2016 – COURSE WORK ASSIGNMENT

Named entity recognition (NER) is one of the first steps in most information extraction tasks aimed at finding and classifying text elements into certain categories, most commonly names of people, places and organizations, time expressions, quantities, monetary values, percentages, etc. For example:

30-year-old Howard signed a $88 million contract with the Rockets in 2013. -> 30-year-old [$_{PERS}$ Howard] signed a [$_{MONEY}$ $88 million] contract with the [$_{ORG}$ Rockets] in [$_{TIME}$ 2013].

In this course work you should implement and evaluate a simple rule-based NER system. The usual approach to NER is based on the statistical sequence labeling techniques, but your programme does not need to involve machine learning or statistical methods – it may use heuristic rules and external data / information sources or files. The texts that will be given to your programme are in the English language.

You should use the following tags:

1. ENAMEX for names of:

- people (e.g. <ENAMEX TYPE="PERSON">Steve Jobs</ENAMEX>)
- places (e.g. <ENAMEX TYPE="LOCATION">California</ENAMEX>)
- organisations (e.g. <ENAMEX TYPE="ORGANIZATION">Apple Inc.</ENAMEX>)

2. TIMEX for:

- date (e.g. <TIMEX TYPE="DATE">1976</TIMEX>,

  <TIMEX TYPE="DATE">Thursday</TIMEX>,

  <TIMEX TYPE="DATE">last year</TIMEX>,
- time (e.g. <TIMEX TYPE="TIME">14:50</TIMEX>,
  <TIMEX TYPE="TIME">2 p.m. EST</TIMEX>

3. NUMEX for:

- money<NUMEX TYPE="MONEY">$500 million</NUMEX>

Your recognition system will be tested against recognizing the following:

1. Date, time and money expressions. Take care that your expressions are greedy enough (e.g. “the end of 1996” and “1996” are not the same expressions).

2. Locations. You’ve probably made some lists for recognising time/date expressions. You can use larger lists for locations called gazetteers. One suggestion is www.geonames.org, but you can use another source.

3. Person names. Lists of first names and surnames can still be helpful. You can try www.census.gov. Think of other indicators of person names, list them in your report.

4. Organisations. Implement some obvious rules that recognise names of companies. What other heuristics can you suggest?

When your programme is tested, it will be given a file (*untagged.txt*) and a reference solution (*tagged.txt*) also called as the “gold-standard corpus” for this task (meaning that results of the related input can be benchmarked with *tagged.txt*). *tagged.txt* contains named entities marked up according to the conventional XML style format described above.

Your programme shall annotate the text in *untagged.txt* using the format described above and put the result into *result.txt* file. Then your programme shall compare *result.txt* and *tagged.txt*, calculating the recall, precision and $F1$ measure as explained below, and print those to the screen.

Your programme should be named "coursework.hs". It should not take any arguments when started.

The statistics to be produced by your programme are:

- recall (the ratio of the number of correctly labelled entities to the total that should have been labelled);
- precision (the ratio of the number of correctly labelled entities to the total labelled);
- $F1$ measure (the weighted average of the precision and recall); the formula is

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Your recognition doesn't need to have a recall of 100%. Describe each step that your programme performs in the report, to be submitted with your programme.

Your submission should contain the main Haskell code, a report explaining your solution, and additional files if you use them (if they can be downloaded from the web just provide links).

You are given sample input files containing tagged and untagged sentences (taken from different texts). If your programme can manage those input files, it will be sufficient, but other files will also be used in testing. Note that you should implement a genuine named entity recognition, not a one that can only solve the example files.