PFL054 2007/08

Term project

Named entity type classification

Named Entity (NE)

A named entity is a word or a sequence of words that can be classified as a name of a person, a geographical place, a product, a company, time etc.

The annotation of texts according to the well prepared methodology is needed to study NEs. For illustration, the MUC, MUC-6 and BBN hierarchies of named entity types have been proposed. In our task, we will work with the hierarchy proposed by Magda Ševčíková and her colleagues. The short description is provided in the enclosed `NE_methodology.pdf file`; see the technical report TR-2007-36 *Zpracování pojmenovaných entit v českých textech*. for more details.

Example: *Dnes sehrají fotbalisté* ***Slavie*** *na* ***Strahově*** *od* ***17.30*** *hodin utkání* ***Interpoháru*** *s* ***Bayerem Leverkusen****, v jehož barvách by se měl představit i bývalý olomoucký útočník* ***Pavel Hapal***.

Slavie	"Institutions"	ic
Strahově	"Geographical"	gq
17.30	"Time"	th
Interpoháru	"Institutions"	ia
Bayerem Leverkusen	"Institutions"	ic
Pavel	"Personal"	pf
Hapal	"Personal"	ps

Data

Data are split into the train, development and evaluation part: `train*xml`, `dtest*xml` and `etest*xml`, respectively.

The `[train,dtest,etest].m.xml` consist of texts from the Czech National Corpus automatically annotated on the morphological layer according to the Prague Dependency Treebank annotation guidelines (`http://ufal.mff.cuni.cz/pdt2.0`). For each word token, word form, lemma (`http://ufal.mff.cuni.cz/rest/CAC/doc/cac-guide/eng/html/chapter11.html`) and morphological tag (`http://ufal.mff.cuni.cz/rest/CAC/doc/cac-guide/eng/html/chapter12.html`) are listed. Each sentence and each word token are uniquely identified.

Example:

```
...
<s id='train-s86'>
<m id='train-s86m1'>
  <form>Dnes</form>
  <lemma>dnes</lemma>
  <tag>Db-------------</tag>
</m>
<m id='train-s86m2'>
  <form>sehrají</form>
  <lemma>sehrát</lemma>
  <tag>VB-P---3P-AA--1</tag>
</m>
```

```
 ...
 ...
 <m id='train-s86m27'>
   <form>Pavel</form>
   <lemma>Pavel-1_;Y</lemma>
   <tag>NNMS1-----A----</tag>
 </m>
 <m id='train-s86m28'>
   <form>Hapal</form>
   <lemma>Hapal_;S</lemma>
   <tag>NNMS1-----A----</tag>
 </m>
 <m id='train-s86m29'>
   <form>.</form>
   <lemma>.</lemma>
   <tag>Z:-------------</tag>
 </m>
 </s>
 ...
```

The `[train,dtest,etest].ne.oneword.xml` files consist of info on the one-word named entities. Link to the morphological info (lemma and tag) is uniquely determined by the word token id.

Example:

```
...
<ne type='ic' start='train-s86m4' end='train-s86m4'/>
<ne type='gq' start='train-s86m6' end='train-s86m6'/>
<ne type='th' start='train-s86m8' end='train-s86m8'/>
<ne type='ia' start='train-s86m11' end='train-s86m11'/>
<ne type='pf' start='train-s86m27' end='train-s86m27'/>
<ne type='ps' start='train-s86m28' end='train-s86m28'/>
...
```

Goal

Simplification: We will restrict the task of named entity type classification into a classification of ONE-WORD NE type – no other NEs (for ex. multi-word NE (*Bayerem Leverkusen*), addresses) are present in the data.

Classify NE type using the info available. Do not detect NEs in the data – you are provided with the data containing NEs already detected. Apply at least two methods discussed during the course. Design of the features needed by a given method is up to you. The R environment is strongly recommended for the project solving.

You can play with the NE hierarchy types in two ways. Either design more general one-level classification ("Personal"-p, "Institutions"-i, "Time"-t etc.), or design more refined two-level classification ("first names"-pf, "surnames"-ps etc.). In special cases (for ex. you are half-way through and the results call for change;-)) it is allowed to define new NE types (for ex. by merging two original NE types). **Note**: Redefining is possible, but not necessary.

Example:

- Merge "Specific number usages"-n & "Quantitative expressions"-t into one class.
- Merge "first names"-pf & "surnames"-ps into one class and the remaining "Personal"-p[^fs] types into one class.

Train methods on the training data ENTIRELY. Use the development data if you need to train an additional parameter. The results on the evaluation data are the most important ones. Their comparison with the results on the training data will be useful as well.

Note: More NE types

The NE types listed below are not specified in the `NE_methodology.pdf` file but they are present in the data. Handle them in the way described above.

f	word from foreign language
segm	wrong lowercase instead of uppercase caused by wrong segmentation of the text
cap	word written in CAPITALS, mainly beacause of typographical reasons
lower	wrong lowercase instead of uppercase
upper	wrong uppercase instead of lowercase
?	unspecified type

Helpdesk

The `counts_train_data.log` and `counts_all_data.log` files contain the numbers of NE type concurrencies in the train and the all data respectively. These files are generated by the following commands:

```
cut -f2 -d' ' train.ne.oneword.xml |grep 'type'|sort|uniq -c >
counts_train_data.log
cut -f2 -d' ' *.ne.oneword.xml |grep 'type'|sort|uniq -c >
counts_all_data.log
```

Deadlines

December 14, 2007, 24:00

You will need to turn in electronically to {hladka,ribarov,schlesinger}@ufal.mff.cuni.cz

- a **programming** (R code)
- a **short report** describing methods, results and comments. The short report for the project should be 1 page (A4) in length, excluding figures.

December 17, 2007, 10:40

- 10 minutes presentation during the seminar (On Monday December 17, the seminar will start at 10:40 and the lecture at 9:00.)

January 13, 2008, 24:00

You will need to turn in electronically to {hladka,ribarov,schlesinger}@ufal.mff.cuni.cz

- a **final programming** (R code)
- a **final report** written according to the guidelines specified at the http://ufal.mff.cuni.cz/~hladka/ML.html -> Project

Pavel Schlesinger will consult with you on the problems you will meet while doing the project. E-mail him to make appointment.