Information Extraction Exercise III

Sequence Classification

Fabian Dreer

Centre for Information and Speech Processing, LMU München

dreer@cip.ifi.lmu.de

February 2017

Objective

We want to extract beginnings of locations from a collection of seminar announcements.

This shall be done with sequence classification.

Overview

Setup

- Download the exercise tarball from
  `http://www.cip.ifi.lmu.de/~dreer/IE_exercise3.tgz`
- Unpack it with
  `$ tar -xzf IE_exercise3.tgz`
- Change into the directory and prepare the setup with
  `$ make`
- Now you should be good to go.

Comparison

Use
`$@diff -b --context=2 seq_dev.txt tmp_check_dev`
to see a diff of your results and the gold standard.

The lines with the `!` are the differences. Make an error analysis.
Try the `-y` option for a side by side view.

Data Sets

This time we are going to use four different data sets:

- **TRAIN** for training the model.
- **DEV** for choosing the right model and parameter optimization.
- **TEST** held-out data for tests.
- **BLINDTEST** for the final score after the model is fixed. Special script: `run_blindtest.sh`

Data-Header

```
<0.26.4.95.11.09.31.hf08+@andrew.cmu.edu.0>
Type:       cmu.andrew.academic.bio
Topic:      "MHC Class II: A Target for Specific
            Immunomodulation of the Immune Response"
Dates:      3-May-95
Time:       <stime>3:30 PM</stime>
Place:      <location>Room 406A</location>
PostedBy:   Helena R. Frey on 26-Apr-95
            at 11:09 from andrew.cmu.edu
Abstract:
```

Data-Text

The text of the announcement could for instance look like:

```
<paragraph><sentence>The lecture is scheduled for
<stime>5:00 p.m</stime>. on Thursday, March 30, in
<location>Doherty Hall, Room 2315</location>
</sentence>.  </paragraph>
```

Files

Your exercise directory gets crowded with many files.

These are simply representing the splits into *training*, *development*, *test* and *blindtest* sets.

Respectively the input data to *Wapiti* generated from these files.

RUN_*.SH

No need for modifications.

Most often you're going to run
`bash ./run_seq_train_test.sh`

This time we're using the `BIEWO` format.

EXTRACT_SEQ_004.(PY|PL)

Feature Extractor

In this file you can add features by appending them to the output line. If you prefer the perl version, change the script in `run_seq_train_test.sh` as well as in `run_blindtest.sh`.
The output could look like:

```
START_OF_FILE NULL Type:  0
NULL Type:  cmu.andrew.academic.bio 0
Type:  cmu.andrew.academic.bio Topic:  0
cmu.andrew.academic.bio Topic:  "MHC 0
```

Abstract Feature Format

The lines produced by the extractor must look like:

```
Feature0 Feature1 Feature2 [Label]
```

where `Label` is printed by the line

```
print(cl,end='\n\n')
```

Wapiti Input

Wapiti reads one feature per column separated by whitespace. Be careful to have only a single space as separator.

column	0	1	2
description	token	capitalized	isPunctuation
concrete example	Mellon	upperM	NOM

In sequence classification there is no need to put previous/next tokens in as explicit features. We do have access to the whole sequence.

B_OFFSET_PATTERN.TXT

To let *Wapiti* use the new feature you need to register it here.

The first character must be one of: `u` (unigram), `b` (bigram), `*` (both). After a colon, the *observation string* follows `%x[column,offset]`

```
u1:%x[0,0] #-> 1:Place:
u2:%x[0,1] #-> 2:Mellon
u3:%x[0,2] #-> 3:Institute
```

But this time we have access to the full sequence of features!

B_OFFSET_PATTERN.TXT (CONT.)

Example:

```
      abstract                        concrete
  a1      b1      c1    #e.g.    Doherty upperD NOD
  a2      b2      c2    #e.g.    (       NO(    punct(
  a3      b3      c3    #e.g.    Large   upperL NOL
```

Pattern `u:%x[-1,0]/%x[+1,2]` applied at position 2 produces the feature representation: `u:a1/c3`.

The full documentation is available at:
`https://wapiti.limsi.fr/manual.html`

REMOVE_COL_NOWARN.PL

This script removes the classification (0/1) for the *diff*.

Known Bugs

Wapiti seems to have problems with features that have few possible values. For example a feature that indicates capitalization gives an F1-score of $-nan$.

A workaround is to concatenate the first character of the token to the feature indicator (tested), or to concatenate a random number to the indicator (untested).