Language Understanding Systems

Sequence Labeling with CRF

Evgeny A. Stepanov

SISL, DISI, UniTN & VUI, Inc.
evgeny.stepanov@unitn.it

Conditional Random Fields (CRF)

- discriminative approach to sequence labeling
- trained for a specific task (accurate)
- allows adding features without making additional independence assumptions
- training time is increased because of complex optimization procedure

CRF++

Open source implementation of Conditional Random Fields for segmenting/labeling sequential data.

Link

`http://taku910.github.io/crfpp/`

- Download
- Compile
- Install

CRF++: CoNLL Data Format

- token-per-line format
- empty line for as EOS (End-Of-Sentence)
- feature per column: token, POS-tag [**fixed** number!]
- label is always the **last** column

```
who       WP     O
plays     VBZ    O
luke      NN     B-character.name
on        IN     O
star      NN     B-movie.name
wars      NNS    I-movie.name
new       JJ     I-movie.name
hope      NN     I-movie.name
```

CRF++: Template File

```
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-1,0]/%x[0,0]
U06:%x[0,0]/%x[1,0]

U07:%x[0,0]/%x[0,1]

# Bigram
B
```

- One *feature template* per line
- `%x[row,column]`
  - *row* relative positions w.r.t. **current token**
  - *column* absolute position of the column

CRF++: Feature Templates

```
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-1,0]/%x[0,0]
U06:%x[0,0]/%x[1,0]

U07:%x[0,0]/%x[0,1]

# Bigram
B
```

```
who     WP    O
plays   VBZ   O
luke    NN    B-character.name << CURRENT
on      IN    O
star    NN    B-movie.name
wars    NNS   I-movie.name
new     JJ    I-movie.name
hope    NN    I-movie.name
```

```
# Template                : Feature
U00:%x[-2,0]              : who
U01:%x[-1,0]              : plays
U02:%x[0,0]               : luke
U03:%x[1,0]               : on
U04:%x[2,0]               : star
U05:%x[-1,0]/%x[0,0]      : plays/luke
U06:%x[0,0]/%x[1,0]       : luke/on

U07:%x[0,0]/%x[0,1]       : luke/NN
```

CRF++: Template Types

- CRF++ automatically generates feature function using macros.
- There are 2 kinds of templates: *Unigram* and *Bigram*
- The ngram is w.r.t. the previous output label
  - unigram (U): output_tag * (all possible strings expanded with a macro)
  - bigram (B) : output_tag * output_tag * (all possible strings expanded with a macro)

Using CRF++: Tools

Training

```
crf_learn template_file train_file model_file
```

Testing

```
crf_test -m model_file test_files ...
```

CRF++: Some Training Options

```
Usage: crf_learn [options] files
 -f, --freq=INT                 use features that occuer
                                no less than INT(default 1)
 -a, --algorithm=(CRF|MIRA)     select training algorithm
 -p, --thread=INT               number of threads
                                (default auto-detect)
 -v, --version                  show the version and exit
 -h, --help                     show this help and exit
```

See the rest with `crf_learn -h`

CRF++: Testing Options

```
Usage: crf_test [options] files
 -m, --model=FILE          set FILE for model file
 -n, --nbest=INT           output n-best results
 -v, --verbose=INT         set INT for verbose level
 -c, --cost-factor=FLOAT   set cost factor
 -o, --output=FILE         use FILE as output file
 -v, --version             show the version and exit
 -h, --help                show this help and exit
```

CRF++: Evaluation

CoNLL evaluation script:

`https://github.com/tpeng/npchunker/blob/master/conlleval.pl`

Usage

```
conlleval.pl [-d delimiterTag] [-o oTag] < file
```

Options

- **r**: raw tags (without B-, I-)
- **d**: alternative delimiter tag (default is single space)
- **o**: alternative outside tag (default is `O`)

- Modify script or data for sentence boundary (default '`-X-`')
- Understanding results: Evaluation slides

Alternative CRF Implementation: CRFSuite

- http://www.chokkan.org/software/crfsuite/
- https://python-crfsuite.readthedocs.io/
- https://sklearn-crfsuite.readthedocs.io/

CRF++: Exercises

- Download data from `https://github.com/teropa/nlp/tree/master/resources/corpora/conll2000`
- Train & Evaluate Chunker
  - vary window size: 1,2
  - ngrams: 1,2,3
  - label bigrams: with or without

1. unigram model in window ± 1
2. unigram model in window ± 2
3. bigram model in window ± 1
4. ...