

Language Understanding Systems

Sequence Labeling with CRF

Evgeny A. Stepanov

SISL, DISI, UniTN

`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	0
plays	VBZ	0
luke	NN	B-character.name
on	IN	0
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      0
plays    VBZ     0
luke     NN      B-character.name << CURRENT
on       IN      0
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 ...
```



```
-f, --freq=INT          use features that occur
                        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

<code>-m, --model=FILE</code>	set FILE for model file
<code>-n, --nbest=INT</code>	output n-best results
<code>-v, --verbose=INT</code>	set INT for verbose level
<code>-c, --cost-factor=FLOAT</code>	set cost factor
<code>-o, --output=FILE</code>	use FILE as output file
<code>-v, --version</code>	show the version and exit
<code>-h, --help</code>	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 0)
- Modify script or data for sentence boundary (default '-X-')
- **Understanding results**: Evaluation slides

CRF++: Exercises

- Download data from
<http://www.cnts.ua.ac.be/conll2000/chunking/>
- 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 ...