Natural Language Research Group

Session requirements

PoS Models

Free Time

Introduction to Human Language Technologies
Lab.4: Part of Speech

Natural Language Research Group

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Course 2018/19

Outline

1 Session requirements

2 PoS Models
- HMM
- TnT
- Perceptron
- CRF
- Saving models
- Exercise

3 Free Time

Session requirements

Pen treebank corpus:

- Both Linux & Windows (via python shell)

```
> import nltk
> nltk.download('treebank')
```

dill & crf:

- Linux (via shell)

```
> pip3 install python-crfsuite
> pip3 install dill
```

- Windows (via cmd)

```
> pip install python-crfsuite
> pip install dill
```

No attached resources.

Outline

1 Session requirements

2 PoS Models
- HMM
- TnT
- Perceptron
- CRF
- Saving models
- Exercise

3 Free Time

PoS models

Different options:

- Use the default POS tagger (averaged perceptron) or a predefined one
- Learn a POS tagger
  - Statistical: HMM, TnT, perceptron, CRF (requires `pip3 install python-crfsuite`)
  - Rule based: Brill
- Use third-parties' code

  Senna, Stanford, hunpos

HMM in NLTK

Example:

In [1]:
```
from nltk.tag.hmm import HiddenMarkovModelTrainer
from nltk.corpus import treebank

train_data = treebank.tagged_sents()[:30]
test_data = treebank.tagged_sents()[3000:]

trainer = HiddenMarkovModelTrainer()
HMM = trainer.train_supervised(train_data)

'accuracy:' + str(round(HMM.evaluate(test_data), 3))
```

Out[1]:
```
'accuracy:0.106'
```

In [2]:
```
HMM.tag(['the', 'men', 'attended', 'to', 'the', 'meetings'])
```

Out[2]:
```
[('the', 'DT'),
 ('men', 'NNP'),
 ('attended', 'NNP'),
 ('to', 'NNP'),
 ('the', 'NNP'),
 ('meetings', 'NNP')]
```

TnT in NLTK

Example:

In [1]:
```
from nltk.corpus import treebank
from nltk.tag import tnt

train_data = treebank.tagged_sents()[:30]
test_data = treebank.tagged_sents()[3000:]

TnT = tnt.TnT()
TnT.train(train_data)

'accuracy: ' + str(round(TnT.evaluate(test_data), 3))
```

Out[1]:
```
'accuracy: 0.457'
```

In [2]:
```
TnT.tag(['the', 'men', 'attended', 'to', 'the', 'meetings'])
```

Out[2]:
```
[('the', 'DT'),
 ('men', 'NNS'),
 ('attended', 'Unk'),
 ('to', 'TO'),
 ('the', 'DT'),
 ('meetings', 'Unk')]
```

Perceptron in NLTK

Example:

In [1]:
```
from nltk.tag.perceptron import PerceptronTagger
from nltk.corpus import treebank

train_data = treebank.tagged_sents()[:30]
test_data = treebank.tagged_sents()[3000:]

PER = PerceptronTagger(load=False)
PER.train(train_data)

'accuracy: ' + str(round(PER.evaluate(test_data), 3))
```

Out[1]:
```
'accuracy: 0.651'
```

In [2]:
```
PER.tag(['the', 'men', 'attended', 'to', 'the', 'meetings'])
```

Out[2]:
```
[('the', 'DT'),
 ('men', 'NN'),
 ('attended', 'RB'),
 ('to', 'TO'),
 ('the', 'DT'),
 ('meetings', 'NN')]
```

CRF in NLTK

Example:

In [1]:
```
from nltk.tag import CRFTagger
from nltk.corpus import treebank

train_data = treebank.tagged_sents()[:30]
test_data = treebank.tagged_sents()[3000:]

CRF = CRFTagger()
CRF.train(train_data,'crf_tagger_model')

'accuracy: ' + str(round(CRF.evaluate(test_data), 3))
```

Out[1]:
```
'accuracy: 0.685'
```

In [2]:
```
CRF.tag(['the', 'men', 'attended', 'to', 'the', 'meetings'])
```

Out[2]:
```
[('the', 'DT'),
 ('men', 'NN'),
 ('attended', 'VBD'),
 ('to', 'TO'),
 ('the', 'DT'),
 ('meetings', 'NNS')]
```

Saving & loading models

Save/Load a learned model:

- CRF uses their own.
  - Training and save: `CRF.train(train_data,"file_name")` as saw before
  - Load: `CRF.set_model_file("file_name")`
- HMM, Perceptron and TnT can use `dill`. Perceptron and TnT can also use `pickle` using, in both cases, `dump` and `load` functions.

Example:

```python
import dill

# saving
with open("tnt_treebank_pos_tagger", "wb") as f:
    dill.dump(TnT, f)

# loading
with open("tnt_treebank_pos_tagger", "rb") as f:
    TnT = dill.load(f)
```

Mandatory exercise

1. Consider Treebank corpus. Train HMM, TnT, perceptron and CRF models using the first 500, 1000, 1500, 2000, 2500 and 3000 sentences. Evaluate the resulting 24 models using sentences from 3001.

2. Provide a figure with four learning curves, each per model type (X=training set size; Y=accuracy). Which model would you select? Justify the answer.

Upload the jupyter file of the exercise to the Raco.

Outline

1 Session requirements

2 PoS Models
- HMM
- TnT
- Perceptron
- CRF
- Saving models
- Exercise

3 Free Time

Past optional exercises

Time to work on past optional exercises

1. Session 3: SMS Spam Filtering
2. Session 3: Spelling Corrector
3. Session 2: Language Identifier