IHLT Laboratory

Jordi Turmo
TALP Research Center
turmo@cs.upc.edu

Session 5

Text level in nltk
Lexical level in nltk
Exercise

Text level in nltk library

- Depending on the needs, text can be splitted into sentences before tokenizing or it can be directy tokenized.

  (http://www.nltk.org/_modules/nltk/tokenize.html)

- Standard functions (recommended by nltk):

```
s_list = nltk.sent_tokenize(T, [language='LANG'])
t_list = nltk.word_tokenize(s, [language='LANG'])
```

  LANG can be:

  czech, danish, ducth, english, estonian, finnish, french, german, greek, italian, norwegian, polish, portuguese, slovene, spanish, swedish or turkish

  Transform the text previously when it is a Unicode string:

```
T.decode("utf8")
```

Text level in nltk library

Example:

```
import nltk

T='Men want children. They get relaxed with kids.'
s_list = nltk.sent_tokenize(T)
print("s_list=",s_list)
t_list = [nltk.word_tokenize(s) for s in s_list]
print("t_list=",t_list)
```

```
s_list= ['Women want children.', 'Men want children.']
t_list= [['Women', 'want', 'children', '.'],
         ['Men', 'want', 'children', '.']]
```

Session 5

Text level in nltk
Lexical level in nltk
Exercise

Lexical level in nltk library

- Words

  Tokenization achieves words. Multiwords (e.g. "Even though") are not recognized. `MWETokenizer` is not useful.

- POS tags

  ```
  t_POS_list = nltk.pos_tag(t_list)
  ```

- Lemmas

  ```
  from nltk.stem import WordNetLemmatizer
  wnl = WordNetLemmatizer()
  wnl.lemmatize(token, pos=[POS]
  ```
  POS can be: 'n','v', ...

- Senses

  See session 1 related to WordNet. For the moment, we will take the first synset of the list of synsets per word+POS as the correct sense, until we study WSD

Lexical level in nltk library

Example

```
import nltk
from nltk.stem import WordNetLemmatizer
wnl = WordNetLemmatizer()

def lemmatize(x):
    if x[1][0] in {'N','V'}:
        return wnl.lemmatize(x[0], pos=x[1][0].lower())
    return x[0]

t_list = ['Women','want','children']
t_POS_list = nltk.pos_tag(t_list)
print('t_POS_list=',t_POS_list)
toks = [lemmatize(x) for x in t_POS_list]
print('toks=',toks)
```

```
t_POS_list=[('Women','NNS'),('want','VB'),('children','NNS')]
toks=['Woman', 'want', 'child']
```

Exercise

Read the three first pair of sentences of the training file within the evaluation framework of the project.

Compute their similarities by considering the following approaches:

- words and Jaccard coefficient
- lemmas and Jaccard coefficient
- correct senses and Jaccard coefficient

Which one of these approaches, if any, do you think that could perform better for any pair of texts? Justify the answer.