Introduction à NLTK

INF8460 - TP1

Polytechnique Montréal

Automne 2020

Rappel

Lire un fichier avec Python :

```
with open("your/file/name.txt", "r") as f:
    data = f.read()
```

Écrire dans un fichier :

```
with open("your/file/name.txt", "w") as f:
    f.write(data)
```

Gestion de l'encodage avec le paramètre optionnel `encoding`, et des erreurs avec `errors="strict", "ignore", "replace"`...

Avant de commencer

NLTK 3.4.4 devrait être installé sur vos machines :

```
>>> import nltk
>>> nltk.__version__
"3.4.4"
```

Vous devez télécharger certaines données avant de commencer :

```
import nltk

nltk.download("punkt")
nltk.download("wordnet")
```

Prétraitement

Segmentation et tokenization

Segmenter un texte en phrases : `sent_tokenize`

```
>>> nltk.sent_tokenize("Alice est là. Bob est ici.")
["Alice est là.", "Bob est ici."]
```

Tokenizer une phrase en mots : `word_tokenize`

```
>>> nltk.word_tokenize("Alice est là")
["Alice", "est", "là"]
```

D'autres tokenizers sont disponibles dans le module `nltk.tokenize`.

Lemmatisation

On initialise un lemmatiseur, puis on appelle sa méthode `lemmatize` :

```
>>> lemmzer = nltk.WordNetLemmatizer()
>>> lemmzer.lemmatize("dogs")
"dog"
>>> lemmzer.lemmatize("wolves")
"wolf"
```

Et pour lemmatiser une phrase entière ?

```
>>> tokens = "of mice and women".split()
>>> [lemmzer.lemmatize(token) for token in tokens]
["of", "mouse", "and", "woman"]
```

Stemming

Comme précédemment, on initialise un *stemmer*, puis on appelle sa méthode `stem` :

```
>>> stemmer = nltk.PorterStemmer()
>>> stemmer.stem("decided")
"decid"
>>> stemmer.stem("connections") ==
...     stemmer.stem("connecting")
True
```

D'autres stemmers sont disponibles dans le module `nltk.stem`.

Enlever les *stopwords*

Si besoin, on peut enlever les mots vides : le, la, du, des (*stopwords* en anglais). NLTK fournit une liste de tels mots :

```
>>> nltk.download("stopwords")
>>> from nltk.corpus import stopwords
>>> stopwords.words("french")
['au', 'aux', 'avec', 'ce', 'ces', 'dans', ...]
```

On peut alors les retirer pendant la phase de tokenization :

```
>>> sentence = "ces vacances ont été méritées"
>>> stopwords_fr = set(stopwords.words("french"))
>>> [token for token in nltk.word_tokenize(sentence)
...     if token not in stopwords_fr]
["vacances", "méritées"]
```