

Fouille de textes

TD – Pré-traitements textuels et représentation par sac de mots

Solen Quiniou

`solen.quiniou@univ-nantes.fr`

Master 2 ATAL – Université de Nantes

Année 2020-2021

Plan du cours

- 1 Principaux frameworks pour la fouille de texte
- 2 Exercice pratique en utilisant NLTK et spaCy

Plan du cours

- 1 Principaux frameworks pour la fouille de texte
- 2 Exercice pratique en utilisant NLTK et spaCy

Principaux frameworks pour la fouille de texte

- Le langage Python est le plus couramment utilisé pour faire de la fouille de texte
 - Les bibliothèques les plus connues et les plus couramment utilisés sont Gensim, NLTK et spaCy¹





- 2009
- Tokenization,
- Topic Modeling (LDA, HDP, LSI),
- Stemming,
- Word Embedding.

- 2001
- Tokenization de mots et de phrases,
- POS-Tagging,
- NER,
- Analyse de sentiments,
- Stemming,
- Algorithmes de classifications,
- Corpora de données.

- 2015
- Tokenisation,
- POS-Tagging,
- NER,
- Analyse de sentiments (toujours en développement),
- Lemmatisation,
- Vecteurs de mots pré-entraînés.

1. <https://www.ekino.com/articles/introduction-au-nlp-partie-ii>

Comparaison de NLTK et spaCy²

	⊕ PROS	⊖ CONS
	<ul style="list-style-type: none">+ The most well-known and full NLP library+ Many third-party extensions+ Plenty of approaches to each NLP task+ Fast sentence tokenization+ Supports the largest number of languages compared to other libraries	<ul style="list-style-type: none">- Complicated to learn and use- Quite slow- In sentence tokenization, NLTK only splits text by sentences, without analyzing the semantic structure- Processes strings which is not very typical for object-oriented language Python- Doesn't provide neural network models- No integrated word vectors
	<ul style="list-style-type: none">+ The fastest NLP framework+ Easy to learn and use because it has one single highly optimized tool for each task+ Processes objects; more object-oriented, comparing to other libs+ Uses neural networks for training some models+ Provides built-in word vectors+ Active support and development	<ul style="list-style-type: none">- Lacks flexibility, comparing to NLTK- Sentence tokenization is slower than in NLTK- Doesn't support many languages. There are models only for 7 languages and "multi-language" models

2. <https://activewizards.com/blog/comparison-of-python-nlp-libraries/>

Plan du cours

- 1 Principaux frameworks pour la fouille de texte
- 2 Exercice pratique en utilisant NLTK et spaCy

Mise en place de l'environnement de travail

- Installation de spaCy et des modèles de langue

- ▶ Le site `https://spacy.io/usage` donne toutes les informations nécessaires pour installer spaCy ainsi que les modèles à utiliser, en anglais et en français

- Installation de NLTK

- ▶ Le site `http://www.nltk.org` donne toutes les informations nécessaires pour installer NLTK

- Données à utiliser

- ▶ Récupérez les données présentes sur `https://gitlab.univ-nantes.fr/m2atal.fouille_de_textes/ft_tdl`

Travail à réaliser

❶ Pré-traitements sur des données en anglais : dépêches Reuters

- ▶ Effectuez les pré-traitements vus en cours, avec NLTK, sur les données en anglais : segmentation en phrases et en mots, lemmatisation, étiquetage grammatical (*POS tagging*), racinisation, suppression des mots vides
- ▶ Effectuez les mêmes pré-traitements, avec spaCy, sur ces mêmes données
- ▶ Comparez les résultats obtenus avec les 2 bibliothèques

❷ Pré-traitements sur des données en français : petites annonces Le Bon Coin

- ▶ Effectuez les mêmes pré-traitements que précédemment, avec NLTK, sur les données en français
- ▶ Effectuez les mêmes pré-traitements, avec spaCy, sur ces mêmes données
- ▶ Comparez les résultats obtenus avec les 2 bibliothèques