



IA - langage : NLP, traduction, analyse

IA051

Durée: 2 jours

1 670 €

27 au 28 janvier
28 au 29 avril

1 au 2 septembre
1 au 2 décembre

Public :

toute personne intéressée par le NLP: Ingénieurs, Analystes, Data Scientists, Data Analysts, Data Steward, Développeurs...

Objectifs :

Comprendre les principes du NLP et savoir les mettre en oeuvre avec Python.

Connaissances préalables nécessaires :

Connaissances de base de Python et du deep learning.

Objectifs pédagogiques :

Programme :

NLP Introduction

Objectifs : comprendre le langage humain, et savoir générer des réponses

Différentes étapes : reconnaissance de caractères, ou de la voix, conversion des données en texte, décomposition en éléments de phrase, nettoyage des données, traitement de l'ambiguïté d'un mot, reconnaissance d'une entité nommée (NEM), traitement des multiples références pour une entité, extraction des informations subjectives, etc ...

Les outils de NLP et historique :

outils statistiques, de machine learning, de deep learning,

Watson NLU, Python et le NLTK.

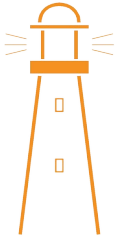
Applications actuelles : solutions de détection de spam, traduction automatique, assistants virtuels, chatbots, analyses d'opinions, de sentiments, etc ...

Python et le NLTK

Introduction : plate-formes supportées, versions de Python,

Présentation des textes et modèles fournis avec le NLTK

Atelier : installation du package NLTK et des datasets



Phirio

Traitements de textes

Etude des différentes fonctions fournies par le NLTK.
Découpage d'un texte en mots ou en phrases, avec `nltk.tokenize()`,
nettoyage de textes avec le filtrage de mots,
stemming avec `nltk.stem`, alertes sur les risques d'un mauvais usage,
étiquetage des différentes parties d'un texte avec `nltk.pos-tag()`,
lemmatisation, pour identifier les formes canoniques des mots,
identification de phrases avec le chunking

Atelier : réalisation d'exemples sur des datasets simples

Analyses de textes

Description de `nltk.ne_chunk()` pour la reconnaissance d'identités nommées,
Présentation des fonctions `concordance()`, `dispersion_plot()`, `FreqDist`

Atelier : Import de corpus de textes, analyse, mise en évidence de
l'utilisation de termes caractéristiques

Etude de cas

Analyse de sentiments avec `nltk.sentiment`
Présentation des fonctions disponibles

Atelier : mise en oeuvre sur un corpus. Utilisation de `polarity.scores()`

Intégration de scikit-learn

Import des algorithmes de classification de scikit-learn

Atelier : exemple d'utilisation des algorithmes de scikit-learn depuis nltk