Traitement de texte de base

Jean Barré

4 février 2025

Doctorant École Normale Supérieure - Université PSL - LaTTiCe

Expressions Régulières

Introduction aux Expressions Régulières

Pourquoi les Expressions Régulières ?

- Utilisées dans presque toutes les tâches de traitement du texte.
- Outil puissant pour la recherche et la transformation de texte.
- Indispensables en prétraitement des données textuelles.
- Utilisées dans l'analyse des données textuelles.
- Employées dans les pipelines de NLP et d'intelligence artificielle.

Syntaxe de Base des Expressions Régulières

Principaux Éléments de Syntaxe

- `.` : Correspond à n'importe quel caractère.
- `*` : Répète zéro ou plusieurs fois.
- `+` : Répète une ou plusieurs fois.
- `?` : Rendu optionnel (zéro ou une occurrence).
- `|` : Alternance entre deux motifs.

Classes de Caractères et Ancres

Classes Utiles

- `\d` : Un chiffre (0-9).
- `\w` : Un caractère alphanumérique.
- `\s` : Un espace.
- `^` : Début de ligne.
- `$` : Fin de ligne.

Exemples d'Utilisation des Regex

Cas d'Usage

- Extraction des dates : `\d{2}/\d{2}/\d{4}`
- Filtrage des adresses email : `\w+@\w+\.\w+`
- Suppression de la ponctuation : `[\^\w\s]`

Les Expressions Régulières : un processus itératif

Deux Types d'Erreurs Courantes

- **Faux Négatifs** : Absence de détection de motifs valides.
  - Exemple : Recherche du mot "le" ne capturant pas "Le".
- **Faux Positifs** : Correspondances incorrectes.
  - Exemple : Détection de "le" dans "atelier" ou "baleine".

Fonctions de Base en Python

rappel

- `re.search()`
- `re.match()`
- `re.sub()`
- `re.findall()`
- `re.split()`

Normalisation du Texte

Problèmes liés à la tokenisation

Exemples

- Ne pas retirer la ponctuation à l'aveugle : *Ph.D., AT&T*
- Gestion des prix : *45.55€*
- Dates : *01/02/06*
- URLs : *https ://psl.eu/*
- Hashtags : *#tall*
- Emails : *someone@ens.psl.eu*
- Clitiques : *je t'aime, l'honneur*
- Expressions multi-mots : *New York, rock 'n' roll*

Tokenisation

```
>>> text = 'That U.S.A. poster-print costs $12.40...'
>>> pattern = r'''(?x)    # set flag to allow verbose regexps
...     ([A-Z]\.)+        # abbreviations, e.g. U.S.A.
...   | \w+(-\w+)*        # words with optional internal hyphens
...   | \$?\d+(\.\d+)?%?  # currency and percentages, e.g. $12.40, 82%
...   | \.\.\.            # ellipsis
...   | [][.,;"'?():-_`]  # these are separate tokens; includes ], [
... '''
>>> nltk.regexp_tokenize(text, pattern)
['That', 'U.S.A.', 'poster-print', 'costs', '$12.40', '...']
```

Figure 1 – Tokenisation par NLTK en 2006

Tokenisation en sous-mots et Byte Pair Encoding

Pourquoi la Tokenisation en Sous-Mots ?

- Les modèles de langue modernes ne traitent pas les mots comme des unités fixes.
- La segmentation en sous-mots permet de mieux gérer :
  - Les mots rares et inconnus.
  - La morphologie des langues agglutinantes.
  - La compression de vocabulaire pour améliorer l'efficacité du modèle.

Byte Pair Encoding (BPE)

Principe du BPE

- Algorithme itératif qui fusionne les paires de caractères les plus fréquentes.
- Réduit la fragmentation excessive du texte tout en conservant une granularité fine.
- Utilisé dans de nombreux modèles de NLP, tels que GPT, BERT et T5.

Exemple de Byte Pair Encoding (BPE) sur ”programmation”

- **Initialisation :** `p r o g r a m m a t i o n`
- **Fusion 1 :** `m m` $\rightarrow$ `mm` $\rightarrow$ `p r o g r a mm a t i o n`
- **Fusion 2 :** `a mm` $\rightarrow$ `amm` $\rightarrow$ `p r o g r amm a t i o n`
- **Fusion 3 :** `r amm` $\rightarrow$ `ramm` $\rightarrow$ `p r o g ramm a t i o n`
- **Fusion finale :** `a t i o n` $\rightarrow$ `ation` $\rightarrow$ `programm ation`

Résultat : Segmentation en `programm` + `ation`

Lemmatisation

Lemmatisation en Français

Représenter tous les mots sous leur lemme

- Transformation en forme canonique = entrée de dictionnaire.
- Exemples :
  - **mange, mangeons, mangé $\rightarrow$ manger**
  - **cheval, chevaux $\rightarrow$ cheval**
  - **je suis, tu es, il est $\rightarrow$ être**
  - **courais, couru, courra $\rightarrow$ courir**
  - **nouveau, nouvelle, nouveaux $\rightarrow$ nouveau**

Pourquoi la Lemmatisation ?

Applications

- Analyse sémantique et stylométrique.
- Recherche et indexation dans les corpus.
- Études lexicographiques et lexicologiques.

Le Cas de l'Ancien Français

Variabilité Graphique

- 36 graphies différentes pour "cheval".
- Influence des substitutions phonétiques et orthographiques.
- Exemples : cheval, cheual, chevaux, chival.

forme	occurr.	forme	occurr.
cheval	785	ceuax	10
cheual	375	cevax	10
chevaus	248	ceuaus	9
ceval	98	chiuau	9
chevax	92	cheuaux	8
chevals	84	kevaus	6
ceual	66	chevau	5
cheuaus	65	cevaux	3
chival	34	chivals	3
chevaux	30	cheuas	2
chivaus	27	keval	2
cheuax	23	chaval	1
chiual	23	chavaux	1
cevaus	19	cheua	1
chevas	19	cheualx	1
cheuals	14	cheuau	1
cevals	12	chevalx	1

Méthodes Modernes : Pie, Spacy, Stanza

Apprentissage Profond

- Utilisation des *word embeddings*.
- Réseaux neuronaux convolutifs et récurrents (CNN et RNN).
- Approches adaptées aux langues non standard.

Types et Tokens

Différences entre Types et Tokens

Définition

- **Token** : Occurrence d'un mot dans un texte.
- **Type** : Une forme unique de mot.
- Le ratio Types/Tokens donne une estimation de la variété linguistique d'un texte.