

Année Universitaire : 2023/2024 Master 2 : SII Module : Recherche d'Information	Université des Sciences et de la Technologie Houari Boumediene Faculté d'Informatique Département d'Intelligence Artificielle et Sciences des Données	TP N°1 Représentation de l'Information : Indexation Partie 1
---	---	---

Support :

1. Extraction automatique des termes

Pour l'extraction automatique de termes, nous pouvons utiliser la méthode `split()` comme suit :

```
>>> Texte = "That D.Z. poster-print costs 120.50DA..."
>>> Termes = Texte.split()
>>> Termes
>>> ['That', 'D.Z.', 'poster-print', 'costs', '120.50DA...']
```

Pour l'extraction automatique de termes, il est recommandé d'utiliser la bibliothèque NLTK (Natural Language ToolKit) avec Python :

```
>>> import nltk
```

Pour l'extraction automatique de termes avec NLTK, il faut définir des expressions régulières à l'aide de la méthode `nltk.RegexpTokenizer()` comme suit :

```
>>> ExpReg = nltk.RegexpTokenizer('\w+') # \w : équivalent à [a-zA-Z0-9_]
>>> Termes = ExpReg.tokenize(Texte)
>>> Termes
>>> ['That', 'D', 'Z', 'poster', 'print', 'costs', '120', '50DA']

>>> ExpReg = nltk.RegexpTokenizer('\w+|(?:[A-Z]\.)+') # ?: nécessaire pour l'utilisation des parenthèses
>>> Termes = ExpReg.tokenize(Texte)
>>> Termes
>>> ['That', 'D', 'Z', 'poster', 'print', 'costs', '120', '50DA']

>>> ExpReg = nltk.RegexpTokenizer('(?:[A-Z]\.)+|\w+')
>>> Termes = ExpReg.tokenize(Texte)
>>> Termes
>>> ['That', 'D.Z.', 'poster', 'print', 'costs', '120', '50DA']

>>> ExpReg = nltk.RegexpTokenizer('(?:[A-Z]\.)+|\w+|\.{3}')
>>> Termes = ExpReg.tokenize(Texte)
>>> Termes
>>> ['That', 'D.Z.', 'poster', 'print', 'costs', '120', '50DA', '...']

>>> ExpReg = nltk.RegexpTokenizer('(?:[A-Z]\.)+|\d+(?:\.\d+)?DA?|\w+|\.{3}') # \d : équivalent à [0-9]
>>> Termes = ExpReg.tokenize(Texte)
>>> Termes
>>> ['That', 'D.Z.', 'poster', 'print', 'costs', '120.50DA', '...']
```

Pour plus de détails sur l'extraction automatique de termes à l'aide de NLTK, veuillez consulter le livre *Natural Language Processing with Python*.

2. Suppression des mots-vides

Pour la suppression des mots-vides, il est recommandé d'utiliser la bibliothèque NLTK (Natural Language ToolKit) avec Python :

```
>>> import nltk
```

Télécharger et installer la liste des mots-vides à l'aide de la méthode `nltk.download()` :

```
>>> nltk.download()
```

Suppression des mots-vides :

```
>>> Texte = "That D.Z. poster-print costs 120.50DA..."
>>> ExpReg = nltk.RegexpTokenizer('(?:[A-Z]\.)+|\d+(?:\.\d+)?DA?|\w+|\.{3}')
>>> Termes = ExpReg.tokenize(Texte)
>>> Termes
>>> ['That', 'D.Z.', 'poster', 'print', 'costs', '120.50DA', '...']
>>> MotsVides = nltk.corpus.stopwords.words('english')
>>> TermesSansMotsVides = [terme for terme in Termes if terme.lower() not in MotsVides]
>>> TermesSansMotsVides
>>> ['D.Z.', 'poster', 'print', 'costs', '120.50DA', '...']
```

3. Normalisation (stemming) des termes extraits

Pour la normalisation des termes extraits, il est recommandé d'utiliser la bibliothèque NLTK (Natural Language ToolKit) avec Python :

```
>>> import nltk
```

Normalisation à l'aide de la méthode `nltk.PorterStemmer()`:

```
>>> Porter = nltk.PorterStemmer()
>>> TermesNormalisation = [Porter.stem(terme) for terme in TermesSansMotsVides]
>>> TermesNormalisation
>>> ['d.z.', 'poster', 'print', 'cost', '120.50da', '...']
```

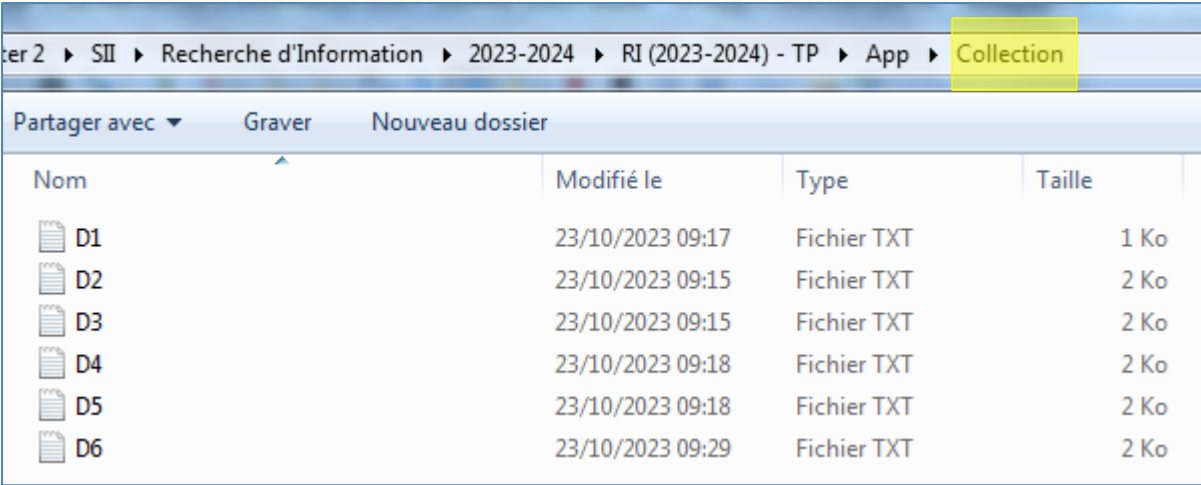
Normalisation à l'aide de la méthode `nltk.LancasterStemmer()`:

```
>>> Lancaster = nltk.LancasterStemmer()
>>> TermesNormalisation = [Lancaster.stem(terme) for terme in TermesSansMotsVides]
>>> TermesNormalisation
>>> ['d.z.', 'post', 'print', 'cost', '120.50da', '...']
```

Exercice :

I. Collection :

Créez un dossier « Collection » contenant un ensemble de documents (voir en pièce jointe). Le ième document est nommé « Di »



The screenshot shows a Windows File Explorer window. The address bar at the top displays the path: "er 2 > SII > Recherche d'Information > 2023-2024 > RI (2023-2024) - TP > App > Collection". The "Collection" folder is highlighted in yellow. Below the address bar, there are buttons for "Partager avec", "Graver", and "Nouveau dossier". The main area of the window displays a table of files within the "Collection" folder.

Nom	Modifié le	Type	Taille
D1	23/10/2023 09:17	Fichier TXT	1 Ko
D2	23/10/2023 09:15	Fichier TXT	2 Ko
D3	23/10/2023 09:15	Fichier TXT	2 Ko
D4	23/10/2023 09:18	Fichier TXT	2 Ko
D5	23/10/2023 09:18	Fichier TXT	2 Ko
D6	23/10/2023 09:29	Fichier TXT	2 Ko

Fig.1 – Collection de documents

II. Création des index :

- . Extraire les termes à l'aide des deux méthodes :

```
split()  
nltk.RegexpTokenizer('expression régulière à définir').tokenize()
```

- . Supprimer les mots vides à l'aide de la méthode :

```
nltk.corpus.stopwords.words('english')
```

- . Normaliser les termes extraits à l'aide des deux méthodes :

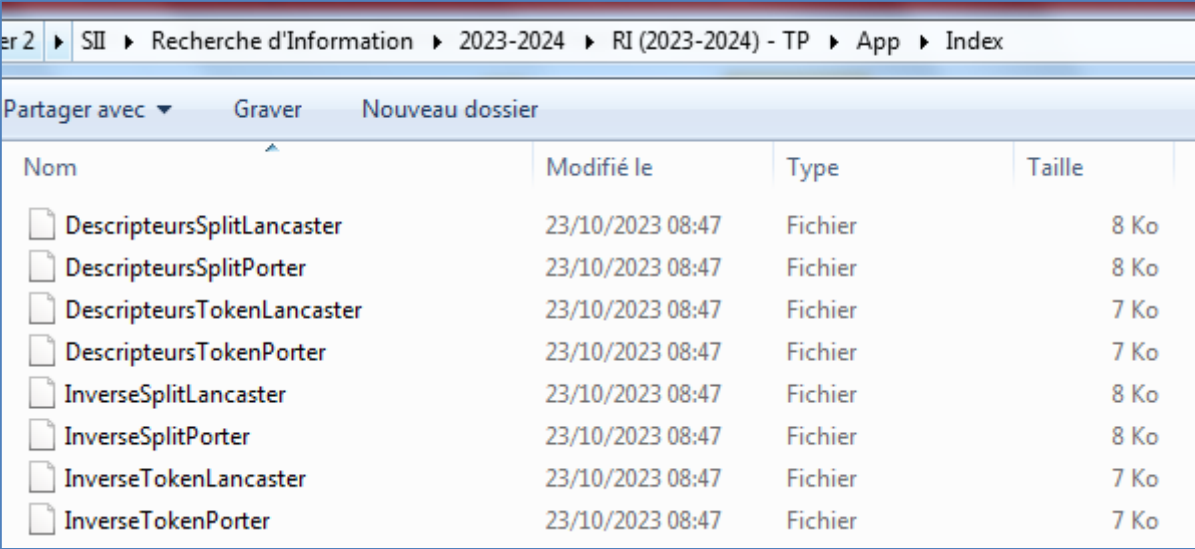
```
nltk.PorterStemmer().stem()  
nltk.LancasterStemmer().stem()
```

- . Créer les fichiers descripteurs, définis comme suit :

```
<N° document> <Terme>
```

- . Créer les fichiers inverses, définis comme suit :

```
<Terme> <N° document>
```











er 2 > SII > Recherche d'Information > 2023-2024 > RI (2023-2024) - TP > App > Index				
Partager avec ▼ Graver Nouveau dossier				
Nom	Modifié le	Type	Taille	
 DescripteursSplitLancaster	23/10/2023 08:47	Fichier	8 Ko	
 DescripteursSplitPorter	23/10/2023 08:47	Fichier	8 Ko	
 DescripteursTokenLancaster	23/10/2023 08:47	Fichier	7 Ko	
 DescripteursTokenPorter	23/10/2023 08:47	Fichier	7 Ko	
 InverseSplitLancaster	23/10/2023 08:47	Fichier	8 Ko	
 InverseSplitPorter	23/10/2023 08:47	Fichier	8 Ko	
 InverseTokenLancaster	23/10/2023 08:47	Fichier	7 Ko	
 InverseTokenPorter	23/10/2023 08:47	Fichier	7 Ko	

Fig.2 – Index à créer

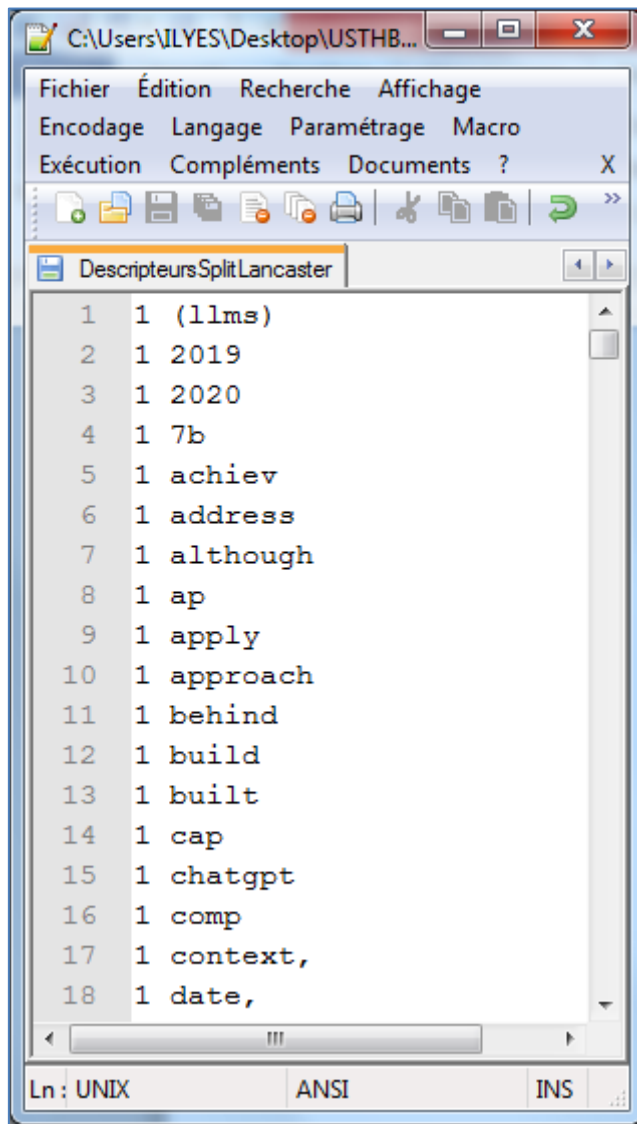


Fig.3 (a) – DescripteursSplitLancaster

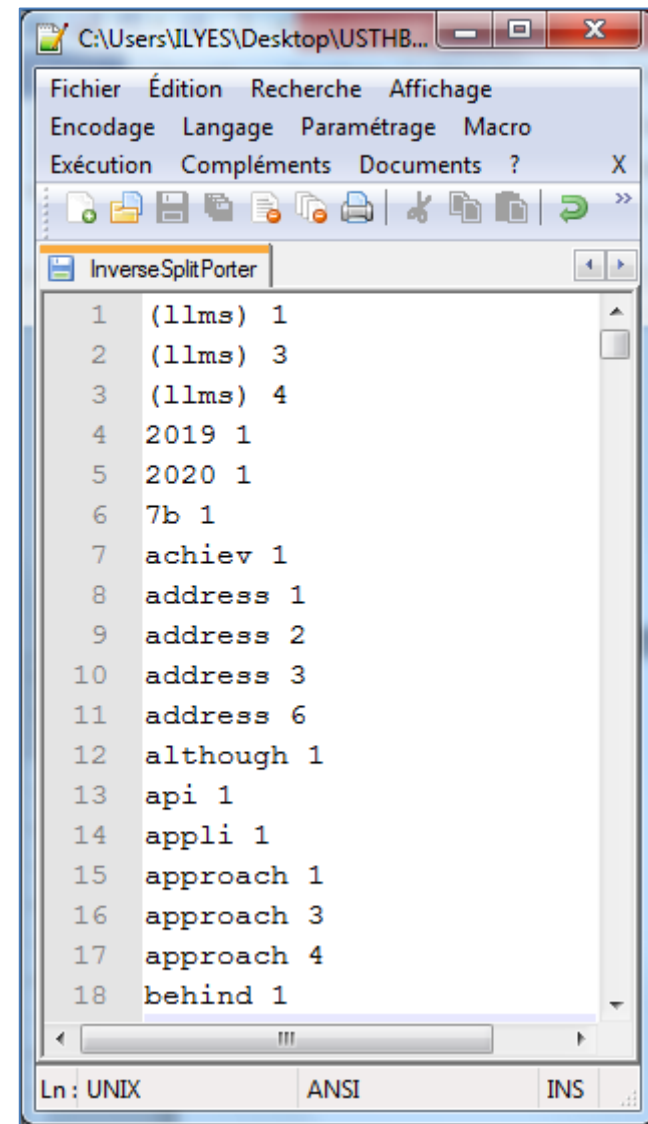


Fig.3 (b) – InverseSplitPorter