

Année Universitaire : 2024/2025 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', '...']
```

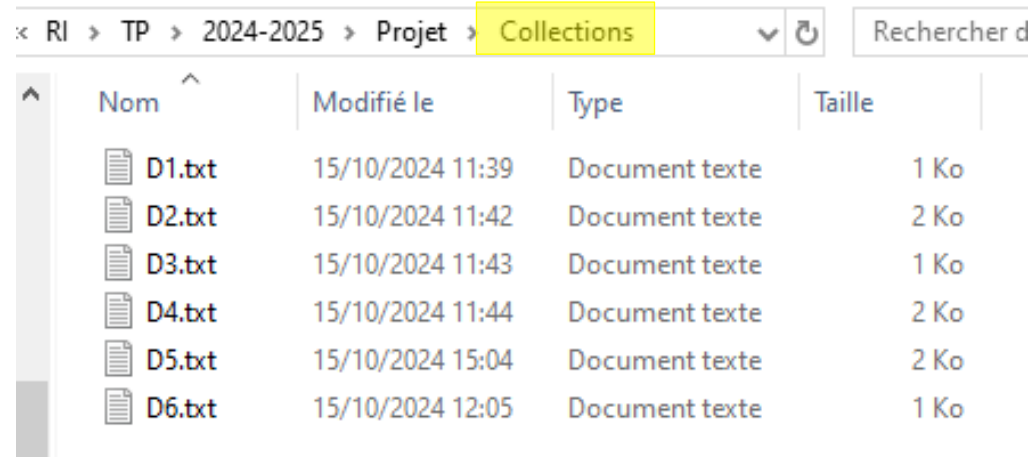
Normalisation à l'aide de la méthode `nltk.LancasterStemmer()`:

```
>>> 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 file explorer interface. The breadcrumb path at the top is « RI » > « TP » > « 2024-2025 » > « Projet » > « Collections ». The 'Collections' folder is highlighted in yellow. To the right of the path is a search bar with the placeholder text 'Rechercher d'. Below the path is a table with four columns: 'Nom', 'Modifié le', 'Type', and 'Taille'. The table contains six rows of document information, each preceded by a document icon. The documents are D1.txt, D2.txt, D3.txt, D4.txt, D5.txt, and D6.txt, all of which are 'Document texte' type.

Nom	Modifié le	Type	Taille
D1.txt	15/10/2024 11:39	Document texte	1 Ko
D2.txt	15/10/2024 11:42	Document texte	2 Ko
D3.txt	15/10/2024 11:43	Document texte	1 Ko
D4.txt	15/10/2024 11:44	Document texte	2 Ko
D5.txt	15/10/2024 15:04	Document texte	2 Ko
D6.txt	15/10/2024 12:05	Document texte	1 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 :

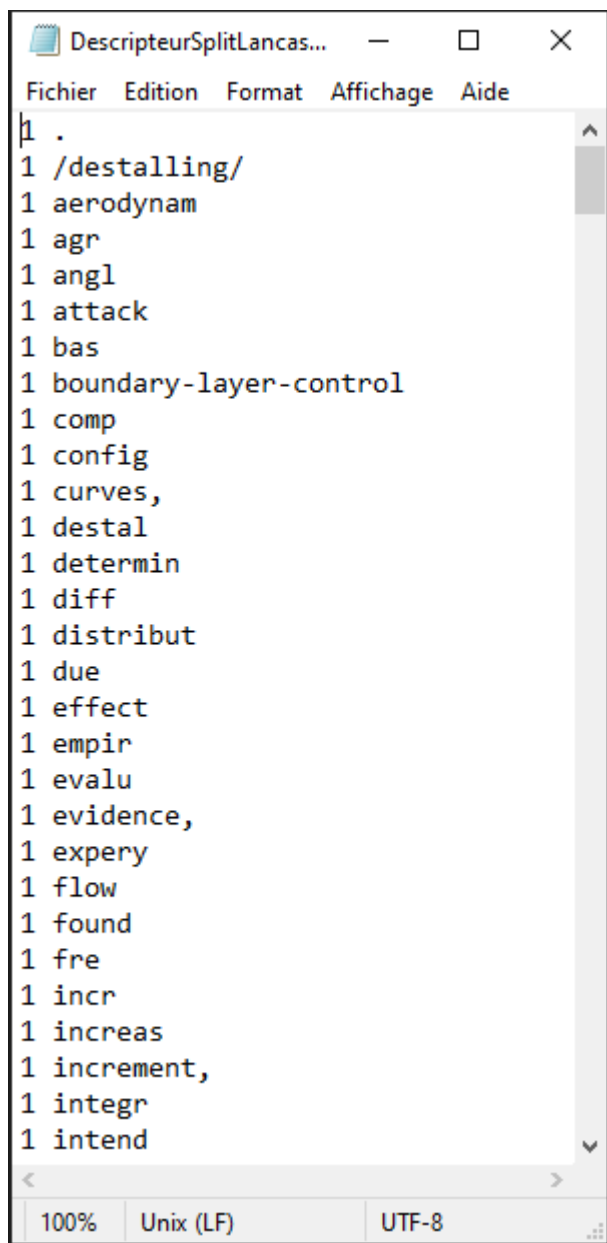
```
<N° document> <Terme>
```

- . Créer les fichiers inverses, définis comme suit :

```
<Terme> <N° document>
```

« RI » TP » 2024-2025 » Projet » Index					Rechercher dans : Index
	Nom	Modifié le	Type	Taille	
	DescripteurSplit	15/10/2024 17:10	Fichier	5 Ko	
	DescripteurSplitLancaster	15/10/2024 17:10	Fichier	4 Ko	
	DescripteurSplitPorter	15/10/2024 17:10	Fichier	4 Ko	
	DescripteurToken	15/10/2024 17:10	Fichier	5 Ko	
	DescripteurTokenLancaster	15/10/2024 17:10	Fichier	4 Ko	
	DescripteurTokenPorter	15/10/2024 17:10	Fichier	4 Ko	
	InverseSpli	15/10/2024 17:10	Fichier	5 Ko	
	InverseSplitLancaster	15/10/2024 17:10	Fichier	4 Ko	
	InverseSplitPorter	15/10/2024 17:10	Fichier	4 Ko	
	InverseToken	15/10/2024 17:10	Fichier	5 Ko	
	InverseTokenLancaster	15/10/2024 17:10	Fichier	4 Ko	
	InverseTokenPorter	15/10/2024 17:10	Fichier	4 Ko	

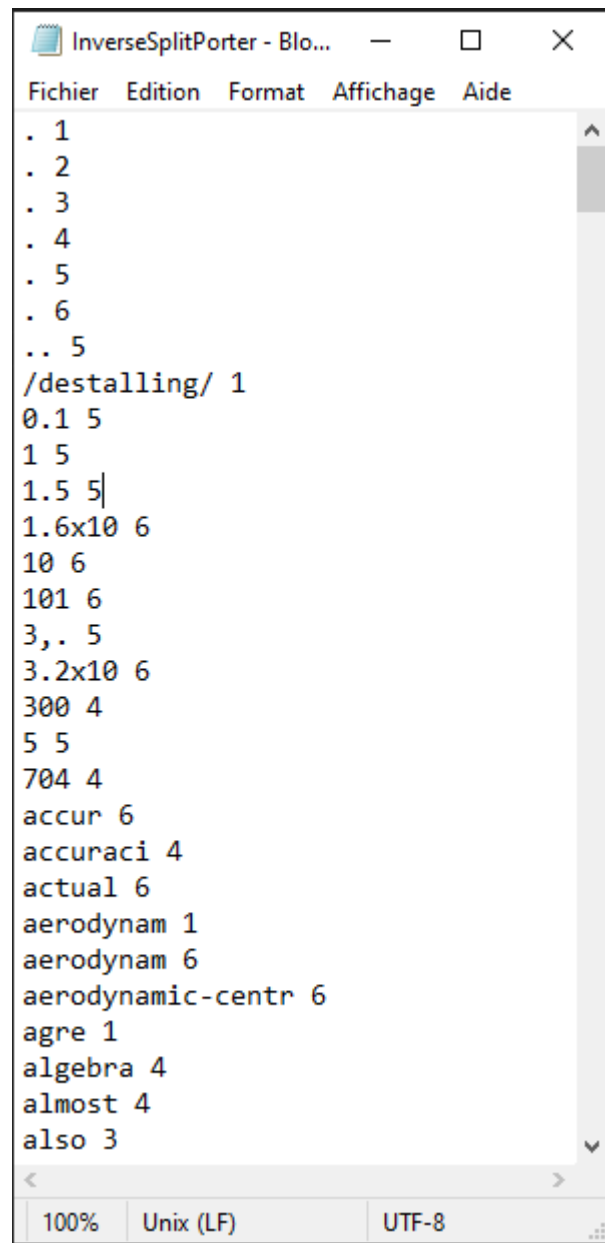
Fig.2 – Index à créer



```
1 .
1 /destalling/
1 aerodynam
1 agr
1 angl
1 attack
1 bas
1 boundary-layer-control
1 comp
1 config
1 curves,
1 destal
1 determin
1 diff
1 distribut
1 due
1 effect
1 empir
1 evalu
1 evidence,
1 experty
1 flow
1 found
1 fre
1 incr
1 increas
1 increment,
1 integr
1 intend
```

100% Unix (LF) UTF-8

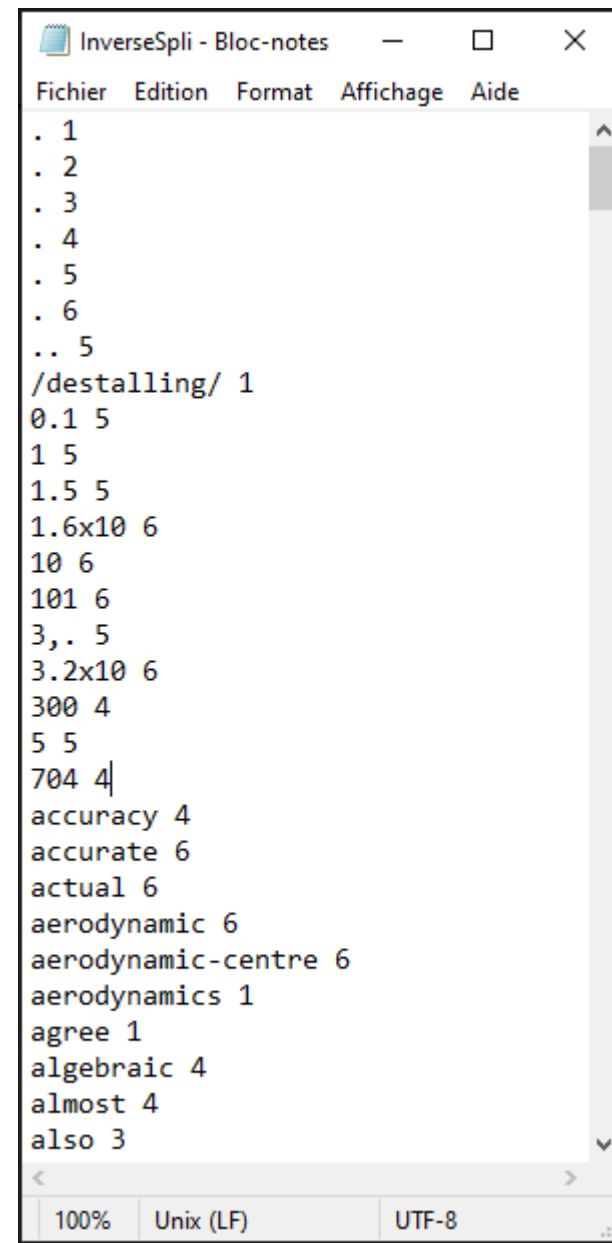
Fig.3 (a) – DescripteursSplitLancaster



```
. 1
. 2
. 3
. 4
. 5
. 6
.. 5
/destalling/ 1
0.1 5
1 5
1.5 5
1.6x10 6
10 6
101 6
3,. 5
3.2x10 6
300 4
5 5
704 4
accur 6
accuraci 4
actual 6
aerodynam 1
aerodynam 6
aerodynamic-centr 6
agre 1
algebra 4
almost 4
also 3
```

100% Unix (LF) UTF-8

Fig.3 (b) – InverseSplitPorter



```
. 1
. 2
. 3
. 4
. 5
. 6
.. 5
/destalling/ 1
0.1 5
1 5
1.5 5
1.6x10 6
10 6
101 6
3,. 5
3.2x10 6
300 4
5 5
704 4
accuracy 4
accurate 6
actual 6
aerodynamic 6
aerodynamic-centre 6
aerodynamics 1
agree 1
algebraic 4
almost 4
also 3
```

100% Unix (LF) UTF-8

Fig.3 (c) – InverseSplit