Compte-rendu

TD1 - Web Scraping sur le texte

Exercice 1 :

1.1 Taille totale en nombre de lignes

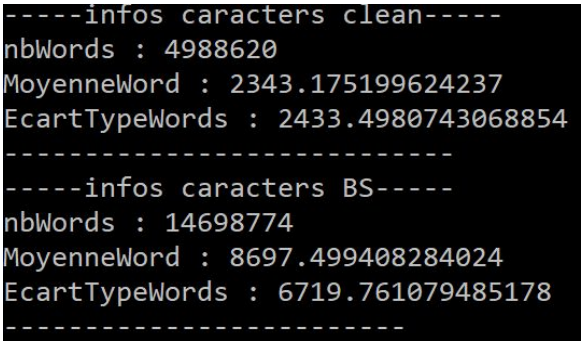
```
-----infos lignes clean-----
nbLignes : 29136
MoyenneLigne : 13.68529826209488
EcartTypeLignes : 10.889377753346407
---------------------------
-----infos lignes BS-----
nbLignes : 9545
MoyenneLigne : 5.64792899408284
EcartTypeLignes : 8.096499026003768
-------------------------
```

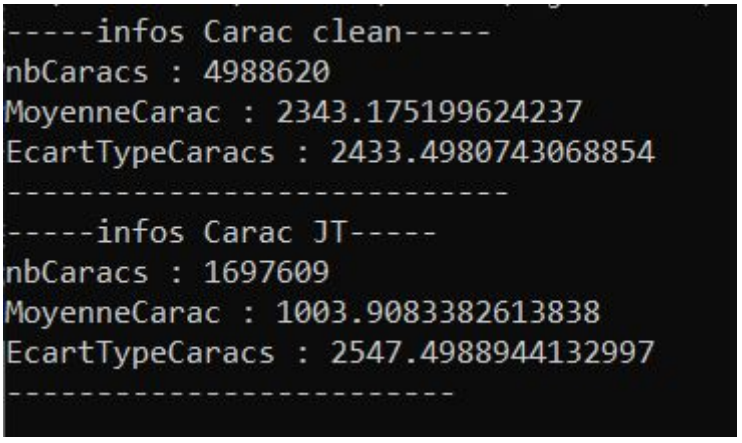
```
-----infos lignes clean-----
nbLignes : 29136
MoyenneLigne : 13.68529826209488
EcartTypeLignes : 10.889377753346407
---------------------------
-----infos lignes JT-----
nbLignes : 9542
MoyenneLigne : 5.642814902424601
EcartTypeLignes : 13.851671127282081
-------------------------
```

Nous avons seulement effectué les vérifications sur “JusText” et “BeautifulSoup” car “BoilerPipe” ne fonctionnait pas sur notre machine.

Les résultats sont très éloignés des résultats attendus si on ne définit pas les langues de chaque document à analyser (9500 lignes en sortie contre 29000 attendues). Cependant, pour “JusText”, nous avons dû renseigner “english” comme langue par défaut car nous sommes obligés de renseigner une langue pour faire une analyse. On constate que ce détail augmente considérablement l’écart-type entre le fichiers JT et clean (écart type de 13 pour JT contre 8 pour BS). Les résultats sont peu précis.

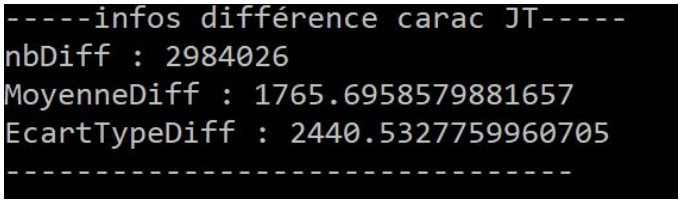
1.2 Taille totale en nombre de caractères

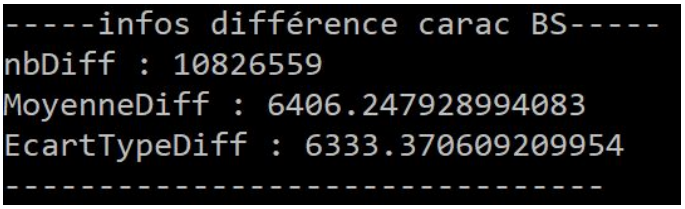




Concernant le nombre de caractères, nous sommes encore loin du résultat attendu. Par ailleurs, on remarque que l'écart-type est très grand ce qui implique que les résultats sont très variables d'un fichier à l'autre que ce soit pour "JusText" ou "BeautifulSoup". De plus, on voit que "JusText" possède un nombre de caractères très faible comparé à "BeautifulSoup". En effet, comme la langue par défaut est l'anglais, plusieurs fichiers ne retournent aucune donnée après analyse.

Différence totale en nombre de caractères





Avec une différence de 1765 caractères en moyenne et avec un écart-type de 2440 pour JT, on peut affirmer que la première analyse n'est pas concluante. Les résultats sont encore plus faibles avec "BeautifulSoup".

Exercice 2 :

	JT	BS	JT_langid	JT_trueLg
Différence (caractères)	2984026	10826559	1561735	1561735
Moyenne de la différence	1765.7	6406.2	924.1	924.1
Écart-type de la différence	2440.5	6333.4	2065.3	2065.3

Dans cet exercice, on utilise maintenant la gestion de la langue avec "JusText" afin de réaliser un traitement de meilleure qualité. La colonne "JT_langid" correspond à l'analyse de langue en utilisant l'outil de recherche de langue "langid" et la colonne "JT_trueLg" correspond à l'analyse en utilisant les langues effectives des fichiers (avec fichier de vérité terrain "doc_lg.json").

On remarque que faire l'analyse en utilisant la bonne langue permet une analyse plus performante (la différence de caractères diminue de 1 422 291). L'écart-type diminue également.

Cependant, on observe aucune différence entre "JT_langid" et "JT_trueLg" car le module "langid" à presque toujours trouvé la langue qui correspond au fichier. On peut nuancer notre résultat car nous avons été obligé de renseigner la langue "anglais" si le document est en "chinois" car cette langue n'est pas prise en compte dans le module.

Exercice 3 :

	el			en			pl			ru			zh			all		
	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P
JT	0.85	0.43	98	76	80.7	79	0.7	0.3	99	1.38	0.8	99	8.4	4.4	100	23	24	93
JT_langid	84.5	84.9	88.1	76	80.7	78.9	76.4	83.2	74.7	69.8	79.5	68.6	8.4	4.43	100	60.3	63.4	83.2
JT_trueLg	84.5	84.9	88.1	76	80.7	78.9	76.4	83.2	74.7	69.8	79.5	68.6	8.4	4.43	100	60.3	63.4	83.2
BS	50.5	93	36.7	48	91.2	35.9	48	89	35	34	89	23	7.6	33	5.1	36.9	77.1	26.7

On remarque que notre première analyse (JT) possède des F-mesures très faible sauf pour la langue anglaise. En effet, celle-ci était choisie par défaut ce qui explique les écarts important. Par ailleurs “BS” possède des résultats faibles mais plus élevé que “JT”. En effet BeautifulSoup ne réalise pas une analyse fine sur la langue.

Les résultats de “JT_langid” et “JT_trueLg” sont identiques comme pour l’exercice précédent et ils sont beaucoup plus précis que “JT” et “BS”. En effet, on possède des F-mesure n’allant pas en dessous de 68 pour toutes les langues sauf le chinois. L’étude de la langue des documents est donc un point fondamental pour la récupération des données textuelles.

Nous pouvons aussi ajouter que l’étude des données chinoises est difficile pour l’ensemble de ces outils (les F-mesures tournent autour de 8). Cette langue étant très différente, il est difficile pour des outils globaux de les récupérer. Un outil spécifique à cette langue serait plus pertinent.

Concernant l’analyse par site et par langue, vous pouvez trouver un aperçu du résultat sur le dossier “JT_trueLg” grâce à la capture ci-dessous. Le résultat se présente sous la forme d’un fichier json contenant 1 partie pour chaque langue (Greek, English…) qui elle même contient chaque site web lui appartenant.


```
{
    "Greek":
        [["www.express.gr", {"F": 90.24209729959391, "R": 87.14539669141502, "P": 95.59377614841652}],
        ["www.iatronet.gr", {"F": 43.25850861607938, "R": 50.02848019417591, "P": 42.008938071024964}],
        ["www.ethnos.gr", {"F": 85.05555823078363, "R": 92.29429978666758, "P": 82.32663241665351}],
        ["www.tanea.gr", {"F": 69.55852215674916, "R": 69.75330065162773, "P": 94.17493323111741}],
        ["health.in.gr", {"F": 89.54623315693483, "R": 84.51080994889512, "P": 95.30668178373547}],
        ["www.tovima.gr", {"F": 74.16475076301717, "R": 84.09270908012788, "P": 70.00344476512754}],
        ["www.rizospastis.gr", {"F": 96.96279352609936, "R": 95.05429584623275, "P": 98.96701945464599}],
        ["www1.rizospastis.gr", {"F": 87.32990586481998, "R": 85.910159206259, "P": 97.19074470520586}],
        ["www2.rizospastis.gr", {"F": 93.14088976486114, "R": 92.14192522133338, "P": 96.1686275736901}],
        ["www.amna.gr", {"F": 95.67147613762486, "R": 94.10480349344978, "P": 97.29119638826185}],
        ["news.in.gr", {"F": 85.16746411483254, "R": 78.76106194690266, "P": 92.70833333333334}],
        ["www.imerisia.gr", {"F": 79.28253746731691, "R": 94.76001562029934, "P": 71.03427998290437}],
        ["www.kathimerini.com.cy", {"F": 87.45944100290619, "R": 80.95034595034595, "P": 95.13352169419234}],
        ["www.ekathimerini.gr", {"F": 96.04863221884499, "R": 95.75757575757575, "P": 96.34146341463415}],
        ["www.avgi.gr", {"F": 82.48885392079728, "R": 93.79084967320262, "P": 73.94996653279784}],
        ["www.gazzetta.gr", {"F": 78.76676169359096, "R": 83.56545298982793, "P": 74.50035891042351}],
        ["ygeia.tanea.gr", {"F": 92.3180272566594, "R": 90.16903819407297, "P": 94.57260148860945}],
        ["news247.gr", {"F": 93.72937293729372, "R": 94.54061251664447, "P": 92.93193717277487}],
        ["www.madata.gr", {"F": 43.92935090130143, "R": 40.14689578713969, "P": 96.60574412532637}],
        ["www.sigmalive.com", {"F": 95.11530143660887, "R": 95.73539583726341, "P": 94.50385221163585}],
        ["www.kavalanet.gr", {"F": 94.21052631578948, "R": 92.74611398963731, "P": 95.72192513368985}],
        ["www.healthierworld.gr", {"F": 85.0, "R": 77.27272727272727, "P": 94.44444444444444}]],

    "English":
        [["ibnlive.in.com", {"F": 82.28736338625116, "R": 82.1932783073254, "P": 96.2362390642575}],
        ["www.deccanchronicle.com", {"F": 76.84964200477327, "R": 83.85416666666666, "P": 70.92511013215858}],
        ["timesofindia.indiatimes.com", {"F": 95.19845527692685, "R": 95.13953092498535, "P": 95.26150864386159}],
        ["www.mid-day.com", {"F": 90.302066772655, "R": 87.92569659442725, "P": 92.81045751633987}],
```


Exercice 4 :

Pour cette exercice nous avons choisi d’utiliser l’outil Unfluff. Cet outil permet de récupérer le contenu d’une page html en le séparant en plusieurs catégories comme “title” ou “text”. On scrappe les fichiers grâce à la commande “extractor” puis on effectue quelques transformations sur la propriété “text” afin d’ajouter des balises “p”. En effet, la propriété “text” nous donne le texte présumé de la page mais ne nous donne pas les informations sur les balises utilisées.

```
var data = extractor(fs.readFileSync(dirHtml+"\\"+siteName),langues[json[siteName]])

var allInfos = (data.text).replace(new RegExp('\n','g'),"</p>\n<p>")
allInfos = "<p>"+allInfos+"</p>"
allInfos = allInfos.replace(new RegExp('\n<p></p>','g'),"")
```

Pour la suite de l'exercice nous avons utilisé le code python réalisé dans l'exercice 3. On obtient donc le tableau ci-dessous concernant les mesures par langues :

	el			en			pl			ru			zh			all		
	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P
Unfluff	2	1	82	86	86	89	71	73	76	3	1	81	18	10.8	82	41.2	39	83
JT_trueLg	84.5	84.9	88.1	76	80.7	78.9	76.4	83.2	74.7	69.8	79.5	68.6	8.4	4.43	100	60.3	63.4	83.2

Si on compare avec les résultats obtenus grâce à JusText, on remarque que de manière globale, Unfluff est moins performant que JusText. Néanmoins, on observe que "Unfluff" possède de meilleurs résultats avec les fichiers anglais et chinois.

On retrouve ci-dessous l'analyse par site et par langue pour les fichiers provenant de "Unfluff" :

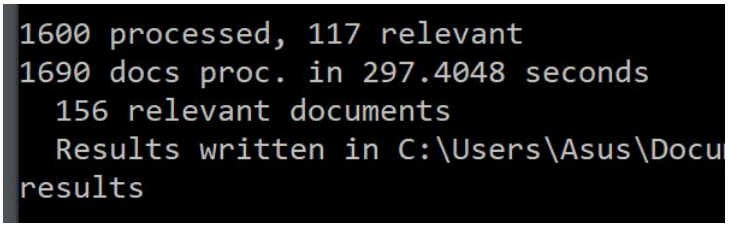

```
{
    "Greek":
        [["www.express.gr", {"F": 3.0534508549568122, "R": 2.143577076639301, "P": 92.51393133895219}],
        ["www.iatronet.gr", {"F": 6.2090219548098045, "R": 3.26783652108375, "P": 85.71428571428572}],
        ["www.ethnos.gr", {"F": 1.3765558329472771, "R": 0.6964289581379223, "P": 72.2222222222222}],
        ["www.tanea.gr", {"F": 1.9894567853956375, "R": 1.0107220063596514, "P": 75.0}],
        ["health.in.gr", {"F": 1.247608613606636, "R": 0.6630286043878383, "P": 57.31884057971014}],
        ["www.tovima.gr", {"F": 1.5222782426205639, "R": 0.7864446199627662, "P": 74.35897435897438}],
        ["www.rizospastis.gr", {"F": 2.115964307613832, "R": 1.071842051478971, "P": 83.33333333333331}],
        ["www1.rizospastis.gr", {"F": 1.9678854254936367, "R": 1.0074224784102355, "P": 80.00000000000001}],
        ["www2.rizospastis.gr", {"F": 1.8870417351394286, "R": 0.9696486117147473, "P": 80.20833333333336}],
        ["www.amna.gr", {"F": 0.8676789587852495, "R": 0.43668122270742354, "P": 66.66666666666666}],
        ["news.in.gr", {"F": 1.2578616352201257, "R": 0.8849557522123894, "P": 2.1739130434782608}],
        ["www.imerisia.gr", {"F": 0.7889867049946101, "R": 0.39660875773435045, "P": 77.77777777777777}],
        ["www.kathimerini.com.cy", {"F": 4.434265492281953, "R": 2.2680097680097684, "P": 100.0}],
        ["www.ekathimerini.gr", {"F": 1.2012012012012012, "R": 0.6060606060606061, "P": 66.66666666666666}],
        ["www.avgi.gr", {"F": 2.480203197370387, "R": 1.256127450980392, "P": 100.0}],
        ["www.gazzetta.gr", {"F": 4.06478102189781, "R": 2.07481699781348, "P": 100.0}],
        ["ygeia.tanea.gr", {"F": 1.3301466117154248, "R": 0.6701615491601584, "P": 100.0}],
        ["news247.gr", {"F": 0.5305039787798409, "R": 0.2663115845539281, "P": 66.66666666666666}],
        ["www.madata.gr", {"F": 3.5176977718414175, "R": 1.8154101995565408, "P": 83.33333333333333}],
        ["www.sigmalive.com", {"F": 1.6647160877930112, "R": 0.8443824615301695, "P": 88.88888888888887}],
        ["www.kavalanet.gr", {"F": 1.0282776349614395, "R": 0.5181347150259068, "P": 66.66666666666666}],
        ["www.healthierworld.gr", {"F": 2.247191011235955, "R": 1.1363636363636365, "P": 100.0}]],

    "English":
        [["ibnlive.in.com", {"F": 92.41234856323241, "R": 91.68086292660965, "P": 93.57502809505205}],
        ["www.deccanchronicle.com", {"F": 94.27792915531334, "R": 90.10416666666666, "P": 98.85714285714286}],
        ["timesofindia.indiatimes.com", {"F": 81.49707707323978, "R": 86.12183797589398, "P": 79.089821661592}],
        ["www.mid-day.com", {"F": 95.59748427672955, "R": 94.11764705882352, "P": 97.12460063897763}],
        ["www.deccanherald.com", {"F": 97.66949152542374, "R": 95.64315352697096, "P": 99.78354978354979}],
        ["twocircles.net", {"F": 96.88013136288998, "R": 93.94904458598727, "P": 100.0}],
        ["www.radionz.co.nz", {"F": 92.55846064199032, "R": 86.3771205062846, "P": 100.0}], ["www.stuff.co.nz",
```

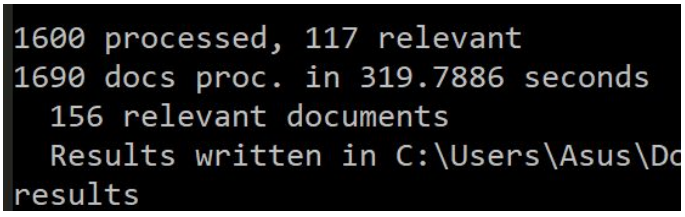
Exercice 5 :

Nous avons utilisé l'outil de classification DANIEL afin d'analyser nos fichiers HTML et déterminer si on parle de maladie dans ceux-ci. Nous obtenons les résultats suivants :

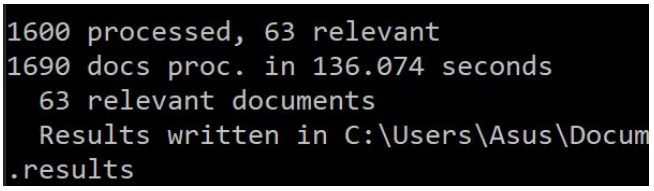
JT_trueLg



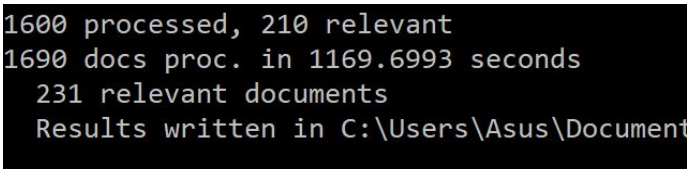
JT_langid



JT



BS



Unfluff

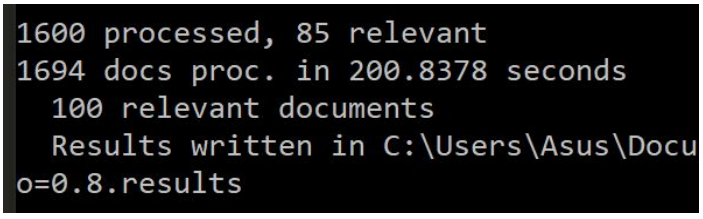


Tableau récapitulatif :

Nom méthode	Résultat
JT	63 fichiers parlent de maladie
JT_langid	156 fichiers parlent de maladie
JT_trueLg	156 fichiers parlent de maladie
BS	231 fichiers parlent de maladie
Unfluff	100 fichiers parlent de maladie

On remarque qu'on trouve plus de fichiers traitant de maladies en utilisant "JusText" (avec la notion de langue) et avec "BeautifulSoup". Les traitement avec Unfluff et JusText (sans gestion de la langue) permettent de trouver moins de fichiers. On trouve un nombre plus important de fichiers avec "BeautifulSoup".

Voici les résultats de l'évaluation entre les fichiers de résultats obtenus et le fichier de vérité terrain :

JT

```
{'TP': 25, 'FP': 38, 'FN': 99, 'TN': 1528, 'Missing_GT': []}
{'Recall': 0.2016, 'Precision': 0.3968, 'F1-measure': 0.2674}
  0 annotations missing
en {'Recall': 0.7143, 'Precision': 0.3968, 'F1-measure': 0.5102}
cn {'Recall': 0, 'Precision': 0, 'F1-measure': 0}
ru {'Recall': 0, 'Precision': 0, 'F1-measure': 0}
el {'Recall': 0, 'Precision': 0, 'F1-measure': 0}
pl {'Recall': 0, 'Precision': 0, 'F1-measure': 0}
```

JT_trueLg

```
{'TP': 87, 'FP': 69, 'FN': 37, 'TN': 1497, 'Missing_GT': []}
{'Recall': 0.7016, 'Precision': 0.5577, 'F1-measure': 0.6214}
  0 annotations missing
en {'Recall': 0.7143, 'Precision': 0.3968, 'F1-measure': 0.5102}
cn {'Recall': 0, 'Precision': 0, 'F1-measure': 0}
ru {'Recall': 0.8621, 'Precision': 0.7143, 'F1-measure': 0.7813}
el {'Recall': 0.8235, 'Precision': 0.5185, 'F1-measure': 0.6364}
pl {'Recall': 0.8519, 'Precision': 0.7419, 'F1-measure': 0.7931}
```

JT_langid

```
{'TP': 87, 'FP': 69, 'FN': 37, 'TN': 1497, 'Missing_GT': []}
{'Recall': 0.7016, 'Precision': 0.5577, 'F1-measure': 0.6214}
  0 annotations missing
en {'Recall': 0.7143, 'Precision': 0.3968, 'F1-measure': 0.5102}
cn {'Recall': 0, 'Precision': 0, 'F1-measure': 0}
ru {'Recall': 0.8621, 'Precision': 0.7143, 'F1-measure': 0.7813}
el {'Recall': 0.8235, 'Precision': 0.5185, 'F1-measure': 0.6364}
pl {'Recall': 0.8519, 'Precision': 0.7419, 'F1-measure': 0.7931}
```

BS

```
{'TP': 71, 'FP': 160, 'FN': 53, 'TN': 1406, 'Missing_GT': []}
{'Recall': 0.5726, 'Precision': 0.3074, 'F1-measure': 0.4}
  0 annotations missing
en {'Recall': 0.5143, 'Precision': 0.1765, 'F1-measure': 0.2628}
cn {'Recall': 0.9375, 'Precision': 0.2542, 'F1-measure': 0.4}
ru {'Recall': 0.5172, 'Precision': 0.5172, 'F1-measure': 0.5172}
el {'Recall': 0.4118, 'Precision': 0.5833, 'F1-measure': 0.4828}
pl {'Recall': 0.5926, 'Precision': 0.5517, 'F1-measure': 0.5714}
```

Unfluff

```
{'TP': 45, 'FP': 55, 'FN': 79, 'TN': 1515, 'Missing_GT': []}
{'Recall': 0.3629, 'Precision': 0.45, 'F1-measure': 0.4018}
  0 annotations missing
en {'Recall': 0.8571, 'Precision': 0.3896, 'F1-measure': 0.5357}
cn {'Recall': 0, 'Precision': 0, 'F1-measure': 0}
ru {'Recall': 0, 'Precision': 0, 'F1-measure': 0}
el {'Recall': 0, 'Precision': 0, 'F1-measure': 0}
pl {'Recall': 0.5556, 'Precision': 0.6522, 'F1-measure': 0.6}
```

Tableau récapitulatif :

	el			en			pl			ru			zh (cn)			all		
	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P
JT	0	0	0	0.5	0.71	0.4	0	0	0	0	0	0	0	0	0	0.27	0.2	0.4
JT_langid	0.63	0.8	0.5	0.5	0.71	0.4	0.8	0.85	0.75	0.79	0.86	0.71	0	0	0	0.62	0.7	0.56
JT_trueLg	0.63	0.8	0.5	0.5	0.71	0.4	0.8	0.85	0.75	0.79	0.86	0.71	0	0	0	0.62	0.7	0.56

BS	0.48	0.4	0.5	0.2	0.51	0.18	0.57	0.6	0.55	0.51	0.51	0.51	0.4	0.94	0.2	0.4	0.57	0.3
Unfluff	0	0	0	0.5	0.86	0.39	0.6	0.56	0.65	0	0	0	0	0	0	0.4	0.36	0.45

Tout d'abord, on remarque que la gestion de la langue est essentielle pour obtenir un bon résultat. En effet, pour JT (sans langue) et Unfluff, beaucoup de colonnes sont à 0 car il n'est pas possible de récupérer les mots liés à la bonne langue. Ce problème apparaît aussi pour JT (avec gestion de la langue) pour la langue chinoise (zh) qui n'est pas gérée par JusText.

Concernant les résultats, on remarque que même si "BeautifulSoup" trouve plus de fichiers traitant de maladies, il reste moins précis que JusText (0.4 de F-mesure pour "BeautifulSoup" contre 0.62 pour JusText). Les données trouvées par cet outil sont donc plus nombreuses mais moins précises. En revanche, JusText possède de très bon résultats quand la langue est correcte (plus de 0.5 en F-mesure pour chaque langue sauf le chinois). On obtient même une F-mesure de 0.8 pour le polonais et le russe. On détecte moins de fichiers parlant de maladie que "BeautifulSoup" mais on obtient des résultats plus précis.