Projet - Chunks

YANG
Heng
(Numéro d'étudiant – 11920185)

QIU
Zhiyuan
(Numéro d'étudiant – 10503044)

LI
Tong
(Numéro d'étudiant – 12001093)

UFR LLASIC
Département d'Informatique intégrée en Langues, Lettres et Langage

Master 1 Mention Sciences du Langage Parcours Industries de la Langue

UE: Modélisation des produits Industries de la langue (NPID8U00)
EC:Formalismes pour le TAL (NPID8X00)
&
UE: Réalisation de projet (NSX8U002)
EC :Représentation des connaissances (NSX8X002)

Enseignant responsable : Monsieur Thomas Lebarbé

Année universitaire
2020 – 2021

I. Présentation générale

Le présent rapport synthétise l'évaluation des scripts que nous avons créés pour l'analyse syntaxique automatique (Les chunks). À cette fin, on tentera d'analyser un article français sélectionné, en vue d'obtenir les règles correspondant aux chunks de cet article. Par la suite, nous utiliserons un autre article pour tester l'exactitude de ces règles des chunks, car elles peuvent ne pas s'appliquer correctement à d'autres articles. Une mesure de la précision des règles s'accompagne finalement de notre bilan sur les phénomènes linguistiques et autres observations tirées des étapes précédentes. Nous commencerons par détailler quelques explications sur la manière d'obtenir des règles de chunks avant de développer nos évaluations.

II. Description

L'article français, nous avons sélectionné sur *France Culture*. Pour l'article, nous donnons la catégorie du tout ainsi que de ses éléments, et le chunking de l'article a d'abord été fait manuellement. Déterminant les règles pour les chunks, et la catégorie correspondant à chaque mot. Nous avons choisi ces catégories ou règles dans l'espoir que les chunks soient toutes correctement identifiées. Après ces préparations, nous constituerons des fichiers. Après la tokenisation manuelle de l'article, nous obtenons un fichier (nommé *"Token.txt"*), et les sorties de *"Token.txt"* sont faites les catégories manuellement, afin d'obtenir un fichier *"Lexique.txt"*. Le fichier *"Règles.txt"*, contient toutes les règles nécessaires pour effectuer le chunking dans cet article. A la fin du programme, il y aura un fichier de sortie, nommé *"Output.txt"*

Exprimé sous la forme la plus simple, l'appliqué au chunking de notre projet peut être exécuté selon les étapes suivantes :

1. Faire le prétraitement du texte (sur le texte 1, la tokenisation)
2. Identifier les mécanismes linguistiques (Chunks & Catégories)
3. Développer le formalisme adapté (Règles & Catégories)
4. Écrire des algorithmes
5. Appliquer au texte
6. Évaluer la précision de résultat (pour texte 1)
7. Appliquer au 2ème texte (étapes 1 à 5). Il est important de noter que les mêmes règles sont utilisées dans la 3ème étape que précédemment. (Règles pour le texte 1)
8. Évaluer la précision de résultat (pour texte 2)

A. Domaine linguistique

Nous expliquerons dans cette partie les tâches linguistiques effectuées au cours du processus de chunk ainsi qu'un aperçu des règles.

Dans l'analyse linguistique, d'une part, nous avons analysé le texte sélectionné par un réseau, analysé la lexicalité, obtenu la structure *"Marqueur => Tête"* des règles de chunking, et effectué l'analyse fondamentale des données des règles et des catégories. Sur la base desquelles nous avons organisé le fichier.txt, le dictionnaire des règles *liste_regles.csv* et le dictionnaire des catégories de chunking: *liste_cat.csv.*

D'autre part, en raison de la complexité des règles dans cet article, les règles simples et générales sont très sujettes à l'ambiguïté et aux erreurs dans le processus de reconnaissance automatique. Nous avons donc affiné les règles existantes et les avons analysées afin de générer un dictionnaire d'expressions régulières pour les tâches de programmation ultérieures.

- Lexique

En termes de vocabulaire, cet article est tiré du texte d'une émission radiophonique de culture française, de longueur courte à moyenne, sans mots hors norme et contenant quelques mots étrangers (les mots anglais: *and /the/nom personnel*), avec un discours standardisé. Après le traitement automatique avec vérification manuelle, le texte complet comporte 125 tokens de mots, y compris les ponctuations. Par l'analyse linguistique, basée sur la théorie de chunking "Marqueur => Tête", nous avons analysé 107 chunks, dont *PN* représente le plus grand occurrence, 21/107=19,8%, suivi de *N* 16,98%, *PCTNF* 12,26%, *V* 10,38%, et *SV* 9,43%. Les autres catégories apparaissent avec très peu d'occurrence dans cet article, cependant nous ne pouvons pas exclure la petite taille de l'article et la complexité faible du lexique.

Nous avons donc essayé de comparer d' autres articles, et il est très évident que les catégories qui apparaissent le plus fréquemment, n'importe quel type ou quelle longueur de l'article, conservent des occurrences élevées. Comparant le résultat de cet article et de celui d'articles utilisés en cours : *PN* (28/149= 18.92%), suivie de *N* 12.84%, *PCTNF* 10.81%, *SV* 10.81% et *V* 9.46%. On pourrait conclure provisoirement que sous les règles de chunking du

français, à l'exclusion de la ponctuation, les catégories de chunks les plus fréquents sont *PN*, *N*, *V* et *SV*.

- Règles

En termes de règle, dont l'analyse et la génération se font en deux étapes. Dans la première étape, nous procédons d'abord au chunk selon le théorie Parsing By Chunks. Sauf verbes, noms et adverbes qui apparaissent séparément sont enregistrés en règle avec "||", les autres règles peuvent réaliser la structure *"Marqueur => Tête"*, en même temps la distribution de l'occurrence des règles est à peu près cohérente avec la distribution des catégories de chunking. Nous présentons donc ici un bref résumé de la première version des règles, basée sur la classification de la catégorie du marqueur/tête.

Préposition en tant que marqueur:

1. **Prep=>[PN ,** 3 type de marqueur dans ce règle: PRP_det/PRP/KON
2. **Prep=>[PVer ,** 1 type de marqueur dans ce règle: PRP
3. **Prep=>[PVant** , 1 type de marqueur dans ce règle: PRP

Verbe en tant que marqueur:

1. ***verbe=>[V*** , 7 type: structure d'un gramme/VER_subp/VER_mod/VER_englais/ VER_impf/ VER_pres/VER_pres_mod
2. ***verbe=>[VSe*** , 1 type: PRO_PER

Pronom en tant que marqueur:

1. ***PPS=>[SV***, 2 type: PRO_PER/PRO_DEM
2. ***PRels =>[PRelS]*** , 1 type: PRO_REL

Nom en tant que Tête:

1. ***det=>[N***, 4 type: DET_ART/DET_englais/DET_POST/NUM
2. ***pro =>[N*** , 1 type: PRO_IND
3. ***nom ||*** , 2 type: NOM/NAM

Le reste des règles est composé de mots simples ou d'expressions, *PCTNF, PCTF, ADV, CO, CS, GF, GO*, qui ne sont pas répétés ici car ils ont moins occurrences et structure simple de “un gramme”. Cependant, la première version de la règle était ambiguë dans le processus de reconnaissance automatique.

Afin de lever l'ambiguïté pour obtenir des résultats plus précis, dans la deuxième étape, nous avons généré des expressions régulières basées sur les règles existantes et la catégorie des tous les lexiques. (Les expressions régulières présentées dans cette section sont uniquement destinées à être généralisées et seront expliquées en détail dans la section suivante.)

La plupart des reconnaissances de règles ont bonnes performances, le texte reconnu possédant un ordre correct et une structure bigramme. Cependant, l'ambiguïté pendant la reconnaissance est due à 1. Certains chunks partiels avec des structures multi gram, 2. Certaine tête qui n'est pas correctement à la fin du chunk. 3. Des cas particuliers où les mots forment des chunks seulement avec soi-même.

Le problème d'ambiguïté 1 est particulièrement présent dans les SV avec structure complexe. Il y a un minimum de 2 éléments et un maximum de 6 éléments dans la structure SV. La raison pour laquelle le chunk est impossible est que dans certaines structures il existe *négation+verbe+adverbe+verbes* auxiliaires, ou verbes mod+adv+adj. Par exemple:”il faut être encore plus fort”. Les règles analysées dans la première étape s'appliquent mal aux structures aussi complexes. En généralisant les 10 chunks de structure SV, nous obtenons les expressions régulières suivantes:

SV:{<PRO_.*><PRO_PER>?<ADV_neg>?<PRO_PER>?<VER_pres>

<ADV_neg>?<KON>?<VER_infi>?<ADJ>?}

Le deuxième problème d'ambiguïté se pose dans la structure *V, PN* et *SV*. Dans la structure *V* et *PN*, quelques chunks particuliers se terminant par adjectifs, par exemple *V*: “est parfois difficile”, *PN*: “au charme musical”. Et dans la structure *SV*, il se termine souvent par un adjectif/adverbe ou “pas”. Nous avons donc corrigé les expressions régulières suivantes:

V: {<VER_.*><ADV>?<VER_pper|VER_infi>?<ADJ>?}

PN: {<KON|PRP.*>+<PRO_DEM|DET_ART>?<ADJ>?<NOM|NAM>+<ADJ>?}

Le troisième problème d'ambiguïté réside dans la tendance des chunks de la structure un gram à se composer avec les éléments suivants. Par exemple, le chunk *KON* n'a qu'un seul élément et est facilement combiné avec le *N* ultérieur, ensuite générer nouveau chunk *PN*, “et” *KON* + “la vie” *N* => *PN*. Pour résoudre ces problèmes, Nous avons modifié les expressions régulières pour structure un gram:

ADV: {<ADV>} PCTNF: {<PUNIF>} PCTF: {<SENT|PUNF>}

CO: {<KON.*>} CS: {<ADV|KON>?<KON>}

GF: {<GUILLEMET_F>} GO: {<GUILLEMET_O>}

N: {<PRO_IND>?<DET.*>?<NUM>?<NOM|NAM>} PRelS: {<PRO_REL>}

En généralisant les expressions régulières, les problèmes ci-dessus sont tous résolus. La reconnaissance automatique des règles est plus performante et plus précise. Les expressions régulières pour les autres structures ont toutes été mises à jour lors du traitement automatique, détaillé dans la section suivante.

B. Domaine informatique

Pour ce projet, nous choisissons de mettre en œuvre le programme avec Python. Dans la partie informatique, nous présentons un script qui permet de lire l'article après la tokenisation et de faire le chunking selon les catégories des mots et les règles.

Chaque type de chunk a une règle de base, généralement écrite par une expression régulière en Python. Et nous avons créé un tableau des marqueurs, principalement des déterminants. Ensuite, nous commençons à parcourir le texte déjà tokenisé en Python et le sauvegarder dans un tableau. De plus, on parcourt le tableau par chaque indice de 0 à longueur du tableau, un chunk est localisé par le tableau des marqueurs, c'est-à-dire que si le token existe dans le tableau des marqueurs, alors nous pouvons déterminer si la combinaison des tokens avant et après lui correspond au règle. Enfin, il est nécessaire de désambiguïser et d'éviter des bruits par l’algorithmes et de programmation basés sur les différentes structures d'un type de chunks.

En outre, les règles qui suivent sont basées sur l'analyse de l'article sélectionné, et dépendent davantage du lexique que nous avons créé et de l'article. Étant donné qu'un chunk peut avoir plusieurs structures différentes, et qu'une ambiguïté peut surgir entre les chunks à

cause des structures similaires, pour chaque chunk, après avoir analysé la règle de sa composition, nous devons également utiliser des algorithmes et la programmation pour désambiguïser et éviter des bruits.

Après notre analyse du chunk--N, nous avons constaté que deux structures principales existent pour ce chunk:

1. Un chunk--N composé d'un seul token, qui est un nom de personne unique, ou un nom qui apparaît seul. Il apparaît généralement comme une apposition.
2. Un chunk--N composé de deux ou trois tokens,le noyau de ce token est toujours un nom, contrairement au cas 1, il y a un déterminant dans le chunk. Et il faut tenir compte du cas où le déterminant est précédé d'un tout(e), et du cas où le nom est suivi d'un adjectif comme modificateur.

En ce qui concerne le chunk--PN, nous n'avons rédigé qu'une seule règle, mais il y aura trois cas. Un algorithme est nécessaire pour déterminer la classification de ces trois cas. La règle de base est la suivante: *(PRP|KON).*?(NOM|NAM).*?(ADJ)?*, Les trois cas dérivés de cette règle sont les suivant:

1. Le premier cas est celui où il n'y a pas d'adjectif comme modificateur, mais seulement une préposition ou(et) un déterminant devant le nom.
2. Ce cas est le même que le premier, sauf qu'il a du adjectif comme modificateur.
3. Le troisième cas exclut principalement l'ambiguïté et le bruit de l'existence du verbe, de la conjonction et du premier token en tant que nom dans la structure *chunk - PN*.

Quant au *chunk--SV*, il est plus complexe que les deux derniers cas. Tout d'abord, nous analysons l'existence de structures *SV* avec des expressions négatives, alors que la structure de négation doit prendre en compte l'occurrence de *y* à l'intérieur. Ensuite, il y a trois cas dans les structures *SV* restantes avec des phrases affirmatives.

1. La première est la structure la plus fondamentale, c'est-à-dire *pronom personnel + verbe.*
2. Le deuxième est basé sur le premier avec verbe à l'infinitif qui peut suivre le verbe auxiliaire, ainsi que les modificateurs.
3. La troisième est la structure dans laquelle un verbe pronominal réfléchi existe.

Les *Chunk--PVant*, *Chunk--PVer* et *Chunk--V* suivants sont les structures simples avec le noyau commun, verbe, donc il suffit d'identifier le token qui correspond à cette règle, puis de déterminer la catégorie du mot qui le précède et le suit par l'algorithme afin de préciser le type de chunk.

Avec l'identification des chunks les plus complexes avec les fréquences les plus hautes, les suivants «*Chunk-- ADV, CO/CS, PRelS, ponctuation, Vse*» sont essentiellement des chunks comportant une seule catégorie, qui peuvent être identifiés en déterminant leur catégorie et le token qui les précède et les suit.

III. Évaluation des résultats

Les étapes de la partie d'évaluation manuelle ont été réalisées à l'aide du dictionnaire Le Petit Robert (édition en ligne, 2021) ainsi que du UDPipe (2017, version informatisée, UD 2.6). Toutes deux sont disponibles dans la bibliographie.

A. Résultats de stats

1. Chunks

Nous constatons que, dans ce cas (**Figure.1**, le texte de *Musique* => L'article que nous avons utilisé pour créer la règle), nous avons obtenu un total de 111 Chunks, dont le pourcentage le plus élevé (25.23%) était *"Chunk:PCT(N)F"*【la ponctuation】, suivie de près par *[Chunk:PN]* (20.72%). Il n'est pas difficile de constater que les trois chunks qui représentent le plus faible pourcentage (1.8%) sont les suivants: *[Chunk:ADV]*、*[Chunk:Vse]* et *[Chunk:PVant]*. Nous avons cependant essayé de comparer un autre article (**Figure.2.3** => l'article pour tester, *"Texte 2"*. Si le "*Nombre de Chunk*:" est *110*, c'est-à-dire qu'on a changé les marqueurs. Sinon (c'est *99*), c'est-à-dire qu'on a pas changé les marqueurs, il a utilisé les même marqueurs de texte "*Musique*") et il est très évident que les catégories qui apparaissent le plus fréquemment, indépendamment du type ou de la longueur de l'article, maintiennent un taux d'apparition élevé. Nous pouvons provisoirement conclure que, selon les règles de chunking du français, à l'exclusion de la ponctuation, les chunks les plus fréquents sont *PN, N, V et SV.*

```
Nombre de Chunk :111
Nombre de 'Chunk:N':18    Pourcentage: (16.22%)
Nombre de 'Chunk:V':10    Pourcentage: (9.01%)
Nombre de 'Chunk:PVer':6          Pourcentage: (5.41%)
Nombre de 'Chunk:PCT(N)F':28      Pourcentage: (25.23%)
Nombre de 'Chunk:PN':23   Pourcentage: (20.72%)
Nombre de 'Chunk:PRelS':4         Pourcentage: (3.60%)
Nombre de 'Chunk:CO':3    Pourcentage: (2.70%)
Nombre de 'Chunk:CS':3    Pourcentage: (2.70%)
Nombre de 'Chunk:SV':10   Pourcentage: (9.01%)
Nombre de 'Chunk:ADV':2   Pourcentage: (1.80%)
Nombre de 'Chunk:Vse':2   Pourcentage: (1.80%)
Nombre de 'Chunk:PVant':2         Pourcentage: (1.80%)
```

Figure 1 (Musique.txt)

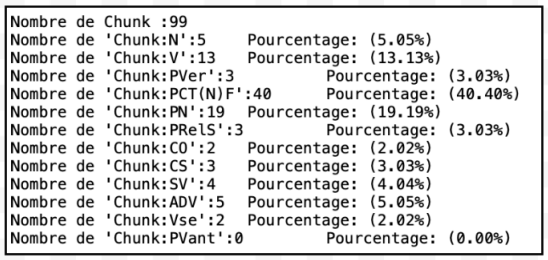


Figure 2 (l'article pour tester, pas de modifier des marqueurs)

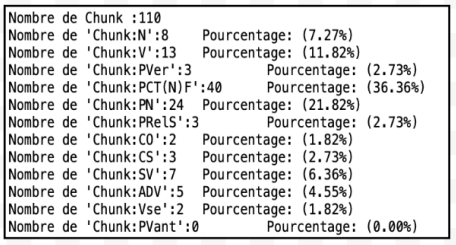


Figure 3 (l'article pour tester => changer les marqueurs)

2. Règles

Même méthode de comparaison que pour le chunk, on peut voir par ***Figure 4*** (*"Musique.txt"*) que la règle la plus utilisée est "*PCT(N)F*", qui représente 25.23% de l'ensemble. En outre, il est facile de voir que, en cas de changement d'article*(* ***Figure 5 et 6*** => l'article pour tester), la règle "*(PRP|KON).*?(NOM|NAM).*?(ADJ)?"* a également un bon taux d'utilisation. Nous pouvons également conclure que la règle *"VER"* n'est pratiquement pas affectée par les modifications du fichier (Bon taux d'application). Dans l'article testé, les règles *"NOM|NAM"* et *"PRO.*?ADV_negVER.*? ADV"* ne sont pas identifiées pour être utilisées. Ce résultat n'est pas très surprenant, parce que dans le texte original (*image noire, "Musique.txt"*), ces deux règles ne sont pas très bien utilisées (2.7%).

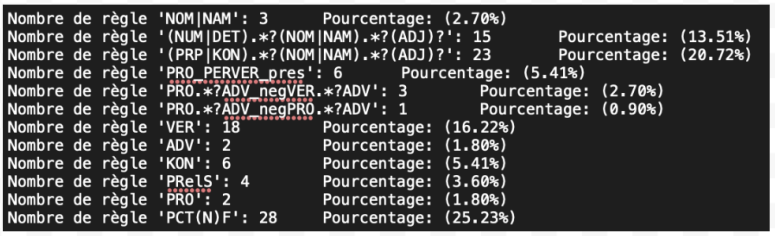


Figure 4 (Musique.txt)

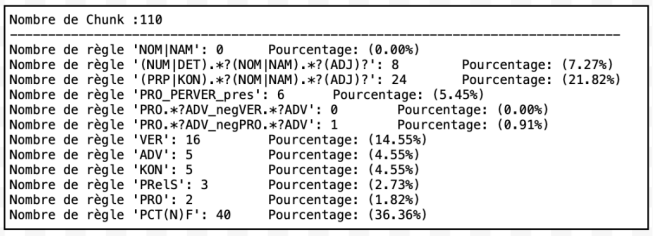


Figure 5

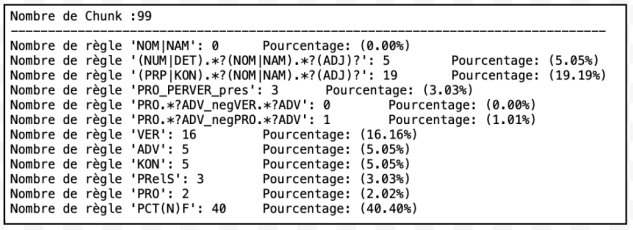


Figure 6

B. Résultats de durée

Les trois images (***Figure 7*** => *"Musique.txt"*)montrent que la durée du programme est liée à la longueur de l'article et au nombre de chunks. Plus la longueur de l'article est courte, plus le temps nécessaire est court. Dans le cas d'un même article, si le traitement est

différent (règles ou marqueurs), alors le temps requis est différent. En comparant les deux dernières images (***Figure 8 et 9)***, nous pouvons conclure que plus le nombre de Chunk est élevé, plus le temps nécessaire est important.

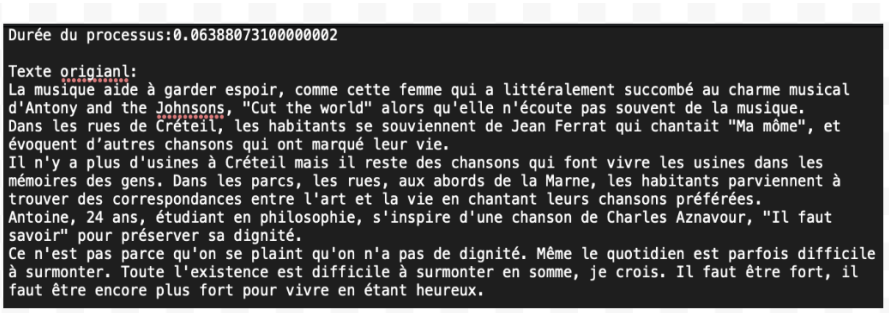


Figure 7

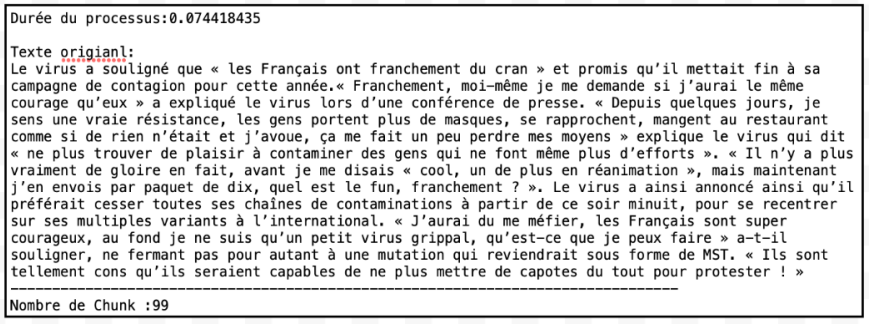


Figure 8 (l'article pour tester, pas de modifacation de marqueurs)

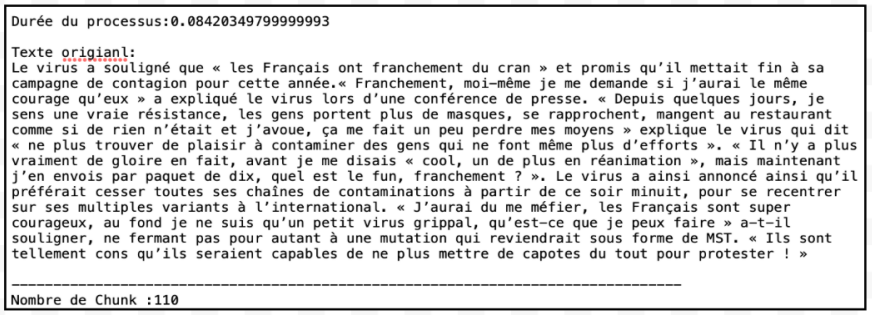


Figure 9 (l'article pour tester => changer les marqueurs)

C. Complexité linéaire

Un algorithme est de complexité linéaire si le temps de calcul croît de façon linéaire en fonction du nombre de données. Si on traite deux fois plus de données, le temps d'exécution sera multiplié par deux. Pour la conformité de notre algorithme, nous pouvons faire un calcul simple.

On sait que le nombre total de mots du premier texte ("*Musique.txt*", image noire,) est de 209 et que le temps requis est de *0,063* environ. Le nombre total de mots dans le texte

testé (*"Texte 2"*) est de 269, et si notre algorithme correspond à la complexité linéaire, alors le temps de traitement obtenu devrait être : ***(269 ÷ 209)×0.063 ≈ 0.081***. Nous pouvons voir sur l'image blanche (*Figure 8 et 9*) que le résultat est conforme à la demande. Nous concluons donc que notre algorithme est conforme à la complexité linéaire.

IV. Discussion et conclusions

En général, le taux d'application des règles que nous avons établies est relativement satisfaisant. Au cours de notre chunking, nous n'obtenons pas des résultats parfaits. Les règles que nous avons créées n'ont pas toutes bien fonctionné pour le chunking du nouvel article. En générale, des "oublis" de segmentation ont empêché la prise en considération de certaines marques de ponctuation ou d'expression polylexicale ce qui aura quelques répercussions sur l'analyse morphosyntaxique et chunk. Évidemment, cela n'a pas été le cas dans notre affaire. Ceci est dû au fait que nous avons utilisé une segmentation manuelle, ce qui réduit considérablement l'occurrence des "oubli" de la segmentation. Mais nous ne pouvons toujours pas garantir un taux d'exactitude de 100 %, cela ne fera que réduire relativement l'impact de cet aspect sur le résultat. En plus de cela, il y a des catégories des mots sont "unknown" (*"qu'est-ce"* et *"quel"*) dans le nouvel article, qui ont un effet sur le chunking d'article.

Nous constatons en outre que la différence de marqueur a également un effet direct sur le chunking. Par exemple, dans notre article de test. Avec les mêmes règles, le nombre de chunks a augmenté (*de 99 à 110*) lorsque les marqueurs ont été modifiés.

Nous conclurons ce rapport en admettant que les règles que nous avons créées sont imparfaites dans certains cas. La segmentation, l'inclusivité des règles et étiquette correcte des unités lexicales françaises nous est clairement apparue comme cruciale au cours de ce travail, et l'identification des catégories est d'autant plus complexe pour ceux d'entre nous qui ne travaillent pas sur notre langue maternelle.

Bibliographie

Dictionnaire Le Petit Robert (2021). Paris : Le Robert. lerobert.com. Repéré le 12 Décembre 2020, à l'adresse https://petitrobert-lerobert-com.sid2nomade-1.grenet.fr/robert.asp

Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Czech Republic. (2017). *UDPip*e (Stable, version 1.2, 2.0), à l'adresse http://lindat.mff.cuni.cz/services/udpipe/run.php

France Culture (19/03/2020). *En passant, quelques mélodie*s à l'adresse https://www.franceculture.fr/emissions/les-pieds-sur-terre/la-clinique-de-lamour-55-le-debut-de-la-fin-0

Le Gorafi. *Désobéissance civile : Vachement impressionné par le courage des Français, le virus annonce qu'il cesse toute contamination en France*, à l'adresse http://www.legorafi.fr/2021/01/28/desobeissance-civile-vachement-impressionne-par-le-courage-des-francais-le-virus-annonce-quil-cesse-toute-contamination-en-france/

Steven P. Abney (1994). *Parsing By Chunks*. Bell Communications Research Morristown USA.

Emmanuel Giguet (1998). *Méthode pour l'analyse automatique de structures formelles sur documents multilingues*

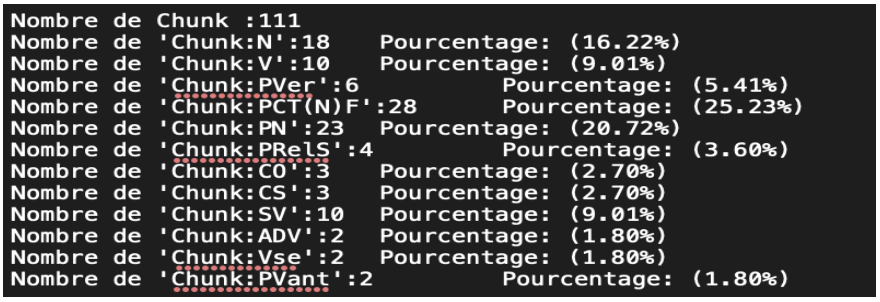
Annexes

Annexe 1 : Liste des fichiers associés

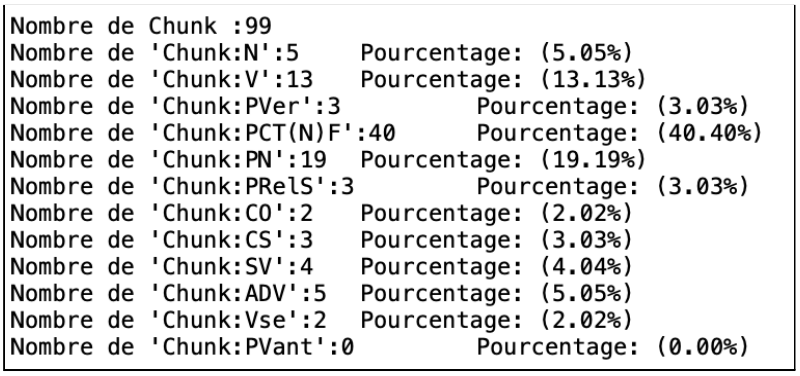
- Script de *traitement.py*: algo de faire des chunk, pour l'article que nous choisissons pour formuler les règles
- Fichier *lexique.py:* des informatiosn de l'article "*Musique.txt*"
- Fichier *Texte original.txt*: le contenu de l'article "*Musique.txt*"
- Fichier *Output(Musique).txt*: le output de algo "*traitement.py*", pour l'article "*Musique.txt*"
- Dossier *resource*: des fichiers liés à l'article "*Musique.txt*". [Token、regles et lexique]
- Dossier *Texte2 (pour tester)*: des fichiers liés à l'article pour tester. [Token、Règles et lexique] Aussi l'algo pour l'article testé "*traitement(texte2).py*" et les Output de "*traitement(texte2).py*".

Annexe 2 : Les figures utilisées dans le rapport

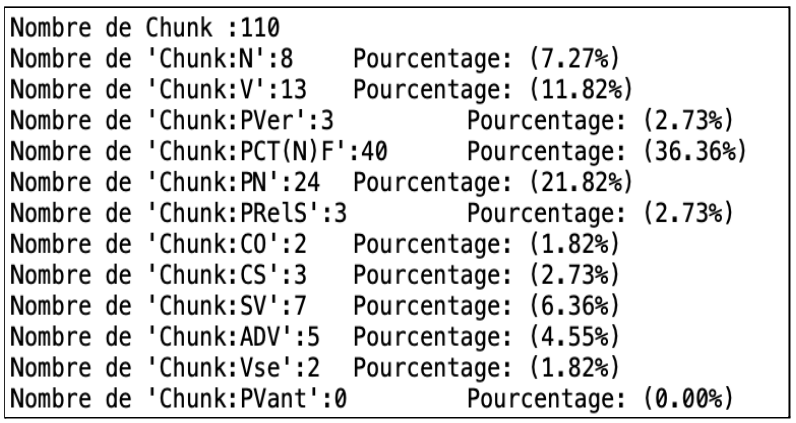
1. Figure 1: nombre de Chunk pour “Musique.txt”



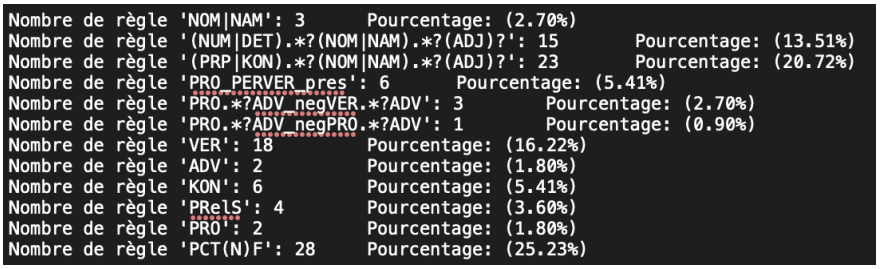
2. Figure 2: nombre de Chunk pour l’article testé (pas de changement de marqueurs)



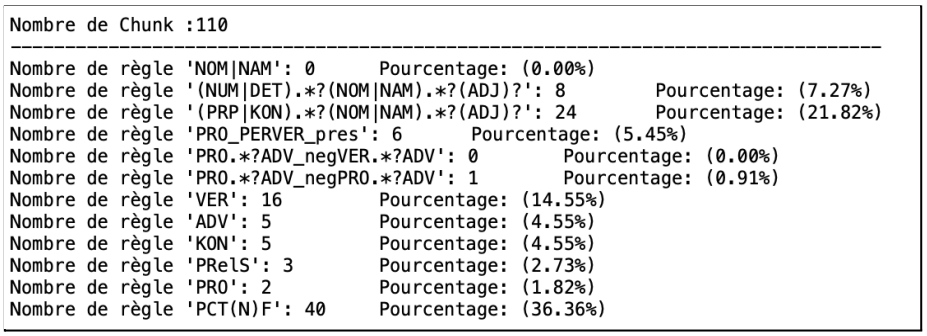
3. Figure 3: nombre de Chunk pour l’article testé (avec changement de marqueurs)



4. Figure 4: Nombre et pourcentage de fois où chaque règle a été utilisée pour “Musique.txt”

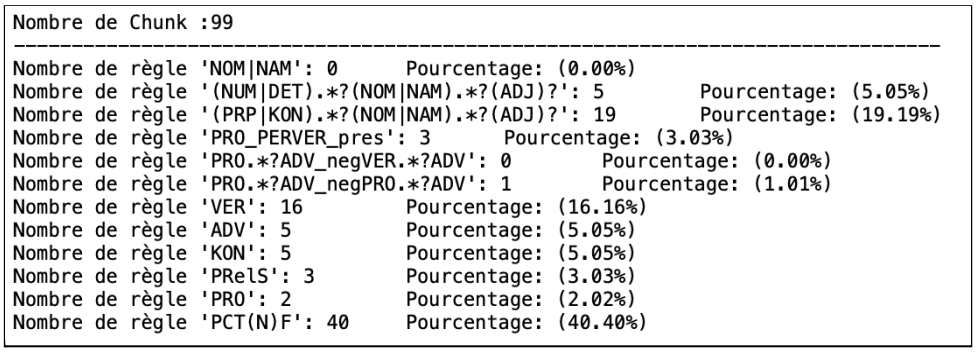


5. Figure 5: Nombre et pourcentage de fois où chaque règle a été utilisée => l'article testé (avec changement marqueurs)



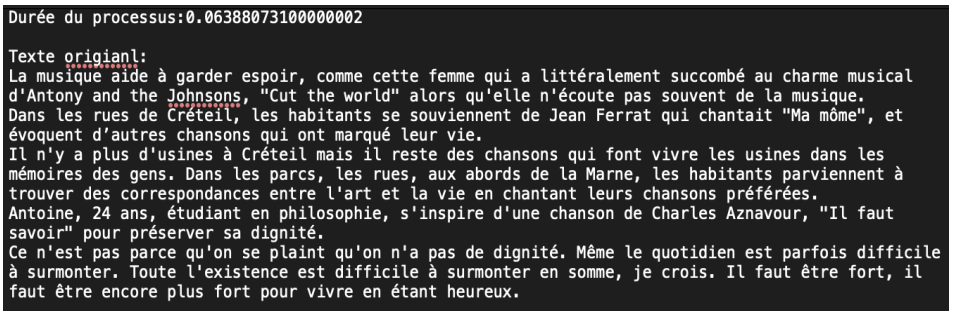
```
Nombre de Chunk :110
--------------------------------------------------------------------------------
Nombre de règle 'NOM|NAM': 0      Pourcentage: (0.00%)
Nombre de règle '(NUM|DET).*?(NOM|NAM).*?(ADJ)?': 8        Pourcentage: (7.27%)
Nombre de règle '(PRP|KON).*?(NOM|NAM).*?(ADJ)?': 24       Pourcentage: (21.82%)
Nombre de règle 'PRO_PERVER_pres': 6      Pourcentage: (5.45%)
Nombre de règle 'PRO.*?ADV_negVER.*?ADV': 0        Pourcentage: (0.00%)
Nombre de règle 'PRO.*?ADV_negPRO.*?ADV': 1        Pourcentage: (0.91%)
Nombre de règle 'VER': 16         Pourcentage: (14.55%)
Nombre de règle 'ADV': 5          Pourcentage: (4.55%)
Nombre de règle 'KON': 5          Pourcentage: (4.55%)
Nombre de règle 'PRelS': 3        Pourcentage: (2.73%)
Nombre de règle 'PRO': 2          Pourcentage: (1.82%)
Nombre de règle 'PCT(N)F': 40     Pourcentage: (36.36%)
```

6. Figure 6: Nombre et pourcentage de fois où chaque règle a été utilisée => l'article testé (pas de changement marqueurs)



```
Nombre de Chunk :99
--------------------------------------------------------------------------------
Nombre de règle 'NOM|NAM': 0      Pourcentage: (0.00%)
Nombre de règle '(NUM|DET).*?(NOM|NAM).*?(ADJ)?': 5         Pourcentage: (5.05%)
Nombre de règle '(PRP|KON).*?(NOM|NAM).*?(ADJ)?': 19        Pourcentage: (19.19%)
Nombre de règle 'PRO_PERVER_pres': 3      Pourcentage: (3.03%)
Nombre de règle 'PRO.*?ADV_negVER.*?ADV': 0        Pourcentage: (0.00%)
Nombre de règle 'PRO.*?ADV_negPRO.*?ADV': 1        Pourcentage: (1.01%)
Nombre de règle 'VER': 16         Pourcentage: (16.16%)
Nombre de règle 'ADV': 5          Pourcentage: (5.05%)
Nombre de règle 'KON': 5          Pourcentage: (5.05%)
Nombre de règle 'PRelS': 3        Pourcentage: (3.03%)
Nombre de règle 'PRO': 2          Pourcentage: (2.02%)
Nombre de règle 'PCT(N)F': 40     Pourcentage: (40.40%)
```

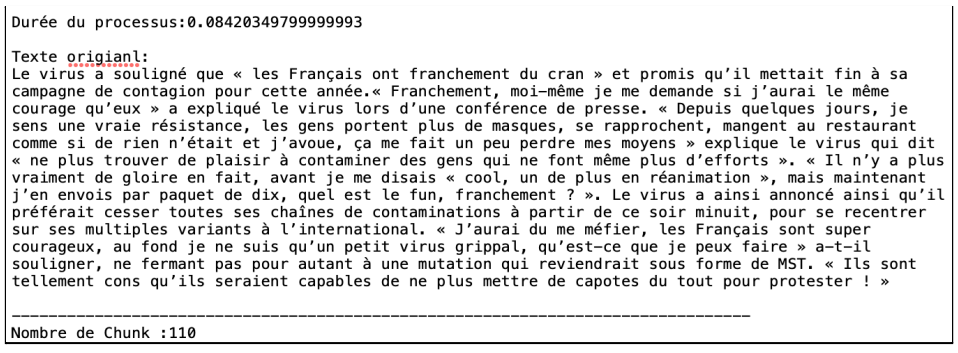
7. Figure 7: la durée du processus pour "*Musique.txt*"



```
Durée du processus:0.06388073100000002

Texte origianl:
La musique aide à garder espoir, comme cette femme qui a littéralement succombé au charme musical
d'Antony and the Johnsons, "Cut the world" alors qu'elle n'écoute pas souvent de la musique.
Dans les rues de Créteil, les habitants se souviennent de Jean Ferrat qui chantait "Ma môme", et
évoquent d'autres chansons qui ont marqué leur vie.
Il n'y a plus d'usines à Créteil mais il reste des chansons qui font vivre les usines dans les
mémoires des gens. Dans les parcs, les rues, aux abords de la Marne, les habitants parviennent à
trouver des correspondances entre l'art et la vie en chantant leurs chansons préférées.
Antoine, 24 ans, étudiant en philosophie, s'inspire d'une chanson de Charles Aznavour, "Il faut
savoir" pour préserver sa dignité.
Ce n'est pas parce qu'on se plaint qu'on n'a pas de dignité. Même le quotidien est parfois difficile
à surmonter. Toute l'existence est difficile à surmonter en somme, je crois. Il faut être fort, il
faut être encore plus fort pour vivre en étant heureux.
```

8. Figure 8: la durée du processus pour l'article testé (avec changement de marqueurs)

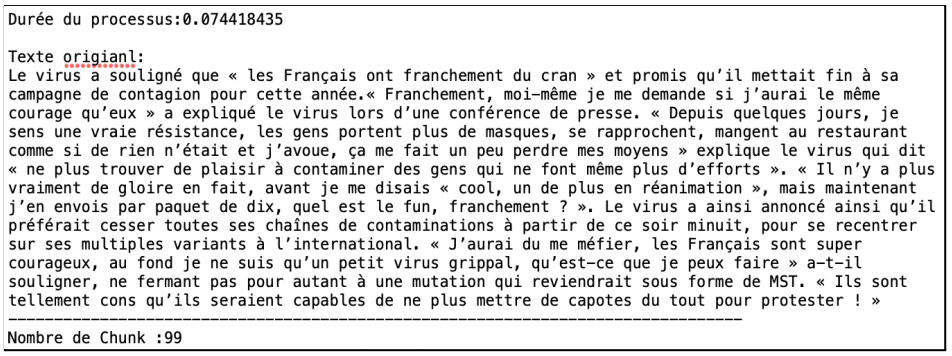


```
Durée du processus:0.08420349799999993

Texte origianl:
Le virus a souligné que « les Français ont franchement du cran » et promis qu'il mettait fin à sa
campagne de contagion pour cette année.« Franchement, moi-même je me demande si j'aurai le même
courage qu'eux » a expliqué le virus lors d'une conférence de presse. « Depuis quelques jours, je
sens une vraie résistance, les gens portent plus de masques, se rapprochent, mangent au restaurant
comme si de rien n'était et j'avoue, ça me fait un peu perdre mes moyens » explique le virus qui dit
« ne plus trouver de plaisir à contaminer des gens qui ne font même plus d'efforts ». « Il n'y a plus
vraiment de gloire en fait, avant je me disais « cool, un de plus en réanimation », mais maintenant
j'en envois par paquet de dix, quel est le fun, franchement ? ». Le virus a ainsi annoncé ainsi qu'il
préférait cesser toutes ses chaînes de contaminations à partir de ce soir minuit, pour se recentrer
sur ses multiples variants à l'international. « J'aurai du me méfier, les Français sont super
courageux, au fond je ne suis qu'un petit virus grippal, qu'est-ce que je peux faire » a-t-il
souligner, ne fermant pas pour autant à une mutation qui reviendrait sous forme de MST. « Ils sont
tellement cons qu'ils seraient capables de ne plus mettre de capotes du tout pour protester ! »

--------------------------------------------------------------------------------
Nombre de Chunk :110
```

9. Figure 9: la durée du processus pour l’article testé (pas de changement marqueurs)



```
Durée du processus:0.074418435

Texte origianl:
Le virus a souligné que « les Français ont franchement du cran » et promis qu’il mettait fin à sa
campagne de contagion pour cette année.« Franchement, moi-même je me demande si j’aurai le même
courage qu’eux » a expliqué le virus lors d’une conférence de presse. « Depuis quelques jours, je
sens une vraie résistance, les gens portent plus de masques, se rapprochent, mangent au restaurant
comme si de rien n’était et j’avoue, ça me fait un peu perdre mes moyens » explique le virus qui dit
« ne plus trouver de plaisir à contaminer des gens qui ne font même plus d’efforts ». « Il n’y a plus
vraiment de gloire en fait, avant je me disais « cool, un de plus en réanimation », mais maintenant
j’en envois par paquet de dix, quel est le fun, franchement ? ». Le virus a ainsi annoncé ainsi qu’il
préférait cesser toutes ses chaînes de contaminations à partir de ce soir minuit, pour se recentrer
sur ses multiples variants à l’international. « J’aurai du me méfier, les Français sont super
courageux, au fond je ne suis qu’un petit virus grippal, qu’est-ce que je peux faire » a-t-il
souligner, ne fermant pas pour autant à une mutation qui reviendrait sous forme de MST. « Ils sont
tellement cons qu’ils seraient capables de ne plus mettre de capotes du tout pour protester ! »
--------------------------------------------------------------------------------
Nombre de Chunk :99
```