Visualisation I

Mathilde Schwoerer

Pour découvrir l'intérêt des méthodes de stylistique numérique et nous familiariser avec leur manipulation, nous les avons mis en oeuvre dans trois projets. D'abord, nous appliquerons le modèle statistique du Latent Dirichlet Allocation accompagnée de l'échantillonnage de Gibbs en R à un corpus de taille très modeste. Parce qu'il nous a semblé important d'apprendre également à utiliser le langage Python, nous proposons une étude de topic-modeling très proche de la première, mais cette fois-ci appliquée à un corpus beaucoup plus vaste, étendu à toute l'oeuvre latine de Tertullien. Enfin, en retournant à R, nous avons cherché à classifier les traités de ce même auteur dans une structure ramifiée.

1 Topic modeling (en R)

Notre projet consiste à dégager les thèmes les plus importants d'un court poème anonyme du Ve siècle inspiré du martyre biblique des frères Maccabées (Macc 2, 6-7) et retenu sous le nom de *Carmen de Martyrio Maccabaeorum* (abrégé en *CdMM*). Il compte 394 hexamètres dactyliques, notre corpus sera donc peu étendu.

Nous avons déjà eu à travailler sur ce texte dans le cadre du mémoire de Master. C'est donc l'occasion confronter nos analyses avec les résultats que peuvent fournir les outils des Humanités Numériques.

1.1 Préparation des données

Notre liste de stopwords reprend la liste éditée sur le site de Perseus [1], complétée en tenant compte du mode de la manière dont LASLA lemmatise les textes. En effet, par exemple, la préposition *ad* apparaît après lemmatisation notée avec un chiffre accolé, comme *ad2*. Nous avons donc mis à jour cette liste .txt en ajoutant des chiffres là où il y en avait dans le texte lemmatisé.

Après avoir lemmatisé le texte latin brut [2] à l'aide de Pyrrha [3], on l'importe au format .csv dans le notebook. Comme il est impossible, semble-t-il, de transformer un dataframe en DocumentTermMatrix, mais que l'opération fonctionne avec une chaîne de caractères, on anticipe le problème en créant une chaîne vide, appelée "texte_long" :

```
df <- read.csv("CdMM_NLP/cdmm.csv", sep=",")
texte_long <- ""
```

1. http ://www.perseus.tufts.edu/hopper/stopwords.

2. Le texte latin est emprunté à Clemens Weidmann dans sa thèse non-éditée. Cl. C.Weidmann, *Das Carmen de martyrio Maccabaeorum*, Universität Wien, Dissertation, 1995. M. Weidmann m'a aimablement fait parvenir le fruit de ses recherches sous forme de fichiers Word

3. T. Clerice, J. Pilla, J.-B. Camps, V. Jolivet, A. Pinche, "Pyrrha, A langage independant post correction app for POS and lemmatization", nov 2019. Doi : 10.5281/zenodo.2325427, url.

Puis, en sélectionnant la colonne "lemma" du .csv, on met en minuscules toutes les majuscules, pour éviter que "Qui" reste dans le corpus si la liste de stopwords le présente sous la typographie "qui". À l'aide d'une boucle, on définit que si le mot n'appartient pas à la liste des stopwords importée auparavant, il est ajouté à la chaîne de caractères "texte-long". Ensuite, la ponctuation est retirée. À l'issue de ce processus, on obtient une longue chaîne de caractères ne contenant que les mots signifiants de la colonne "lemma".

```
for (word in tolower(df$lemma)) {
  if (!word %in% StopW) {
    texte_long <- paste(texte_long, word, sep=" ")
  }
  texte_long <- gsub("[[:punct:]]", "", texte_long)
}
```

Nous allons aborder le texte selon une approche dite *bag of words*. Cette méthode s'appuie sur la fréquence d'apparition des mots du corpus. Un vecteur relie chacun des mots à sa fréquence à l'intérieur d'une matrice vectorielle. Avant d'en créer une, on divise la chaîne de caractères "texte_long" en dix "sacs" (bag), puis on les organise dans une liste étiquettée "extraits" :

```
Nb_sequences <- 10
extraits <- strwrap(texte_long, nchar(texte_long) / Nb_sequences)
```

On peut ensuite transformer cette liste d'extraits en matrice vectorielle (notée "corpus") à l'aide d'une fonction de la library "text mining" [4] :

```
corpus <- Corpus(VectorSource(extraits), readerControl = list(
    language = "lat"))
```

On compte le nombre de colonnes dans la matrice.

```
ncol(as.matrix(DocumentTermMatrix(corpus)))
```

Le résultat est 740. Cette information sera importante par la suite.

On établit enfin un "documentTermMatrix", où chacun de nos extraits forme une ligne.

```
dtmCdMM <- DocumentTermMatrix(corpus)
```

À la lumière de cette information précise sur la taille du corpus, on décide de ne pas éliminer les mots les moins fréquents du texte. Si on ajuste en effet les seuils au volume du corpus, les mots les moins fréquents sont ceux qui se trouvent une seule fois, soit 408 termes sur 740.

1.2 Analyse du corpus

Notre projet consiste à faire émerger les deux ou trois thèmes que l'on estime centraux dans le *Carmen de Martyrio Maccabaeorum*. Dans cet objectif, on mobilise un modèle probabiliste pour données discrètes groupées, la Latent Dirichlet Allocation (LDA) [5], capable d'associer des mots à certains sujets.

4. I. FEINERER, K. HORNIK, D. MEYER, "Text Mining Infrastructure with R", *Journal of Statistical Software* 25/5 (2008), p. 1-54, ici p. 10-11, doi : 10.18637/jss.v025.i05.

5. W. M. Darling, "A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling", School of computer science, University of Guelf, december 2011.

```
k = 2
lda_2 <- LDA(dtmCdMM, k= k, control = list(seed = 1234))
#Puis trois.
lda_3 <- LDA(dtmCdMM, k= k+1, control = list(alpha = 0.1))
```

On renforce ce modèle à l'aide d'un algorithme qui repose sur la construction d'une chaîne markovienne, "l'échantillonnage de Gibbs" (*Gibbs sampler*). Une chaîne se caractérise comme markovienne si la valeur à l'état t dépend exclusivement de la valeur à la date t-1. (Pour ainsi dire, si l'on parle de météo, la prédiction météorologique du lendemain ne dépendrait que de la météo du jour présent.) Appliqué à la LDA, l'échantillonnage de Gibbs calcule la probabilité qu'un sujet s soit associé à un mot m, conditionnellement à toutes les autres associations de sujets à chacun des mots[6].

```
burnin <- 2000
iter <- 2000
thin <- 500
SEED=c(1, 2, 3, 4, 5)
seed <-SEED
nstart <- 5
best <- TRUE
lda_gibbs_2 <- LDA(dtmCdMM, k, method="Gibbs", control=list(nstart=
    nstart, seed=seed, best=best, burnin=burnin, iter=iter, thin=
    thin))
lda_gibbs_3 <- LDA(dtmCdMM, k+1, method="Gibbs", control=list(
    nstart=nstart, seed=seed, best=best, burnin=burnin, iter=iter,
    thin=thin))
```

```{r}
"LDA 2"
termsTopic <- as.data.frame(terms(lda_2,10))
head(termsTopic,11)
"LDA 3"
termsTopic <- as.data.frame(terms(lda_3,10))
head(termsTopic,11)
"LDA GIBBS 2"
termsTopic <- as.data.frame(terms(lda_gibbs_2,10))
head(termsTopic,11)
"LDA GIBBS 3"
termsTopic <- as.data.frame(terms(lda_gibbs_3,10))
head(termsTopic,11)
```

On obtient ainsi quatre tableaux de données. Le plus pertinent pour le sujet nous semble résulter de l'association de la LDA avec l'échantillonnage de Gibbs, à la recherche de deux sujets :

Le premier thème gravite autour des relations qui unissent la mère des Maccabées à ses fils. On y retrouve les discours d'exhortation rattachés à la piété filiale et à l'honneur de la lignée qu'elle adresse à chacun de ses enfants avant leur martyre. Le second sujet, en revanche, cerne les oppositions binaires qui traversent le poème : le conflit frontal entre le roi Antiochus et la mère des Maccabées, l'opposition entre le royaume de dieu et le royaume terrestre du tyran, le triomphe des frères Maccabées sur les flammes de leur bûcher.

Avec la boîte à outils "tidytext", on calcule également la probabilité d'apparition d'un mot par thème, notée "beta". On associe à chaque mot son beta.

6. Le phénomène mathématique est expliqué bien plus clairement dans W. M. Darling, *op. cit.*, p. 3-6.

	Topic 1	Topic 2
1	meus	rex
2	maneo	mater
3	solus	natus1
4	sanctus	deus
5	genus1	uinco
6	iubeo	regnum
7	parens1	ignis
8	supero	frater
9	partus1	uideo
10	muto2	uultus

```
lda_2 <- LDA(dtmCdMM, k= k, control = list(seed = 1234))
...
themes <- tidy(lda_2, matrix = "beta")
themes
```

Par exemple, un terme comme *dolor* présente deux probabilités d'apparitions très différentes dans l'un et l'autre thème. Il a presque cinq fois plus de chance d'être rattaché au second sujet qu'au premier (1 dolor 1.898765.10exposant3 contre 2 dolor 9.236187.10exposant3). Il en va de même pour *flamma*, plus de deux fois plus représenté dans le second sujet, par rapport au premier (1 flamma 3.129793x10exposant3 contre 2 flamma 7.997133x10exposant3).

1.3 Visualisation des résultats et interprétation

À partir des deux thèmes découlant des analyses menées grâce à la Latent Dirichlet Allocation et à l'échantillonnage de Gibbs, on obtient deux nuages de mots. Le premier rassemble les termes dépeignant la relation qui unit la mère des Maccabées à ses fils, dans des discours teintés d'urgence et d'exigence maternelle. On y retrouve les mots-clefs ponctuant les discours d'exhortation qu'elle leur adresse, aussi bien les encouragements motivés par l'accès au séjour des saints après le martyre *uictor, fortis, sanctus, supero* que les ordres dictés par la pression familiale (*pareo, parens, iubeo, gens*). Le lexique de la parole, qui sert à introduire le discours direct de la mère, est également présent dans ce thème *loquor, uox, uerbum)*, confirmant qu'il dépeint bien les relations entre la mère et ses fils. On y lit également les apostrophes hypocoristiques (*puer, meus*) dont elle les gratifie.

Le second thème s'intéresse aux conflits qui éclatent dans le poème. Les trois personnages principaux se trouvent au coeur du nuage de mots : la mère, le roi Antiochus et les fils Maccabées. Il dessine l'opposition entre le royaume de Dieu et les possessions terrestres d'Antiochus, laquelle se dispute pour les jeunes martyres dans le supplice du bûcher (*saeuus, durus, poena, ignis, flamma*) où leur espérance et leur courage (*spes, uirtus*) triomphent de l'effroi qu'ils éprouvent (*timor*).

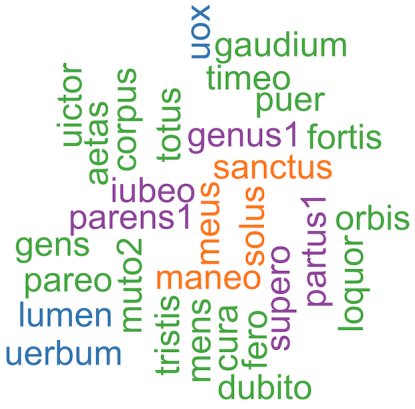

FIGURE 1 – Thème 1 : les discours d'exhortation de la mère destinés à ses fils (LDA et Gibbs)

Sans oublier que le *Carmen de martyrio Maccabaeorum* a connu des transformations importantes au cours du processus (la lemmatisation et la suppression des stopwords) et compte tenu des fonctions utilisées ainsi que des paramètres d'analyse statistique choisis, on observe une répartition des termes les plus fréquents en deux sujets qui tracent une frontière entre deux espaces. Nous avions autrefois émis l'hypothèse que la mère délimitait deux espaces dans le poème : celui du conflit verbal contre le tyran, et un autre réservée à ses fils et à elle-même. Les frères Maccabées ne prononcent pas un seul mot de tout le poème. Ils ne s'adressent jamais qu'à Dieu, dans un moment de recueillement silencieux. Ils paraissent ainsi détachés des joutes verbales entre la mère et le tyran. La mère les garde dans un espace imaginaire où elle est la seule à leur parler. On retrouve quelque peu ces dynamiques dans les deux nuages de mots mis au jour. Le roi est absent du premier, il esquisse un espace dédié à la mère et à ses fils, tandis qu'il se dégage du second nuage de mots une atmosphère bien plus menaçante. Ainsi, cette étude confirmerait en partie les résultats de nos analyses antérieures.

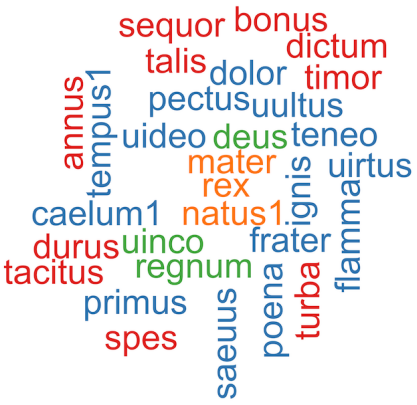

FIGURE 2 – Thème 2 : les caractéristiques des conflits (LDA et Gibbs)

2 Topic modeling (en Python)

Ce travail en Python s'appuie beaucoup sur deux cours de Lino Galiana distribués en ligne [7]. Le questionnement à l'origine de ce projet porte sur les thématiques non-théologiques qu'aborde Tertullien. Nous ne cherchons pas à enrichir les études sur la pensée chrétienne de l'auteur, mais essayons de cerner les problématiques qui traversent son imaginaire d'écrivain.

2.1 Préparation des données

2.1.1 Lemmatisation de l'ensemble des textes

Une fois l'intégralité des oeuvres de Tertullien téléchargées en ligne et enregistrée dans des fichiers séparés [8], on utilise l'outil de lemmatisation pour le latin classique en ligne, Pyrrha [9]. Ensuite, on fond tous les fichiers .tsv en un corpus unique à l'aide d'un site internet. Il en résulte le fichier intitulé "Corpus.csv".

2.1.2 Nettoyage des données

Afin de créer le corpus d'analyse, qui consiste en une liste de chaînes de caractères, on initie une variable sous le nom de "text csv". Il s'agit d'une liste vide dans laquelle seront copiés les mots de la colonne "lemma" du .csv, une fois qu'ils auront été réduits en minuscules et que l'algorithme aura établi qu'ils ne figurent pas dans la liste des stopwords.

7. Adresse du site web : site web. Liens vers les notebooks : Notebook LDA et Notebook Word to Vector

8. Perseus.

9. T. CLERICE, J. PILLA, J.-B. CAMPS, V. JOLIVET, A. PINCHE, "Pyrrha, A langage independant post correction app for POS and lemmatization", nov 2019. Doi : 10.5281/zenodo.2325427, url.

```
#Pour chaque ligne du dataframe, on met en minuscules le mot de la colonne "lemma".
for word in CSV['lemma']:
    #On ne conserve que les chaînes de caractères :
    if type(word)==str:
        word=word.lower()
        #Si le mot n'est pas un stop-words...
        if word not in stop_words:'
            #...alors l'ajoute à notre variable texte
            text_csv+=word + ' '
#On enlève enfin la ponctuation.
text_csv = text_csv.translate(str.maketrans('', '', string.punctuation))
#On affiche les 500 premiers caractères.
print(text_csv[:500])
```

FIGURE 3 – Code photographié pour ne point vexer LaTeX...

Comme l'indique la dernière ligne du code ci-dessus, on élimine également la ponctuation après copie des chaînes de caractères dans la variable, pour finalement obtenir ceci (500 premiers caractères) :

> uarie diabolus aemulor ueritas affecto aliquando defendo concutio unicus dominus uindico omnipotens mundus1 conditor1 unicus haeresis facio pater descendo uirgo nascor patior iesus excido1 coluber iesus baptisma ioannes tento filius aggredior certus filius habeo scriptura tentatio struo filius lapis panis item filius deicio scribo mando2 angelus pater manus1 tollo necubi lapis pes offendo numquid mendacium euangelium exprobro uideo matthaeus lucas accedo1 omnipotens cominus tento accedo1 tento f

Une remarque à propos de la liste des *stopwords* : on recourt à la même qu'auparavant, à deux termes près. Ont été ajoutés en effet deux noms extrêmement représentés chez un auteur chrétien comme Tertullien, *Deus* et *Christus*.

2.2 Premier nuage de mots

On crée une figure à partir de nos données nettoyées, puis on définit les fonctions de Python pour générer un nuage de mots.

```
1 fig = plt.figure()
2 def make_wordcloud(corpus):
3     wc = wordcloud.WordCloud(background_color="white", max_words
          =200, mask=book_mask, contour_width=3, contour_color='
          steelblue')
4     wc.generate(corpus)
5     return wc
```

Afin d'obtenir des résultats plus précis que sur ce nuage de mots, on calcule la fréquence des termes les plus courants en établissant d'une part leur liste, d'autre part leur fréquence d'apparition :

```
1 from nltk import FreqDist
2 fdist = FreqDist(Lemme)
3 Mots_courants= [fdist.most_common(20)[i][0] for i in range (20)]
4 Nb_iterations=[fdist.most_common(20)[i][1] for i in range (20)]
```

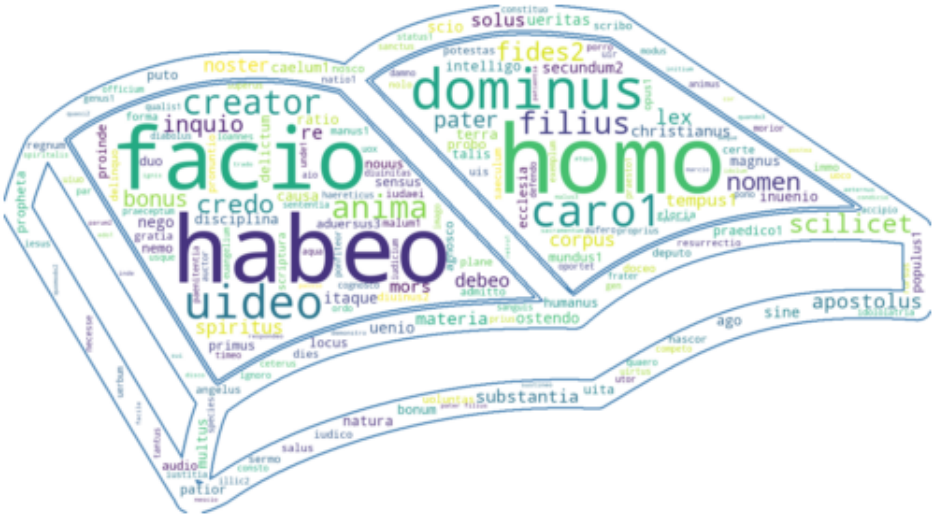

FIGURE 4 – Wordcloud obtenu sur notre corpus entier

À partir de ces données, nous générons le graphique suivant :

2.3 Topic-modeling avec Dirichlet Latent Allocation

Dans l'intention de dégager les thèmes secondaires dans l'oeuvre de Tertullien, nous mobilisons l'Allocation de Dirichlet Latente, un modèle probabiliste génératif qui s'intéresse (très grossièrement) à la distribution des distributions de mots. Il présuppose un nombre fixe de sujet (que l'on définit manuellement) et s'attache à calculer la proximité de tel mot avec tel sujet [10].

Appelons Scikit Learn et définissons neuf sujets de recherche :

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
d = {'Tokens': Lemme}
corpus=pd.DataFrame(data=d)
count_vectorizer = CountVectorizer(stop_words=stop_words)
count_data = count_vectorizer.fit_transform(corpus.apply(lambda s:
    ' '.join(s)))
lda = LatentDirichletAllocation(n_components=9, max_iter=5,
                                learning_method = 'online',
                                learning_offset = 50.,
                                random_state = 0,
                                n_jobs = 1)
lda.fit(count_data)
```

Le résultat s'affiche sous forme de nuages de mots :

10. D. M. BLEI, "Topic Modeling and Digital Humanities", *Journal of Digital Humanities* 2/1 (2012), url

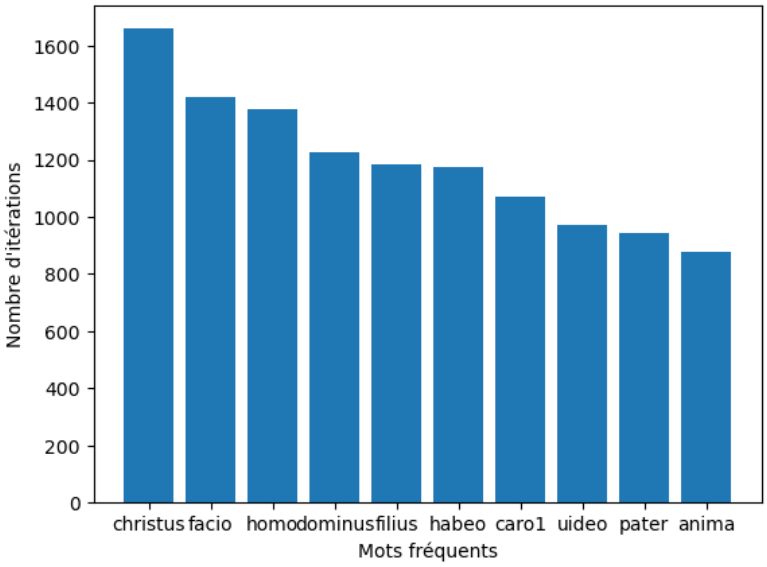

FIGURE 5 – Distribution des mots les plus fréquents

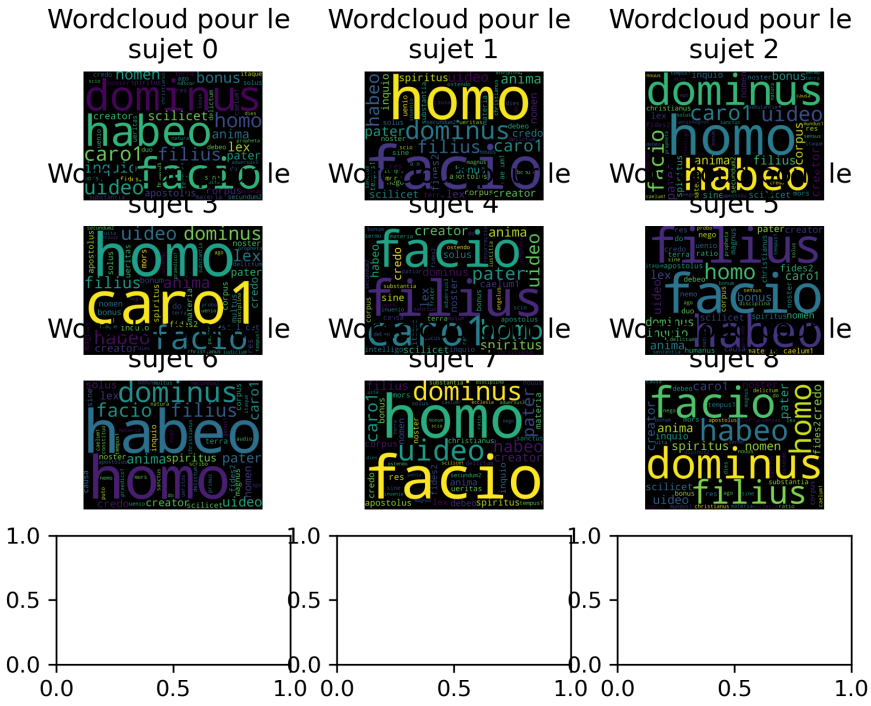

FIGURE 6 – Wordclouds de 9 sujets sur le corpus entier

On observe que les neuf nuages de mots générés contiennent quasi tous les mêmes termes principaux : *homo, facio, habeo, dominus, caro1, anima, creator, filius, pater, uideo, spiritus*. Trois sont des verbes d'usage assez commun (*facio, habeo, uideo*) qui éclairent peu les thématiques qu'aborde Tertullien. Le vocabulaire lié aux personnes divines du christianisme ne nous intéresse pas non plus dans cette perspective (*dominus, creator, filius, pater*). Comme on sait que Tertullien se distingue parmi les théologiens pour sa dialectique très particulière entre la chair et l'esprit [11], les termes *caro1* et *anima* nous intéressent peu. Pour tenter d'affiner l'analyse du corpus, on crée une seconde liste de *stopwords* contenants cette série de termes récurrents, puis on relance le code pour obtenir le résultat suivant :

```
with open('StopwordsLatin_Tertu.txt', 'r') as stop_words_file:
    stop_words_Tertu = stop_words_file.read().splitlines()
```

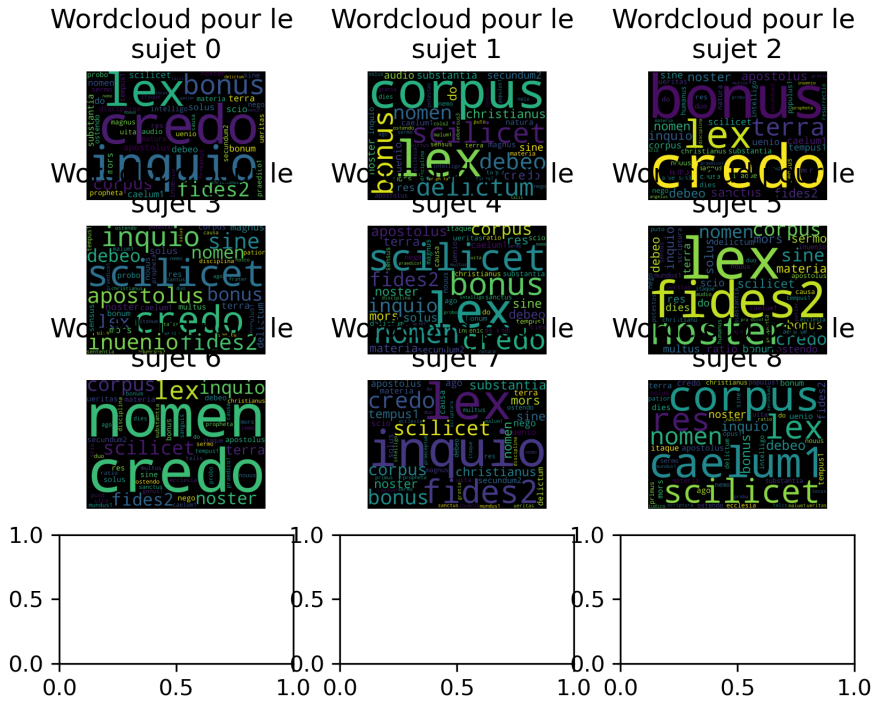

FIGURE 7 – Wordclouds de 9 sujets sur le corpus entier après suppression de mots très fréquents

L'élimination de mots très fréquents n'améliore guère la détection de sujets de notre corpus. De fait, d'autres mots ressortent avec la même force, comme *lex, credo, nomen, corpus, scilicet* et *inquio*, mais on ne peut faire émerger des thèmes précis.

11. J. ALEXANDRE, *Une chair pour la gloire. L'anthropologie réaliste et mystique de Tertullien*, Théologie Historique 115, Paris, 2001.

2.4 Résultats

En somme, les résultats obtenus sont mitigés. Grâce à la Dirichlet Latent Allocation, la machine réussit à caractériser Tertullien. Elle indique en effet les problématiques qui traversent son oeuvre en soulignant les mots qui les représentent, comme la dialectique entre la chair et l'âme, la rigueur de sa pensée qui s'en réfère toujours à la loi, la foi et la défense, capitale pour lui, du nom de chrétien. Cependant, le recours à la Dirichlet Latent Allocation ne permet pas à la machine de cerner **finement** l'auteur en explorant les thèmes secondaires de ses écrits. Elle le caractérise au point d'en brosser une caricature.

Toutefois cette conclusion en demi-teinte ne doit pas nous décourager : si la machine parvient aussi aisément à caractériser Tertullien, il a y a de fortes chances qu'elle puisse identifier sa plume parmi un jeu composé de textes d'autres auteurs, quand on l'aura entraînée à l'aide d'outils de machine-learning. C'est une tâche à laquelle nous comptons nous atteler par curiosité personnelle et pour la suite de ma thèse.

3 Répartition des traités de Tertullien (en R)

Du même Tertullien, nous conservons un héritage de 31 traités, que les spécialistes répartissent en trois catégories. D'un côté, les oeuvres de défense du christianisme contre ses détracteurs païens, désignée communément sous le nom de traités "apologétiques". Une deuxième catégorie se forme des traités "disciplinaires", c'est-à-dire des écrits parénétiques dans lesquels l'auteur prodigue des recommandations pour appliquer au mieux, selon lui, les règles du Nouveau Testament. Il s'adresse alors plutôt à un lectorat chrétien. Une dernière catégorie regroupe enfin les textes émergeant dans un contexte de questionnements sur la nature de Dieu, sur la matière ou bien sur la génération de la divinité. Tertullien vécut lors d'une ère d'inquiétudes métaphysiques qui touchaient aussi bien païens que chrétiens, aussi les derniers imaginaient-ils parfois des raisonnements inspirés de la philosophie qui leur attiraient les foudres de l'Église majoritaire [12]. Tertullien s'est heurté à la pensée de quelques uns de ces hérétiques, le dualiste Marcion, les partisans de la gnose valentinienne, le matérialiste Hermogène ou encore Praxeas, proche du monarchianisme.

À l'aide du package Stylo [13], nous tenons à vérifier si la répartition qu'ont proposée les experts de Tertullien se traduit également dans l'emploi de certains stylèmes. Dans ce cas, on obtiendrait un dendrogramme à trois branches principales.

3.1 Préparation des données

Après avoir téléchargé l'intégralité des textes de Tertullien en format .xml sur le site de Perseus [14] et les avoir placés dans un dossier dédié ("corpus" [15]), nous

12. E. R. DODDS, *Païens et chrétiens dans un âge d'angoisse. Aspects de l'expérience religieuse de Marc-Aurèle à Constantin*, L'Âne d'Or : Belles Lettres, Paris, 2010 [1965].

13. Eder, M., Rybicki, J. and Kestemont, M. (2016). Stylometry with R : a package for computational text analysis. R Journal 8(1) : 107-121. ¡https ://journal.r-project.org/archive/2016/RJ-2016-007/index.html¿.

14. https ://scaife.perseus.org/library/

15. La boîte à outils employée, Stylo, fonctionne nécessairement avec un dossier noté "corpus". Celui-ci ne contient que des fichiers de texte à analyser et tous ces fichiers partagent un

les avons renommés avec une lettre initiale qui indique à quelle catégorie chacun appartient. La lettre "A" marque les traités apologétiques, "D" correspond aux écrits de controverse doctrinale contre les hérétiques et les juifs, tandis que les textes de discipline destinés aux chrétiens se distingue par "Di". Cette notation facilitera la lecture des dendrogrammes en incitant Stylo à afficher dans la même couleur les textes d'une même catégorie :

— en rouge, les écrits apologétiques ;
— en bleu, les traités disciplinaires ;
— en vert, les textes de controverse doctrinale.

3.2 Analyse du corpus

Comme notre projet relève de "l'apprentissage non-supervisé", quelques difficultés se présentent. L'apprentissage non-supervisé s'oppose à l'apprentissage supervisé [16]. Celui-ci présuppose que l'on confie à la machine des données labellisées, c'est-à-dire des données dont on connaît la valeur. Quand la machine propose une réponse à un problème donné, les chercheurs derrière elle sont capables de dire si elle se fourvoie ou non. Prenons par exemple un problème typique de classification : si on entraîne un ordinateur sur les textes de trois auteurs (A, B et C), puis qu'on lui soumet un texte appartenant à C - sans qu'elle le sache -, on peut savoir si sa prédiction est correcte ou non. En revanche, en situation d'apprentissage non-supervisé, on ignore le label des données. On ne peut donc pas effectuer de contrôle de la prédiction aussi nettement que dans l'apprentissage supervisé. Le machine propose des résultats relativistes, en situant les données les unes par rapport aux autres, sans pouvoir fournir des réponses absolues. Comme la métrique de calcul choisie influence beaucoup les résultats, il faut en tester plusieurs.

Mais alors, comment trancher ? Deux solutions s'offrent au chercheur :

— raisonner "en moyenne" en s'appuyant sur les résultats de plusieurs métriques.
— confronter les résultats avec les hypothèses des experts du domaine. On peut se référer à un argument d'autorité pour écarter les résultats les moins probants et peut-être se prononcer plutôt en faveur de telle métrique qui retrouve des conclusions déjà mises au jour par d'autres moyens, à condition de rester prudent dans l'interprétation des résultats.

3.2.1 Distance de Manhattan

Nous commençons par calculer la distance de Manhattan entre les trigrammes de caractères. L'opération est renouvelée automatiquement de 200 à 1000 mots les plus fréquents, en incrémentant de 100 [17]. Comme on peut le constater sur la figure8, représentant le cluster pour 1000 trigrammes, les résultats sont concluants par endroits, surtout pour les textes disciplinaires (en bleu), mais les textes apologétiques (en rouge) ne semblent pas présenter de cohérence d'écriture particulière.

même format.

16. J. Delua, "Supervised vs. Unsupervised Learning : What's the Difference ?", *IBM Cloud*, march 2021, lien.

17. Voir les autres dendrogrammes enregistrés dans le dossier "Resultats".

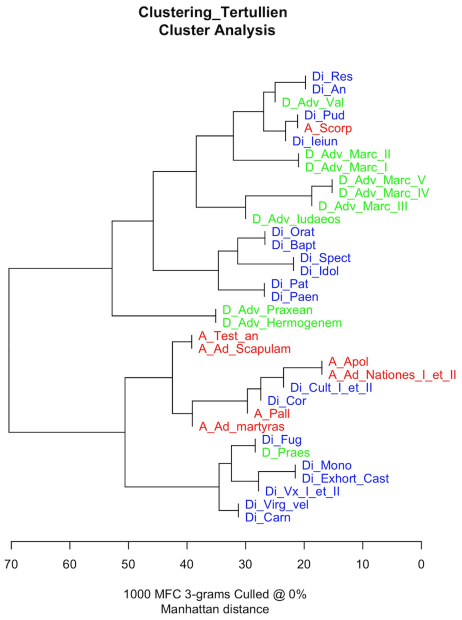

FIGURE 8 – Clustering des oeuvres de Tertullien (Manhattan, 1000 MFC 3-grams)

3.2.2 Consensus trees

Pour synthétiser ces résultats difficiles à interpréter, on produit quatre "consensus tree" sur les trigrammes de caractères, avec un consensus réglé à 0.5. On essaie en utilisant la distance de Manhattan, la distance cosine (figure9, puis le delta de Burrows (figure10) et enfin la métrique euclidienne.

Avec la distance cosine, les oeuvres contre les hérétiques sont mieux groupées qu'avec le delta de Burrows. On retrouve les mêmes textes au niveau des sous-branches, mais le delta de Burrows oppose les derniers livres de l'*Adversus Marcionem* aux premiers volumes. Les textes apologétiques forment également une catégorie plus définie avec la distance cosine qu'avec le delta de Burrows. La proximité unissant l'*Ad nationes* à *Apol* dans les deux consensus trees n'est pas étonnante, puisque le premier est le brouillon de l'autre. Globalement, la distance cosine semble mieux s'appliquer au problème.

3.2.3 Distance de Würzburg

On teste une autre méthode de calcul de distance, en augmentant le nombre de caractères les plus fréquents à prendre en compte. Ainsi, avec la distance de Würzburg appliquée aux 4743 trigrammes de caractères les plus représentés, on obtient un dendrogramme11 et une représentation sous forme de multidimentional scaling12.

Ce dernier stemma regroupe nettement les écrits selon leur catégorie d'appartenance. Pour les textes anti-hérétiques (en vert), seulement l'*Adversus Valentinianos* et, à plus forte raison, le *De Praescriptionibus* sont isolés. Le phénomène

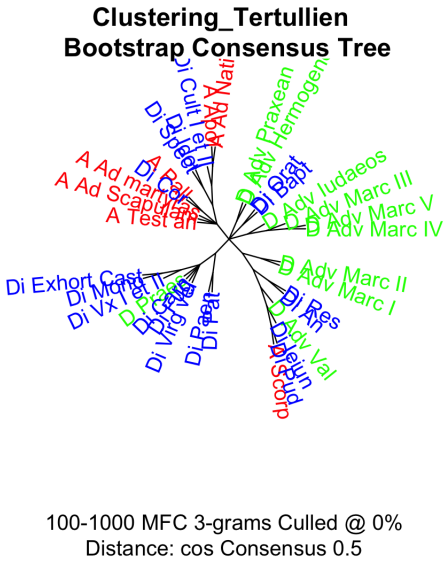

FIGURE 9 – Consensus tree (Cosine, 1000 MFC 3-grams)

se retrouve dans les autres figures (1000 MFW Manhattan, Consensus tree Cosine et Consensus tree Burrow's Delta), ce qui lui confère quelque solidité. Force est de reconnaître que l'*Advsersus Valentinianos* ne ressemble guère aux autres écrits de controverse, rien qu'à la lecture. Si d'habitude Tertullien explique leurs erreurs aux hérétiques à l'aide d'un arsenal logique, il adopte une toute autre stratégie dans l'*Adversus Valentinianos* : la dérision. Il jette le discrédit sur les thèses des Valentiniens en mettant en scène - presque littéralement - la génération des éons d'après la gnose qu'ils professent. Aussi n'est-ce guère surprenant de retrouver ce texte aussi éloigné de ceux dont il partage les visées. En revanche, il faudrait étudier à l'échelle microscopique comment se traduit sa proximité avec le *De anima*.

Un cas fort intéressant pour la recherche sur le Carthaginois est celui du *De Pallio*, considéré comme l'un des textes les plus énigmatiques de l'Antiquité tardive. Dans ce court texte où Tertullien déploie tous les raffinements rhétoriques de la Seconde Sophistique avec un brio qui rappelle celui d'Apulée dans les *Florides*, l'auteur justifie son choix d'avoir troqué la toge du citoyen romain contre le manteau des philosophes. Il ne mentionne qu'une seule fois la religion chrétienne, et à mots couverts. Ce texte qui a fait couler beaucoup d'encre est souvent considéré à part dans l'oeuvre de Tertullien, comme n'appartenant à aucune des trois catégories bien définies de sa production littéraire. Récemment, on a proposé de le lire comme un écrit apologétique [18]. Sa proximité avec les textes apologétiques (en rouge), manifeste non seulement grâce la méthode de Würzburg (3-grams, 4743 MFC), mais aussi avec la distance de Manhattan et, de manière moins nette, sur le consensus tree utilisant la distance cosine, pourrait éventuellement corroborer cette hypothèse interprétative.

18. F. CHAPOT, "Tertullien, *De Pallio* : le conflit des interprétations", Revue des Études latines 91 (2013), p. 191-210.

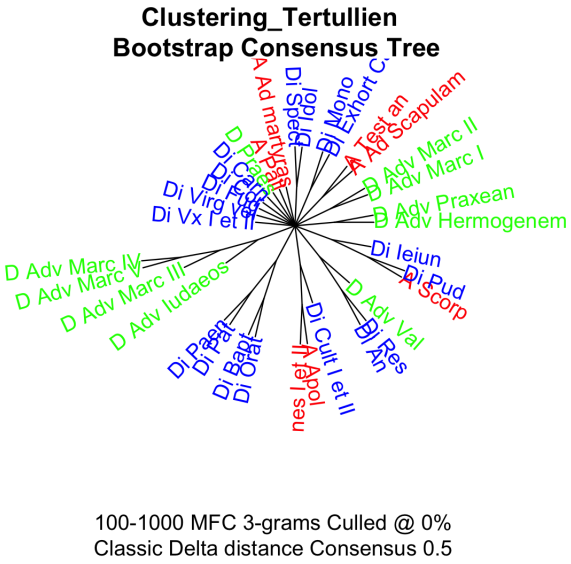

FIGURE 10 – Consensus tree (Burrows' Delta, 1000 MFC 3-grams)

3.3 Résultats

En conclusion, avec des gradients différents, les métriques appliquées successivement tendent à confirmer, *primo*, que la tripartition des oeuvres du Carthaginois se retrouve dans sa manière d'écrire, quand on s'intéresse aux trigrammes de caractères. *Secundo*, plusieurs méthodes (4743 MFC 3-grams Würzburg, 1000 MFC BTS 3-grams Cosine et 1000 MFC BTS 3-grams Burrow's Delta) semblent confirmer le classement des traités qu'ont proposé les spécialistes et pourraient apporter de nouveaux arguments dans les discussions portant sur les traités plus délicats, comme le *De pallio*.

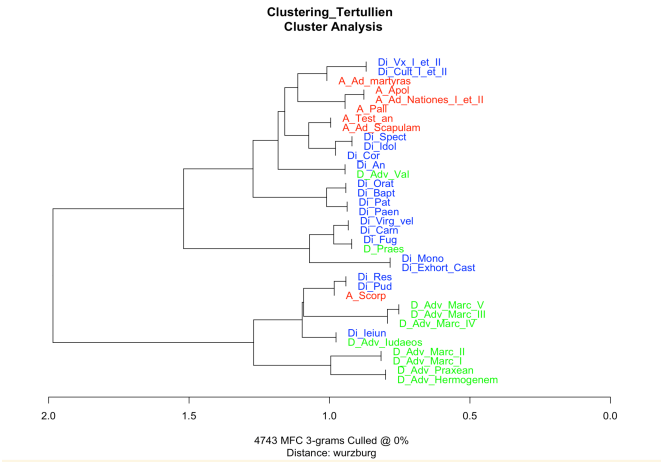

FIGURE 11 – Clustering des oeuvres de Tertullien (Würzburg, 4743 MFC 3-grams)

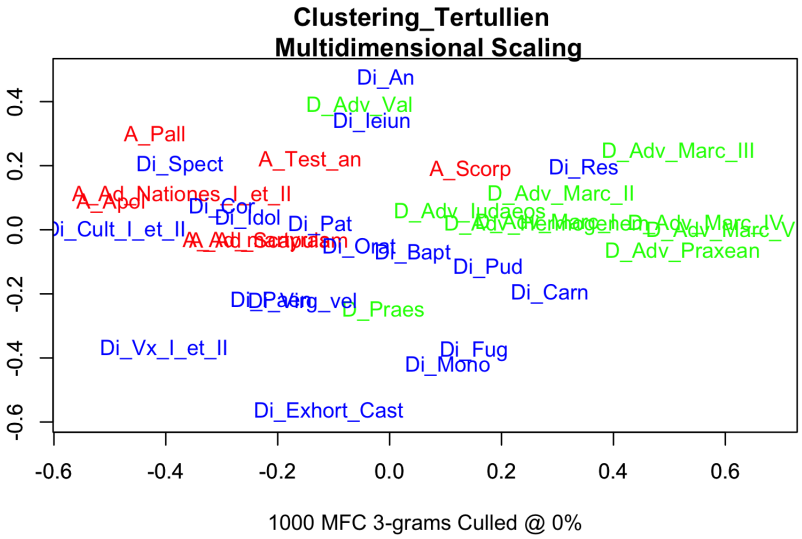

FIGURE 12 – Multidimensional scaling (Würzburg, 4743 MFC 3-grams)