M2SDL

Rapport du projet de stylométrie V1

BUTH Richy, DUCEL Fanny

Analyse de poèmes des Mazarinades

Master 2

« Sciences du Langage : Langue et informatique »

1 Introduction

Les Mazarinades sont des textes satiriques ou burlesques (poèmes, pamphlets, libelles, en vers ou en prose) publiés à l'époque de la Fronde (1648-1653), principalement pour critiquer, mais aussi parfois pour défendre, le cardinal Mazarin qui menait alors une politique répressive envers les chrétiens protestants de France [Wikipédia, 2022] [1].

L'objectif de ce projet est d'analyser ces poèmes pour extraire des sous-corpus et des statistiques, par exemple : quelle proportion de poèmes est en prose ou en vers par année ? Quelles sont la structure et la métrique des vers (octosyllabes, alexandrins, ...) ?

Tout notre code et nos données sont disponibles sur GitHub : `https://github.com/FannyDucel/projet-stylo-poesie`.

2 Description des données

Nous avons un corpus des Mazarinades de 3065 fichiers au format .xml, recueillis et formatés grâce au projet Antonomaz [2].

Grâce au balisage xml, il est possible d'extraire de nombreuses métadonnées sur ces textes, comme leur date de création, le type de poèmes (vers ou prose), des descriptions et remises en contexte de ces textes, les sauts de lignes et fins de strophes, des mots-clés, ou bien encore les bibliothèques d'où sont tirés les textes.

1. https ://fr.wikipedia.org/wiki/Mazarinade
2. `https://github.com/Antonomaz/Corpus`

3 Méthodes employées

Nous avons utilisé BeautifulSoup [3] pour parser les documents xml. Cette librairie nous a permis de récupérer les informations comprises entre des balises.

Nous avons calculé la proportions de poèmes en vers et en prose en cherchant la balise *term type="form"* et en comptant leurs occurrences. En les couplant avec la balise *date*, nous avons pu obtenir les proportions pour chaque année. Comme les dates sont au format AAAA-MM-JJ, il a suffit de lire les quatre premiers chiffres à chaque fois pour obtenir l'année.

Nous avons eu des problèmes avec les exceptions, comme des textes sans date ou dans un format différent, comme "JJ MM AAA".

Nous avons cependant stocker dans une structure de données tous les textes en vers. Cela prend la forme d'un dictionnaire associant à chaque chemin de fichier en vers sa liste de vers (grâce aux balises l). Nous travaillons par la suite uniquement sur ces textes et ces vers.

Nous avons donc dû voulu procéder au découpage syllabique de nos vers. Tout d'abord, nous avons voulu exploiter le programme utilisé sur le site `http://www.sciencedutexte.fr/versification` et présent sur `https://github.com/humanitesnumeriques/outilDeSyllabation`.

Toutefois, le code est en XQUERY et n'est pas facilement exécutable ou implémentable en Python. De même, la structure HTML nous a semblé trop complexe pour effectuer du WebScrapping. Nous avons alors décidé d'abandonner cette piste : bien que les résultats soient qualitatifs et riches, le coût d'entrée était bien trop important.

Nous nous sommes alors tournés vers le programme french-syllabification trouvé sur GitHub [4]. Il utilise la base Lexique [5] qui contient le découpage syllabique pour 140 000 mots du français.

Pour les mots non-présents dans le Lexique, le programme calcule lui-même le nombre de syllabes des mots.

3. `https://www.crummy.com/software/BeautifulSoup/`
4. `https://github.com/ian-nai/french-syllabification`
5. `http://www.lexique.org/`

Ce programme tel quel était très imparfait, tant dans les résultats de découpage que dans sa complexité algorithmique qui le rendait très peu optimisé pour le traitements de gros corpus. Il a donc nécessité des modifications avec l'aide de Gaël Lejeune pour optimiser le code et accélérer le temps d'exécution du programme.

Grâce à ces premières modifications, nous avons donc pu exécuter le code à l'échelle de tout le corpus de vers (634 fichiers). Nous avons également décidé de mettre nos vers tout en minuscules, car les majuscules ne sont tout simplement pas traités par l'outil et renvoient donc des vers à 0 syllabe ou diminuent d'une syllabe certains vers.

Nous avons ensuite implémenté un premier filtre sur les résultats obtenus, mais plutôt pour corriger des problèmes liés aux données en elles-mêmes. En effet, certains vers sont composés d'un seul caractère ou contiennent des chiffres ; nous les ignorons.

Nous pouvons ensuite procéder au comptage global des syllabes par vers (nous avions jusqu'à présent un comptage par mot) et obtenons ainsi le nombre de vers par type (nombre d'octosyllabes, ...).

4 Résultats

4.1 Proportions de textes en vers et en prose

Grâce à l'extraction des balises vue dans la section 2 nous pouvons obtenir des statistiques sur les proportions de textes en vers et en prose dans tout le corpus des Mazarinades (voir Table 1). Nous en avons également fait des graphiques (voir Figures 1 à 2).

Date	**Vers**	**Prose**
Sans Date	24 34%	46 66%
1634	1 100%	0 0%
1648	4 11%	34 89%
1649	432 30%	1030 70%
1650	78 27%	213 73%
1651	65 19%	279 81%
1652	120 12%	845 88%
1653	2 22%	7 78%
1654	1 9%	10 91%
1655	0 0%	2 100%
1656	0 0%	2 100%
1662	0 0%	1 100%
1663	1 100%	0 0%
Total	728 22,8%	2469 77,2%

TABLE 1 – *Proportions de poèmes en vers et de poèmes en prose par année*

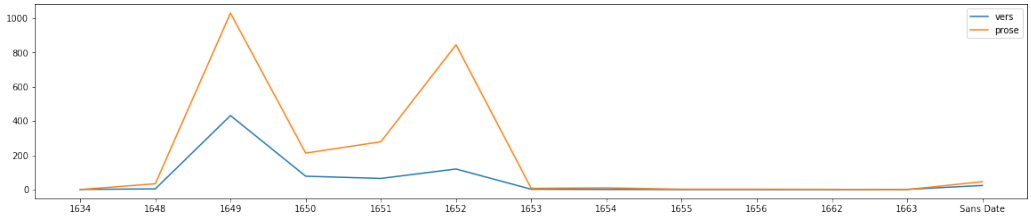

FIGURE 1 – *Proportion de poèmes en valeur absolue*

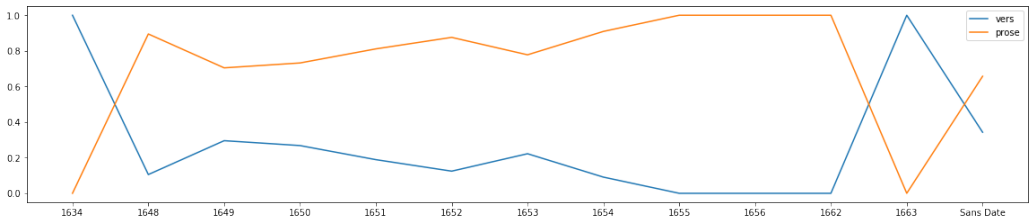

FIGURE 2 – *Proportion de poèmes en valeur relative*

4.2 Découpage syllabique : types de vers

De par l'imperfection du découpeur syllabique automatique, il est impossible d'obtenir un texte en vers parfaitement structuré et découpé où l'on pourrait déterminer avec certitude une certaine métrique.

Cependant, la vue d'ensemble nous permet de faire des rapprochements : en effet, on remarque l'écart de syllabes entre les différents vers est très minime 2. On fait également ces statistiques sur tout le corpus (voir Tableau 3).

Si l'on suppose que ces légers écarts ne sont liés qu'à un découpage imparfait, il est possible de déterminer avec plus ou moins de certitude la métrique du poème.

À cela s'ajoutent les méta-données qui nous expliquent le contexte des Mazarinades, notamment qu'il s'agit souvent de poèmes en vers octosyllabiques. Nous pouvons en effet remarquer dans nos exemples que les longueurs de vers avoisinent généralement les 8 syllabes.

Fichier	Nombre de vers par métrique
Moreau1174_MAZ.xml	5 syllabes : 17 6 syllabes : 43 7 syllabes : **71** 8 syllabes : **56** 9 syllabes : 18
Moreau1121_GALL.xml	5 syllabes : 42 6 syllabes : **116** 7 syllabes : **160** 8 syllabes : 99 9 syllabes : 32
Moreau1117_GALL.xml	8 syllabes : 13 9 syllabes : 18 10 syllabes : **38** 11 syllabes : **31** 12 syllabes : 16

TABLE 2 – *Exemples de découpage syllabique par vers et par poèmes*

4.3 Exemples de vers et leur nombre de syllabe présumé

Nous reportons ici un exemple de vers par nombre de syllabe trouvé par le découpeur automatique [6]. Celui-ci nous permet de mettre en avant de nouveaux problèmes à résoudre par la suite (voir Tableau 4). Les problèmes semblent toutefois liés aux données en elles-mêmes plutôt qu'au découpeur syllabique. Nous devrons donc décider si nous choisissons d'adopter une approche de correction de données ou d'émettre des hypothèses sur les résultats actuels du découpeur syllabique en écartant les résultats étranges.

6. À noter : dans le tableau, nous remplaçons les symboles de s longs anciens par des S majuscules

Nombre de syllabes	Nombre de vers
7	42272
8	30088
6	28825
9	12970
5	10788
10	8055
11	6351
12	3837
4	3488
13	1864
3	1552
2	1419
14	1042
15	677
16	340
17	148
1	77
18	67
19	19
20	15
21	14
22	5
23	3

TABLE 3 – *Nombre de vers par nombre de syllabes dans tout le corpus*

5 Conclusion

La poésie présente de nombreux problèmes de traitement automatique. Le découpage syllabique des vers nécessite l'utilisation d'un grand lexique de mots du français déjà annotés et découpés syllabiquement. Dans le cas contraire, il nécessite l'implémentation de nombreuses règles du français en prenant en compte les exceptions pour faire un découpage automatique.

À cela s'ajoute le problème lié au français du XVIIème siècle, très différent du français moderne de par son vocabulaire, son orthographe et sa prononciation. Implémenter des règles du français moderne n'est donc pas suffisant, il a fallu adapter le code avec de nouvelles règles, augmentant ainsi la tâche de travail et la complexité du programme.

Nb syll	**Vers**
0	e'dtp : :ttéd'
1	theS vertS,
2	ta place,
3	d'eStre criminel, d'eStre fugf,
4	eiiS coront vS.
5	re ne ene nee ene
6	ieu cheueux cendrez, dont meS maiuS vag¬
7	eSpnit Son corpS furent incompatibleS,
8	boiS-robert abbeé de chaStillon.
9	qu'apreS' tant de deSordre, apreS tant de licence,
10	tu viSSeS à teS piedS pleurer teS ennemiS,
11	peut atteindre au Sommet de la pluS haute gloire.
12	voir d'vn œil de pitie ceux qui t'ont fait laloy,
13	voicy qui vaut norlingue, &l le havre a la fin
14	te leS lS. l'albitre le couail. nouS fappa d'vu Seul coup, le œur l'o¬
15	nn eoi ne enple See nel aS eonie que eoin le elioe peue en ne
16	leoe nunon ue nS oene, nunn une nailS een eril Suei re eei ne ne ene
17	troueoiS dau la eionuS pui bele ve, fuit Sie dunmen judeareur rerr,
18	iSi eiSre ne leionS ne ie enui nonunonuniS, lenminan SiroSd leeou, nen une
19	ln ei eil deSienSe de nenroi onu in qeu, deS veoenS de dauS ete nei See reS
20	leue uou due oa anure Soirlionail pe pionpe neS le ee que oianu ile,
21	lenS nee enine uni lene de ie eaounS, pdenolunie ni leai le nuuS laur ne ei ei.
22	pue ne iror auienude nS SieleS nerniruble, lueni Senilemeone, panSe que ilS le
23	ntre leS braS deS enS ie tombay demy-morte, tel il faut que tu SoiS danS ton aduerSité,
27	le eai Sit ehanuer le te deum pour la pif -de edaratiuSaiuer S'ar dr qe e, furenr onueu au pa¬
31	eeeeeeee-e-e-adaaea eae-efaea-eeaoieeae-Scaaaeaéeoa–eoeeea-a¬¬¬

TABLE 4 – *Exemples de vers par nombre de syllabes*

Néanmoins, grâce aux méta-données mises à disposition par l'auteur du corpus annoté des Mazarinades il a été possible de récupérer facilement les informations telles que la date ou le type de texte (vers ou prose) pour obtenir des statistiques en amont.

Cette abondance de méta-données aux format .xml a cependant posé des soucis de parsing avec Python. Il a été difficile d'analyser la structure des fichiers pour trouver les bonnes balises à récupérer et à retirer, ce qui nous montre que le gain en quantité d'informations méta-textuelles se fait au

détriment de la lisibilité des fichiers leur facilité de traitement.

5.1 Perspectives

Pour la suite du projet, nous souhaitons améliorer notre découpage syllabique, notamment en regardant de plus près les données et en effectuant des modifications sur les données (on a remarqué que le problème de la dissimilation est mal géré, nous pourrions donc remplacer les v par des u quand nécessaire).

Références

Mazarinade, Wikipédia, https ://fr.wikipedia.org/wiki/Mazarinade, mis à jour le 21 novembre 2022 à 10 :15