La mort d'Ivan Illitch en modélisation quantitative

Jean Barré

8 juin 2022

Résumé

Ce travail envisage le roman de Tolstoï avec une approche quali-quantitative pour modéliser la narration au fil de l'intrigue et comprendre comment le récit nous amène à la mort d'Ivan Illitch. Nous employons les méthodes du Traitement Automatique des Langues (TAL) et de l'apprentissage machine, et de la modélisation de sujet pour sonder quantitativement le roman. Ces méthodes ne sont rien sans un retour en lecture proche et une analyse fine des inférences statistiques des modèles.

1 Enjeux de recherches

La mort d'Ivan Illitch est un roman de Léon Tolstoï publié en 1886. C'est un ouvrage remarquable qui présente l'histoire d'Ivan Illitch, magistrat et bourgeois ordinaire, et sa lente descente aux enfers. Cette dernière se fait sous deux aspects : la première est d'ordre physique, puisque Ivan Illitch, à la suite d'un accident ordinaire (il tombe en s'occupant de ses rideaux), s'enfonce dans la souffrance qui se termine finalement par sa mort. La seconde est d'ordre psychologique, puisque Ivan Illitch prend conscience de la médiocrité de sa vie lorsqu'il est sur le point de mourir. En effet, le protagoniste principal a tout dédié à son travail et à son paraître pour s'affirmer socialement, et délaisse sa vie de couple, sa vie familiale, et amicale au profit de sa carrière et de son importance sociale. Avec son accident et sa lente périclitation, Ivan Illitch lève le voile sur sa vie et découvre que celle-ci est exclusivement représenté par le mensonge, l'hypocrisie, la solitude et les conventions sociales. Cet aveu terrible d'avoir manqué sa vie est fait au bord de la mort, ce qui l'enfonce dans des souffrances encore plus atroces, dont seule la mort pourra le délivrer.

Dans ce travail, nous allons nous focaliser sur cette lente agonie au fil du récit en prenant comme objet d'étude le chapitre. En effet, nous voulons comprendre la manière dont le récit plonge dans la souffrance et la mort et comment cela s'articule à l'échelle des chapitres. Le chapitre est un élément de structuration textuelle très important. Il permet à l'auteur de mettre différentes étapes dans l'avancée du récit. *La mort d'Ivan Illitch* est composé de douze chapitres, de différentes longueurs.

Le tableau 1 décrit la longueur des chapitres en nombre de tokens, c'est à dire en nombre de mots. La figure 1 montre son évolution au fil des chapitres.

Avec ces premières descriptions statistiques, nous pouvons formuler un premier constat : le récit s'articule en quatre phases. Une première étape d'entrée dans le récit

TABLE 1 – Statistiques du corpus

chap_01	3698 tokens
chap_02	3314 tokens
chap_03	3155 tokens
chap_04	3186 tokens
chap_05	1919 tokens
chap_06	1375 tokens
chap_07	1897 tokens
chap_08	2962 tokens
chap_09	1213 tokens
chap_10	965 tokens
chap_11	1170 tokens
chap_12	996 tokens
Moyenne par chapitre	2154 tokens
Texte entier	25850 tokens

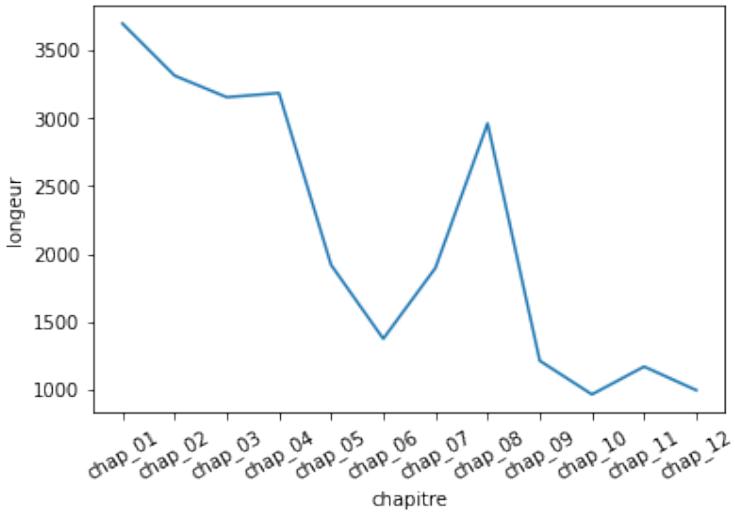

FIGURE 1 –

de quatre chapitres longs, puis une deuxième phase marque une accélération franche dans les chapitres 5, 6 et 7 qui sont en comparaison assez courts. Puis le chapitre 8 marque un dernier ralentissement notable avant la fin du récit. En effet, les chapitres 9, 10, 11 et 12 sont très courts et représentent 1/3 d'une longueur de chapitre moyen. Le narrateur réduit son discours au fur et à mesure que le récit avance comme si Ivan Illitch courrait vers sa mort et le dénouement final.

Ainsi, quelque chose se joue à l'échelle du chapitre et nous voulons faire le diagnostique textuel de l'avancée du récit. Qu'est-ce qui dans le lexique du récit, traduisent cette accélération du récit ? Quel est le programme d'écriture de Léon Tolstoï et comment se justifie la structure des chapitres. Telles sont les questions que l'on se pose dans ce travail, et nous allons essayer d'y répondre à l'aide des techniques quantitatives de TAL et d'apprentissage machine. Plus précisément, nous nous demandons si un modèle statistique est capable de détecter l'avancée narrative du récit, et nous voulons mettre

au point un modèle statistique capable de labelliser automatiquement un morceau de texte comme appartenant à un chapitre précis du roman.

2 Méthode

2.1 Une approche quali-quantitative

Cette présente étude ce trouve au carrefour de plusieurs disciplines, entre sémiotique quantitative, stylométrie et études littéraires computationnelles. Grâce aux progrès de l'informatique et du traitement du langage naturel, l'utilisation d'approches statistiques et mathématiques dans le domaine des études littéraires a considérablement augmenté ces dernières années. Ce nouveau champ de recherches permet d'augmenter la focale et d'apporter des perspectives nouvelles dans ces disciplines. Nous proposons de suivre une approche quali-quantitative, qui utilise les techniques quantitative pour constater et vérifier des éléments en lecture proche. Cette approche se caractérise par un perpétuel mouvement entre ces deux aspects, pour renforcer les hypothèses de base et les résultats au fur et à mesure de la recherche.

2.2 « Operationalisation » et caractéristiques textuelles retenues

Le roman dont nous proposons l'étude quantitative est finalement assez limité en terme de longueur totale, et pour justifier l'implémentation des méthodes d'apprentissage machine nous décidons de découper nos textes en morceaux de 100 mots. A chaque morceaux est associé une étiquette, le chapitre auquel il appartient. Nous augmentons ainsi le nombre de points d'étude dans notre texte.

Nous décidons de prendre comme caractéristiques textuelles les lemmes du texte, avec les 500 uni-grammes les plus fréquents du roman et nous rajoutons aussi les 500 bi-grammes les plus fréquents du roman. Deux raisons à cela, une première pratique, puisque la prise en compte de tous les mots du roman aurait nécessairement amené à des coûts computationnels importants, car la matrice résultante aurait été très éparse, avec beaucoup de fréquences d'apparitions nulles ou proches de zéro. La seconde raison réside dans la nature des mots que nous récupérons. Comme ce sont les mots les plus fréquents du corpus, la plupart sont des déterminants, prépositions et autres *mots-outils*.

Ces derniers relèvent plus d'une écriture inconsciente et automatique des auteurs qu'à des mots moins fréquents relatifs au contenu et aux thèmes du textes. Ces mots-outils sont au coeur de la stylométrie, notamment dans les attributions d'auteur, et dans l'étude des idiolectes, c'est à dire de la signature textuelle d'un écrivain. Ces méthodes ont produit de très bons résultats, allant de Hildegarde de Bingen [1] ou Molière [2].

Si le problème auquel nous faisons face n'est pas de même nature, nous pensons que ces techniques sont pertinentes pour traiter notre problème. En effet, si le programme

1. Mike Kestemont, « Function Words in Authorship Attribution. From Black Magic to Theory ? », dans *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, Gothenburg, Sweden, 2014, p. 59-66, DOI : 10.3115/v1/W14-0908.

2. Florian Cafiero et Jean-Baptiste Camps, « Why Molière most likely did write his plays », *Science Advances*, 5–11 (nov. 2019), eaax5489, DOI : 10.1126/sciadv.aax5489.

d'écriture de Léon Tolstoï prévoit des éléments spécifique chapitre après chapitre, nous devrions pouvoir détecter quantitativement des changements dans la manière d'écrire de l'écrivain.

Pour quantifier les fréquences d'apparition de mots, on réduit les unités lexicales sujettes à flexion (les verbes, les substantifs, les adjectifs) à leur unité lexicale commune. On appelle ce processus lemmatisation. Pour ce traitement, nous utilisons la librairie Spacy. Elle nous permet de tokeniser, lemmatiser et de nettoyer les chapitres en contrôlant l'étiquetage morphosyntaxique des tokens.

Nous décidons de transformer nos ouvrages en sac-de-mots. Les textes sont ainsi décomposés en des listes de mots qui indexent leur fréquence relative (le nombre de fois où l'unité apparaît, divisé par la longueur totale du texte). Chaque unité est traitée comme une caractéristique des textes dans lesquels elle apparaît - une sorte de trait d'identification - et le texte devient un vecteur de ces traits.

2.3 Outils de programmation

Nous fondons notre travail sur le langage de programmation Python et les librairies construites au-dessus. Pour l'analyse des données et leur manipulation nous utilisons Pandas [3] et Numpy [4]. Pour le traitement du texte à proprement dit, nous employons la librairie Spacy [5]. Cette dernière est très performante pour une analyse sur de grandes quantités de données et couvre tous les traitements d'une chaîne de TAL classique. Cette dernière a aussi l'avantage d'être très bien documentée et comporte plusieurs modèles de langages pour le français. Au vu des performances des différents modèles, nous prenons la décision d'utiliser le modèle fr_core_news_lg qui a un très bon rapport d'utilisation de ressources temporelles et matérielles entre exécution et performance. Nous utilisons aussi Scikitlearn [6], qui donne des outils efficaces pour l'analyse prédictive de données. Scikitlearn est assez simple à utiliser et implémente des algorithmes d'apprentissage machine au niveau de l'état de l'art. Pour la visualisation des résultats nous utilisons les librairies seaborn [7] et matplotlib [8].

3 Méthode Supervisée

Nous voulons observer si des différences statistiques majeures existent entre nos chapitres, nous faisons appel pour cela au champ de recherches de l'apprentissage machine.

La classification automatique de textes est un problème très étudié en statistiques. Une famille de modèles retient particulièrement notre attention parce qu'elle obtient de bons résultats pour la classification de textes littéraires : les Machines à Vecteur de Support (SVM) [9]. Les SVM ont pour but de trouver les plans qui séparent les points de données avec les marges maximales entre les frontières de décision. Ils traitent les

3. https ://pandas.pydata.org/
4. https ://numpy.org/
5. https ://spacy.io/
6. https ://scikit-learn.org/stable/index.html
7. https ://seaborn.pydata.org/
8. https ://matplotlib.org/
9. Corinna Cortes et Vladimir Vapnik, « Support-vector networks », *Machine Learning*, 20–3 (sept. 1995), p. 273-297, DOI : 10.1007/BF00994018.

caractéristiques comme des coordonnées dans un espace cartésien à haute dimension et tentent de tracer une ligne qui divise au mieux les caractéristiques uniques d'une classe. Pour l'estimateur SVM, un texte n'est qu'une combinaison de caractéristiques qui tendent à apparaître plus souvent dans une classe de textes que dans une autre. Nous utilisons dans ce travail la famille de SVM développé par l'équipe scikit-learn depuis 2011. Nous avons affaire à une approche très connue en apprentissage machine : l'apprentissage supervisé. Ce dernier est assez simple à comprendre, puisqu'il associe un ensemble de données avec une certaine classe labellisée. Un modèle statistique est évalué sur sa capacité à faires des inférences entre les particularités des données et une certaine classe. Les classes de nos morceaux de textes correspondent à quel chapitre ils appartiennent.

Nous mettons en place les bases de l'apprentissage machine. Le jeu de données est séparée en deux échantillons.

— Un premier sur lequel nous entraînons le modèle statistique, c'est à dire que nous lui donnons le label associé pour chaque roman.
— Un autre sur lequel nous évaluons ses performances de prédictions sur des données qu'il n'a jamais vu.

Nous mesurons ainsi la capacité du modèle à généraliser. Nous voudrions que la taille de l'échantillon du corpus d'entraînement soit la plus large possible, pour donner toutes ses chances au modèle. Pour autant, il est important de garder une taille conséquente pour l'échantillon de test afin de mesurer à quel point le modèle est capable de réaliser de bonnes prédictions sur un grand nombre de données. Nous fixons la taille de l'échantillon test à 20% du total. Nous implémentons grâce à Scikitlearn un pipeline avec un pré-traitement des données, un StandardScaler, et un estimateur, le SVM. Ce pré-traitement permet de normaliser nos données. La normalisation d'un ensemble de données est une exigence commune à de nombreux estimateurs d'apprentissage automatique : ils peuvent mal se comporter si la distribution statistique ne ressemble pas à des données distribuées sous forme d'une loi normale.

Nous évaluons notre modèle grâce à des métriques d'évaluation de la performance : l'efficacité, la précision, le rappel et un f1-score. L'efficacité est la métrique la plus simple à comprendre, puisque c'est le pourcentage d'éléments prédits correctement par le modèle. Nous nous focaliserons sur cette métrique dans notre cas.

4 Résultats

4.1 Première approche

Les résultats de la première approche décrite précédemment sont un peu décevants. Comme on peut le voir dans le tableau 2, le modèle atteint 0.47 d'efficacité, c'est à dire que le modèle arrive à prédire le bon label pour un morceau de texte donné une fois sur deux. Comme nous disposons de 12 classes (nos 12 chapitres), cela reste encourageant.

Pour comprendre un peu mieux ce qui ce joue dans ce résultat, nous projetons la matrice de confusion du modèle. Dans le meilleurs des cas, une matrice de confusion est remplie au niveau de la diagonale, ce qui voudrait dire dans notre cas que chaque chapitre est prédit correctement.

Notre matrice de confusion en figure 2 montre bien que quelques problèmes résident

	precision	recall	f1-score	support	accuracy
chap_01	0.55	0.75	0.63	8	
chap_02	0.50	0.71	0.59	7	
chap_04	0.29	0.40	0.33	5	
chap_04	0.29	0.33	0.31	6	
chap_05	1.00	0.60	0.75	5	
chap_06	0.00	0.00	0.00	1	
chap_07	0.00	0.00	0.00	1	
chap_08	0.40	0.40	0.40	5	
chap_09	0.00	0.00	0.00	2	
chap_10	0.00	0.00	0.00	2	
chap_12	0.00	0.00	0.00	1	
full dataset				21	**0.47**
macro-average	0.71	0.77	0.71	21	
weighted average	0.81	0.76	0.75	21	

TABLE 2 – Résultats de l'évaluation du modèle avec les labels par chapitre

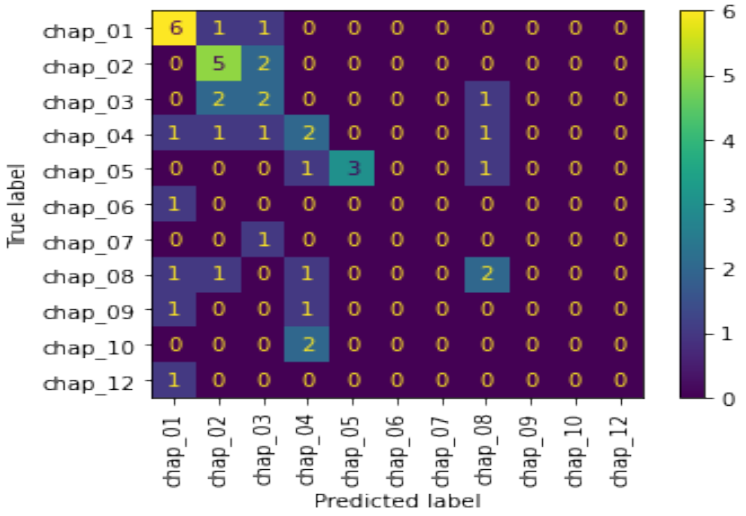

FIGURE 2 –

dans notre prédiction. En effet, seules les chapitres 1, 2 et 5 sont plutôt bien prédits, pour le reste des chapitres c'est bien plus compliqué. Nous faisons face à un manque de données flagrant, et nos chapitres 6, 7, 9, 10, 11 et 12 ne contiennent même pas de texte dans l'échantillon test. Pour palier cela, nous modifions un peu l'approche pour contourner le problème.

4.2 Seconde approche

La seconde approche consiste simplement à fusionner des chapitres. Cela nous permettra d'une part de réduire le nombre de classes pour faciliter le travail au modèle. D'autre part nous augmentons la taille des échantillons des différentes classes, ce qui augmente la probabilité que le modèle s'entraîne sur ces échantillons.

Pour déterminer la meilleure façon de fusionner les chapitres, nous implémentons une réduction de dimensions sur un tableau contenant en colonnes les 1000 caractéristiques respectives et en lignes les 12 chapitres. Le tableau est rempli par les fréquences d'occurrence des mots dans chaque chapitre.

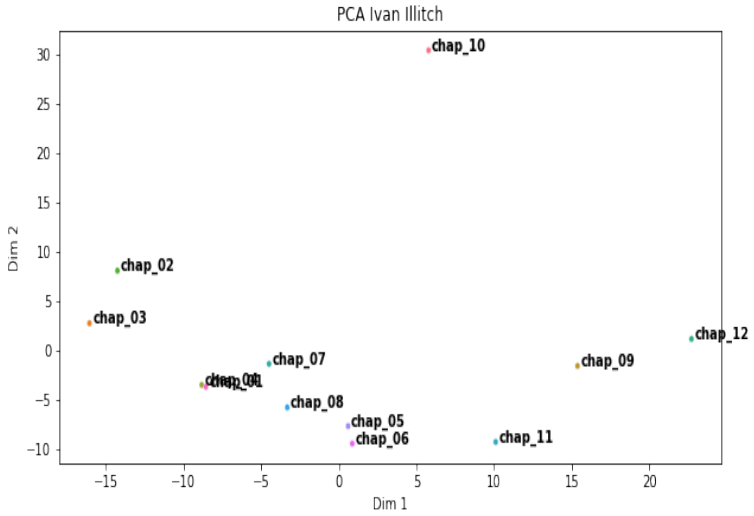

FIGURE 3 – Graphique en ACP de la position relative des Chapitres

Nous obtenons des résultats un peu surprenants, puisque le point du chapitre 1, est confondu avec celui du chapitre 4. Le chapitre 1 est pourtant sensé être assez unique, c'est en effet le seul chapitre qui se passe avant la mort de Ivan Illitch, avec le point de vue de Piotr Ivanovitch, que l'on suit jusqu'aux funérailles d'Ivan Illitch. Le quatrième chapitre est assez différent, il raconte les différentes visites de Ivan Illitch chez le médecin pour trouver l'origine de son mal. Autrement, nous avons comme chapitres similaires le 2 et le 3, qui sont deux chapitres très narratifs reprenant l'histoire détaillée de la vanité de la vie d'Ivan Illitch jusqu'à l'incident des rideaux et s'arrête avant le début des souffrances. Les chapitres 5 et 6 sont aussi similaires, ce sont deux chapitres assez courts qui racontent le début des souffrances d'Ivan Illitch et la longue litanie des rendez-vous de médecins pour soulager la douleur. Les chapitres 7 et 8 marquent la prise en importance dans le récit du personnage Guérassime, valet d'Ivan Illitch, dont la présence permet de soulager un peu la douleur. Nous apposons une étiquette commune à ces quatre couples de chapitres. Les chapitres 9, 10, 11 et 12, sont assez dispersés dans l'espace, et sont tous assez loin des autres chapitres projetés. Ces chapitres sont très courts, on voit bien que le récit accélère et que la mort d'Ivan Illitch est toujours plus proche. Ces chapitres sont trop courts pour être fusionnés à deux, nous prenons donc la décision d'apposer une seule étiquette pour les quatre.

Le tableau 3 montre les résultats du modèle avec la fusion des étiquettes des chapitres. Le modèle atteint 0.76 d'efficacité, ce qui est un très bon résultat par rapport

	precision	recall	f1-score	support	accuracy
chap_01_04	0.71	0.83	0.77	6	
chap_02_03	0.86	1.00	0.92	6	
chap_05_06	0.50	1.00	0.67	1	
chap_07_08	1.00	0.50	0.67	6	
chap_09_10_11_12	0.50	0.50	0.50	2	
full dataset				21	**0.76**
macro-average	0.71	0.77	0.71	21	
weighted average	0.81	0.76	0.75	21	

TABLE 3 – Résultats de l'évaluation du modèle avec les labels fusionnés

à notre première approche. Il est admis qu'une classification multi-classes est statistiquement solide à partir de 75% d'efficacité. Si l'on ressent encore les disparité dans les tailles des échantillons, (chap_05_06 et chap_09_10_11_12 sont très limités), notre modèle gagne en assurance dans la prédiction.

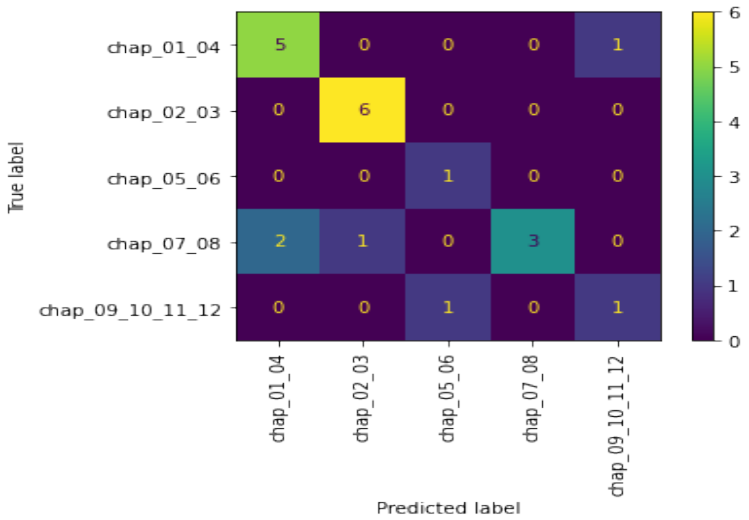

FIGURE 4 –

Nous projetons en figure 4 la matrice de confusion de notre modèle. Nous voyons que les erreurs se font moins nombreuses, et sont quasiment parfaites sur la diagonale pour les étiquettes chap_01_04, chap_05_06 et chap_07_08. Il reste toutefois des petits bugs, avec par exemple deux morceaux de textes appartenant aux chapitres 7 et 8 qui sont prédits comme étant l'étiquette chap_01_04.

5 Analyse qualitative des coefficients en nuage de mots

Il faudrait maintenant comprendre comment notre modèle statistique est véritablement capable de prédire la canonicité. Un des intérêts de l'apprentissage machine est d'ailleurs la possibilité de plonger dans les inférences réalisées par le modèle. En effet, on peut récupérer les coefficients que le modèle assigne à chaque caractéristique pour séparer nos labels. Pour ce faire, nous projetons le nuage de mots spécifique à chaque étiquette. A chaque mot est associé un coefficient, dont on se sert pour établir la taille dans le nuage de mot. Autrement dit, la taille du mot est proportionnelle à son poids dans les coefficients du modèle.

5.1 chapitres 1 et 4

Dans la figure 5, nous projetons le nuage de mots des coefficients les plus discriminants selon le modèle pour assigner l'étiquette chap_01_04.

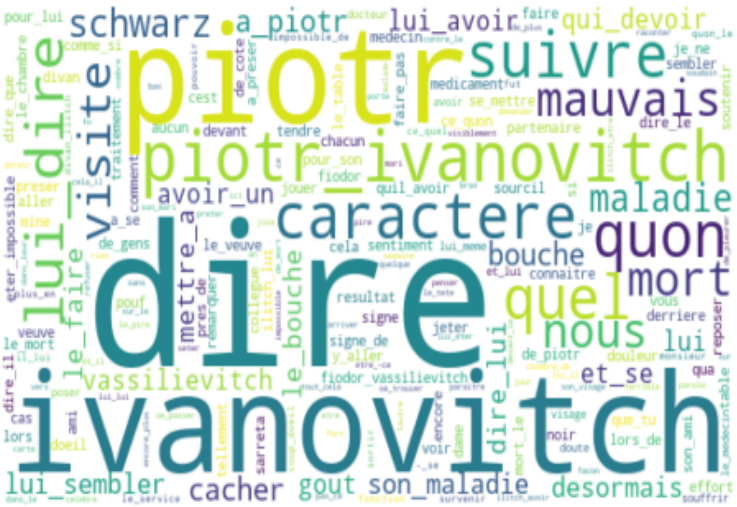

FIGURE 5 –

L'ami d'Ivan Illitch, Piotr Ivanovitch ressort nettement dans ce nuage de mots. Les uni-grammes « piotr » et « ivanovitch » ainsi que le bi-gramme « piotr_ivanovitch ». Cela est assez cohérent puisque ce dernier est mentionné en grande majorité dans le chapitre 1, avec une mention dans le chapitre 4. Il en va de même pour un autre amis d'Ivan Illitch, « Schwartz ». Nous n'avons pas plus d'explication pour le verbe « dire », si ce n'est la présence de dialogues dans le chapitre premier.

5.2 chapitres 2 et 3

Pour les chapitres 2 et 3, les caractéristiques discriminantes en figure 6 sont assez pertinentes, avec par exemple le mot « mariage » qui est très présent dans le chapitre 2 où est narré l'histoire du mariage entre Ivan Illitch et Prascovia Fiodorovna. Pour le mot « ancien », l'explication se trouve au chapitre 3, avec pas moins de 7 mentions sur un total de 10 dans tout le récit. Ces mentions se trouvent lorsque nous est raconté les évolutions professionnelles d'Ivan Illitch et son rapport factice avec ses anciens collègues.

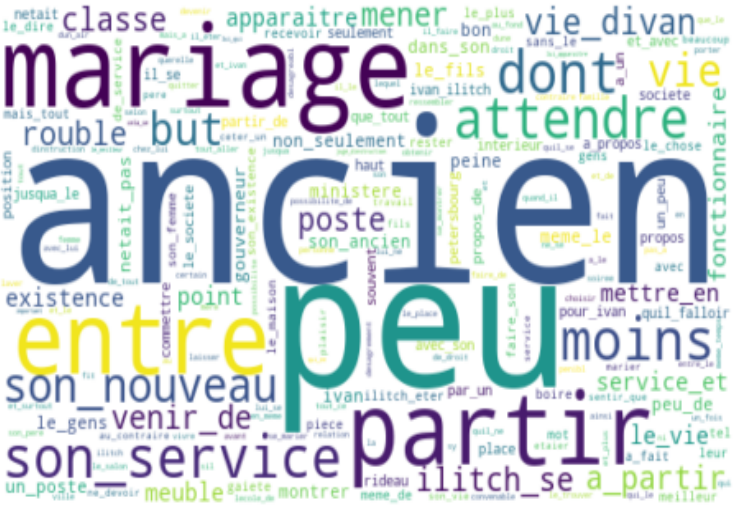

FIGURE 6 –

Il est intéressant de noter la présence du mot rideau dans ce nuage de mots (en petit, en dessous de partir). L'accident anodin à premier abord a lieu dans le chapitre 3, et se trouve être la cause des souffrances d'Ivan Illitch.

5.3 chapitres 5 et 6

Pour les chapitres 5 et 6, les caractéristiques discriminantes en figure 7 sont assez bruitées, avec de nombreux nombreux bigrammes pas vraiment porteur de sens.

FIGURE 7 –

Cela pourrait s'expliquer par les moins bonnes performances du modèle sur cette classe chap_05_06, à cause du manque d'échantillons. Le modèle ne s'est pas suffisamment entraîné dessus et n'a pas pu produire des coefficients pertinents.

5.4 chapitres 7 et 8

Pour les chapitres 7 et 8, les caractéristiques discriminantes en figure 8 sont assez propres. Les mots « malade » et « mensonge » ressortent énormément.

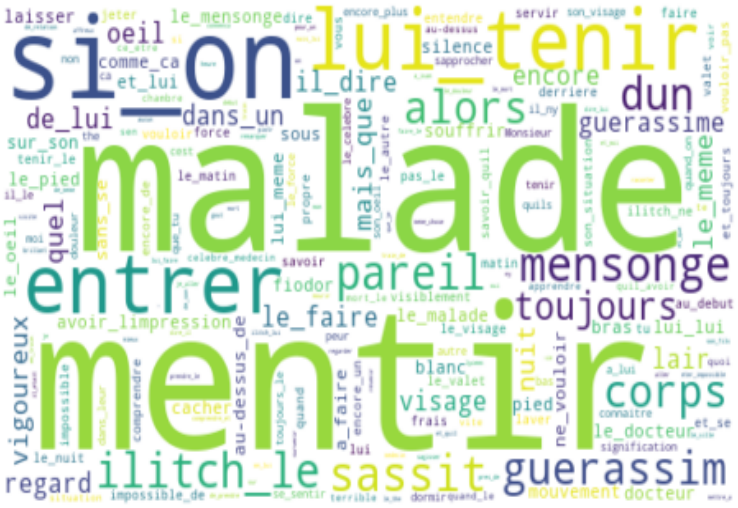

FIGURE 8 –

Cela est cohérent puisque les chapitres 7 et 8 marquent l'aggravation de la maladie d'Ivan Illitch et l'affirmation de Guérassime comme seul être capable de le soulager un peu. Ivan Illitch apprécie son valet car il est le seul à ne pas « mentir », à ne pas faire preuve d'hypocrisie envers lui.

5.5 chapitres 9, 10, 11 et 12

Pour les chapitres 9, 10, 11 et 12, les caractéristiques discriminantes en figure 9 sont encore une fois difficile à comprendre. Cela est dû aux mêmes problèmes que pour les chapitres 5 et 6. Il y a malgré tout un début de réponse, avec le mot « solitude » par exemple, qui est représentatif de la fin du roman.

On a aussi deux éléments intéressants, le mot « souvenir » et le verbe « rappeler ». A la fin du roman, Ivan Illitch est emporté par une vague de souvenirs, notamment aux chapitres 9 et 10. C'est assez significatif de la fin où Ivan Illitch se repasse sa vie et essaie de faire la paix avec lui-même au moment de mourir. L'importance du mot « dossier » est plus énigmatique, mais fait référence au chapitre 10 avec le dossier du divan sur lequel Ivan Illitch meurt à petit feu.

FIGURE 9 –

6 Conclusion

Ainsi, nous avons modéliser avec des techniques quantitatives l'avancée du récit vers la mort d'Ivan Illitch. En annotant des morceaux de 100 mots avec le chapitre dans lequel il se trouvait, nous avons entraîné un modèle de classification automatique qui a pu retrouver la bonne étiquette avec 76% d'efficacité. Ce résultat est bon, mais il n'était pas suffisant pour comprendre exactement à quel point le modèle était performant, et comment il faisait pour choisir entre les labels.

Ainsi, nous avons retrouvé les coefficients pour chaque classe et nous les avons étudiés. Ces derniers sont autant de signes qui, selon le modèle, définissent la spécificité des chapitres fusionnés par nos soins. Les mots que le modèle soulignent sont dans l'ensemble assez cohérent avec notre connaissance de l'œuvre, même si le thème principal du roman, la mort, n'apparaît pas vraiment. Cela est sans doute lié au fait que ce thème est transversal à tout le récit, et qu'il n'est pas relatif à un seul chapitre.

En définitive, les méthodes computationnelles nous offrent un nouveau regard sur Ivan Illitch, et montrent leur utilité même à l'échelle d'un seul roman, ce qui n'était pas évident à première vue. Si elles sont parfois surprenantes (le chapitre 4 similaire au chapitre 1, les mots « ancien » ou « dossier » très importants pour les chapitres 2,3 et 9, 10, 11, 12), elles ont le mérite d'avoir globalement collé à la connaissance qualitative du roman, en apportant une perspective innovante sur *La mort d'Ivan Illitch*.

7 Bonus : Méthode non-supervisée : Modélisation de sujets

Dans cette partie bonus, qui n'est pas le cœur du travail présenté au dessus, nous avons implémentons deux librairies python pour de la modélisation de sujet au niveau de l'état de l'art. Il s'agit de BertTopic [10] et de Top2vec [11]. Les deux librairies sont assez similaires mais se distinguent dans les stratégies mathématiques mises en place pour sortir des thèmes.

Ces techniques appartiennent au domaine du non supervisé, c'est à dire que les algorithmes déterminent eux-même le nombre de thèmes pertinents à sortir et les mots qui composent chaque thème. Le problème principal est que ces deux librairies nécessitent énormément de données pour être performantes. Comme nous n'avons pas assez de données à l'échelle du chapitre, nous implémentons la modélisation de sujet au niveau du récit entier. La librairie Bertopic n'a pas donné de résultats satisfaisants, du fait du manque de données, et nous présentons uniquement les topics formulés par Top2vec. Tous les mots du roman sont pris en compte, la librairie reconnaît automatiquement la présence de mots outils et essaie de le enlever le plus possible.

Voici les quatre thèmes trouvés par la librairie :

Figure 10 –

Le premier thème en figure 10 semble se référer au début du récit, où Ivan Illitch devait « faire son devoir » dans la vie factice qu'il avait construite. Le nom propre « Ivan » est associé à ce topic. On trouve les mots « rouble », « affaire », « droit », « société ». Autant de mots qui sont relatifs à l'hypocrisie sociale et le carriérisme matérialiste dans lesquels vivait Ivan Illitch.

Le deuxième thème en figure 11 est assez explicite : La souffrance d'Ivan Illitch jusqu'à sa mort est détectée. C'est un thème transversal du récit. Les mots « souffrance », « souffrir », « peine » apparaissent au premier plan, tandis qu'au second plan se trouve les mots « chagrin », « douleur », ainsi que le champ lexical de la mort.

10. https ://maartengr.github.io/BERTopic/
11. https ://top2vec.readthedocs.io/en/latest/

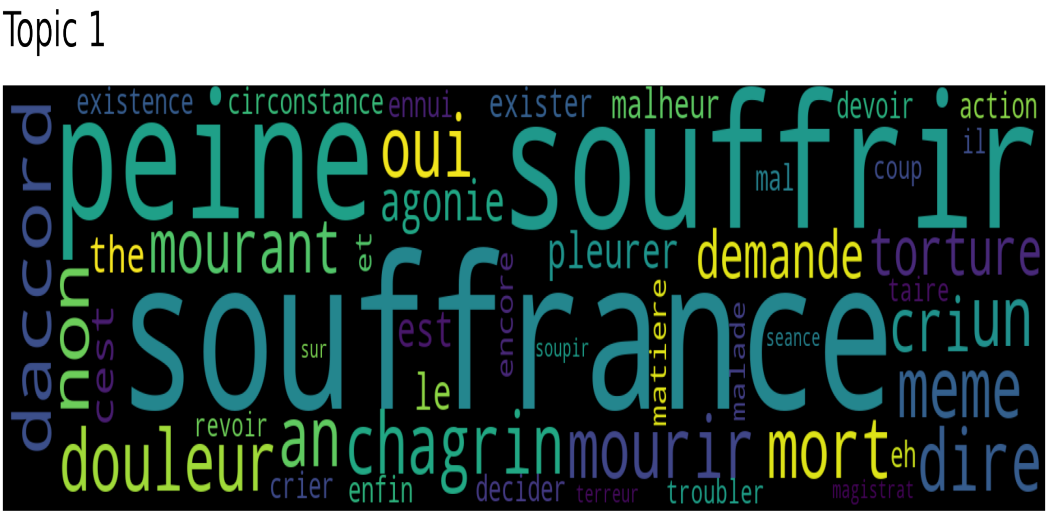

Figure 11 –

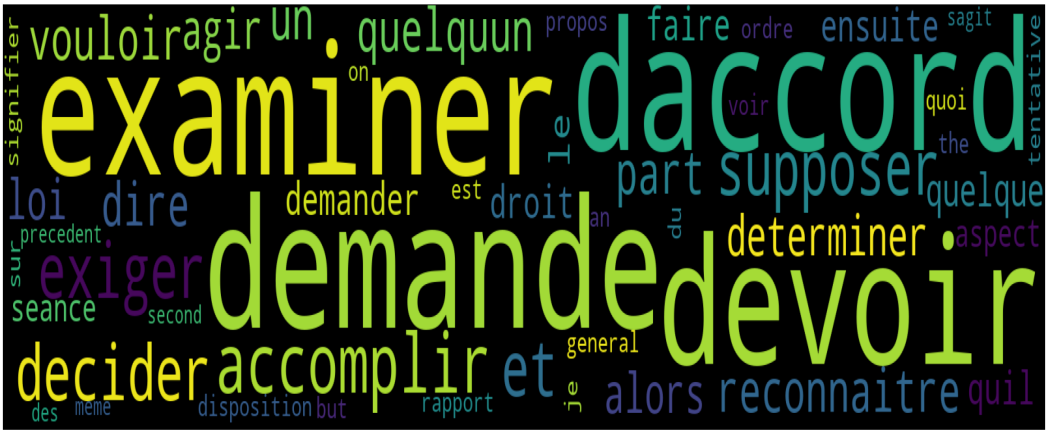

Figure 12 –

Le troisième thème en figure 12 est assez énigmatique, il semble se rattacher aux rapports qu'Ivan Illitch entretient avec les personnages du récit.

Le dernier thème en figure 13 est très explicite et fait référence à la maladie d'Ivan Illitch et à la recherche de moyens pour soulager sa souffrance. Les mots « médecin », « malade », « docteur » ou encore « prescription » et « diagnostique » ressortent nettement.

Topic 3

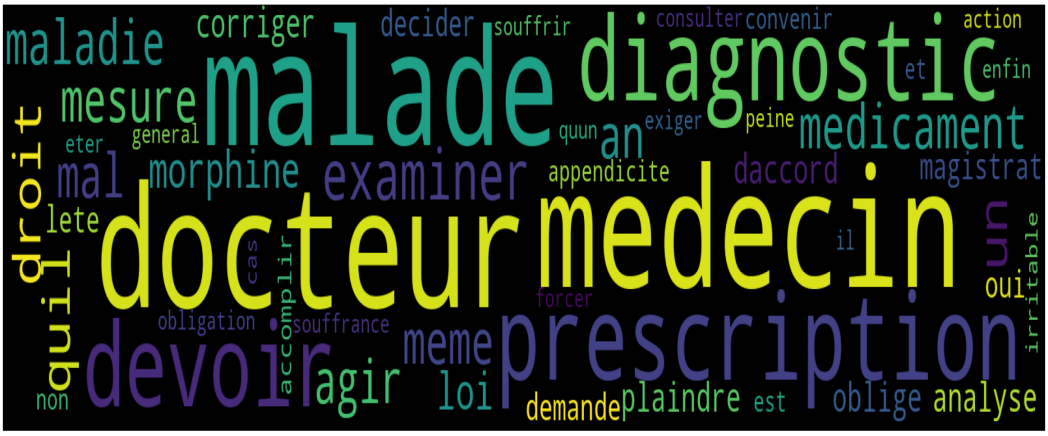

FIGURE 13 –