Actes de CORIA-TALN-RJCRI-RECITAL 2025
Marseille, France, 30 juin au 4 juillet 2025

VERS : Versification Et Représentation de Séquences

Marceau Hernandez[1, 2]
(1) CERES, Sorbonne Université, 28 rue Serpente, 75006 Paris, France
(2) STIH, Sorbonne Université, 28 rue Serpente, 75006 Paris, France
articles@marceau-h.fr

Résumé

L'analyse métrique est une étape importante pour le traitement des textes versifiés. Le résultat d'une telle analyse permet, par exemple, de comparer les textes entre eux, ou, dans le cas de textes chantés, de les comparer avec différents airs. Nous proposons une méthode pour la création d'un modèle produisant diverses analyses métriques pour un vers donné, ainsi qu'une application en diachronie longue de cette méthode sur des données en français produites à partir du 16ème siècle et jusqu'au début du 20ème siècle. Cette méthode repose sur la prédiction des noyaux vocaliques d'un vers. Nous offrirons également un point de comparaison et nous poserons la question de la robustesse à la variation de ces méthodes selon l'état de langue considéré et le bruitage provenant de l'application de reconnaissance optique de caractères en amont.

Abstract

VERSE : Verse Evaluation and Resilience through Syllabic nuclei Estimation

A key step towards processing versified texts resides in the metric analysis of such documents. With this annotation, one could then compare versified texts to one another. In the case of sung texts, this could also enable the comparison of lyrics to melodies. We propose a method to create models producing verse-related metrics, along with an application of this method on French versified texts (produced between the 16th century and the 20th century). This method relies on predicting the nucleus of each syllable. We will also offer a comparison and evaluate the resilience of the metric analysis against variation. The variation comes twofold, from the diachronic language variations, but also from the noise induced by documents produced with automatic transcription tools.

Mots-clés : textes versifiés, métrique, mètre, seq2seq, noyaux vocaliques, poésie, théâtre, paroles, transcriptions bruitées, OCR.

Keywords : versified texts, metrics, meter, seq2seq, syllable nucleus, poetry, theater, lyrics, noisy transcriptions, OCR.

Actes des 18e Rencontres Jeunes Chercheurs en RI (RJCRI) et 27ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL), pages 268–288.

1 Introduction

Nous entendons ici par le terme de “textes versifiés” les documents textuels composés uniquement de vers. Le “vers” est entendu selon la définition proposée par le TLFi (Bernard *et al.*, 2002) : « Segment d'énoncé constituant une unité d'ordre rythmique et phonique fondée sur des règles retenant [...] le nombre des syllabes, et marqué par [...] une disposition unilinéaire. ». Globalement, nous nous intéresserons ici à trois types de textes versifés : (i) la poésie, (ii) le théâtre et (iii) les paroles de chanson. Les deux premiers types sont représentés au sein de nos données d'entraînement alors que le dernier type constitue celui que nous tentons de mesurer.

La métrique définit le processus à travers lequel un texte versifié est mesuré ainsi que les mesures en résultant. Dans cette étude, nous choisissons quatre mesures principales pour nos analyses : (i) la longueur de chaque vers, en nombre de syllabes ; (ii) le genre de rime de chaque vers (féminine ou masculine) ; (iii) le schéma de rimes d'un texte (continues, suivies, embrassées et alternées) ; (iv) et la longueur de chaque strophe, en nombre de vers.

La mesure automatique des différentes métriques associées à un texte versifié requiert diverses connaissances externes au texte, telles que des connaissances de la langue et des usages en versification. Aquien (2018) explique que « les différentes règles classiques du compte des syllabes sont désormais partiellement caduques [...] la prononciation de la poésie a tenu compte des changements de la langue courante. Mais, pour autant, elle ne s'est pas calquée sur elle ». Ainsi, il n'est pas aisé pour un locuteur du français de mesurer un texte versifié sans en connaître les règles. De même une personne formée à la métrique de textes en anglais ne pourra pas mesurer un texte versifié en français sans une connaissance de la langue.

La métrique repose notamment sur une connaissance des règles phonologiques de la langue : afin de pouvoir compter le nombre de syllabes, nous avons besoin de pouvoir les reconnaître et les délimiter (par exemple, passer de “bonjour” à [bɔ̃.ʒuʁ]). De plus, compter le nombre de syllabes ne suffit pas : certains phénomènes peuvent modifier le compte, comme la diérèse [1] et la synérèse [2], mais d'autres phénomènes viennent également affecter ce compte, comme l'élision des “e” instables (/ə/) suivant le contexte droit.

Certaines considérations proviennent également d'une sorte d'attendu quant au nombre de syllabes qu'est supposé faire un vers selon son contexte de production géo-spatial et son genre littéraire. Ainsi, si un locuteur, en comptant le nombre de syllabes, se retrouve avec un compte peu usuel, ce dernier cherchera alors si une règle changeant le compte peut s'appliquer. Par exemple, dans le vers suivant « Vaines précautions ! Cruelle destinée ! » [3], si nous ne prenons pas en compte de diérèse, nous nous retrouverions avec un compte peu habituel de onze syllabes, nous amenant alors à penser qu'une diérèse se situe dans la première syllabe du mot vieillard ([vjɛ.jaʁ] -> [vi.ɛ.jaʁ]).

D'autres règles sont à considérer pour l'analyse de la rime : il faut être capable de détecter la prononciation ou non des “e” instables (/ə/) en fin de vers ou encore d'identifier les vers

1. « Prononciation en deux syllabes distinctes de deux voyelles successives d'un même mot. » (Bernard *et al.*, 2002). Une diérèse augmente alors le compte de syllabes de un.
2. « Fusion en une diphtongue de deux voyelles contiguës. Anton. diérèse. » (Bernard *et al.*, 2002). Une synérèse diminue alors le compte de syllabes de un.
3. Vers issu de la pièce “Phèdre”, 1677 de Jean RACINE, scène III de l'acte I.

finissant par des sons identiques malgré une graphie variante. De ce fait, un outil servant à la production automatique de la métrique de textes versifiés doit prendre en compte ces différents obstacles pour produire un résultat cohérent.

La métrique d'un texte versifié n'est généralement pas le fruit du hasard. L'auteur peut par exemple, chercher à respecter un schéma métrique attendu. Cependant d'autres paramètres peuvent influencer le choix d'un schéma métrique. C'est par exemple le cas en chanson, où, afin de faire correspondre un texte à un air précédemment produit, le parolier peut influer sur la longueur des vers. Il y a alors une correspondance entre un air musical et le schéma métrique des paroles qui peuvent lui être associées. Un travail de classification des airs, en fonction de leur coupe, est ainsi entrepris en parallèle. Pour ce faire, nous sommes actuellement en train de développer une base de données à partir de "La Clé du caveau" (Capelle, 1848) afin de recenser les airs et les informations qui leur sont liées telles que la partition et la coupe associée.

L'analyse automatique de la métrique de paroles de chansons pourra alors, à terme, permettre le lien entre un texte donné et un air, voire entre plusieurs textes. De plus, l'existence d'un tel outil offrira la possibilité d'annoter des ensembles de textes afin d'en permettre l'indexation selon leur métrique, permettant *in fine* de comparer manuellement des textes aux métriques communes.

2 État de l'art

Phonemizer (Bernard & Titeux, 2021), est un outil de phonétisation multilingue proposant une interface unique à partir de quatre outils différents. Parmi ces outils, trois outils de synthèse vocale (Speech-To-Text, STT), sont configurés à la phonétisation d'un plus ou moins grand nombre de langues. Le dernier n'est pas destiné directement au STT mais laisse l'utilisateur créer des correspondances graphèmes/phonème. Selon l'outil sous-jacent choisi, plusieurs fonctionnalités sont disponibles, dont une fonction de découpage en syllabes. Malheureusement, le seul outil capable d'un tel découpage ne supporte que l'anglais américain, le rendant inutilisable sur nos données, qui sont en français. Nous aurions alors pu choisir de phonétiser à l'aide de cet outil avant de gérer le découpage des syllabes. Cependant, cette approche nous semblait soulever de nouvelles problématiques : le découpage en syllabes de textes versifiés peut varier en fonction de nombreux phénomènes qui peuvent poser un problème au découpage par règles, comme nous le verrons en section 4 ; et la phonétisation étant dépendante de l'état de langue [4], nous aurions besoin d'un outil capable de s'adapter à un français non-contemporain, celui de nos corpus [5].

Ces différents obstacles, ainsi que les faibles performances obtenues à l'utilisation de méthodes à base de règles (Cf. section 4), nous ont alors incité à tester d'autres approches. Or, parmi les données qui nous étaient accessibles, nous avons remarqué que Métrique en ligne (Delente & Renault, 2015) nous permettait de rapprocher notre problème d'une autre tâche bien connue, celle de la traduction automatique par réseaux neuronaux. En effet, don-

4. Deux éléments nous viennent alors en tête pour le français : le changement de la graphie, notamment par la dissimilation du u et du v ainsi que du i et du j ; et, comme le présente Aquien (2018), le degré de propension des textes versifiés à être accompagnés de musique.

5. Un exemple de texte versifié de notre corpus est présent en figure 1.

nées de Métrique en ligne, présentées en section 3, permettent de faire correspondre à un vers sa suite de noyaux vocaliques, ces derniers offrant la possibilité de récupérer plusieurs métriques associés aux vers. Notre tâche serait de passer d'une chaîne d'entrée (les caractères du vers) à une chaîne de sortie (les noyaux vocaliques), une tâche alors analogue à celle de la traduction automatique.

Bahdanau *et al.* (2014) décrivent une approche aux modèles de séquence à séquence pour la traduction automatique (*Neural Machine Translation*, NMT). Nous reprendrons deux éléments clés de cette méthode. Premièrement, ils proposent d'entraîner simultanément la partie encodeur et décodeur, formant ainsi un seul réseau neuronal optimisé de bout en bout. Contrairement aux systèmes traditionnels de traduction statistique composés de sous-modules ajustés séparément, cette approche d'entraînement conjoint permet au modèle d'apprendre des représentations interdépendantes : l'encodeur apprend à générer des annotations spécifiquement utiles pour le décodeur, tandis que le décodeur apprend à interpréter ces annotations. Cette optimisation unifiée est rendue possible par la rétro-propagation du flux de gradients à travers le mécanisme d'attention. Dans un second temps, ils introduisent un mécanisme d'attention. Au lieu d'obtenir un vecteur de taille fixe en sortie de l'encodeur, nous obtiendrons une séquence de vecteurs d'annotation de longueur égale à celle de la chaîne d'entrée. Puis, à partir de cette matrice, le décodeur, à l'aide du mécanisme d'attention, décide à quel point il considère chaque vecteur de la matrice pour la prédiction de chaque token cible, jusqu'à produire un token de fin de séquence. Cette approche permet d'éviter à l'encodeur de devoir encoder toute l'information au sein d'un seul vecteur de taille fixe [6]. Le deuxième élément est l'utilisation de réseaux de neurones récurrents (*Recurrent Neural Network*, RNN) bi-directionnels. Au lieu de ne traiter la séquence que dans le sens de lecture traditionnel, cette approche utilise deux RNN afin de venir lire la chaîne à traiter dans les deux sens : le premier lit la chaîne du token 0 au token N ; lorsque le second lit la chaîne du token N au token 0, dans le sens opposé. L'annotation produite par les deux réseaux est alors concaténée pour chaque token. Cela permet d'associer les contextes gauche et droit à l'annotation de chaque token, le premier encodeur ayant vu le contexte précédent au moment de l'encoder lorsque le second a vu le contexte suivant. Cet élément se révèle particulièrement intéressant pour notre tâche, les sons associés à une lettre dépendant souvent des caractères qui la succède, en plus des caractères précédents.

Luong *et al.* (2015) présentent un état de l'art succinct sur l'usage de modèles de type séquence à séquence pour la NMT. Ils comparent les différentes approches choisies, notamment pour la partie cachée des encodeurs et décodeurs. Ces derniers soulignent que plusieurs travaux se fondent sur des *Long Short-Term Memory* (LSTM) ou des architectures proches pour les couches cachées, comme évoqué plus tôt, nous établirons également notre architecture sur des réseaux récurrents avec des LSTM au sein des couches cachées de l'encodeur et du décodeur. Puis, dans un second temps, ces derniers présentent une architecture de modèle fondée sur une attention à petite fenêtre. Au moment de la prédiction d'un token cible, le modèle produit un alignement, informant sur la fenêtre, de longueur fixe, sur laquelle porter l'attention. Selon les auteurs, cette approche permettrait un entraînement moins coûteux. La méthode peut sembler appropriée pour notre tâche, les noyaux vocaliques étant majoritairement déductibles à partir de quelques caractères. Cependant, nous avons choisi de ne pas nous fonder sur cette dernière car nous supposons que la prédiction de phénomènes

6. « By letting the decoder have an attention mechanism, we relieve the encoder from the burden of having to encode all information in the source sentence into a fixed-length vector. » (Bahdanau *et al.*, 2014)

modifiant le compte au sein d'un vers serait facilitée par la prise en compte d'un contexte plus global.

Une méthode analogue à ces tâches de NMT est alors utilisée par Chormai *et al.* (2020). Dans cet article les chercheur·euse·s essaient de trouver les frontières de mots en thaï, dans laquelle les mots ne sont pas constamment délimités par des espaces, en passant par une première étape de segmentation en syllabe. Pour se faire, une des approches de segmentation en syllabes présentées se sert alors d'un BiLSTM (*Bi-directional LSTM*, LSTM bi-directionnel) pour annoter les séquences de caractères selon si elles représentent une frontière à la syllabe. Cette approche, ainsi que la seconde présentée, leur permet d'obtenir un F_1 score d'environ 98 %. Contrairement à l'approche présentée en section 5, les chercheur·euse·s ne cherchent pas à obtenir de représentations phonétiques. Cependant, en plus de se baser sur un jeu de données avec une taille du même ordre de grandeur, la tâche et la méthode sont particulièrement proches de ce que nous réalisons dans cet article et constituent d'un point de vue méthodologique l'article le plus proche que nous avons pu trouver dans la littérature. De ce fait, la qualité des résultats obtenus dans cette étude fournit une base de comparaison raisonnable.

3 Détail de la tâche et des données

3.1 Objet d'étude

Comme évoqué précédemment, nous étudions les textes versifiés. L'article porte son attention sur les textes en français mais la méthode est pensée pour être la plus flexible possible, aussi nous encourageons des réutilisations avec de nouveaux jeux de données.

Les données utilisées pour l'entraînement ainsi que pour l'évaluation de notre méthode nous proviennent de Métrique en ligne (Delente & Renault, 2015). Le site web, ainsi que le corpus au format XML, regroupe un ensemble de textes versifiés en français. On y retrouve ainsi, à date, 19 753 poèmes et 133 pièces de théâtre pour lesquels plusieurs métriques ont été produites. Il y a ainsi un total de 1 038 121 vers analysés. Les textes sont produits par 239 auteurs différents, entre la fin du 16$^{\text{ème}}$ siècle et jusqu'au début du 20$^{\text{ème}}$ siècle.

Comme présentées dans la section précédente (Section 2), les annotations de Métrique en ligne ont été produites à partir d'un outil conçu par la même équipe et semblent avoir été corrigées en partie à la main. Cependant, l'outil n'a pas été rendu accessible, après avoir pris contact avec le laboratoire, il semblerait que l'outil ne soit pas disponible. À partir des données disponibles[7], nous avons alors pu récupérer, pour chaque vers, les informations métriques associées, telles que le nombre de syllabes ou encore les noyaux vocaliques, une donnée que nous évoquerons dans la partie 5.2. Cette vérité de terrain n'est donc pas un *gold standard*, certaines erreurs ont par ailleurs été retrouvées. Pour reprendre la nomenclature proposée par (Rebholz-Schuhmann *et al.*, 2011 ; Petkovic *et al.*, 2025), nous parlerons de *silver standard*. Il nous semblait cependant plus important d'avoir des données que d'avoir des données parfaites.

Cependant, l'outil a été imaginé afin de déterminer les métriques des textes versifiés d'un

7. Les données produites sont accessibles à partir du dépôt suivant https://git.unicaen.fr/malherbe/analyses.

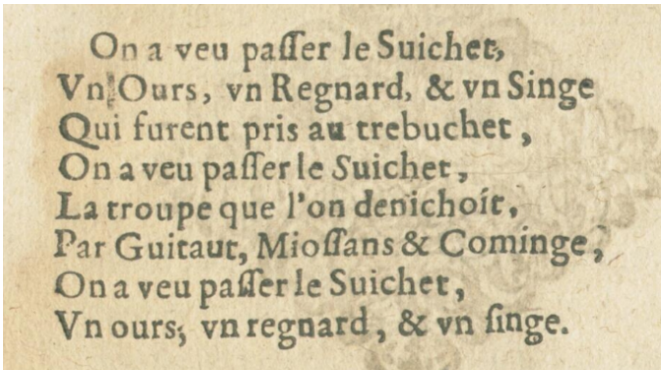
On a veu paſſer le Suichet,
Vn Ours, vn Regnard, & vn Singe
Qui furent pris au trebuchet,
On a veu paſſer le Suichet,
La troupe que l'on denichoit,
Par Guitaut, Mioſſans & Cominge,
On a veu paſſer le Suichet,
Vn ours, vn regnard, & vn ſinge.

FIGURE 1 – Exemple de paroles au sein du corpus des Mazarinades - Triolets pour le temps présent, identifiant Moreau 3861.

autre corpus, le corpus Antonomaz [8]. Ce dernier est composé de "Mazarinades", de courts documents publiés principalement au milieu du 17ème siècle, lors de la Fronde. S'inscrivant au début de la presse artisanale, l'impression permettait la production de documents à bas coûts, influant sur la qualité du papier utilisée ainsi que sur les conditions de conservation. Il était alors interdit d'écrire des chansons (Haffemayer, 2016), "les pouvoirs ayant bien compris la puissance de diffusion des informations par ce moyen" (Abiven, 2023). En effet, ces dernières permettaient une circulation rapide, et ce même auprès de personnes ne sachant pas lire. De ce fait, les auteurs n'indiquaient pas explicitement qu'ils écrivaient des paroles, les laissant paraître comme de simples vers. Cependant, de nombreux indices les laissaient transparaître à l'œil avisé, comme la présence d'alinéas en début de strophe, visible en figure 1. Ne déclarant pas la présence de chansons, les paroliers ne communiquaient pas non plus sur l'air employé, la métrique peut alors nous indiquer une liste d'airs candidats.

3.2 Métrique

Comme évoqué précédemment, afin de pouvoir rapprocher les textes versifiés entre eux ou à un air dans le cas de paroles de chansons, nous nous intéresserons à l'obtention de quatre métriques : (i) la longueur du vers, en nombre de syllabes ; (ii) la longueur de la strophe, en nombre de vers ; (iii) le type de rime d'un vers (rime féminine ou masculine) ; (iv) et motif de rime entre les vers (rimes plates/croisées/…).

L'intérêt de la métrique est double. Elle peut permettre de rapprocher les textes versifiés des métriques proches, mais également de les rapprocher des airs. Dans le contexte du corpus des Mazarinades : (i) il y a un nombre réduit d'airs utilisés ; (ii) un air est souvent associé à une métrique précise [9] ; (iii) une métrique n'est alors associée qu'à une poignée d'airs au plus ; (iv) si nous connaissions la métrique correspondant à chaque air, avoir la métrique d'un texte versifié nous informe directement sur les airs candidats.

De plus, les airs ne sont pas réutilisés de façon anodine, il est souvent fait référence aux chansons d'avant, phénomène qui se retrouve même dans le texte : « La musique peut en

8. Dépôt GitHub du corpus Antonomaz : https://github.com/Antonomaz

9. Par exemple, Capelle (1848) est l'auteur d'un livre de référence pour l'étude des airs et de leurs coupes. Cet ouvrage présente de conséquentes énumérations sur différents aspects des airs, tels que leurs coupes, mais également les partitions, et ce pour plus de 2 000 airs (2 350 pour l'édition de 1948).

elle-même porter le double sens, avec la constante réutilisation de timbres. Cette pratique du recyclage de mélodies constitue en effet un possible levier intertextuel. » (Abiven, 2022).

4 Méthodes de référence

4.1 Présentation des méthodes utilisées

Quatre méthodes ont été expérimentées :

1 La première méthode utilise la bibliothèque *Pyphen*[10]. Cette bibliothèque, destinée à l'ajout de césures dans les documents, se sert d'un dictionnaire pour chaque langue afin de séparer les syllabes ;

2 La seconde méthode, à base de règles, se sert d'expressions régulières afin de reconnaître les sons du français obtenus à l'aide de plusieurs lettres telles que les voyelles nasales, pour finalement séparer sur les *clusters* vocaliques ;

3 La troisième méthode, également à base de règles, utilise une boucle pour séparer sur les voyelles, tout en prévoyant les cas de *clusters* consonantiques ;

4 La dernière méthode se base sur le module Lexique383 de la bibliothèque *pylexique*[11]. Ce lexique permet de récupérer le nombre de syllabes des mots connus. Nous le combinons à la méthode par règles la plus efficace afin d'essayer d'augmenter les résultats de cette dernière.

Les trois premières méthodes ont été reprises du blog de Cyril GROUIN[12] Il est cependant important de noter que, selon l'auteur, ces solutions sont imparfaites ; à défaut de disposer de meilleurs outils pour le français, nous nous servirons de ces outils comme point de référence. La dernière méthode se fonde sur un lexique, en utilisant la meilleure des trois méthodes précédentes dans le cas d'un mot hors-lexique.

4.2 Modalités d'évaluation

Avant d'évaluer ces méthodes de référence, il est important de notifier que, parmi les quatre métriques souhaitées, seulement une d'entre elles n'est produite par ces méthodes, la longueur du vers en syllabes. Il nous manque les informations sur les rimes ainsi que la longueur des strophes.

4.3 Performance

Les évaluations ont été réalisées à partir des vers extraits de la base de Métrique en ligne. 166 095 vers, soit environ 20 % du total ont été choisis aléatoirement pour servir de jeu de données d'évaluation.

10. Lien du dépôt de Pyphen https://github.com/Kozea/Pyphen
11. Lien du dépôt de pylexique https://github.com/SekouDiaoNlp/pylexique
12. Lien vers le blog de Cyril GROUIN : https://perso.limsi.fr/grouin/inalco/.

TABLE 1 – Résultats des méthodes “baseline” sur la métrique “nombre de syllabes par vers”. La colonne “Méthode” contient le nom de la méthode, la colonne “ACCURACY” représente le pourcentage de fois où la méthode trouve la bonne réponse, la colonne “MEAN DIFF” représente l’écart moyen entre la prédiction de la méthode et la réponse attendue et la colonne “ALMOST CORRECT” représente le pourcentage de fois où la méthode trouve la bonne réponse ou se trompe de seulement une syllabe. En gras apparaissent les meilleurs résultats pour une colonne donnée.

Méthode	ACCURACY	MEAN DIFF	ALMOST CORRECT
1 - pyphen	25.4 %	1.70	66,9 %
2 - regex	31.0 %	1.46	75,7 %
3 - while	**54.5 %**	**1.26**	**91,1 %**
4 - lexique383_while	30.4 %	1.49	73,2 %

Le tableau 1 présente : l’exactitude (ACCURACY), le ratio de bonnes réponses rapporté au nombre de vers ; le ratio de bonnes réponses à une syllabe près, que nous appellerons ALMOST CORRECT ; ainsi que l’écart absolu moyen entre la longueur estimée et la longueur réelle de chaque vers [13] sur lequel la méthode s’est trompée (MEAN DIFF).

Plusieurs observations ressortent de ces résultats : (i) les résultats sont loin d’être parfaits, l’exactitude étant comprise entre 25.4 et 54.5 % ; (ii) les méthodes à base de lexique donnent les moins bons résultats ; (iii) lorsque les méthodes se trompent, l’écart est assez petit.

Les deux derniers points nous confortent sur l’influence des phénomènes modifiant le compte au sein d’un vers, tels que les diérèse, synérèse, élision et synalèphe, sur la métrique automatique. Cependant, pallier ces phénomènes n’est pas chose aisée : lorsqu’un locuteur tente de prédire la présence de phénomènes influant le compte, ce dernier, en plus d’utiliser les règles décrivant où et quand ces phénomènes peuvent se produire, se repose sur le fait d’éviter les métriques inhabituelles. Ainsi, il semble étrange de ne pas avoir un schéma de rimes régulier. Cependant, il existe aussi des schémas de rimes plus complexes [14], voire très peu réguliers [15]. À cet endroit, et en raison du nombre d’erreurs que nous aurions par poème, il paraît difficile d’envisager un moyen d’ajuster le nombre de syllabes pour se rapprocher d’un schéma plus “naturel” en ne se basant que sur les informations que renvoient ces méthodes.

4.4 Couverture

Ces méthodes présentent cependant d’autres désavantages. Le plus saillant est la spécificité de ces méthodes à un état de langue précis. Comme nous avons pu l’observer, les méthodes les plus efficaces sont celles se servant de lexiques. De plus, même les méthodes à base de règles se servent de connaissances de la langue et ne sont, de ce fait, pas applicables telles quelles à d’autre langues, il nous faudrait réécrire les règles de syllabation de la langue ciblée. C’est un problème de taille pour notre corpus d’étude, écrit en français pré-moderne, car les règles de syllabation y sont suffisamment différentes pour ne plus correspondre à celles

13. $\sum_{i=1}^{n} |\hat{y}_i - y_i|$ avec : i, la i$^{\text{ème}}$ instance de vers du jeu de données de test ; n la longueur du jeu de données de test ; $\hat{y}_i$ la i$^{\text{ème}}$ longueur estimée ; et y_i la i$^{\text{ème}}$ longueur réelle

14. Exemple de schéma de rimes peu régulier,“Promesses d’un visage”, Les épaves, Galanteries, XI, C. Baudelaire.

15. Exemple en chanson La Farce des courtisans de Pluton, et leur pèlerinage en son royaume, s. l., 1649, p. 17-18. Moreau1372, BnF, Département Littérature et Arts, cote YE-3298.

utilisées dans les systèmes 2 et 3 [16].

Une autre particularité de notre corpus cible est qu'il a été retranscrit automatiquement, à partir d'outils de reconnaissance optique de caractères (*Optical Caracter Recognition*, OCR). Les retranscriptions contiennent des erreurs, pouvant affecter les méthodes de mesure. Afin d'estimer l'influence des erreurs de retranscription, nous avons utilisé la bibliothèque *string-noise* [17] (Miller, 2025) afin de générer artificiellement des erreurs au sein de notre corpus de test. L'influence de différents niveaux d'erreurs [18] des chaînes en entrée, pour les trois mesures d'erreur, sur la prédiction donnée par les différentes méthodes est représentée au sein des figures 2 ;3 ;4 (pages 17 ;18 ;19).

Nous pouvons remarquer que nos quatre méthodes sont sensibles au niveau de bruit, et ce même pour un CER faible, surtout dans le cas de la troisième méthode, la plus performante sur les vers non bruités. Ainsi, avec un CER moyen à $0,030$ [19], le score d'Accuracy passe à $50,0$ %, soit une perte de $4,5$ points par rapport à la version non bruitée.

5 Méthode proposée

Dans cette section, nous allons nous intéresser à la nouvelle méthode que nous avons développé, qui se fonde sur les noyaux vocaliques.

5.1 Les noyaux vocaliques, c'est quoi ?

Mertens (1987) définit le noyau de la syllabe comme : « la partie voisée de la syllabe à la plus haute énergie » [20]. Le noyau vocalique est alors le ou les phonème·s phonétique·s ayant le plus grand degré de voisement au sein d'une syllabe donnée. En français, le noyau est alors généralement un unique phonème vocalique, un phonème produit par une voyelle.

Ainsi, si l'on voulait retranchement le mot "bonjour" en la suite de ses noyaux vocaliques, nous obtiendrons /ɔ̃/ et /u/, avec /ɔ̃/ comme noyau vocalique de [bɔ̃] [21], et /u/ comme noyau vocalique de [ʒuʁ] [22].

16. On peut notamment observer la présence d'une lettre en plus, le "s" long ainsi que la non différenciation entre les voyelles "u" et "i" et les consonnes "v" et "j", respectivement.

17. La bibliothèque *string-noise* est accessible à partir du dépôt suivant https://github.com/dleemiller/string-noise. Cette dernière permet de simuler, entre autres, les erreurs communément produites par des outils d'OCR. À l'aide de cette dernière, nous simulons différents niveaux de bruit en faisant varier le paramètre de probabilité de remplacement (de 0 à 1), nous avons choisi les niveaux suivants : $0,001$, $0,005$, $0,01$, $0,05$, $0,1$, $0,2$, $0,3$, $0,4$, $0,5$, $0,6$, $0,7$, $0,8$, $0,9$ et 1.

18. Niveaux d'erreurs mesurés au taux d'erreur au caractère *Caracter Error Rate*, CER).

19. Un CER à $0,030$ représente un texte assez peu bruité. À titre d'exemple, Petkovic *et al.* (2025) décrivaient un tel score d'« un OCR de bonne qualité » pour un livre de 1868, tapuscrit et en français.

20. « the nucleus corresponds to the high energy voiced part of the syllable » (Mertens, 1987)

21. /b/ étant alors l'attaque de la syllabe.

22. /ʒ/ et /ʁ/ étant respectivement l'attaque et la coda de la syllabe.

5.2 Les noyaux vocaliques, pourquoi ?

Nous avons décidé de produire la suite de noyaux d'un vers car, à partir de ces derniers, plusieurs de nos métriques d'intérêt deviennent accessibles. En effet, à partir de ces derniers, nous obtenons facilement : (i) le nombre de syllabes, en comptant le nombre de noyaux ; (ii) la qualité de la rime, en regardant le dernier noyau ; (iii) et le schéma des rimes, en comparant les derniers noyaux vocaliques de vers successifs.

De plus, comme évoqué précédemment, le corpus de Métrique en ligne (Delente & Renault, 2015) nous fournit, pour chaque vers, l'ensemble des noyaux vocaliques [23]. Nous en obtenons une ressource conséquente, de plus d'un million de vers, nous permettant l'entraînement et l'évaluation de notre méthode.

De plus, passer par les noyaux vocaliques nous permet de constituer le reste de notre chaîne de traitements à partir de méthodes peu dépendantes de l'état de langue ou du genre littéraire. Nous espérons que cette ressource pivot nous permettra de mesurer à la fois des textes versifiés du 19ème siècle issus de la poésie ou du théâtre que des textes du 17ème siècle issus de la chanson.

5.3 Détails du modèle

Le modèle créé est donc de type séquence à séquence. Les modèles séquence à séquence sont, comme évoqués auparavant (Section 2), souvent utilisés pour des tâches de traduction automatique, ou *Machine Translation*. Ils ont alors comme principe de "traduire" une chaîne de caractères d'une langue, vers une chaîne de caractères dans une autre langue.

Notre modèle est ainsi destiné à transposer nos textes versifiés, en français, vers leur suite de noyaux vocaliques, en Alphabet Phonétique International (API). Il a donc pour but de passer d'une séquence A, la suite des lettres constituant un vers, à une séquence B la suite des noyaux vocaliques de ce vers.

Ce modèle repose sur une architecture de type *encoder/decoder*, voici la suite d'étapes que suit la chaîne en entrée pour produire la sortie souhaitée : (i) les *tokens* de la chaîne d'entrée sont convertis par leur indice dans la langue d'entrée, puis ; (ii) la chaîne indices des *tokens* d'entrée passe par une couche d'*embedding*, chaque indice de *token* est converti en un vecteur dense ; (iii) la représentation de la chaîne passe ensuite par l'encodeur, chaque *token* est encodé, récursivement dans une séquence de vecteurs de sortie [24], puis ; (iv) la matrice obtenue passe par une seconde couche d'*embedding* ; (v) la représentation obtenue passe récursivement par le décodeur : un vecteur est alors généré pour chaque passage de la représentation dans le décodeur ; (vi) une couche linéaire fait correspondre la sortie du LSTM du décodeur à l' espace du vocabulaire ; (vii) pour finir, nous convertissons les indices par les *tokens* associés.

23. Les textes, annotés au format XML, contiennent de nombreuses informations. Nous en extrayons les noyaux vocaliques afin de produire un couple vers/noyaux. Ce fichier est disponible au sein du dépôt du projet (https://github.com/Marceau-h/VERS-Models/tree/master/donnees_article) sous la même licence que le corpus d'origine (Cf. https://crisco4.unicaen.fr/verlaine/index.php?navigation=docPdf&idPdf=Pr%C3%A9sentation_du_corpus.pdf).

24. Chaque vecteur de sortie résulte alors de la concaténation des états finaux cachés et cellule des deux RNN (un par direction) afin de former l' état initial pour le décodeur, ce pour chaque token donné.

Les couches des encodeur et décodeur sont constituées de BiLSTM, ou LSTM bi-directionnels. L'architecture de ce modèle, dont l'utilisation de RNN bi-directionnels est directement inspirée des travaux de Bahdanau *et al.* (2014). L'architecture du modèle est présentée en annexe .1.

L'ensemble du code ayant servi à la réalisation de ce modèle a été pensé pour en permettre la réutilisation. Le code du modèle et de ses différentes interfaces ont été réalisées à l'aide de la librarie PyTorch (Ansel *et al.*, 2024). Le projet est, à l'instar du jeu de données employé, disponible sous licence libre [25].

5.4 Entraînement

L'entraînement est réalisé à partir des données de Métrique en ligne (Delente & Renault, 2015), une fois converties en couple entrée/sortie souhaitée, que nous appellerons x_i et y_i. Le nombre d'*epochs*, c'est-à-dire, le nombre de fois que les poids du modèle ont été ajustés à l'aide de l'ensemble du jeu d'entraînement [26], est de 800. L'entraînement s'est fait à partir de zéro, cela signifie qu'au début de ce dernier, les poids et biais des différentes couches du modèle ont été initialisés avec des valeurs aléatoires. La phase d'entraînement comporte également un *teacher-forcing*, remplaçant alors occasionnellement un token prédit par le token cible. Cette méthode permet au modèle de converger plus rapidement. Cependant, comme l'observe Lu (2023), elle peut aussi induire la diminution des performances du modèle.

L'entraînement s'est déroulé sur les serveurs de la plateforme SACADO MeSU de l'Université de la Sorbonne, il a duré 14 heures et 32 minutes sur une carte graphique NVIDIA A40 partagée. Afin d'estimer le coût environnemental de l'entraînement, nous considérerons que le modèle a utilisé la carte graphique dans ses capacités maximales [27]. Pour cet entraînement, nous estimons le coût énergétique total de l'entraînement à $4,360$ kWh, soit environ 300 Wh par heure d'utilisation du serveur. Selon la moyenne française d'émission de eq CO_2 par kWh estimée à 9,75 grammes eq CO_2 par kWh [28], l'entraînement de ce modèle a résulté en une émission d'environ $42,51$ g eq CO_2.

Les poids du modèle seront proposés au téléchargement, sous licence libre, à partir du dépôt de code [25].

5.5 Inférence

L'inférence peut se faire à l'aide du code proposé au sein du dépôt. Le temps moyen pour 1 000 inférences est de $4,30$ secondes [29] pour ce modèle. Le coût pour 1 000 inférences est alors estimé, selon les mêmes calculs que pour l'entraînement, à $1,19.10^{-3}$ Wh et $1,16.10^{-2}$

25. Dépôt regroupant le code servant à l'entraînement et l'emploi du modèle présenté, sous AGPL-3.0+ https://github.com/Marceau-h/VERS-Models
26. Le jeu d'entraînement se compose d'environ 80 % des données récupérées depuis le corpus Métrique en ligne, soit 872 026 vers.
27. Selon la fiche constructeur, la consommation maximale de la carte graphique est de 300 W : https://images.nvidia.com/content/Solutions/data-center/a40/nvidia-a40-datasheet.pdf
28. Moyenne de l'émission d'un grammeeq CO_2 par kWh consommé pour l'année 2024 obtenue depuis un rapport de EDF : https://www.edf.fr/sites/groupe/files/2025-03/edf-france_bilan-ges_2014-2024.pdf
29. *Benchmarks* réalisés sur le même serveur que l'entraînement

TABLE 2 – Échantillon de couples vers / noyaux vocaliques obtenus à partir de la base Métrique en ligne (Delente & Renault, 2015), ces vers dont ceux de la première strophe de "Promesses d'un visage", Les épaves, Galanteries, XI, C. Baudelaire.

vers	noyeaux_syll_text
J' aime, ô pâle beauté,|tes sourcils surbaissés,	ɛ | o | a | ə | o | e | ɛ | u | i | y | ɛ | e
D' où semblent couler des ténèbres ;	u | ɑ̃ | ə | u | e | ɛ | e | ɛ | ə
Tes yeux, quoique très-noirs,|m' inspirent des pensers	ɛ | ø | wa | ə | ɛ | wa | ɛ̃ | i | ə | ɛ | ɑ̃ | e
Qui ne sont pas du tout funèbres.	i | ə | ɔ̃ | a | y | u | y | ɛ | ə

grammes eq CO_2.

À titre d'exemple, le tableau 2 présente un échantillon d'inférences obtenues et mises en regard avec la séquence d'entrée ainsi que la sortie désirée.

5.6 Quelle entrée / sortie

Une classe est dédiée à la gestion des langues, cette dernière prend deux formats en entrée : (i) un fichier au format *plain text* avec un caractère séparant chaque couple x, y (par défaut le saut de ligne, U+000A) et un autre séparant la chaîne d'entrée x de la chaîne de sortie y ; (ii) un fichier au format JSON représentant une liste de clés, contenant la valeur de y, avec chaque clé contenant une liste de chaîne de caractères, contentant les valeurs de x correspondantes.

Dans les deux cas, il est possible de spécifier comment séparer les chaînes de caractère d'entrée et de sortie en *tokens*. Les utilisat·eur·rice·s peuvent spécifier une chaîne de caractères sur laquelle découper, une liste de chaînes de caractères, une expression régulière, ou laisser le séparateur vide pour séparer chaque caractère.

6 Résultats

6.1 Performances

Le jeu de test est composé de 166 095 couples vers/noyaux vocaliques, représentant environ 20 % des 1 038 121 vers analysés par le projet Métrique en ligne (Delente & Renault, 2015). Pour chaque chaîne d'entrée (chaîne de caractères du vers), nous demandons à notre modèle de prédire le couple correspondant (chaîne des noyaux vocaliques, potentiellement composés de plusieurs caractères tels que /wa/ ou encore /ɔ̃/).

Sur l'ensemble des données du test, 3 343 prédictions différaient de la sortie attendue, ce qui représente une *Accuracy* d'environ 97, 99 %. Ce résultat semble surprenant en comparaison aux 4 méthodes de référence explorées précédemment. Cependant, Chormai *et al.* (2020) présentent dans leur article un modèle de segmentation de syllabes avec un F_1-score d'environ 98, 6%, au sein d'une langue qui ne sépare les mots par des espaces qu'occasionnellement.

Sur les 3 343 prédictions erronées, 304 prédictions produisaient une suite de noyaux voca-

liques de longueur égale et 3 278 produisaient le même dernier noyau. Nous avons alors un modèle capable de prédire la longueur des vers en syllabes à 98, 17 % d'exactitude et le noyau vocalique de la dernière syllabe à 99, 96 % d'exactitude. À noter cependant que pour prédire le schéma des rimes et longueurs de vers, il ne suffit pas de prédire correctement ces informations pour un vers, mais pour tous ceux de la strophe. Ainsi, pour une strophe de 8 vers, la probabilité de n'avoir aucune erreur sur la longueur des vers serait de 86, 26 %, sur la rime, cette probabilité serait de 99, 68 %.

Un autre phénomène que nous avons pu remarquer est la présence d'erreurs dues au jeu de test. Après analyse manuelle, il s'avère que plusieurs cas de discordance entre notre Vérité de Terrain (VT) et nos prédictions ne proviennent pas d'une erreur de prédiction mais d'une erreur au sein de la VT. Par exemple, pour le vers « Et ma parole d'or allégerait vos pas. » [30], la VT y couple la suite de noyaux vocaliques suivante « e | a | a | ɔ | ə | ɔ | a | e | ə | ɛ | a » lorsque le modèle prédit « e | a | a | ɔ | ə | ɔ | a | e | ə | ɛ | **o** | a », ajoutant le noyau [o], manquant à la référence. Nous pensons donc, à terme, ré-annoter manuellement les 3 343 couples pour lesquels il y a désaccord entre notre modèle et la VT afin d'estimer l'influence de ce phénomène sur nos résultats.

6.2 Métrique manquante : nombre de vers par strophe

Comme nous l'avons évoqué auparavant (Cf. sous-section 5.2), l'intérêt de prédire la suite de noyaux vocaliques se trouve dans le fait que ces derniers nous permettent par la suite de calculer aisément presque toutes les mesures qui nous intéressent. Cependant, une mesure nous manque celle de la longueur des strophes par vers. En effet, au sein des retranscriptions automatiques obtenues par OCR, aucune information de structure, mis à part les sauts de ligne. Sans avoir la structure encodée, il nous semble alors difficile de quantifier les informations sur les strophes. Nous réfléchissons actuellement à plusieurs pistes en vue de rajouter cette structure au sein de nos retranscriptions automatiques.

6.3 Robustesse face à la variation

Comme pour les méthodes de référence, un autre point qui nous intéresse réside dans la robustesse face à la variation de la chaîne d'entrée (Cf. sous-section 4.4). Pour ce faire, nous avons évalué l'influence de différents niveaux d'erreurs sur les performances du modèle, avec la même méthodologie que pour les méthodes de référence [31].

Les performances de notre modèle se retrouvent fortement impactées, même à un faible niveau de bruit. Dans le cas d'un CER moyen de 0.038 [32], le taux d'exactitude passe à 19, 45 % pour la prédiction des noyaux vocaliques, à 51, 43 % pour la prédiction de la longueur du vers et 89, 88 % pour la prédiction du dernier noyau. Cela nous amène à penser que le modèle a sur-appris la tâche donnée, potentiellement en faisant correspondre de manière systématique une suite de caractères précise à une sortie précise. Après une analyse manuelle des erreurs,

30. Vers 117 de "La lyre morte", Les Cariatides (1842), livre deuxième, I, Ceux qui meurent et qui combattent.
31. Nous avons utilisé la bibliothèque *string-noise* (Miller, 2025) afin de générer artificiellement différents niveaux de bruit.
32. Le niveau de CER a légèrement augmenté par rapport aux tests sur les méthodes de référence. Les versions bruitées ont cependant été produites en utilisant *string-noise* et à partir des mêmes paramètres.

le modèle semble être particulièrement sensible à deux types d'erreurs : les retraits d'espaces, qui provoquent souvent une perte d'un noyau, et les modifications de consonnes en voyelles, par exemple, un "t" devenant un "i". L'exactitude en contexte bruité de notre modèle est alors inférieure à celle de la meilleure méthode de référence à niveau de CER comparables.

Dans le but de pallier la faible résistance à la variation de la chaîne d'entrée de notre modèle, nous avons entrepris un affinage (ou *finetuning*) de notre modèle précédemment produit. En reprenant les paramètres obtenus à la suite des 800 itérations (ou *epochs*), nous continuons l'entraînement du modèle à partir de données du jeu d'entraînement auxquelles du bruit a été ajouté artificiellement pendant 50 itérations [33]. À l'aide de ce modèle affiné, nous procédons à une nouvelle évaluation, en bruitant une nouvelle fois le jeu de données de test. Dans le cas d'un CER moyen de 0.039 [34], le taux d'exactitude passe à 23, 58 % pour la prédiction des noyaux vocaliques, à 62, 12 % pour la prédiction de la longueur du vers et 92, 23 % pour la prédiction du dernier noyau. Ces premiers résultats d'affinage sont loin d'êtres parfaits, nous observons cependant qu'il semble possible d'améliorer la robustesse de notre modèle au travers de cette méthode. Cet affinage a été réalisé avec un mélange équivalent de tous les niveaux de bruit, il pourrait alors être intéressant d'explorer les résultats d'un affinage plus ciblé, en n'utilisant que des vers bruités à des taux de CER proches d'un CER attendu au sein des retranscriptions automatiques ou au moins moins distribués.

7 Conclusion et perspectives

Dans cet article, nous avons présenté une nouvelle approche pour l'analyse métrique de textes versifiés. Cette dernière nous permet de trouver le nombre de syllabes dans un vers avec un taux d'exactitude de 98, 17 %. Aucune approche similaire n'a été trouvée, nous avons cependant deux points de comparaison au travers d'outils de segmentation en vers : (i) une méthode à base de règles qui obtient un taux d'exactitude de 54.5 % sur notre jeu de données de textes en français ; (ii) une méthode à partir d'un modèle qui obtient un F_1-score d'environ 98 % sur leur jeu de données, en thaï (Chormai *et al.*, 2020).

De plus, cette méthode nous permet également de trouver le noyau vocalique de la dernière syllabe à 99, 96 % d' exactitude. Sachant que la prédiction du dernier noyau vocalique nous permet de produire deux autres métriques : le genre de la rime ; et le schéma des rimes.

Nous avons également pu observer la résistance au bruit (généré artificiellement) des approches de référence ainsi que de notre modèle et une version affinée pour le bruit. La version affinée permettait d'avoir un certain de degré de tolérance au bruit, suggérant une possible exploitabilité de la méthode sur des retranscriptions OCR de bonne qualité.

Nous avons également rendu accessible le code ayant permis l'entraînement et l'utilisation de ce modèle. Nous avons donc pensé ce projet pour être réutilisable pour, par exemple, entraîner sur d'autres langues ou pour des tâches légèrement différentes. Les paramètres du modèle seront également disponibles. L'ensemble est téléchargeable[25] et modifiable sous licence libre (AGPL-3.0+).

33. Cet entraînement supplémentaire a pris 40 minutes. En reprenant les calculs réalisés en sous-section 5.4, l'affinage représente une consommation de 200 kWh, ce qui représente une émission d'environ 1,950 g eq CO_2.

34. Versions bruitées en utilisant *string-noise* et à partir des mêmes paramètres que pour les méthodes de référence et le modèle non affiné.

Ainsi, notre méthode nous permet d'obtenir trois métriques, la longueur des vers, le genre de la rime et le schéma de rimes. Cependant, à défaut d'avoir des informations de structure au sein de nos retranscriptions, nous sommes toujours incapables de compter la longueur des strophes en vers. Nous avons pensé à plusieurs manières de pallier ce manque d'informations. Clérice (2023) présente dans son article une méthode de détection et étiquetage de zones de textes reposant sur YOLOv5 (Jocher *et al.*, 2022). Une telle méthode permettrait de détecter, à partir des documents sources, les différentes strophes, pour ainsi ajouter l'information au sein des retranscriptions. D'autres solutions sont en cours d'exploration, notamment à partir d'indices visuels de changement de strophe tels que : la variation de l'interligne, les alinéas et les décorations inter-strophiques.

En parallèle du découpage en strophes, nous aimerions explorer l'apport du contexte sur les métriques : en utilisant le contexte de la strophe, serait-il possible de mieux prédire les métriques ? Nous pourrions alors proposer un système de contraintes, favorisant les suites de noyaux d'une certaine longueur ou d'un certain dernier noyau en fonction des suites de noyaux vocaliques dans une même strophe.

Nous souhaitons également explorer davantage la robustesse de notre modèle face à la variation, notamment en annotant manuellement une partie de notre corpus d'étude. Ce dernier, issu de retranscriptions automatiques, nous permettrait d'exploiter un bruit non synthétique évaluer la robustesse en conditions réelles ainsi que pour affiner le modèle sur des données naturellement bruitées et de l'état de langue cible. Pour finir, nous explorons également des méthodes pour identifier et trouver la métrique des paroles associées aux airs à partir de la partition.

Remerciements

Le travail présenté dans cet article a bénéficié du soutien du CERES [35] par lequel j'ai pu avoir de précieux conseils lors de discussions et relectures, ainsi que le programme "Méthodes numériques pour les LSHS" qui finance mon contrat doctoral.

35. Centre d' expérimentation en méthodes numériques pour les recherches en Sciences Humaines et Sociales - https://ceres.sorbonne-universite.fr/

Annexes

.1 Détails du modèle

Le modèle entrainé est alors composé de 5 parties principales :

- Une couche de plongements de caractères, créant la représentation abstraite de ces derniers, de dimensions $(164, 512)$;
- Le BiLSTM d'encodage est de dimensions $(512, 512)$;
- La couche de plongements de l'encodeur au décodeur est de dimensions $(48, 512)$;
- Le second BiLSTM, de décodage est de dimensions $(512, 1024)$;
- Pour finir, la couche dense est de dimensions $(1024, 48)$.

Sachant que : la taille maximale de la chaîne d'entrée est de 162 caractères[36], les chaines plus courtes étant complétées de *tokens* de remplissage (*padding*) ; et la taille maximale de la chaîne de sortie est de 46 noyaux vocaliques[37], les chaînes plus courtes étant de nouveau complétées à l'aide de tokens de remplissage.

36. 164 *tokens* en comptant les *tokens* de début et fin de phrase
37. 48 en comptant les *tokens* de début et fin de phrase

.2 Figures

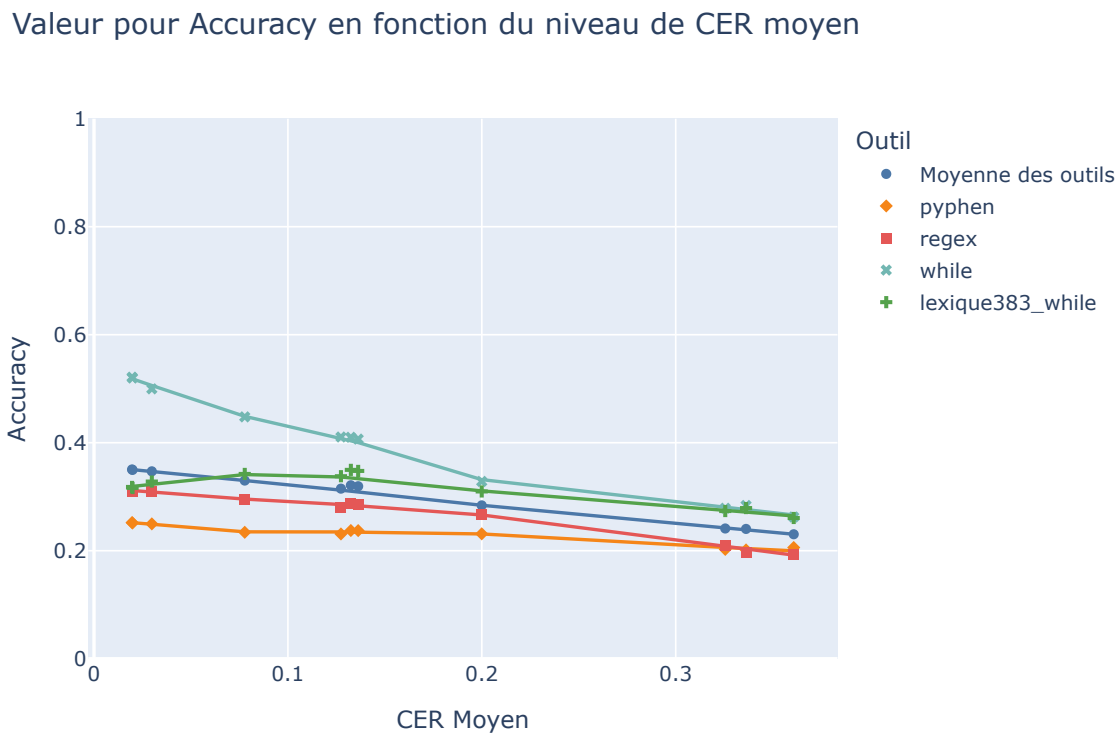

FIGURE 2 – Variation de l'exactitude ("ACCURACY") en fonction du niveau de bruit (généré artificiellement) de la chaîne d'entrée. Les séquences ont été bruitées artificiellement à l'aide de la bibliothèque *string-noise* (Miller, 2025) avec le mode "OCR" et plusieurs niveaux de probabilité[17]. Les niveaux de bruit sont ensuite mesurés au taux d'erreur au caractère *Caracter Error Rate*, CER) pour chaque niveau de probabilité testé.

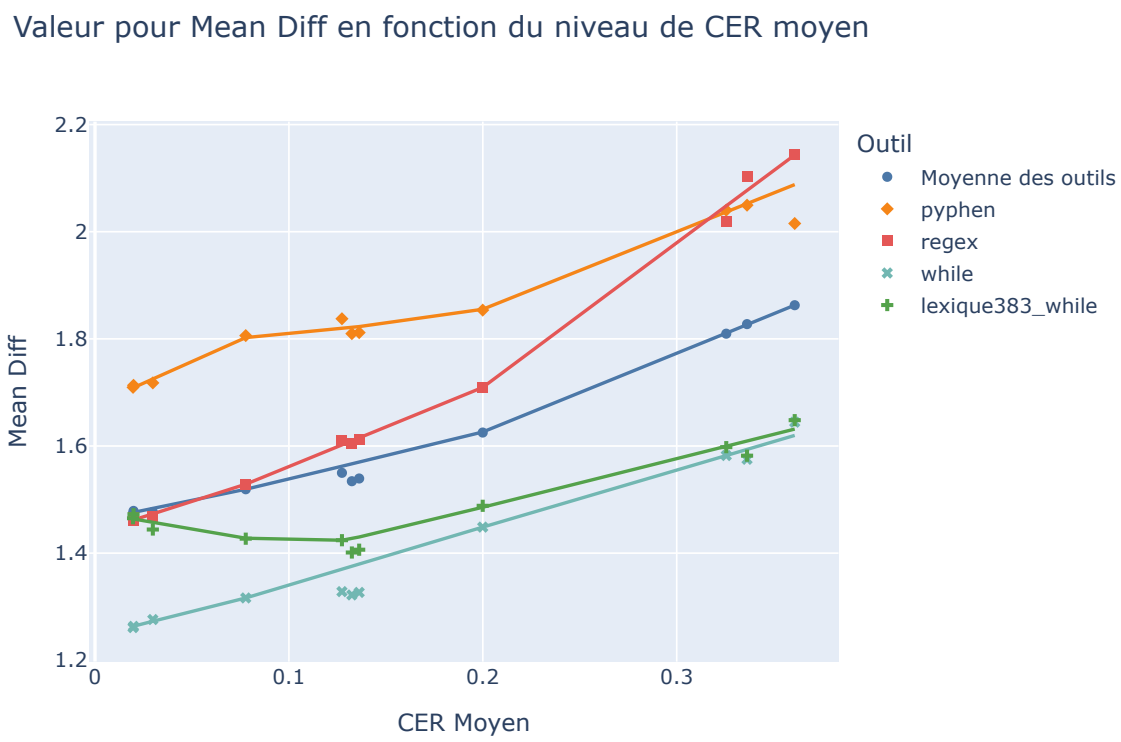

FIGURE 3 – Variation de l'écart du nombres de syllabes au sein des mauvaises réponses ("MEAN DIFF") en fonction du niveau de bruit (généré artificiellement) de la chaîne d'entrée. Les séquences ont été bruitées artificiellement à l'aide de la bibliothèque *string-noise* (Miller, 2025) avec le mode "OCR" et plusieurs niveaux de probabilité[17]. Les niveaux de bruit sont ensuite mesurés au taux d'erreur au caractère *Caracter Error Rate*, CER) pour chaque niveau de probabilité testé.

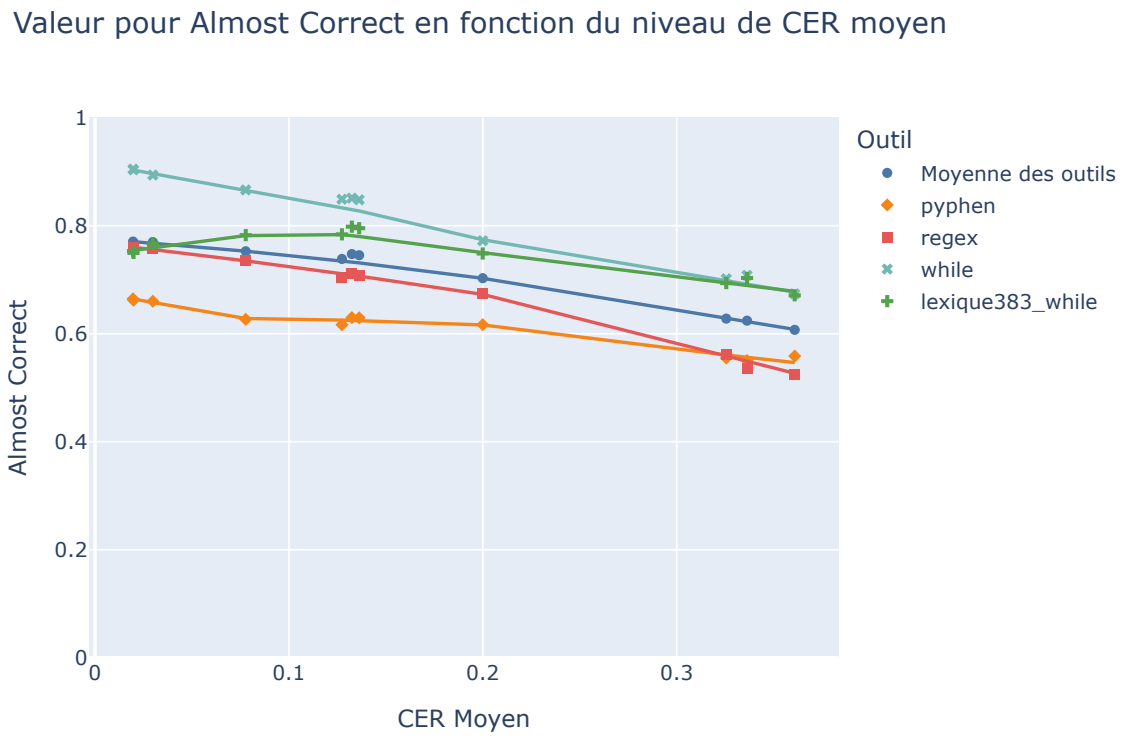

FIGURE 4 – Variation du taux de bonnes réponses ou de réponses incorrectes de seulement une syllabe ("ALMOST CORRECT") en fonction du niveau de bruit (généré artificiellement) de la chaîne d'entrée. Les séquences ont été bruitées artificiellement à l'aide de la bibliothèque *string-noise* (Miller, 2025) avec le mode "OCR" et plusieurs niveaux de probabilité[17]. Les niveaux de bruit sont ensuite mesurés au taux d'erreur au caractère *Caracter Error Rate*, CER) pour chaque niveau de probabilité testé.

Références

Abiven K. (2022). «À quoi sert une chanson, si elle est désarmée ?» les chansons pendant la fronde : armes ou récits ? *Écrire l'histoire. Histoire, Littérature, Esthétique*, (22), 131–133.

Abiven K. (2023). Viralité des mazarinades chantées et écrites : tubes et/ou éléments de langage ? *Tubes, teasing et viralité : Grand Siècle et pop' culture, Les recettes du succès. Stéréotypes compositionnels et littérarité au XVIIe siècle, dir. Anna Fouqué-Legros, Miriam Speyer, Tony Geerhaert, dans Fabula/Les colloques.*

Ansel J., Yang E., He H., Gimelshein N., Jain A., Voznesensky M., Bao B., Bell P., Berard D., Burovski E., Chauhan G., Chourdia A., Constable W., Desmaison A., DeVito Z., Ellison E., Feng W., Gong J., Gschwind M., Hirsh B., Huang S., Kalambarkar K., Kirsch L., Lazos M., Lezcano M., Liang Y., Liang J., Lu Y., Luk C., Maher B., Pan Y., Puhrsch C., Reso M., Saroufim M., Siraichi M. Y., Suk H., Suo M., Tillet P., Wang E., Wang X., Wen W., Zhang S., Zhao X., Zhou K., Zou R., Mathews A., Chanan G., Wu P. & Chintala S. (2024). PyTorch 2 : Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)* : ACM. doi : 10.1145/3620665.3640366.

Aquien M. (2018). *La versification :«Que sais-je ?» n° 1377*. Presses Universitaires de France.

Bahdanau D., Cho K. & Bengio Y. (2014). Neural machine translation by jointly learning to align and translate. *ArXiv*, **1409**.

Bernard M. & Titeux H. (2021). Phonemizer : Text to phones transcription for multiple languages in python. *Journal of Open Source Software*, **6**(68), 3958. doi : 10.21105/joss.03958.

Bernard P., Lecomte J., Dendien J. & Pierrel J.-M. (2002). Un ensemble de ressources informatisées et intégrées pour l'étude du fançais : FRANTEXT, TLFi, Dictionnaire de l'Académie et logiciel Stella, présentation et apprentissage de leur exploitation. In *Actes. 9ème Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN). Nancy, palais des congrès. 24-27 juin 2002*, volume 2, p. 3–36. Comité de lecture. Conférences conjointes co-organisées par le Laboratoire ATILF, UMR 7118 CNRS Université Nancy 2 et le LORIA, UMR 7503 CNRS-INRIA-Université de Nancy., hal : halshs-00004837.

Capelle P. A. (1848). *La clé du caveau, à l'usage de tous les chansonniers français et étrangers, des amateurs, auteurs, acteurs du Vaudeville, et de tous les amis de la chanson.* Janet et Cotelle.

Cho K., van Merrienboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H. & Bengio Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Chormai P., Prasertsom P., Cheevaprawatdomrong J. & Rutherford A. (2020). Syllable-based neural Thai word segmentation. In D. Scott, N. Bel & C. Zong, Éds., *Proceedings of the 28th International Conference on Computational Linguistics*, p. 4619–4637, Barcelona, Spain (Online): International Committee on Computational Linguistics. doi : 10.18653/v1/2020.coling-main.407.

Clérice T. (2023). You Actually Look Twice At it (YALTAi): using an object detection approach instead of region segmentation within the Kraken engine. *Journal of Data Mining*

and Digital Humanities, **Historical Documents and...** DOI : 10.46298/jdmdh.9806, HAL : hal-03723208.

DELENTE E. & RENAULT R. (2015). Traitement automatique des formes métriques des textes versifiés. In J.-M. LECARPENTIER & N. LUCAS, Éds., *Actes de la 22e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts*, p. 116–122, Caen, France : ATALA.

HAFFEMAYER S. (2016). *Mazarin face à la fronde des mazarinades, ou comment livrer la bataille de l'opinion en temps de révolte (1648-1653)*, volume vol. 12. Stéphane HAFFEMAYER, Patrick REBOLLAR, Yann SORDET , Droz. HAL : hal-01401574.

JOCHER G., CHAURASIA A., STOKEN A., BOROVEC J., NANOCODE012, KWON Y., MICHAEL K., TAOXIE, FANG J., IMYHXY, LORNA, 曾逸夫 Z. Y., WONG C., V A., MONTES D., WANG Z., FATI C., NADAR J., LAUGHING, UNGLVKITDE, SONCK V., TKIANAI, YXNONG, SKALSKI P., HOGAN A., NAIR D., STROBEL M. & JAIN M. (2022). ultralytics/yolov5 : v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. DOI : 10.5281/zenodo.3908559.

LU C. (2023). Performance analysis of attention mechanism and teacher forcing ratio based on machine translation. In *Journal of Physics : Conference Series*, volume 2580, p. 012006 : IOP Publishing.

LUONG T., PHAM H. & MANNING C. D. (2015). Effective approaches to attention-based neural machine translation. In L. MÀRQUEZ, C. CALLISON-BURCH & J. SU, Éds., *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 1412–1421, Lisbon, Portugal : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1166.

MERTENS P. (1987). Automatic segmentation of speech into syllables. In *Proceedings of the European Conference on Speech Technology*, volume 2, p. 9–12 : CEP Consultants ; Edinburgh.

MILLER L. (2025). string-noise. https://github.com/dleemiller/string-noise.

PETKOVIC L., KOUDORO-PARFAIT C., DESMAREST M.-S. & LEJEUNE G. (2025). Quelle solution pour améliorer les performances de la reconnaissance d' entités nommées sur des données bruitées, corriger l' entrée ou filtrer la sortie ? *Corpus*, (26).

REBHOLZ-SCHUHMANN D., YEPES A. J., LI C., KAFKAS S., LEWIN I., KANG N., CORBETT P., MILWARD D., BUYKO E., BEISSWANGER E. *ET AL.* (2011). Assessment of ner solutions against the first and second calbc silver standard corpus. *Journal of biomedical semantics*, **2**, 1–12.