Détection de la variation graphique dans une langue non standardisée : le cas des dialectes alsaciens

Heng Yang

Détection de la variation graphique dans une langue non standardisée : le cas des dialectes alsaciens

Heng
YANG

Tuteur universitaire : Claude Ponton

Réalisé au sein du laboratoire LiLPa - Linguistique, Langues et Parole à Strasbourg

Tuteur de stage : Pablo Ruiz Fabo

Stage co-encadré par Alice Millour et Delphine Bernhard

UFR LLASIC
Département Sciences du langage

Mémoire de master 2 mention Sciences du langage - 30 crédits

Parcours : Industries de la Langue

Année universitaire 2021-2022

Détection de la variation graphique dans une langue non standardisée : le cas des dialectes alsaciens

Heng
YANG

Tuteur universitaire : Claude Ponton

Réalisé au sein du laboratoire LiLPa - Linguistique, Langues et Parole à Strasbourg

Tuteur de stage : Pablo Ruiz Fabo

Stage co-encadré par Alice Millour et Delphine Bernhard

UFR LLASIC
Département Sciences du langage

Mémoire de master 2 mention Sciences du langage - 30 crédits

Parcours : Industries de la Langue

Année universitaire 2021-2022

Remerciements

Je profite de ces quelques lignes pour témoigner ma gratitude à ceux qui m'ont aidé tout au long de mes études en France. La réalisation de ce mémoire n'aurait pas été possible sans le support des certaines personnes.

Tout d'abord, je tiens à adresser toute ma reconnaissance à Alice Millour et Delphine Bernhard, mes deux co-encadrants du stage, et à Pablo Ruiz Fabo, mon tuteur du stage. Malgré la distance, vous m'avez encadré de façon remarquable par votre clarté et votre disponibilité. Un grand merci à vous pour vos patiences, vos suivis, vos encouragements et vos judicieux conseils.

Je désire aussi remercier les professeurs du laboratoire LIDILEM pour leurs guides précieux et leurs cours de qualité pendant ces deux années.

Je tiens à remercier spécialement Claude Ponton, qui a été particulièrement attentif et patient avec chaque étudiant et m'a aidé lorsque j'étais perdu.

Je remercie ma famille pour leur soutien inconditionnel et elle me manque profondément.

Mes derniers remerciements sont adressés à Fang Chen qui ont toujours été là pour moi.

DÉCLARATION ANTI-PLAGIAT

1. Ce travail est le fruit d'un travail personnel et constitue un document original.
2. Je sais que prétendre être l'auteur d'un travail écrit par une autre personne est une pratique sévèrement sanctionnée par la loi.
3. Personne d'autre que moi n'a le droit de faire valoir ce travail, en totalité ou en partie, comme le sien.
4. Les propos repris mot à mot à d'autres auteurs figurent entre guillemets (citations).
5. Les écrits sur lesquels je m'appuie dans ce mémoire sont systématiquement référencés selon un système de renvoi bibliographique clair et précis.

PRENOM : Heng

NOM : GANG

DATE : 07/09/2022

Sommaire

Introduction

De nos jours, la recherche d'informations (en anglais *information retrieval*, *IR*) est le principal moyen que nous utilisons pour rechercher et obtenir des informations. Par exemple, dans un document textuel, la recherche d'informations nous permet de repérer un mot ou un segment de mot donné et nous indique précisément combien il y en a. Toutefois, s'il existe de multiples graphies équivalentes du mot que nous voulons rechercher, comment obtenir des résultats précis ? Ces différentes graphies sont également connues sous le terme de variantes orthographiques.

Les variantes orthographiques n'apparaissent pas seulement en abondance sur les plateformes en ligne ou dans les messages textuels, mais on les retrouve aussi souvent dans les textes historiques. En outre, la variation orthographique est l'un des défis majeurs pour le TAL des textes historiques. Dans ce mémoire, notre travail porte sur la détection de la variation graphique dans un corpus de théâtre dialectal alsacien pour la période 1870-1940.

Les parlers dialectaux d'Alsace, qui font partie des dialectes du haut allemand, sont toujours parlés par environ 500,000 locuteurs en Alsace (INSEE et al.,1999, cité dans Steiblé & Bernhard, 2018). D'un point de vue du traitement automatique, disposant de peu de ressources informatisées, les dialectes alsaciens appartiennent aux langues peu dotées, et le manque de ressources textuelles volumineuses complique l'utilisation d'approches empiriques. De plus, les dialectes alsaciens sont principalement utilisés à l'oral et il n'existe pas de convention d'orthographe consensuelle pour l'alsacien. Par conséquent, l'énorme variation orthographique de l'alsacien complique les analyses thématiques ou textométriques. Comme il n'existe pas de norme orthographique, nous ne pouvons pas faire correspondre directement des variantes orthographiques à une norme, ce qui complique nos recherches. Nous planifions de présenter nos recherches selon la structure suivante.

La première partie décrit le contexte du stage et la problématique de notre recherche. La deuxième partie présente l'état de l'art en lien avec notre sujet de recherche. Cette partie porte sur les méthodologies et techniques qui nous semblent les plus prometteuses et inspirantes. Dans la troisième partie, nous présenterons le corpus original et la méthode d'extraction et de préparation des données à partir de ce corpus. Dans la quatrième partie, nous présenterons notre méthode pour l'extraction de règles de variation orthographique. Celle-ci se déroule en quatre étapes : le prétraitement des données, l'analyse contrastive, l'alignement des formes et l'extraction de motifs de substitution. Nous

détaillerons l'objectif de chaque partie et les démarches associées, ainsi que les résultats de chaque étape. Enfin, nous analyserons les résultats produits et discuterons également des perspectives et des limitations liées à notre travail.

Partie 1

-

Contexte et Problématique

Chapitre 1. Contexte

Ce mémoire s'inscrit dans le cadre de mon stage de master 2 Sciences du Langage parcours Industries de la Langue mené au sein du laboratoire Linguistique, langues et parole (LiLPa) de l'Université de Strasbourg[1]. Ce stage est co-encadré par Alice Millour (LIASD, Université Paris 8), Delphine Bernhard et Pablo Ruiz Fabo (tuteur de stage).

Le laboratoire Linguistique, langues et parole (LiLPa) est une unité de recherche fédératrice née en 2003. Actuellement la quasi-totalité des linguistes de l'Université de Strasbourg y est rassemblée et travaille dans trois axes thématiques :

- Thème 1 - Lexique(s), discours et transposition(s) ;
- Thème 2 - Langage, parole et variation ;
- Thème 3 - Langue/s et société.

LiLPa constitue ainsi une concentration de chercheurs, d'horizons divers mais de formations et de compétences complémentaires, qui en fait une unité de recherche unique en son genre dans le Grand Est français.

Le projet « MeThAL : Vers une macroanalyse du théâtre en alsacien »[2], initié par le laboratoire LiLPa, vise à rendre possibles des études quantitatives en analyse dramatique et en sociolinguistique historique de l'alsacien.

Pour le théâtre alsacien, une telle analyse nécessite un corpus numérique approprié. Dans le cadre du projet, un corpus large de théâtre dialectal alsacien encodé selon les recommandations de la Text Encoding Initiative (TEI) a été publié, sur la base de l'ensemble représentatif de pièces en mode image numérisées par la Bibliothèque nationale et universitaire de Strasbourg (Bnu).

[1] Source : http://lilpa.unistra.fr
[2] Source : https://methal.pages.unistra.fr/

Chapitre 2. Problématique

Dans le cadre du projet MeThAL, il serait intéressant d'aborder l'analyse thématique ou textométrique du corpus et de comparer le contenu de ses textes, qui sont variés en origine géographique et sous-genre dramatique et qui couvrent plusieurs décennies. Néanmoins, en raison de la variété des dialectes alsaciens, l'analyse informatique de ce corpus présente des défis spécifiques en Traitement automatique des langues (TAL). Ces défis soulignent des besoins imparfaitement couverts par les outils d'analyse textuelle existants, orientés prioritairement vers les langues majoritaires et conduisent à la problématique de ce mémoire. Dans ce chapitre, nous allons décrire les dialectes alsaciens et présenter notre problématique.

1. Description des dialectes alsaciens

Les dialectes alsaciens, qui appartiennent au groupe du haut allemand, sont parlés dans la région d'Alsace, située dans le Nord-Est de la France à la frontière avec l'Allemagne et la Suisse. D'après une étude sur le dialecte alsacien, parmi les habitants de la région Alsace, 43% de la population déclarent bien savoir parler l'alsacien. Cependant, la proportion de locuteurs alsaciens est en déclin (OLCA / EDinstitut, 2012).

En outre, les dialectes alsaciens sont principalement oraux et manquent de ressources textuelles importantes. Il arrive toutefois que ces dialectes soient mis à l'écrit, comme la première pièce alsacienne en 1816, *DerPfifingstmontag* de Jean-Georges-Daniel Arnold. Bien que cette production écrite puisse être retracée depuis deux siècles, il existe peu d'outils et ressources informatisées pour les dialectes d'Alsace, qui font donc partie des langues dites peu dotées (Steiblé & Bernhard, 2018).

L'alsacien est qualifié de dialecte dans le sens où il se caractérise par son oralité et constitue en fait un continuum de dialectes (Bernhard, 2014). Il varie dans l'espace et n'a pas de graphie codifiée. Comme chaque sous-système dialectal de l'alsacien obéit souvent à des lois propres (Hudlett, 2009) en fonction de son origine géographique ou linguistique, il existe beaucoup de variabilité en l'absence d'une orthographe standardisée. En conséquence, il existe plusieurs variantes différentes pour un lexème donné[3]. Par exemple,

[3] Suivant Bernhard (2014), nous utilisons lexème dans le sens de lexème chez Bauer (2003) : Une unité abstraite du vocabulaire, réalisée par des mot-formes représentant le lexème et sa morphologie flexionnelle ; une des formes est choisie par convention afin de nommer le lexème dans une entrée de dictionnaire.

> « le verbe « jouer » est présent sous quatre formes différentes dans notre lexique : *spìela*, *spiele*, *speele*, *schpeela* » **(Steiblé & Bernhard, 2016)**.

Nous pouvons également observer cette variabilité dans le tableau 1.

	Forme (pour *Tag, jour*)	**Variables sociales**
(a)	Daö	variante « rurale » (Kochersberg)
(b)	Daa	variante strasbourgeoise
(c)	Tag	allemand standard (dans une lettre)

D'r Herr Maire (G. Stoskopf, 1898)

Tableau 1. Exemple de variantes (Ruiz Fabo et al., 2021)

2. *Problématique*

Dans le cadre du projet MeThAL, nous disposons d'un corpus encodé en TEI de pièces de théâtre alsaciennes (de 1870 à 1940) à exploiter. Toutefois, la comparaison du contenu des pièces ou l'analyse de données textuelles en alsacien se heurte souvent au problème de l'énorme variation orthographique rencontrée dans les textes, en l'absence de norme graphique établie. La manière de traiter cette variabilité est par conséquent essentielle pour cette étude.

Certains chercheurs locaux affirment que l'allemand standard devrait être considéré comme le standard de référence de ces parlers dialectaux en Alsace. Dans son discours de 1985, le recteur Pierre Deyon définit la langue régionale comme suit :

> « Il n'existe en effet qu'une seule définition scientifiquement correcte de la langue régionale en Alsace, ce sont les dialectes alsaciens dont l'expression écrite est l'allemand […] ».

Cette normalisation permettrait d'éliminer les différences entre les dialectes parlés dans le nord-est de la France, et de faire disparaître les micro-variations entre eux. Cependant, les Alsaciens n'écrivent pas leurs dialectes avec l'orthographe allemande, mais avec des orthographes spécifiques (Steiblé & Bernhard, 2017). Et l'existence d'un certain nombre de textes historiques constitue également une preuve objective que l'alsacien possède ses propres orthographes, comme les pièces de théâtre.

Des propositions telles que le système ORTHAL (Crévenat-Werner & Zeidler, 2008) et le système GRAPHAL-GERIPA (Hudlett & Groupe d'Etudes et de Recherches

Interdisciplinaires sur le Plurilinguisme en Alsace et en Europe, 2003) ont été établies pour définir les conventions orthographiques. Même s'ils gardent la variabilité, la dissémination et l'utilisation réelle de ces conventions sont difficiles à estimer. D'ailleurs, beaucoup d'internautes alsaciens ne les connaissent pas, comme le montre une récente enquête (Millour, 2019).

Dans ce contexte, les questions qui se posent globalement sont les suivantes : comment peut-on identifier automatiquement les variantes orthographiques ? Quelles composantes du contexte peut-on analyser pour les identifier ? Comment peut-on s'assurer que les paires de variantes que l'on identifie sont les variantes orthographiques correctes ? Qu'est ce que cette étude peut apporter à l'analyse du corpus de projet MeThAL ?

Au plan informatique, enfin, quels algorithmes peut-on mettre en œuvre pour la tâche de l'identification ? Pour analyser des documents TEI et construire un corpus, quelle structure et quels outils sont les plus pratiques ?

Dans la section suivante, nous effectuons un tour d'horizon de l'état de l'art, en vue d'examiner des approches dédiées à la détection automatique de la variation graphique et de dégager les pistes les plus prometteuses qui nous ont inspiré dans nos travaux.

Partie 2

-

État de l'art

Chapitre 3. Identification des variantes graphiques

Comme expliqué précédemment, ce travail vise à proposer des méthodes pour aider à détecter la variation graphique dans les dialectes alsaciens. Plus précisément, pour pouvoir comparer le contenu des pièces du corpus mentionné ci-dessus et effectuer des analyses textuelles, une représentation orthographique homogène du vocabulaire est nécessaire, ainsi qu'une neutralisation des variantes graphiques.

Plusieurs méthodes ont été proposées pour la détection de la variation graphique. En fonction de leur finalité, nous pouvons les résumer en deux approches principales : soit les variantes sont transformées en une norme choisie, soit elles sont tout simplement reconnues comme étant des variantes, sans qu'il y ait pour autant une normalisation explicite (Ruiz Fabo et al., 2020).

1. Normalisation orthographique automatique

La normalisation de textes, en tant que tâche de Traitement Automatique des Langues (TAL), consiste à préparer les textes pour effectuer un traitement automatique du contenu de plus haut niveau. Cette tâche est nécessaire lorsque les textes fournis aux outils de TAL proviennent de sources peu fiables quant à la forme du texte.

Les techniques courantes de Traitement Automatique des Langues naturelles se concentrent davantage sur les textes formels. Cependant, la communication est devenue plus dynamique avec la popularisation des réseaux sociaux et des applications qui permettent aux gens de s'exprimer et de communiquer instantanément, et de plus en plus de textes apparaissent sur les réseaux sociaux, souvent sous une forme informelle, ce qui complique le traitement des langues naturelles. Les internautes utilisent souvent des expressions spéciales au lieu des mots formels pour plusieurs raisons : euphémisme ; exprimer des émotions fortes ; exprimer le sarcasme ou l'humour ; rendre un texte plus concis ; rendre une description plus distincte et intéressante ; échapper à la censure officielle. Évidemment, une autre partie des variantes dans les textes des plateformes en ligne provient des fautes d'orthographe des utilisateurs, et les variantes et leurs formes canoniques correspondantes (les mots originaux) coexistent généralement dans le texte.

D'un point de vue linguistique, les variantes orthographiques dans les plateformes en ligne peuvent être considérées comme de l'anti-langue. L'anti-langue est un terme proposé par le linguiste Michael Halliday et signifie une forme de langage différente de la

langue courante et ayant ses propres connotations. Selon Halliday (1976), l'anti-langue se caractérise par les éléments suivants :

- L'anti-langue recode les mots dans le processus de création ;
- L'anti-langue est globalement conforme à la grammaire de la langue majoritaire ;
- Les expressions de l'anti-langue sont comme des codes, qui ne peuvent être compris que par ceux qui se situent dans leur contexte.

D'un point de vue du TAL, ces variantes sont généralement considérées comme des mots hors vocabulaire (MHV, en anglais *out-of-vocabulary*, *OOV*), i.e., les mots inconnus du système et n'ayant pas été rencontrés dans les données d'apprentissage. Les mots hors vocabulaire sont responsables de la dégradation de la qualité des applications du TAL.

Yoon et al. (2010) présentent une approche pour détecter toutes les variations d'un mot vulgaire avec modification des phonèmes en appliquant un alignement de chaînes basé sur les phonèmes. La méthode utilise la propriété d'espace métrique de la distance d'édition des chaînes de caractères. Par exemple, il convertit le signe de ponctuation « ! » en lettre « i », et lorsque le mot « *sh!t* » est rencontré, le mot est converti en « *shit* » pour être traité. Li (2014) effectue le calcul de la similarité phonologique et morphologique des textes pour le latin ou l'anglais, et détecte les variantes sur la base de la similarité. Doval et al. (2020) ont proposé un plongement de mots (ou *word embedding* en anglais) pour traiter des mots hors vocabulaire. En outre, pour la normalisation des textes historiques, l'idée principale est de transformer les variantes historiques en une forme standard moderne (Bollmann, 2019).

Cependant, la normalisation n'est pas applicable aux dialectes alsaciens :

> « *First, there is no consensus on the writing norm for Alsatian dialects and it is thus difficult to decide which form should prevail. Moreover, even though Alsatian is closely related to German, there are a number of lexical and syntactic differences which have to be taken into account* » (Bernhard, 2014).

Mais grâce à ces travaux, nous comprenons mieux le problème de variation graphique et nous pouvons nous inspirer de certaines des méthodes utilisées dans leur travail.

2. *Myriadisation (crowdsourcing)*

La plupart des tâches de traitement de langues non standardisées ne peuvent pas être résolues uniquement par des algorithmes basés sur des machines. Par conséquent, la myriadisation a suscité l'intérêt des chercheurs, qui profitent des connaissances des locuteurs à la création des ressources langagières pour certaines langues peu dotées.

Sood et al. (2012) ont opté pour le crowdsourcing afin d'étiqueter leur corpus. En particulier, ils ont utilisé « *Amazon Mechanical Turk* », une plateforme web de crowdsourcing, pour leur fournir un ensemble de données étiquetées, et des jugements sur chaque commentaire comme mesure. Enfin, ils ont utilisé des machines à vecteurs de support (en anglais *support-vector machine*, *SVM*) pour la détection de blasphème (ou *profanity* en anglais).

Millour (2020) a développé la plateforme « Recettes de Grammaire » pour la collecte de variantes graphiques auprès des locuteurs de l'alsacien. Ensuite, Chaque paire de variantes alignées est utilisée pour extraire automatiquement des règles de variation. De plus, ces règles sont contraintes par différentes correspondances des contextes,

> « c'est-à-dire forçant ou non l'alignement des contextes gauche et droit [...] soit la correspondance des contextes gauches (L), soit celle des contextes droits (R), ou soit celle des deux (L + R) » (Millour, 2020, 154-155).

Enfin, elle a appliqué les règles de variation à l'appariement automatiquement des graphies alternatives potentielles. Dans cette étude, nous utilisons la méthode de Millour (2020) pour aligner les variantes et extraire les règles.

3. *Partitionnement de données (clustering)*

Le clustering est le processus qui consiste à regrouper des données similaires en fonction d'un critère particulier (par exemple distance de Levenstein), sans qu'il soit nécessaire d'attribuer une étiquette aux données, mais dans le but d'agréger des données similaires : après le clustering, les données de la même classe sont regroupées autant que possible, et les données de classes différentes sont séparées autant que possible. Le clustering est une méthode d'apprentissage non supervisée.

Une telle approche est applicable aux langues non standardisées, il consiste à repérer les variantes sans les normaliser (Ruiz Fabo et al., 2020). Par exemple, Dasigi et Diab (2011) ont abordé le problème de l'identification des variantes orthographiques dans

l'arabe dialectal : si deux chaînes de caractères sont similaires, ils les regroupent. Le travail de Bernhard (2014) ne cherche pas à normaliser les variantes graphiques en alsacien mais à les identifier et les agréger sous forme de cluster. Ne disposant pas d'informations sur les formes standard dans le jeu de données, Rafae et al. (2015) ont proposé un algorithme phonétique, *« UrduPhone »*, adapté à l'urdu romain afin de produire un premier groupement de différentes variantes orthographiques d'un mot.

Les méthodes de clustering sont souvent utilisées lorsque l'objectif n'est pas de produire la meilleure normalisation, mais de regrouper un ensemble de variantes orthographiques.

4. Classification

La classification consiste à prédire automatiquement une classe pour des données. Il s'agit d'une méthode d'apprentissage supervisée. Elle obtient un classificateur en entraînant le jeu de données et utilise ensuite le classificateur pour prédire les données inconnues.

Barteld et al. (2019) ont proposé une chaîne de traitement pour détecter des variantes orthographiques en moyen bas allemand à l'aide de méthodes supervisées. Tout d'abord, ils génèrent des variantes candidates par deux voies, soit à partir d'une liste de variantes connues, soit à partir de la distance de Levenshtein modifiée. Ensuite, les variantes candidates générées par la distance de Levenshtein sont filtrées au moyen d'un SVM. Enfin, ils utilisent un réseau neuronal convolutif (CNN) pour classer les paires de variantes.

Dans ce chapitre, nous avons fourni un bref état de l'art des méthodes existantes liées à l'identification des variantes graphiques. Les chercheurs ont adopté différentes approches en fonction de la ressource langagière. Dans le cadre de nos travaux, vu qu'il s'agit de corpus non contemporains, notamment le théâtre en alsacien, la variation orthographiques dépend de la zone dialectale, le dramaturge, la période, etc., il est nécessaire d'extraire des formes caractéristiques des différentes divisions sur la base d'une comparaison. Ensuite, nous pouvons nous orienter vers les approches mentionnées par Bernhard (2014) et Millour (2020) pour nous aider à repérer les variantes orthographiques à partir des formes caractéristiques de chaque sous-corpus. Nous allons par la suite décrire la mesure de discriminativité que nous avons utilisée pour extraire des formes caractéristiques.

Chapitre 4. Extraction des caractéristiques via Zeta

Les mesures de discriminativité permettent d'analyser notre corpus dans une perspective contrastive et d'extraire des formes qui sont caractéristiques des différentes divisions du corpus selon les métadonnées(zone dialectale, la période, le dramaturge, etc.). Il existe de nombreuses méthodes, et dans cette étude, nous nous référons principalement à la mesure de Zeta.

Zeta a été utilisé pour la stylométrie ou l'attribution d'auteur qui consiste à identifier l'auteur le plus probable d'un texte parmi un ensemble de candidats. Zeta a été proposé pour la première fois par Burrows (2007), et il existe plusieurs variantes de Zeta proposées par Kinney et Craig (2009) et par Schöch et al. (2018). En général, cette mesure quantifie les degrés de dispersion d'une caractéristique dans deux corpus et les compare. La formule de base de Zeta est simple :

$$zeta_t = sp_t(G_1) - sp_t(G_2)$$

$$sp_t(G_1) = \frac{s_t(G_1)}{s(G_1)},\ sp_t(G_2) = \frac{s_t(G_2)}{s(G_2)}$$

- $zeta$: le score zeta de chaque terme dans un document ;
- t : un terme ou une forme ;
- sp_t : la proportion de segments dans lesquels ce terme (une forme ou une caractéristique) apparaît au moins une fois (binaire) ;
- $G1$: un groupe de textes (corpus 1) ;
- $G2$: un autre groupe de textes (corpus 2) ;
- s : Nombre de tous les segments d'un corpus ;
- s_t : Nombre de tous les segments d'un corpus qui ont au moins une occurrence de la forme.

À partir de cette formalisation, Kinney et Craig (2009) utilisent les fréquences relatives (rf) au lieu des proportions de segments (sp) ; Schöch et al. (2018) appliquent une transformation logarithmique aux valeurs *rf* et *sp* au lieu de les utiliser directement.

Zeta divise chaque document d'un groupe en segments égaux, s'il y a plus de termes dans le document (c'est-à-dire avec un document plus long), plus il y aura de termes dans chaque segment pour le même nombre de segments. Selon Schöch et al. (2018), la longueur du contenu d'un seul segment affecte le score zeta, ils ont ainsi évalué l'effet de différents paramètres sur les performances de zeta dans une tâche de classification. Schöch et al. (2018) ont résume que le Zeta de Burrows (2007) est particulièrement peu performant avec les petits segments (50, 100 termes) et particulièrement performant avec les grands segments (>10000 termes)[4]. La variante de Zeta proposée par Schöch et al. (2018) donne de meilleurs résultats que la Zeta de Burrows et est plus robuste en ce qui concerne la taille des segments. Nous résumons les informations de zeta et créons une carte mentale (voir l'annexe 1).

[4] la taille du segment : combien de termes sont dans un segment.

Chapitre 5. Alignement de texte

Comme nous l'avons vu au chapitre 3, la plupart des études d'identification de variantes ont utilisé des méthodes d'alignement. Dans le chapitre 3, nous décrivons la méthode de Bernhard (2014)**.** Elle a repéré les variantes en alsacien en effectuant de l'alignement à partir de ressources externes et le double metaphone. En particulier, Bernhard (2014) a proposé l'algorithme Double Metaphone adapté aux dialectes alsaciens dont nous profitons dans cette mémoire. En outre, Prokić et al. (2009) a utilisé ALPHAMALIG (Alonso et al., 2004) avec des transcriptions phonétiques de dialectes bulgares. ALPHAMALIG est un outil MSA (*Multiple Sequence Alignment*) pour la détection de motifs dans un ensemble de séquences comparables.

Pour ALPHAMALIG, l'entrée est un ensemble de séquences et un critère de similarité qui décrit la similarité entre les différents symboles qui constituent les séquences. Tout d'abord, la paire de séquences la plus similaire selon le critère de similarité est trouvée, et elles sont alignées. Ensuite, une nouvelle séquence contenant les différents symboles qui constituent les séquences alignées est créée. Enfin, les séquences alignées sont enlevées de l'ensemble de séquences, et sont remplacées par la nouvelle séquence résultant de l'alignement. Le processus est répété jusqu'à ce que l'on trouve une seule séquence contenant tous les symboles constituant l'ensemble de séquences. Le résultat est un ensemble de séquences alignées et une similarité relative entre elles.

Millour (2020) a adopté l'outil ALPHAMALIG dans son travail pour aligner les variantes myriadisées. Suivant les recommandations de Millour (2020), cet outil a également été utilisé dans notre travail pour l'alignement des n-grammes de caractères.

Partie 3

-

Collecte et préparation des données

Chapitre 6. Construction du corpus

Dans ce chapitre, nous allons présenter nos sources et la structure de document TEI du corpus original. Notre procédure de récolte de jeu de données utilisé dans cette recherche est ensuite abordée, ainsi que la description du corpus final.

1. Corpus original

Le corpus original encodé en format TEI et les métadonnées des pièces pour cette étude sont proposés par le projet MeThAL[5], et une interface pour explorer le corpus est disponible sur internet[6]. La figure 1 présente l'interface d'exploration de corpus.

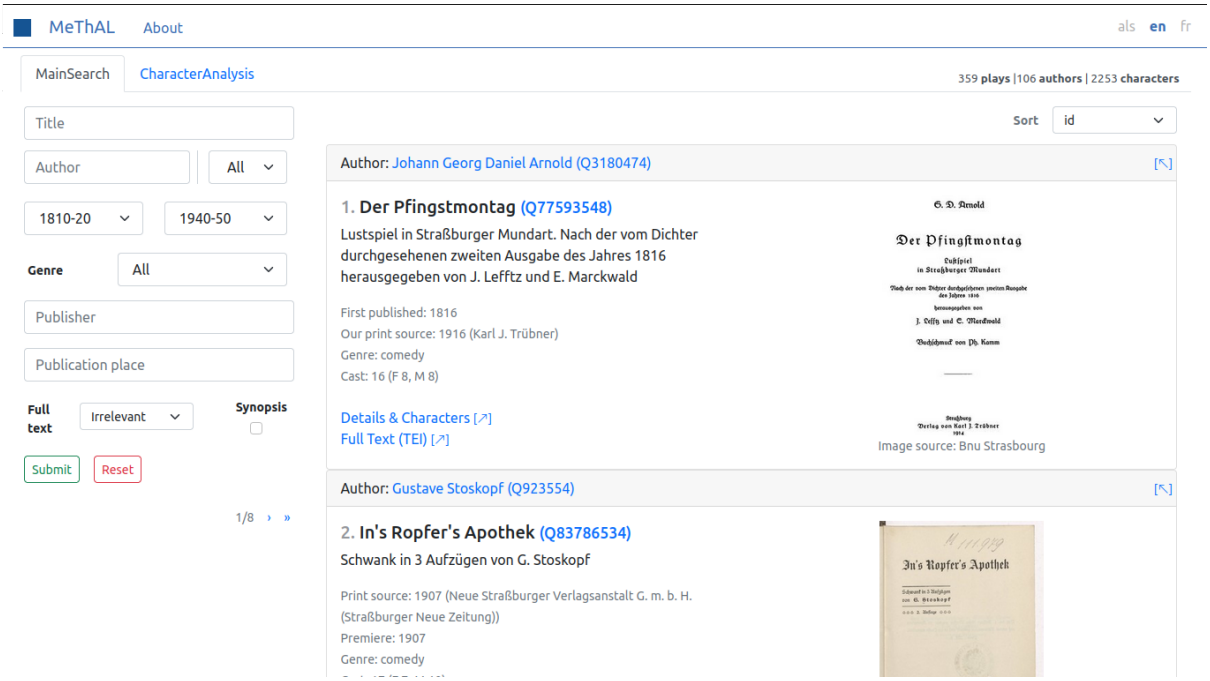


Figure 1. MeThAL Corpus Explorer

1.1. Sources du corpus

La source première du corpus, disponible sur Numistral[7], est une collection représentative d'environ 150 pièces de théâtre en alsacien. C'est une ressource numérisée en 2019 par la Bibliothèque nationale et universitaire (Bnu) à Strasbourg, mais la source est une numérisation en mode image qui demande des améliorations en vue de faciliter la recherche pluridisciplinaire. Le projet MeThAL a sélectionné un sous-ensemble des pièces couvrant principalement la période 1870-1940 et a effectué une océrisation

[5] Le corpus est sur https://gitlab.huma-num.fr/methal/corpus-methal-all
[6] Source : https://methal.eu/ui/
[7] Source : https://numistral.fr/fr/theatre-alsacien (lien [Découvrir] pour explorer la collection)

(reconnaissance automatique des caractères ou OCR) sur la base des images disponibles sur Numistral, et après la correction de l'OCR, les a encodées en TEI.

Une autre partie de la source du corpus provient de Wikisource[8], en format wiki-markup. Cette source est une collection qui comprend des œuvres complètes d'un seul auteur, August Lustig, et les œuvres ont été transformées en TEI par script lors d'un stage en 2021 dans le projet MeThAL. Enfin, il y a un certain nombre de pièces océrisées et encodées en TEI, mais elles ne sont pas encore publiées par le projet.

Le volume du corpus encodé en TEI atteint 77 pièces, et le nombre de documents par source est indiqué dans le tableau 2 ci-dessous :

Source	Nombre de documents
Numistral	25
Wikisource	26
Numistral (pas publiées)[9]	26
Total	77

Tableau 2. Distribution des sources

1.2. Structure de texte TEI

Ces documents XML-TEI comportent de nombreux éléments qui détaillent les métadonnées, les informations sur les acteurs, les informations sur les personnages, les répliques, etc. Par conséquent, nous avons besoin d'analyser la structure des textes TEI dans le but d'en extraire les données utiles à notre recherche.

Chaque document TEI du corpus est enregistré en format XML et le nom du fichier est défini par l'auteur et le titre principal de la pièce. Pour le corpus venant de Wikisource, les pièces sont toutes de l'auteur August Lustig, donc les noms de fichiers dans cette source sont nommés par le titre uniquement, et leurs dossiers parents sont nommés par le nom de l'auteur. L'exemple de la structure du corpus est présenté dans la figure 2.

[8] Source : https://als.wikipedia.org/wiki/Text:August_Lustig/A._Lustig_S%C3%A4mtliche_Werke:_Band_2
[9] Versions de travail pas encore publiées sur un entrepôt de données par le projet, et sans DOI. Utilisables sachant que : Il y a eu moins de validation que pour les pièces publiées ; Il y aura donc plus d'erreurs que dans ces dernières ; Il manque de gérer les traits d'union en fin de ligne.

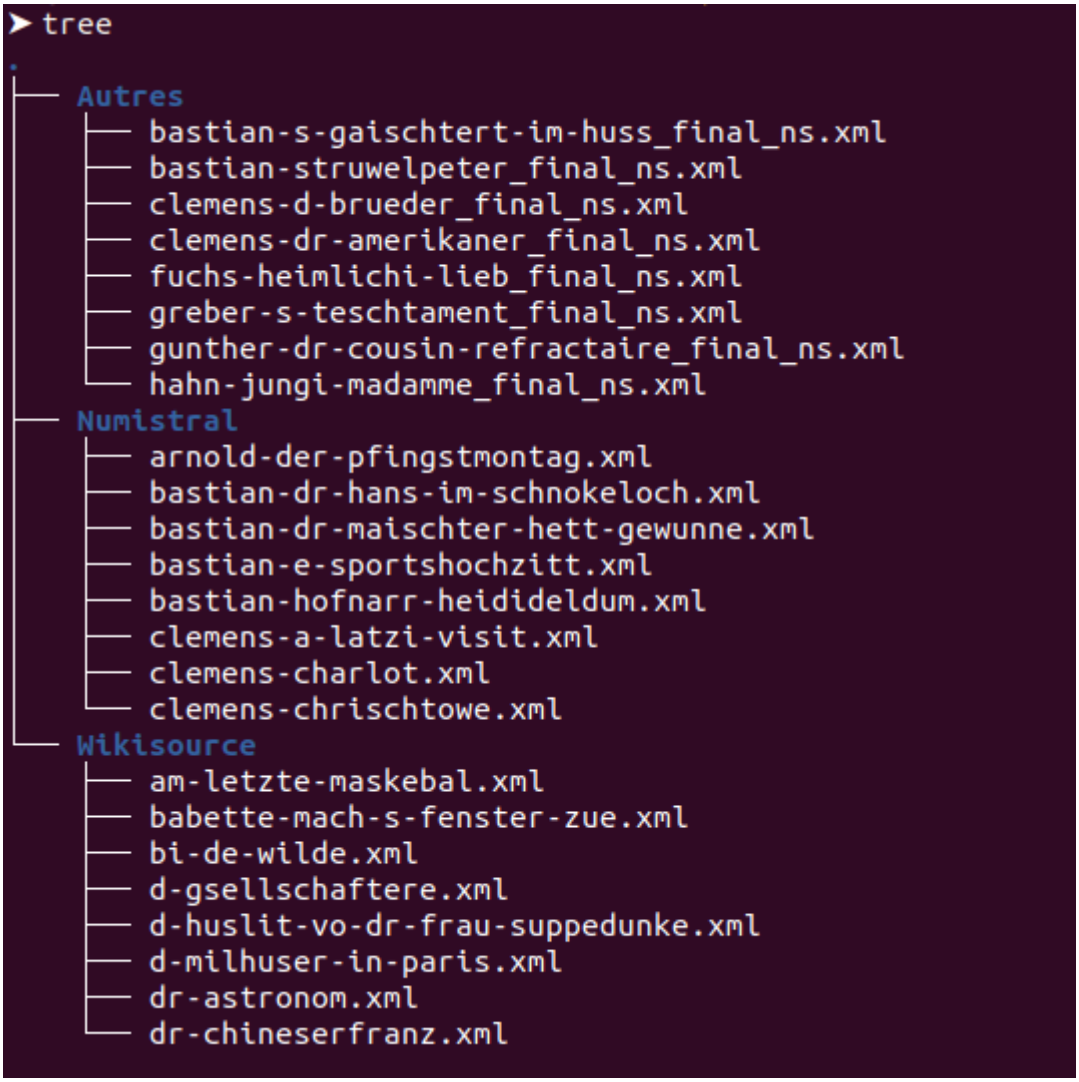


Figure 2. Exemple de la structure du corpus

Tous les fichiers TEI possèdent un élément racine qui encadre l'ensemble du fichier : l'élément <TEI> qui se compose ensuite de deux sous-parties :

- L'en-tête permettant d'indiquer les métadonnées du document : représenté par l'élément <teiHeader> ;
- Le texte du document : représenté par l'élément <text>.

Un texte sera encodé en utilisant une structure initiale comme celle-ci :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="../work/css/tei-
drama.css"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
   <teiHeader>
   ...
   </teiHeader>
   <text>
   ...
   </text>
</TEI>
```

Parmi ces éléments, il y a des informations qui sont importantes mais non exploitables pour notre recherche, nous devons d'abord préciser de quelle partie des données nous avons besoin, puis nous nous concentrer sur la partie pertinente.

Dans notre étude, les données d'intérêt dans ces éléments sont les textes bruts en alsacien, et nous avons également besoin des métadonnées des pièces pour classer les données en vue de l'analyse.

L'en-tête permet d'indiquer toutes les métadonnées associées au document numérique lui-même, de manière analogue à la page de titre d'un livre imprimé. Dans les fichiers XML de notre corpus, il se compose de trois parties :

- L'élément <fileDesc> contient une description bibliographique complète du fichier électronique ;
- L'élément <profileDesc> fournit les éléments de description des aspects non bibliographiques du texte, qui dans notre cas contient la description des personnages sur scène et des relations entre eux ;
- L'élément <encodingDesc> décrit l'encodage effectué, en particulier le choix des balises ou des ponctuations.

Dans les métadonnées, nous avons besoin d'informations telles que l'auteur de la pièce, la date, le lieu de publication, le lieu de naissance, etc. Évidemment, nous pouvons obtenir ces informations dans l'élément <fileDesc> (voir la figure 3), par exemple, nous pouvons obtenir le titre principal de la pièce dans l'élément <title type="main"> et l'auteur dans <author>.

L'élément <text> encadre le texte du document, il est composé de deux éléments :

- L’élément <front> contient une liste des acteurs et des informations sur le contexte de la pièce ;
- L'élément <body> est la partie la plus importante, contenant les textes des acteurs, et constitue le corps de la pièce.

La figure 4 montre clairement la structure hiérarchique de l'élément <text> dans l'un des fichiers, qui est essentiellement la structure des autres fichiers XML.

```
<fileDesc>
  <titleStmt>
  <publicationStmt>
    <publisher>LiLPa - Université de Strasbourg</publisher>
    <availability>
    <idno type="doi" xml:base="https://doi.org/">10.34847/nkl.7a84q6be</idno>
    <idno type="methal" xml:id="mtl-020"/>
    <idno xml:base="https://www.wikidata.org/entity/" type="wikidata">Q106592612</idno>
  </publicationStmt>
  <sourceDesc>
    <bibl type="digitalSource">
      <name>Numistral</name>
      <idno type="URL">https://www.numistral.fr/ark:/12148/bpt6k9109742m</idno>
      <bibl type="originalSource">
        <author key="wikidata:Q106592612">Ferdinand Bastian</author>
        <title type="main">D'r Hans im Schnokeloch</title>
        <title type="sub">Volksspiel in 4 Aufzügen mit Musik, Gesang und Tanz von Ferd. Bastian</title>
        <series>Elsässische Volksbucher, No 12</series>
        <edition>2</edition>
        <publisher>Druck und Verlag: Imprimerie du Nouveau Journal de Strasboug S. à. r. l.</publisher>
        <date when="1930" type="print">1930</date>
        <date when="1903" type="written">1903</date>
        <note>The first edition was published by C. A. Vomhoff (Strasbourg) in 1903.</note>
      </bibl>
    </bibl>
  </sourceDesc>
</fileDesc>
```

Figure 3. Exemple de l'élément <fileDesc>

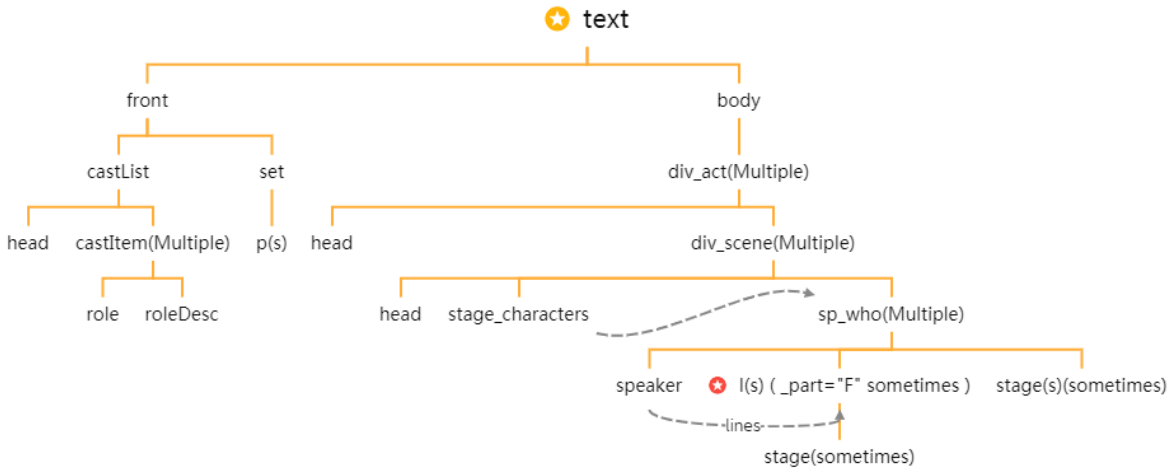
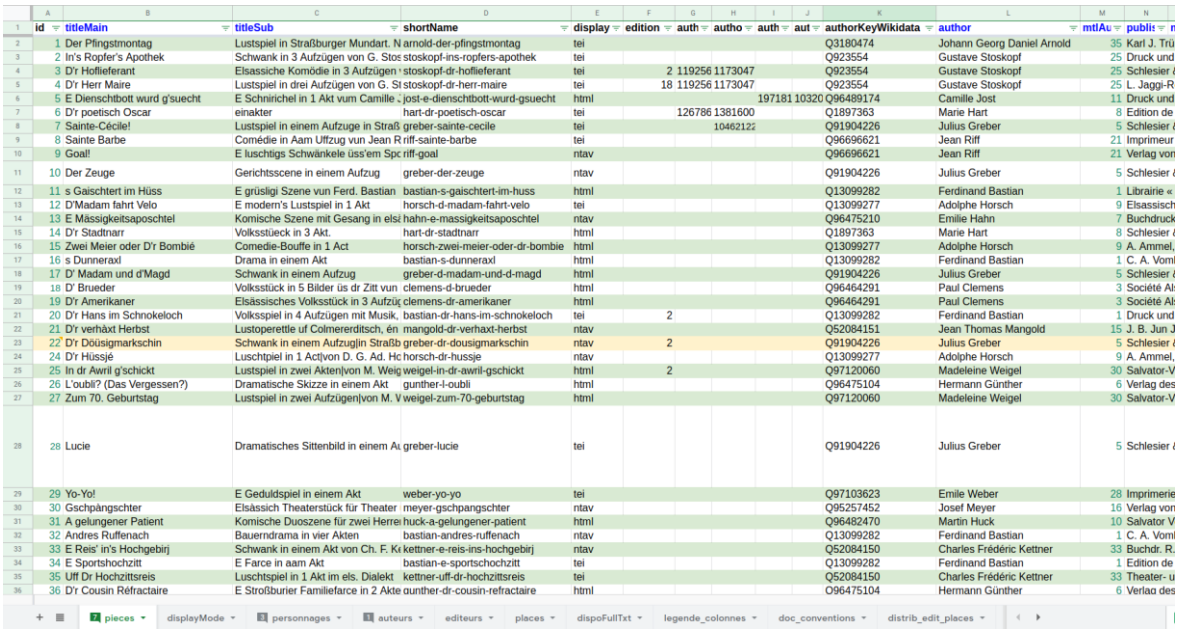


Figure 4. Exemple de la structure de l'élément <text>

2. *Métadonnées pour les pièces*

Bien que chaque document XML-TEI contienne des métadonnées pour la pièce, certains documents ont des métadonnées incomplètes. Il faut obtenir les métadonnées manquantes avec un classeur (voir figure 5) qui contient des informations complètes sur toutes les pièces. La procédure exacte pour obtenir des métadonnées par ce classeur sera détaillée dans la section suivante.



id	titleMain	titleSub	shortName	display	edition	auth	autho	auth	aut	authorKeyWikidata	author	mtlAu	publis
1	Der Pfingstmontag	Lustspiel in Straßburger Mundart. N	arnold-der-pfingstmontag	tei						Q3180474	Johann Georg Daniel Arnold	35	Karl J. Trü
2	In's Ropfer's Apothek	Schwank in 3 Aufzügen von G. Sto	stoskopf-ins-ropfers-apothek	tei						Q923554	Gustave Stoskopf	25	Druck und
3	D'r Hoflieferant	Elsassiche Komödie in 3 Aufzügen '	stoskopf-dr-hoflieferant	tei	2	119256	1173047			Q923554	Gustave Stoskopf	25	Schlesier
4	D'r Herr Maire	Lustspiel in drei Aufzügen von G. St	stoskopf-dr-herr-maire	tei	18	119256	1173047			Q923554	Gustave Stoskopf	25	L. Jaggi-R
5	E Dienschtbott wurd g'suecht	E Schnirichel in 1 Akt vum Camille _	jost-e-dienschtbott-wurd-gsuecht	html				197181	10320	Q96489174	Camille Jost	11	Druck und
6	D'r poetisch Oscar	einakter	hart-dr-poetisch-oscar	tei		126786	1381600			Q1897363	Marie Hart	8	Edition de
7	Sainte-Cécile!	Lustspiel in einem Aufzuge in Straß	greber-sainte-cecile	tei			10462122			Q91904226	Julius Greber	5	Schlesier
8	Sainte Barbe	Comédie in Aam Uffzug vun Jean R	riff-sainte-barbe	tei						Q96696621	Jean Riff	21	Imprimeur
9	Goal!	E luschtigs Schwänkele üss'em Spo	riff-goal	ntav						Q96696621	Jean Riff	21	Verlag von
10	Der Zeuge	Gerichtsscene in einem Aufzug	greber-der-zeuge	ntav						Q91904226	Julius Greber	5	Schlesier
11	s Gaischtert im Hüss	E grüsligi Szene vun Ferd. Bastian	bastian-s-gaischtert-im-huss	html						Q13099282	Ferdinand Bastian	1	Librairie «
12	D'Madam fahrt Velo	E modern's Lustspiel in 1 Akt	horsch-d-madam-fahrt-velo	tei						Q13099277	Adolphe Horsch	9	Elsassisch
13	E Mässigkeitsaposchtel	Komische Szene mit Gesang in elsä	hahn-e-massigkeitsaposchtel	ntav						Q96475210	Emilie Hahn	7	Buchdruck
14	D'r Stadtnarr	Volksstüeck in 3 Akt.	hart-dr-stadtnarr	html						Q1897363	Marie Hart	8	Schlesier
15	Zwei Meier oder D'r Bombié	Comedie-Bouffe in 1 Act	horsch-zwei-meier-oder-dr-bombie	html						Q13099277	Adolphe Horsch	9	A. Ammel,
16	s Dunneraxl	Drama in einem Akt	bastian-s-dunneraxl	html						Q13099282	Ferdinand Bastian	1	C. A. Voml
17	D' Madam und d'Magd	Schwank in einem Aufzug	greber-d-madam-und-d-magd	html						Q91904226	Julius Greber	5	Schlesier
18	D' Brueder	Volksstück in 5 Bilder üs dr Zitt vun	clemens-d-brueder	html						Q96464291	Paul Clemens	3	Société Al
19	D'r Amerikaner	Elsässisches Volksstück in 3 Aufzü	clemens-dr-amerikaner	html						Q96464291	Paul Clemens	3	Société Al
20	D'r Hans im Schnokeloch	Volksspiel in 4 Aufzügen mit Musik,	bastian-dr-hans-im-schnokeloch	tei	2					Q13099282	Ferdinand Bastian	1	Druck und
21	D'r verhäxt Herbst	Lustoperette uf Colmererditsch, én	mangold-dr-verhaxt-herbst	ntav						Q52084151	Jean Thomas Mangold	15	J. B. Jun J
22	D'r Döüsigmarkschin	Schwank in einem Aufzug(in Straßb	greber-dr-dousigmarkschin	ntav	2					Q91904226	Julius Greber	5	Schlesier
23	D'r Hüssjé	Luschtpiel in 1 Act\von D. G. Ad. Hc	horsch-dr-hussje	ntav						Q13099277	Adolphe Horsch	9	A. Ammel,
24	In dr Awril g'schickt	Lustspiel in zwei Akten\von M. Weig	weigel-in-dr-awril-gschickt	html	2					Q97120060	Madeleine Weigel	30	Salvator-V
25	L'oubli? (Das Vergessen?)	Dramatische Skizze in einem Akt	gunther-l-oubli	html						Q96475104	Hermann Günther	6	Verlag des
26	Zum 70. Geburtstag	Lustspiel in zwei Aufzügen\von M. V	weigel-zum-70-geburtstag	html						Q97120060	Madeleine Weigel	30	Salvator-V
27	Lucie	Dramatisches Sittenbild in einem Au	greber-lucie	tei						Q91904226	Julius Greber	5	Schlesier
28	Yo-Yo!	E Geduldspiel in einem Akt	weber-yo-yo	tei						Q97103623	Emile Weber	28	Imprimerie
29	Gschpängschter	Elsässich Theaterstück für Theater	meyer-gschpangschter	ntav						Q95257452	Josef Meyer	16	Verlag von
30	A gelungener Patient	Komische Duoszene für zwei Herre	huck-a-gelungener-patient	html						Q96482470	Martin Huck	10	Salvator V
31	Andres Ruffenach	Bauerndrama in vier Akten	bastian-andres-ruffenach	ntav						Q13099282	Ferdinand Bastian	1	C. A. Voml
32	E Reis' in's Hochgebirj	Schwank in einem Akt von Ch. F. K	kettner-e-reis-ins-hochgebirj	ntav						Q52084150	Charles Frédéric Kettner	33	Buchdr. R.
33	E Sportshochzitt	E Farce in aam Akt	bastian-e-sportshochzitt	tei						Q13099282	Ferdinand Bastian	1	Edition de
34	Uff Dr Hochzittsreis	Luschtspiel in 1 Akt im els. Dialekt	kettner-uff-dr-hochzittsreis	tei						Q52084150	Charles Frédéric Kettner	33	Theater- u
35	D'r Cousin Réfractaire	E Straßburier Familiefarce in 2 Akte	gunther-dr-cousin-refractaire	html						Q96475104	Hermann Günther	6	Verlag des

pieces | displayMode | personnages | auteurs | editeurs | places | dispoFullTxt | legende_colonnes | doc_conventions | distrib_edit_places

Figure 5. Exemple du classeur google docs

Chapitre 7. Récolte de données[10]

Ayant ainsi précisé la définition de la ressource originale, il a ensuite fallu récolter les données. Les documents XML-TEI du corpus original ne peuvent pas être exploités directement ; ce dont nous avons besoin, c'est du texte brut du dialecte alsacien dans les documents XML et des métadonnées associées. De plus, une structure de données plus appropriée pour représenter le corpus faciliterait l'étude des variantes graphiques.

Cela aurait été un travail immense et irréalisable que d'extraire et d'enregistrer les données au bon format à la main. Pour automatiser cette tâche, le Document Object Model, ou « DOM », a répondu parfaitement à nos besoins, le DOM est une API inter-langage du World Wide Web Consortium (W3C) pour accéder et manipuler les documents XML. Plus précisément, une implémentation DOM présente le document XML comme un arbre et les feuilles de l'arbre sont définies comme des nœuds. Tous les éléments sont accessibles à travers l'arborescence du DOM, leur contenu peut être modifié ou supprimé, et de nouveaux éléments peuvent être créés. Les éléments, leur texte et leurs attributs sont considérés comme des nœuds. Dans cette étude, nous ne faisons que consulter ces contenus sans les modifier. Cela permet de manipuler les documents XML en manipulant l'arbre et les nœuds, fournissant ainsi un cadre conceptuel pour traiter tous les aspects du document.

Voici un exemple d'arborescence de XML-DOM (figure 6) :

[10] Lien de script : https://gitlab.huma-num.fr/methal/corpus-methal-all/-/tree/main/code/script/text_extract

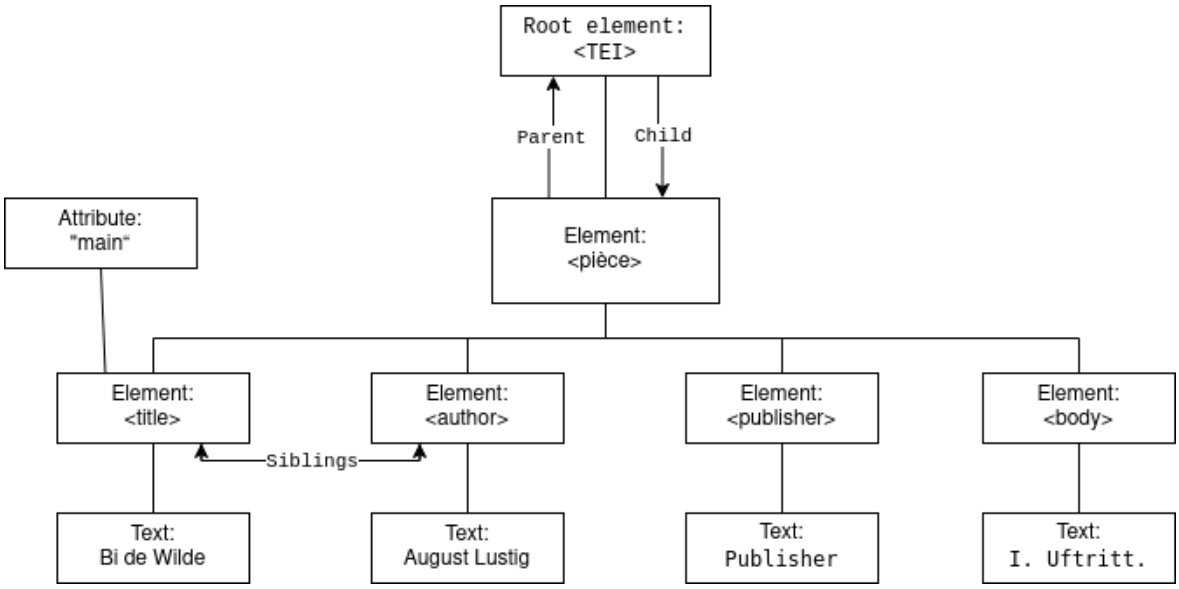


Figure 6. Exemple d'arborescence de XML DOM

1. Analyseurs syntaxiques XML (parseurs XML)

Le XML-DOM contient des méthodes (fonctions) permettant de parcourir l'arbre XML, d'accéder aux nœuds, de les insérer et de les supprimer. Toutefois, avant de pouvoir accéder à un document XML et le manipuler, il doit être chargé dans un objet XML-DOM, un analyseur syntaxique utilisant DOM prend en entrée un document XML et construit, à partir de cela, un arbre formé d'objets : chaque objet appartient à une sous-classe de nœud et des opérations sur ces objets permettent de créer de nouveaux noeuds, ou de naviguer dans le document.

Dans cette recherche, nous employons les bibliothèques Python « lxml[11] » et « Beautiful Soup[12] » pour l'objectif de nous aider à analyser les documents XML. Ces bibliothèques sont faciles à utiliser, rapides avec les documents volumineux et permettent une conversion facile des données, ce qui facilite la manipulation des fichiers. En outre, dans l'environnement Python, nous pouvons ensuite utiliser d'autres bibliothèques puissantes pour analyser les données.

[11] Source : https://lxml.de/
[12] Source : https://www.crummy.com/software/BeautifulSoup/

2. Data mapping avec Beautiful Soup

Beautiful Soup fournit quelques fonctions simples, de style Python, pour gérer la navigation, la recherche, la modification des arbres d'analyse, etc. Il s'agit d'une boîte à outils qui fournit aux utilisateurs les données dont ils ont besoin en analysant les documents, et en raison de sa simplicité, il ne nécessite pas beaucoup de code pour écrire une application complète.

Beautiful Soup est compatible avec les parseurs de la bibliothèque standard de Python, ainsi qu'avec certains parseurs tiers. Comme mentionné ci-dessus, nous avons adopté le parseur lxml. Par rapport à l'API ElementTree, lxml est plus rapide et plus facile à utiliser.

Concernant l'utilisation concrète de cette bibliothèque, nous l'avons réalisé sur les documents TEI du corpus original en exécutant des scripts en Python. Premièrement, nous mettons en correspondance un document TEI et un objet Python. Deuxièmement, nous profitons de cette implémentation pour générer un dataframe pandas et pour stocker les informations extraites sous un fichier CSV. Finalement, les métadonnées manquantes dans le fichier TEI sont complétées à l'aide du classeur mentionnées dans la section 1.3, et les pièces sont ensuite classées en fonction des métadonnées pour construire un nouveau corpus.

Python est un langage qui permet la Programmation Orientée Objet (POO) et la POO permet de créer des entités (objets) que l'on peut manipuler. Dans notre cas, un document TEI peut être d'abord considéré comme un objet en Python. La figure 7 montre que le nom du fichier TEI et ses éléments XML peuvent être considérés comme des propriétés de l'objet Python. Ensuite, le contenu textuel des éléments liés au XML est récupéré par l'API de BeautifulSoup et attribué aux propriétés de l'objet Python :

```
class TEIFile(object):
    def __init__(self, filename):
        self.filename = filename
        self.soup = read_tei(filename)
        self._title = ''
        self._author = ''
        self._front = None
        self._body = None

    # sourceDesc
    def idnoMtl(self):
        tag_temp = self.soup.find('idno', type = 'methal')
        if tag_temp:
            idmtl = tag_temp.attrs.get('xml:id')
            self._idnoMtl = idmtl
        else:
            self._idnoMtl = tag_temp
        return self._idnoMtl

      @property
    def title(self):
        if not self._title:
            title_list = []
            for title in self.soup.bibl.find_all('title'):
                title_list.append(title.getText())
            titles = "\n".join(title_list)
            self._title = titles
        return self._title

      @property
    def author(self):
        if not self._author:
            self._author = self.soup.author.getText()
        return self._author
```

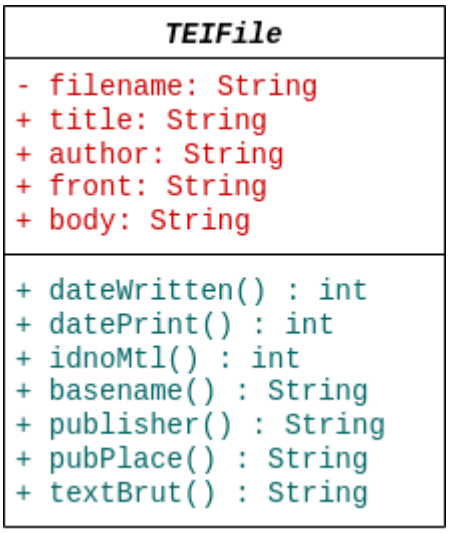


Figure 7. Diagramme UML de classe de fichier TEI

3. Corpus final

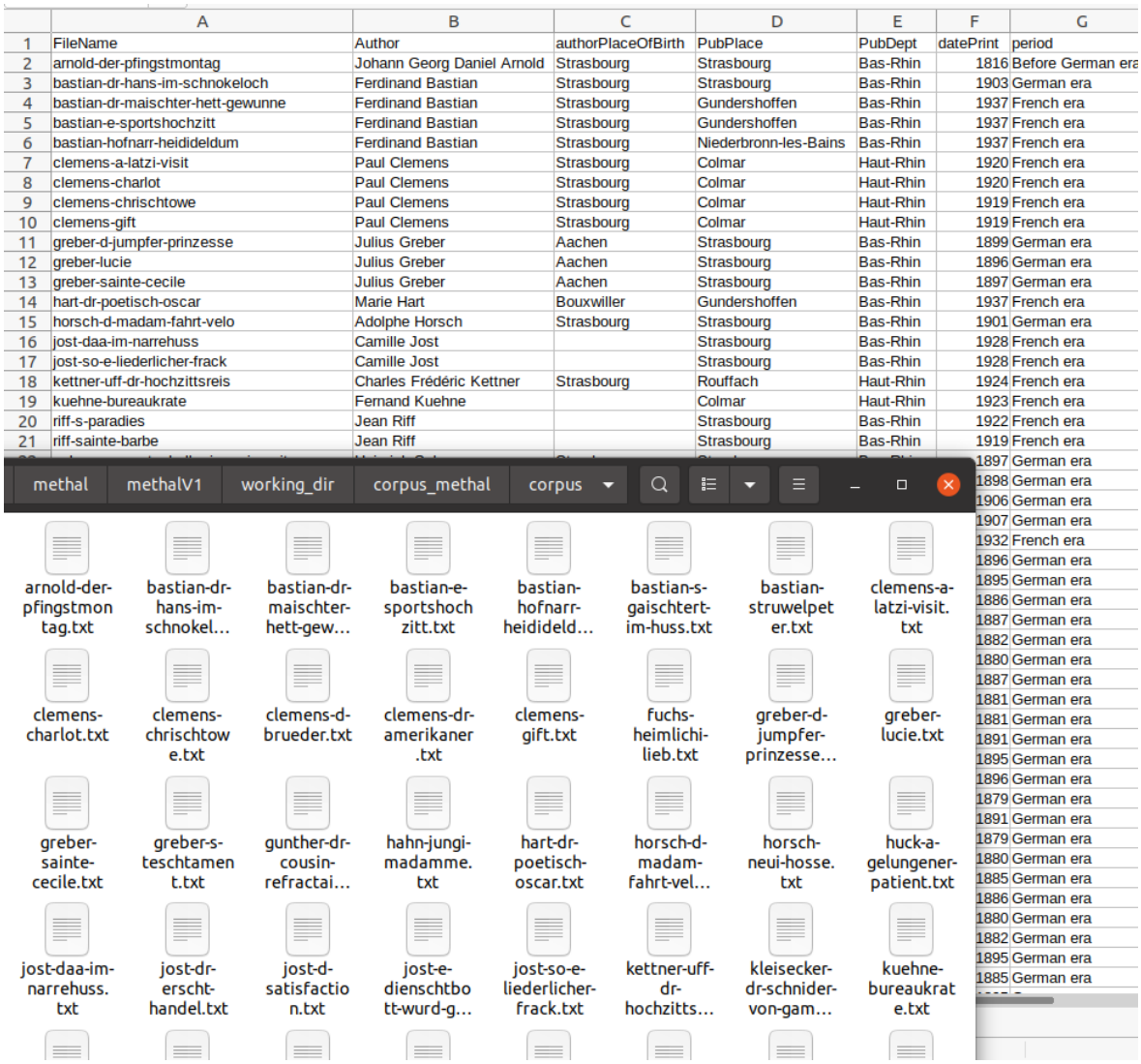
Il ne reste plus qu'à regrouper, classifier en fonction des métadonnées, et à structurer les données collectées. Les métadonnées que nous avons extraites pour le corpus sont :

- Idno (identifier) ;
- FileName ;
- Title ;
- Author ;
- authorPlaceOfBirth ;
- Publisher ;
- PubPlace : lieu de publication ;
- PubDept : département du lieu de publication, e.g., Bas-Rhin ou Haut-Rhin ;
- datePrint.

Nous avons choisi la date de publication plutôt que la date de rédaction pour l'information de la pièce, car la date de création n'est pas disponible pour certaines pièces. Parmi ces métadonnées, nous avons enlevé l'identifiant, le titre de la pièce et le nom de l'éditeur. En conséquence, nous disposons d'un corpus contenant uniquement du texte brut

en dialecte alsacien, ainsi que les métadonnées importantes pour ce corpus. La figure 8 présente la capture d'écran partielle du corpus final et de ses métadonnées.

Il est à noter que, comme on peut le voir dans la capture d'écran, des métadonnées peuvent manquer dans certains cas : dans la capture nous voyons que le lieu de naissance des auteurs n'est pas toujours disponible. Ceci est une limitation du corpus puisque pour l'étude de la variation il fallait assigner une zone dialectale aux pièces. Nous avons essayé de surmonter cette limitation de la façon suivante : lorsque le lieu de naissance de l'auteur manquait, nous avons assigné la région de la maison d'édition ayant publié la pièce. Il s'agit d'une méthode imparfaite mais que nous avons jugée suffisante pour les tâches d'exploration de méthodes d'extraction de règles de variation dans cette étude.



	A	B	C	D	E	F	G
1	FileName	Author	authorPlaceOfBirth	PubPlace	PubDept	datePrint	period
2	arnold-der-pfingstmontag	Johann Georg Daniel Arnold	Strasbourg	Strasbourg	Bas-Rhin	1816	Before German era
3	bastian-dr-hans-im-schnokeloch	Ferdinand Bastian	Strasbourg	Strasbourg	Bas-Rhin	1903	German era
4	bastian-dr-maischter-hett-gewunne	Ferdinand Bastian	Strasbourg	Gundershoffen	Bas-Rhin	1937	French era
5	bastian-e-sportshochzitt	Ferdinand Bastian	Strasbourg	Gundershoffen	Bas-Rhin	1937	French era
6	bastian-hofnarr-heidideldum	Ferdinand Bastian	Strasbourg	Niederbronn-les-Bains	Bas-Rhin	1937	French era
7	clemens-a-latzi-visit	Paul Clemens	Strasbourg	Colmar	Haut-Rhin	1920	French era
8	clemens-charlot	Paul Clemens	Strasbourg	Colmar	Haut-Rhin	1920	French era
9	clemens-chrischtowe	Paul Clemens	Strasbourg	Colmar	Haut-Rhin	1919	French era
10	clemens-gift	Paul Clemens	Strasbourg	Colmar	Haut-Rhin	1919	French era
11	greber-d-jumpfer-prinzesse	Julius Greber	Aachen	Strasbourg	Bas-Rhin	1899	German era
12	greber-lucie	Julius Greber	Aachen	Strasbourg	Bas-Rhin	1896	German era
13	greber-sainte-cecile	Julius Greber	Aachen	Strasbourg	Bas-Rhin	1897	German era
14	hart-dr-poetisch-oscar	Marie Hart	Bouxwiller	Gundershoffen	Bas-Rhin	1937	French era
15	horsch-d-madam-fahrt-velo	Adolphe Horsch	Strasbourg	Strasbourg	Bas-Rhin	1901	German era
16	jost-daa-im-narrehuss	Camille Jost		Strasbourg	Bas-Rhin	1928	French era
17	jost-so-e-liederlicher-frack	Camille Jost		Strasbourg	Bas-Rhin	1928	French era
18	kettner-uff-dr-hochzittsreis	Charles Frédéric Kettner	Strasbourg	Rouffach	Haut-Rhin	1924	French era
19	kuehne-bureaukrate	Fernand Kuehne		Colmar	Haut-Rhin	1923	French era
20	riff-s-paradies	Jean Riff		Strasbourg	Bas-Rhin	1922	French era
21	riff-sainte-barbe	Jean Riff		Strasbourg	Bas-Rhin	1919	French era

Figure 8. Capture d'écran partielle du corpus final et de ses métadonnées

Partie 4

-

Méthode

Chapitre 8. Pré-traitement

1. Découpage en tokens[13]

Le texte brut de notre corpus est constitué de phrases, que nous devons découper en plus petites unités à l'aide, tout d'abord, de la tokénisation. Cette tâche est particulièrement complexe pour l'alsacien en raison de l'absence de convention orthographique stable.

Le tokéniseur adapté à l'alsacien (Bernhard, 2018) permet à une application de découper la chaîne de caractères en tokens séparés par des délimiteurs. Grâce au paquet multiprocessing de Python[14], nous avons ajouté une fonction à l'application qui permet au tokéniseur de traiter plusieurs fichiers en même temps. Cela nous donne également un corpus au niveau des tokens[15].

2. N-grammes de caractères[16]

Afin d'explorer les habitudes de scripturalisation du corpus et de détecter les variantes orthographiques sur des données à un niveau de granularité plus bas, un corpus au niveau des n-grammes de caractères est nécessaire. Par exemple, dans la détection de variantes effectuée sur des textes du Moyen Bas Allemand, Barteld et al. (2019) génèrent des variantes candidates qui sont ensuite filtrées à partir des n-grammes de caractères qu'elles contiennent.

L'algorithme d'obtention de n-grammes de caractères est complexe et ne fait pas partie des priorités de notre recherche ; nous recourons à *stylo* (Eder et al., 2016), un package dans R, pour générer des n-grammes. Ce package fournit un certain nombre de fonctions, complétées par une interface graphique, pour effectuer diverses analyses dans le domaine de la stylistique computationnelle, de l'attribution d'auteur, etc., nous profitons de leur implémentation d'une série de pipelines standard pour le traitement de texte, ainsi que d'un certain nombre de métriques de similarité. Nous utilisons la fonction dans *stylo* qui effectue les étapes de prétraitement nécessaires à la sélection des caractéristiques :

```
txt.to.features(tokenized.text, features = "w", ngram.size = 1)
```

- `tokenized.text` : un vecteur de mots tokénisés ;

[13] Script du tokéniseur: https://gitlab.huma-num.fr/methal/corpus-methal-all/-/tree/main/code/script/token
[14] Source : https://docs.python.org/fr/3/library/multiprocessing.html
[15] https://gitlab.huma-num.fr/methal/corpus-methal-all/-/tree/main/code/working_dir/tokens
[16] Script pour N-grammes: https://gitlab.huma-num.fr/methal/corpus-methal-all/-/tree/main/code/Rstylo

- features : une option permettant de spécifier le type de caractéristique souhaité : w pour les mots, c pour les caractères (par défaut : w) ;
- ngram.size : un argument optionnel (entier) indiquant la valeur de n, ou la taille des n-grammes à créer. Si cet argument est absent, la valeur par défaut de 1 est utilisée.

Cette fonction nous permet de convertir un texte d'entrée dans le type de séquences nécessaires (n-grammes etc.) et retourne un nouveau vecteur d'éléments. Toutefois, elle a une limite du format du texte d'entrée. Nous remarquons que l'entrée à chaque fonction du pipeline fourni dans *stylo* dépend de l'objet retourné à l'étape précédente (sauf pour l'importation des données initiales), et nous ne pouvons pas importer directement un corpus sous l'autre format dans la fonction à une étape intermédiaire.

En premier lieu, il faut importer notre corpus au niveau des tokens dans *stylo* comme données initiales[17] :

```
raw.corpus <- load.corpus(files = "all",
                          corpus.dir = "corpus_als_token",
                          encoding = "UTF-8")
```

En second lieu, nous transformons les tokens sous un format valide pour *stylo*, sans changer le contenu des données, en utilisant le tokéniseur intégré à *stylo* :

```
tokenized.corpus <- txt.to.words(raw.corpus,
                                 splitting.rule="[[:space:]]+",
                                 preserve.case = FALSE)
```

En dernier lieu, un vecteur au niveau des n-grammes de caractères est généré via txt.to.features, voici un exemple de 4-grammes[18] :

[17] Les codes présentés dans cette partie sont uniquement pour la démonstration et peuvent différer du code réel, veuillez consulter le lien gitlab pour plus de détails.
[18] Le tiret bas "_" indique le début et la fin du mot.

```
_ d o _
d o _
o _ _
_ _ p
_ p a
_ p a t
p a t r
a t r o
t r o n
r o n _
o n _
n _ _
_ _ s
_ s i
_ s i n
s i n n
i n n _
```

Cependant, il y a des éléments qui ne sont pas des 4-grammes et ne devraient pas se trouver dans la liste comme `d o _` et `o _ _`. Dans le script R, stylo considère l'espace entre les mots comme un caractère, e.g., `d o _` <espace>, `o _` <espace> `_`, et <espace> `_ p a`, et afin d'éviter des n-grammes de longueur incorrecte, il faut supprimer tous les n-grammes qui commencent, se terminent ou contiennent une espace inter-mots. La liste des n-grammes de caractère est donc filtrée à partir de l'expression régulière :

```
regexpattern = "^(?!.*[ ](?=[ ]))\\S.*?\\S$"
```

Ensuite, nous pouvons voir sur la figure 9 que l'expression régulière correspond aux n-grammes valides et que les n-grammes de longueur incorrecte ne sont pas sélectionnés.

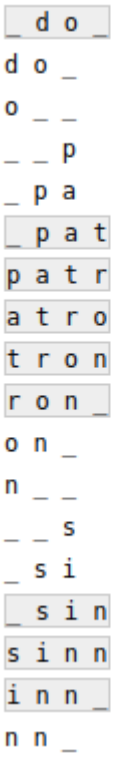


Figure 9. Capture d’écran partielle de console R

Chapitre 9. Analyse contrastive

Dans l'objectif d'assister l'induction de règles de variation en alsacien, nous proposons tout d'abord de concevoir une méthode visant à extraire automatiquement des exemples de variantes potentielles au sein du corpus. Cela peut être considéré comme un problème consistant à extraire des éléments caractéristiques sur la base d'une comparaison de leur degré de représentativité statistique dans différents sous-corpus.

Ainsi, nous avons divisé le corpus en deux parties par département, Haut-Rhin et Bas-Rhin, dans une dimension relativement importante pour la variation scripto-linguistique, puis nous effectuons une analyse contrastive entre ces deux ensembles de textes en nous appuyant sur les méthodes disponibles dans *stylo*. Dans ce chapitre, les corpus divisés seront présentés et la méthode d'extraction des éléments caractéristiques de chaque sous-corpus sur la base d'une comparaison statistique sera détaillée.

1. Sous-corpus

Nous avons regroupé ces pièces alsaciennes en deux sous-corpus, Haut-Rhin et Bas-Rhin, en fonction du lieu de naissance de leurs auteurs (si disponible dans les métadonnées) ou du lieu de publication des pièces.

Département	Haut-Rhin	Bas-Rhin	Total
Nombre de documents	32	45	77
Nombre de tokens	185 497	427 024	612 521

Tableau 3. Chiffres des sous-corpus

De plus, même si on divise le corpus en deux sous-corpus par région pour notre tâche, il faut savoir que la variation graphique peut varier d'un auteur à l'autre. On a donc compté le nombre de tokens par auteur de chaque sous-corpus et créé un graphique circulaire pour représenter la distribution. En observant le graphique, nous notons que le corpus de Haut-Rhin couvre prioritairement la production de Lustig (voir la figure 10 et la figure 11).

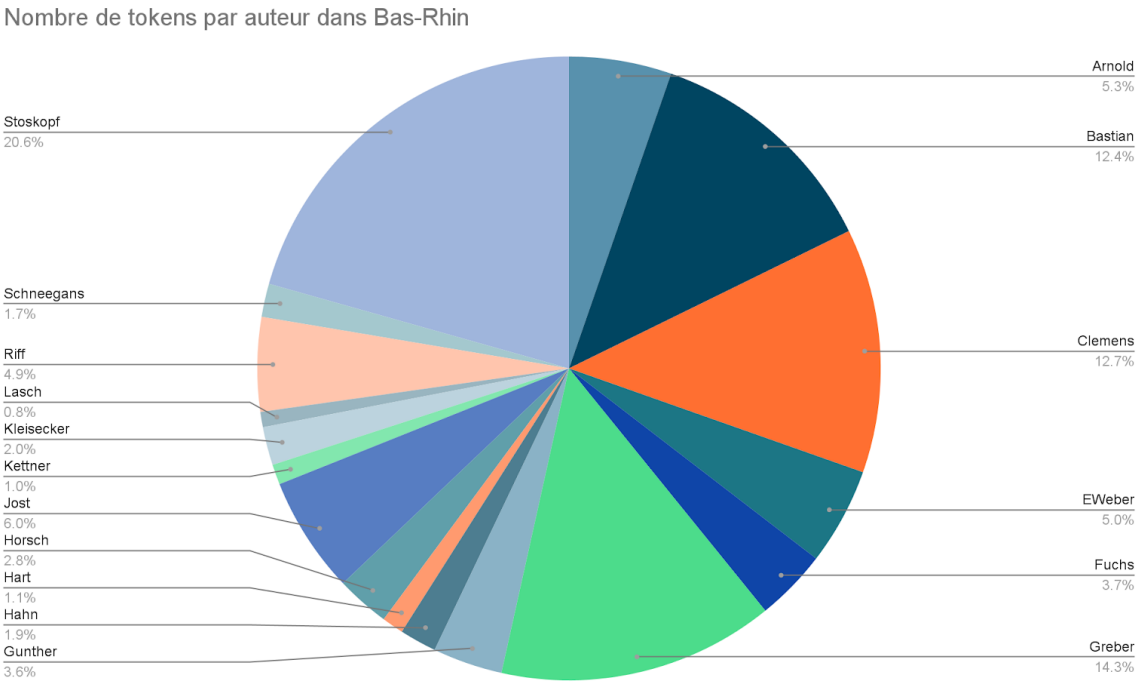


Figure 10. Distribution du nombre de tokens par auteur dans le Bas-Rhin

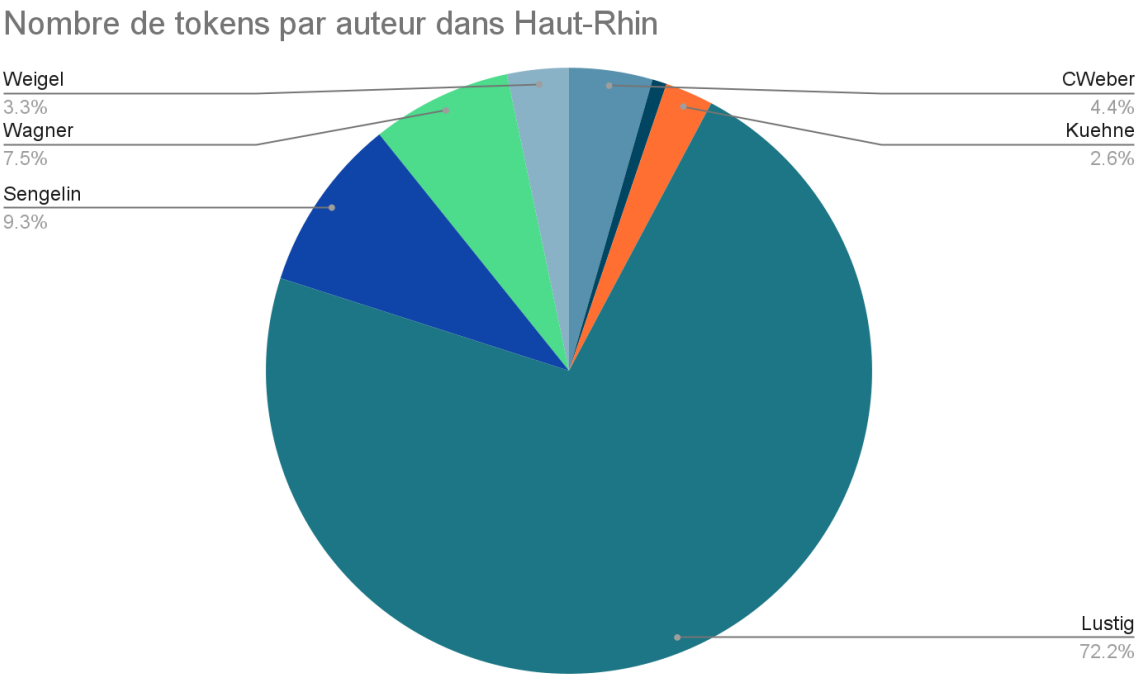


Figure 11. Distribution du nombre de tokens par auteur dans le Haut-Rhin

2. *Extraction d'éléments caractéristiques*

En plus des méthodes de prétraitement mentionnées ci-dessus, le paquet *stylo* offre des méthodes d'analyse contrastive des textes. Avec la méthode oppose() dans *stylo*, nous pouvons contraster deux ensembles de documents et extraire les traits les plus caractéristiques dans les deux ensembles de textes. Afin d'obtenir plus de détails sur la variation et d'explorer les habitudes de scripturalisation du corpus, deux sous-corpus au niveau des n-grammes de caractères ont été choisis comme les données d'entrée :

```
oppose(primary.corpus = basrhin.ngram.corpus,
       secondary.corpus = hautrhin.ngram.corpus,
       rare.occurrences.threshold = 0,
       zeta.filter.threshold = 0,
       oppose.method = "craig.zeta"
                       (option:"eder.zeta", "chisquare.zeta"),
       ...)
```

- primary.corpus : un corpus pré-traité ;
- secondary.corpus : un corpus pré-traité ;
- Ces paramètres (par défaut 0.1) de rare.occurrences.threshold et zeta.filter.threshold permettent de se débarrasser des caractéristiques ayant une faible force de discrimination ; plus le nombre est élevé, moins de mots apparaissent dans les listes de mots finales. On le fixe à 0 afin d'obtenir le plus grand nombre possible d'exemples de variation ;
- oppose.method : Plusieurs métriques sont implémentées pour sélectionner les caractéristiques qui présentent une différence statistiquement significative dans les distributions entre les deux ensembles. Nous utilisons ici la métrique Zeta de Craig qui est une extension de la métrique Zeta proposée à l'origine par Burrows (Burrows, 2007), qui reste un choix populaire dans la communauté de la stylistique pour sélectionner des caractéristiques stylistiques discriminantes dans des contextes de classification binaire (Kinney & Craig, 2010).

Nous disposons ainsi d'une liste de n-grammes de caractères significativement préférés par le premier ensemble de textes, i.e., le corpus de Bas-Rhin, et d'une autre liste contenant les n-grammes de caractères significativement évités par le premier (ou, préféré dans le Haut-Rhin). On a aussi modifié uniquement la valeur du paramètre oppose.method

afin de comparer les résultats obtenus avec différentes métriques Zeta. Les idées principales de ces trois méthodes sont dérivées de Zeta, le but est de diviser les textes d'entrée en segments de tailles égales. Les différences entre les méthodes est :

- craig.zeta : le nombre de segments dans lesquels une forme apparaît dans les sous-corpus A et B est ensuite comparé à l'aide de la formule de Zeta de Craig.
- chisquare.zeta : le nombre de segments dans lesquels une forme apparaît dans les sous-corpus A et B est ensuite comparé à l'aide du test de Chisquare (si la valeur p dépasse 0,05, une différence est considérée comme significative).
- eder.zeta : le nombre de segments dans lesquels une forme apparaît dans les sous-corpus A et B est ensuite comparé en utilisant une distance dérivée de la mesure de similarité de Canberra.

Ces résultats sont résumés dans le tableau dessous[19] :

Métrique	chisquare.zeta	eder.zeta	craig.zeta
Nombre d'éléments dans Bas-Rhin	493	10 856	53 880
Nombre d'éléments dans Haut-Rhin	577	11 308	28 496
Total	1 070	22 164	82 376

Tableau 4. Nombre de résultats obtenus en utilisant différentes zetas

On peut remarquer que la productivité de craig.zeta est plus forte que les autres métriques, et on trouve que les données obtenues avec craig.zeta couvrent largement les résultats obtenus avec les autres métriques. Nous allons donc utiliser les deux sous-corpus obtenus par cette mesure pour extraire des règles de variation, mais nous conservons les résultats obtenus avec les autres métriques.

[19] Prenons l'exemple de 5-grammes de caractères.

Chapitre 10. Alignement

Le travail du nôtre est proche de celui décrit par Millour et Fort (2019), qui ont développé une plateforme de myriadisation, Recettes de Grammaire[20] pour la collecte de variantes graphiques en alsacien. La plateforme permet aux participants d'ajouter les différentes graphies pour un seul mot ou pour une séquence de mots grâce aux deux systèmes (voir la figure 12 et la figure 13).

> « Au total, 215 paires de graphies alternatives ont été myriadisées sur Recettes de Grammaire » (Millour, 2020).

À partir des variantes collectées, elle a utilisé l'outil ALPHAMALIG[21] (Alonso et al., 2004) pour effectuer l'alignement.

Figure 12. Ajout d'une orthographe alternative via le nuage de mots (Millour & Fort, 2019)

[20] Source : http://bisame.paris-sorbonne.fr/recettes.
[21] Source : http://alggen.lsi.upc.es/recerca/align/alphamalig/intro-alphamalig.html.

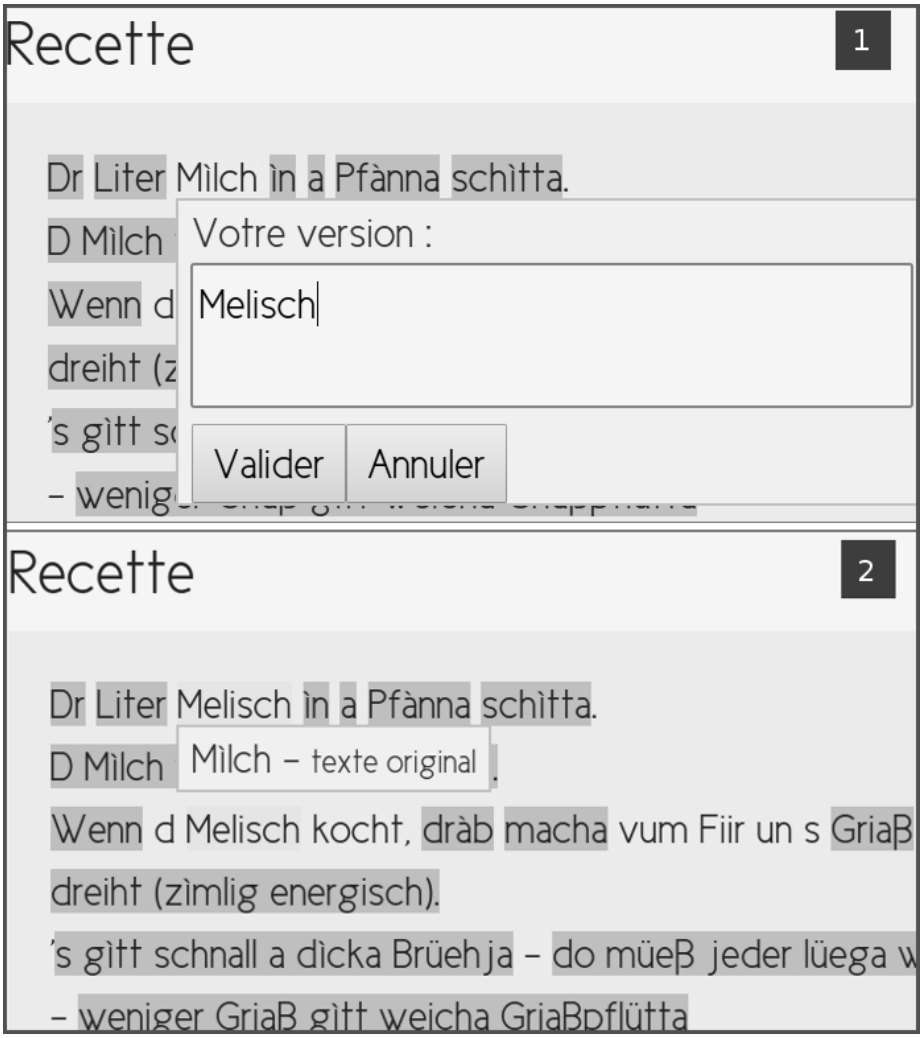


Figure 13. Ajout d'une variante (1) et visualisation (2) (les mots surlignés présentent au moins une variante alternative) (Millour & Fort, 2019)

Dans la collection de ressources de Millour, les variantes alternatives d'un même mot sont déjà définies par la myriadisation. Plus précisément, les orthographes (ou les segments) qui sont l'alternative les uns des autres sont mis en correspondance par les propositions des locuteurs de l'alsacien (voir figure 14), c'est-à-dire, les orthographes alternatives d'un mot donné sont alignées dans ce processus.

Toutefois, les variantes de notre corpus sont dispersées, et doivent être associées et alignées par un pivot. Ensuite, nous pouvons disposer d'un jeu de données tel que celui de la figure 14, sur lequel se base l'alignement des variantes dans nos corpus.

Cas de l'alsacien

mot original	variante 1	variante 2	variante 3	variante 4	variante 5
'r	er				
Dr	D'r	De	Der		
Dreiha	Drahja	Dreihe	draje		
drüs	d'rüs				
e	a				
Galrìewle	Galerewle	Galerieble	Galriawla	Galerìewle	Galriawla
Griaß	Grees	Gress	Greß		
Griaßpflütta	Greespflüdde	Greßpflütte	Griesbap	Griespflüdde	GrussFlutta
güet	gùt	güat	guet		
kàt	kààt	kànt	kànn	kàt's	

Figure 14. Capture d'écran du tableau dans la thèse de Millour - Extrait des graphies alternatives myriadisées sur Recettes de Grammaire (Millour, 2020)

1. *Double metaphone pour les dialectes alsaciens*

La recherche d'un mot dans une base de données est un défi particulier. Selon la source et l'ancienneté des données, il se peut que l'on ne puisse pas s'attendre à ce que l'orthographe soit standardisée et les mots peuvent ne pas être orthographiés de la même manière s'ils apparaissent plusieurs fois. Les divergences entre les données stockées et les termes de recherche peuvent également être dues à des préférences personnelles ou à des différences culturelles dans l'orthographe, à des homophones, à des fautes d'orthographe ainsi qu'à l'analphabétisme ou simplement à l'absence d'orthographe normalisée à certaines périodes. Ces problèmes sont particulièrement fréquents dans les transcriptions de documents historiques manuscrits utilisés par les historiens et autres chercheurs, notamment pour l'alsacien où la convention orthographique consensuelle est absente.

Une solution commune à ces problèmes de recherche de chaînes de caractères consiste à trouver des valeurs similaires à la cible de la recherche. Cependant, l'utilisation de la mesure classique de correspondance approximative de chaînes (ou *fuzzy match* en anglais) afin de calculer la similarité des chaînes de caractères arbitraires peut être coûteuse et ne convient pas aux dialectes alsaciens, comme l'illustre l'exemple de [güet – gùt – güat – guet] (adjectif ou adverbe « bien ») (Millour, 2020) dans l'avant-dernière ligne du tableau de la figure 14.

Une meilleure solution est le calcul préalable des valeurs de hachage pour chaque élément dans le corpus. Il existe des algorithmes de hachage spécialisés conçus à cet effet. Ces algorithmes phonétiques nous permettent de comparer deux chaînes de caractères sur la base de leur prononciation, plutôt que de leur orthographe exacte.

L'un des algorithmes les plus pertinents pour notre problématique est le double metaphone adapté aux dialectes alsaciens (Bernhard, 2014), une variante de l'algorithme du double metaphone initialement proposé par Phillips (2000). Le double metaphone pour l'alsacien retourne une clé identique et une clé alternative pour les chaînes de caractères qui se prononcent de manière proche. Par exemple, pour les trois variantes « *güet* », « *gùt* », « *güat* » et « *guet* », la clé produite est KT[22] :

```
$ python metaphone_als.py

güet    ('KT', None)
gùt     ('KT', None)
güat    ('KT', None)
guet    ('KT', None)
```

Suivant Bernhard (2014), nous avons utilisé l'algorithme du double metaphone pour les dialectes alsaciens afin d'effectuer l'alignement des n-grammes de caractères de chaque sous-corpus. Pour commencer, s'ils partagent une de leurs clés metaphone, les n-grammes de caractères seront alignés. En conséquence, nous obtenons une structure de données organisée comme un arbre où chaque nœud représente une clé metaphone, et ses feuilles représentent toutes les chaînes de caractères qui partagent la même clé, comme l'illustre la figure 15.

[22] Source : https://gitlab.huma-num.fr/methal/corpus-methal-all/-/tree/main/code/script/metaphone

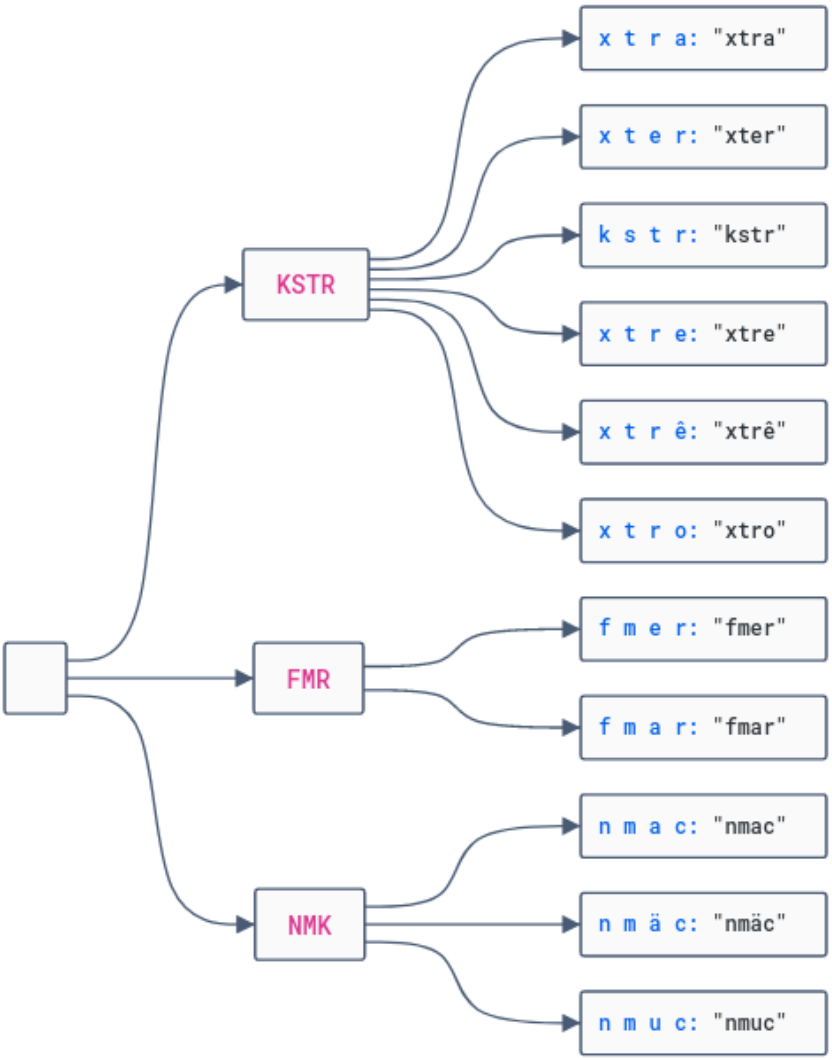


Figure 15. Extrait d'arborescence des 4-grammes de caractères alignés

De plus, en appliquant cette méthode au sous-corpus du Haut-Rhin et au sous-corpus du Bas-Rhin, deux arborescences sont produites, et on ne conserve que les clés metaphone communes aux deux sous-corpus et les valeurs correspondant à leurs clés. Afin de faciliter l'exploitation et la présentation des sorties, nous les reformatons et les enregistrons dans un fichier CSV[23].

[23] Source : https://gitlab.huma-num.fr/methal/corpus-methal-all/-/tree/main/code/working_dir/metaphone

mesure	CléMetaphone Communes	BasRhinPreferredNgrams[24]	BasRhinAvoidedNgrams[25]
Craig	SLM	s ä l m; s l i m	s e l m; s a l m; s à l m
Craig	NRF	n e r v; n e r f	n a r v
Craig	LSL	l ä s l; l e ß l; l ü ß l	l a s l; l s a l; l i s l
Eder	NSR	n s a r; n s r i; n s e r	n s r e
Khi2	HRS	h e r z	h a r z

Tableau 5. Extrait de sorties des clés-valeurs (le fichier *4grams.csv* dans gitlab)[26]

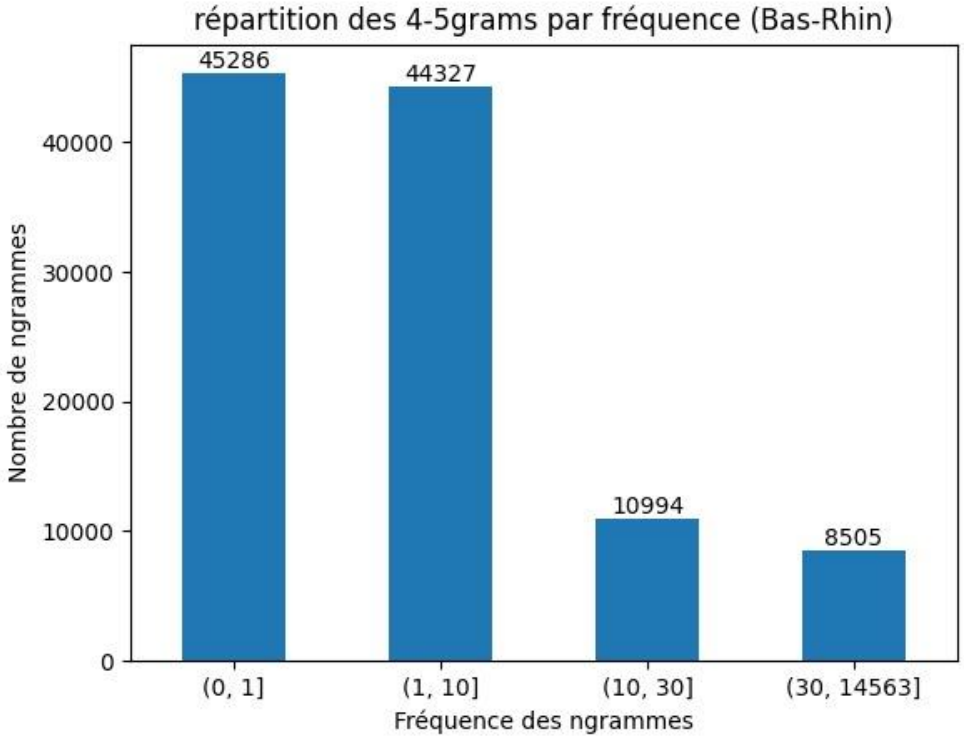
2. *Alignement avec ALPHAMALIG*

Suivant les recommandations de Millour (2020), nous avons utilisé l'outil ALPHAMALIG[27] (ALPHAbet Multiple ALIGnment) (Alonso et al., 2004) pour aligner les n-grammes de caractères.

2.1. Préparation du jeu de données

En premier lieu, nous combinons les 4-grammes de caractères alignés par double metaphone avec les 5-grammes de caractères alignés. En second lieu, nous créons un graphique qui montre la fréquence d'apparition des n-grammes de caractères dans le but de les filtrer en fonction du seuil de fréquence (voir figure 16). En dernier lieu, nous avons dû convertir les données d'entrée dans un format particulier pour utiliser l'outil d'alignement[28].

[24] Les n-grammes de caractères caractéristiques dans le sous-corpus Bas-Rhin.
[25] Les n-grammes de caractères caractéristiques dans le sous-corpus Haut-Rhin.
[26] Nous avons gardé les résultats des mesures telles que Eder et Khi2 (chisquare).
[27] Source : http://alggen.lsi.upc.es/recerca/align/alphamalig/intro-alphamalig.html.
[28] Pour code/ressources, voir le dossier [to_variant_list] du dépôt dans gitlab.



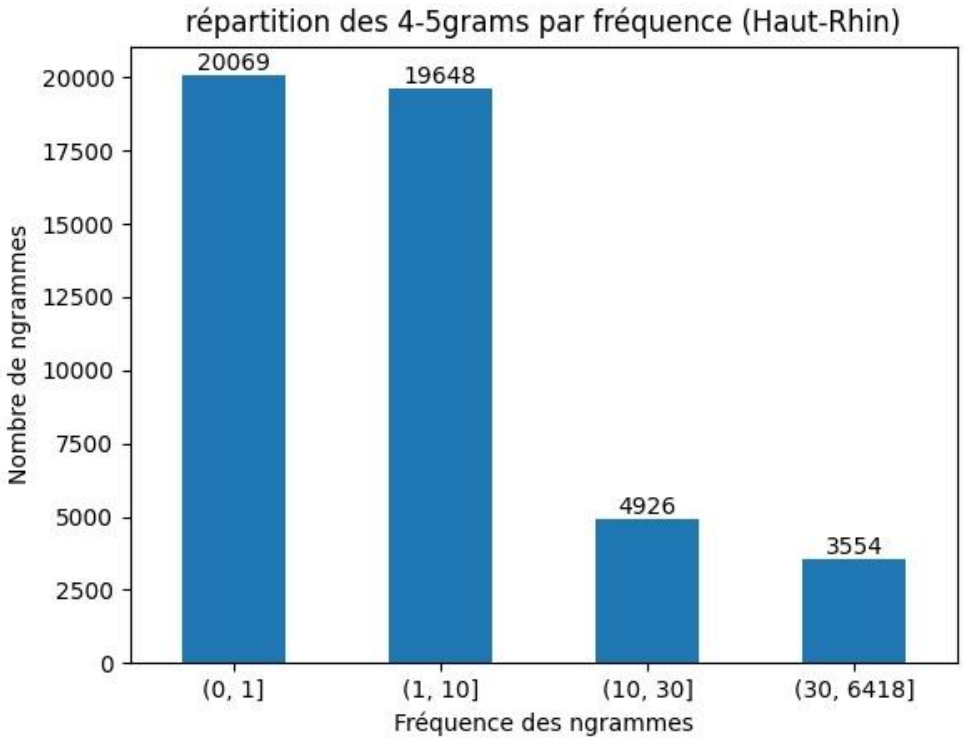


Figure 16. Fréquence d'apparition des n-grammes de caractères

Ainsi, nous générons le jeu de données à l'aide de scripts selon le seuil de fréquence et la mesure :

```
#Exemple :
python to_variant_almf.py [-h] mesure(Craig/Eder/Khi2) seuil
```

Le jeu de données nécessaire pour l'alignement se compose de deux fichiers CSV, dans lesquels les variantes d'une même ligne correspondent à la même clé metaphone. Par exemple, on peut voir dans la figure 17 que les données dans la première ligne des deux dataframes partagent la même clé ([rwär - rawer - rwer - rwur] et [rwar - rwàr]).

```
yh@yh-pc:~/methal/methalV1/working_dir/metaphone_json$ python to_variant_almf.py Craig 10
variants de bas-rhin:
         0      1      2      3      4    ...   160   161   162   163   164
1      rwär  rawer   rwer   rwur   None  ...  None  None  None  None  None
2     rlaub   rleb   rlob   None   None  ...  None  None  None  None  None
3     babbl   None   None   None   None  ...  None  None  None  None  None
4     alasc   elsc  alisc  ellsc  ollsc  ...  None  None  None  None  None
5     mietl   None   None   None   None  ...  None  None  None  None  None
...     ...    ...    ...    ...    ...  ...   ...   ...   ...   ...   ...
1355   trez   türz  derzu   None   None  ...  None  None  None  None  None
1356  chwei   chwy   gewö  chwöu  geweh  ...  None  None  None  None  None
1357  _gebe  _gewe  _gebo  _gewi   _qu'  ...  None  None  None  None  None
1358   jule   jele  jlich   None   None  ...  None  None  None  None  None
1359   _jul  _jule   None   None   None  ...  None  None  None  None  None

[1359 rows x 165 columns]
variants de haut-rhin:
          0      1      2      3     4    ...    50    51    52    53    54
1      rwar   rwàr   None   None  None  ...  None  None  None  None  None
2     rlieb  rlaüb   None   None  None  ...  None  None  None  None  None
3      publ  publi   None   None  None  ...  None  None  None  None  None
4      alsc   None   None   None  None  ...  None  None  None  None  None
5     maidl   None   None   None  None  ...  None  None  None  None  None
...     ...    ...    ...    ...   ...  ...   ...   ...   ...   ...   ...
1355   driz  derzü  drize   derz  None  ...  None  None  None  None  None
1356   g'wi   chwu   chwi   chwö  chwà  ...  None  None  None  None  None
1357  _g'wu   _qua  _g'we  _g'wo  _gwa  ...  None  None  None  None  None
1358  julie  jalou   juli   jalo  None  ...  None  None  None  None  None
1359  _juli  _jalo   _jal   None  None  ...  None  None  None  None  None

[1359 rows x 55 columns]
```

Figure 17. Capture d'écran de bash

2.2. Alignement des n-grammes de caractères

Nous avons modifié le code d'alignement de Millour (2020) pour l'adapter à nos besoins. Étant donné qu'il y a deux fichiers d'entrée, il faut donc aligner chaque variante de Bas-Rhin avec les autres variantes de Haut-Rhin (elles partagent la même clé metaphone). En outre, l'outil ALPHAMALIG ne supporte pas directement les lettres accentuées et certains signes de ponctuation, il faut convertir les séquences d'entrée pour les rendre compatibles avec le format FASTA[29] utilisé par ALPHAMALIG. La correspondance utilisée est la suivante :

```
# Map towards C compatible strings
map = {'à': '0', 'è': '1', 'ì': '2', 'ò': '3', 'ù': '4'
, 'á': '5', 'é': '6', 'í': '7', 'ó': '8', 'ú': '9'
, 'ä': '!', 'ë': 'ø', 'ï': '#', 'ö': '%', 'ü': '='
, 'â': '[', 'ê': '(', 'î': ')', 'ô': '*', 'û': '+'
, 'ß': '{', '\'': '~', '\'': '~', '-':']', 'æ':'}'}
```

Pour ALPHAMALIG, l'entrée est un ensemble de séquences et un critère de similarité, et les scores sont déterminés en fonction du critère de similarité, de sorte que les séquences les plus similaires soient assignées à un score plus élevé. Puisque nous n'avons pas la connaissance a priori de tels critères pour rendre compte de la similarité entre tous les symboles de l'alphabet, entre chaque symbole et lui-même, et entre chaque symbole et un vide, nos critères hypothétiques sont les suivants :

- Entre deux caractères identiques : poids de 4 qui doivent être alignés ;
- Entre deux voyelles, entre deux consonnes, ou entre une consonne et un tiret bas[30] : poids de 3 ;
- Entre une consonne et une voyelle ou entre une voyelle et un tiret bas : poids de 2 ;
- Entre une consonne et un vide ou entre une voyelle et un vide : poids de 1.

Cela est représenté dans le script comme :

[29] Source : https://fr.wikipedia.org/wiki/FASTA_(format_de_fichier)
[30] Un tiret bas représente le début ou la fin d'un mot.

```
# alpha file weight map
char_count=len(charlist)
for n in range(char_count):
    for m in range(n+1):
        charlist[n]
        charlist[m]
        if n==m:
            f.write("4 ")
        else:
            if voyelle(charlist[n]) & voyelle(charlist[m]):
            f.write("3 ")
            elif (not voyelle(charlist[n])) & (not
voyelle(charlist[m])):
                f.write("3 ")
            else:
                f.write("2 ")
    f.write("\n")
# alpha file final line (gap symbol)
for n in range(char_count+1):
    f.write("1 ")
f.write("\n")
```

Le tableau 6 donne un exemple de résultat fourni par ALPHAMALIG pour l'alignement des n-grammes de caractère :

_	H	Ä	R	-	-	(1)
_	H	I	R	-	-	(2)
_	H	Ö	C	-	H	(3)
_	H	E	R	-	R	(4)
_	H	Ö	R	E	-	(5)
_	H	I	R	O	-	(6)
_	H	Ö	R	-	-	(7)
_	H	E	R	-	-	(8)

Tableau 6. Alignement de huit variantes des n-grammes de caractère en alsacien

Chapitre 11. Extraction de règles

À partir des alignements produits, nous avons utilisé la méthode dans la thèse de Millour (2020) pour extraire les règles, qui a été modifiée pour s'adapter aux n-grammes de caractères[31]. Ensuite, toutes les lettres apparaissant dans des motifs de substitution sont mises en correspondance par quatre catégories : un umlaut, un diacritique, une consonne ou une voyelle :

- **V**oyelle : a, e, i, o, u (abréviation en «V») ;
- **U**mlaut : ä, ö, ü (abréviation en «U») ;
- **D**iacritique : é, ê, è, à (abréviation en «D») ;
- **C**onsonne : b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z (abréviation en «C») .

Nous générons une nouvelle règle en conservant les lettres qui changent et en enlevant le reste des caractères, s'il n'y a aucun caractère sur un côté, il est assigné à « *EMPTY* ». Le tiret bas représente le début et la fin d'un mot. Enfin, il y a un total de 1 171 correspondances dans les règles extraites, et les 15 premières règles extraites sont présentées dans la figure 18 suivante[32] :

[31] Source : https://gitlab.huma-num.fr/methal/corpus-methal-all/-/tree/main/code/aligne_variants_alsa/bin
[32] Pour les ressources/codes des règles, voir le dossier [rules_Craig_10] du dépôt dans gitlab.

Unnamed: 0	index	var1	var2	size	rule
0	189	dr	der	30	EMPTY >> V
1	234	er_	e_	23	C >> EMPTY
2	1027	t_	te_	13	EMPTY >> V
3	402	ic	isc	13	EMPTY >> C
4	207	e_	er_	12	EMPTY >> C
5	279	g_	ge_	10	EMPTY >> V
6	679	ner	nr	10	V >> EMPTY
7	1073	vr	ver	10	EMPTY >> V
8	437	l_	le_	9	EMPTY >> V
9	1071	ver	vor	9	V >> V
10	306	gs	g's	9	EMPTY >> '
11	1068	v'r	ver	8	' >> V
12	152	bàw	bew	8	D >> V
13	978	so_	se_	8	V >> V
14	976	so_	s_	8	V >> EMPTY

Figure 18. Capture d’écran partielle de règles extraites

Partie 5

-

Résultats et discussion

Chapitre 12. Exploitation des règles extraites

Nous extrayons les paires de variantes en alignant les n-grammes des caractères, mais ces variantes telles que « *här* » et « *har* » sont probablement des parties des mots, et peuvent être des n-grammes de caractères dans des mots. Cependant, il nécessite des exemples de variation qui sont caractéristiques des différentes divisions du corpus en fonction des métadonnées. Dans ce chapitre, nous allons présenter la méthode d'extraction des exemples concrets de variations et le résultat final que nous obtenons.

1. Extraction des exemples concrets

Afin d'extraire les exemples de variation à partir des paires de n-grammes de caractère, nous avons recherché les mots qui contiennent les n-grammes de caractère dans les sous-corpus au niveau des tokens. Chaque motif de substitution est considéré comme une clé de tableau de hachage, et une fois qu'une paire de mots correspondant à la règle est trouvée, la paire est ajoutée à cette clé comme une valeur.

Plus précisément, pour les deux sous-séquences « *här* » et « *har* », nous extrayons les mots contenant la sous-séquence « *här* » dans le corpus du Bas-Rhin et les mots contenant la sous-séquence « *har* » dans le corpus du Haut-Rhin. Nous « soustrayons » la sous-séquence « *här* » ou la sous-séquence « *har* » du mot, et si les deux séquences restantes sont identiques, ces deux mots sont identifiés comme les paires de variantes que nous recherchons. En ce qui concerne les deux chaînes de caractères « *här* » et « *har* », les résultats que nous avons extraits sont les suivants :

```
"här(0.73) >> har(0.78)": {
    "_härе_": "_harе_",
    "_härze_": "_harze_",
    "_härmoniere_": "_harmoniere_",
    "_härzig_": "_harzig_",
    "_härz_": "_harz_",
    "_härre_": "_harre_",
    "_härr_": "_harr_",
    "_härzhaft_": "_harzhaft_"
    }
```

En outre, pour chaque forme, nous nous intéressons à quelle proportion d'auteurs différents elle apparaît. Comme nous avons regroupé les pièces de chaque dramaturge dans un document séparé, on peut obtenir le pourcentage de dramaturges dans lesquels

apparaissent chaque forme. Par exemple, dans le résultat ci-dessus, här(0.73) signifie que cette chaîne de caractères apparaît chez 73% des dramaturges du corpus de Bas-Rhin.

2. Résultat final[33]

Le résultat final est enregistré dans un fichier JSON. D'après la hiérarchie dans la figure 19, on peut voir que la structure JSON du résultat comporte trois niveaux, dans lesquels se trouvent les règles concernant des catégories de lettres, des règles initiales que nous avons extraites ainsi que des exemples de variation.

[33] Pour les résultats, voir le dossier [rules_Craig_10] du dépôt dans gitlab.

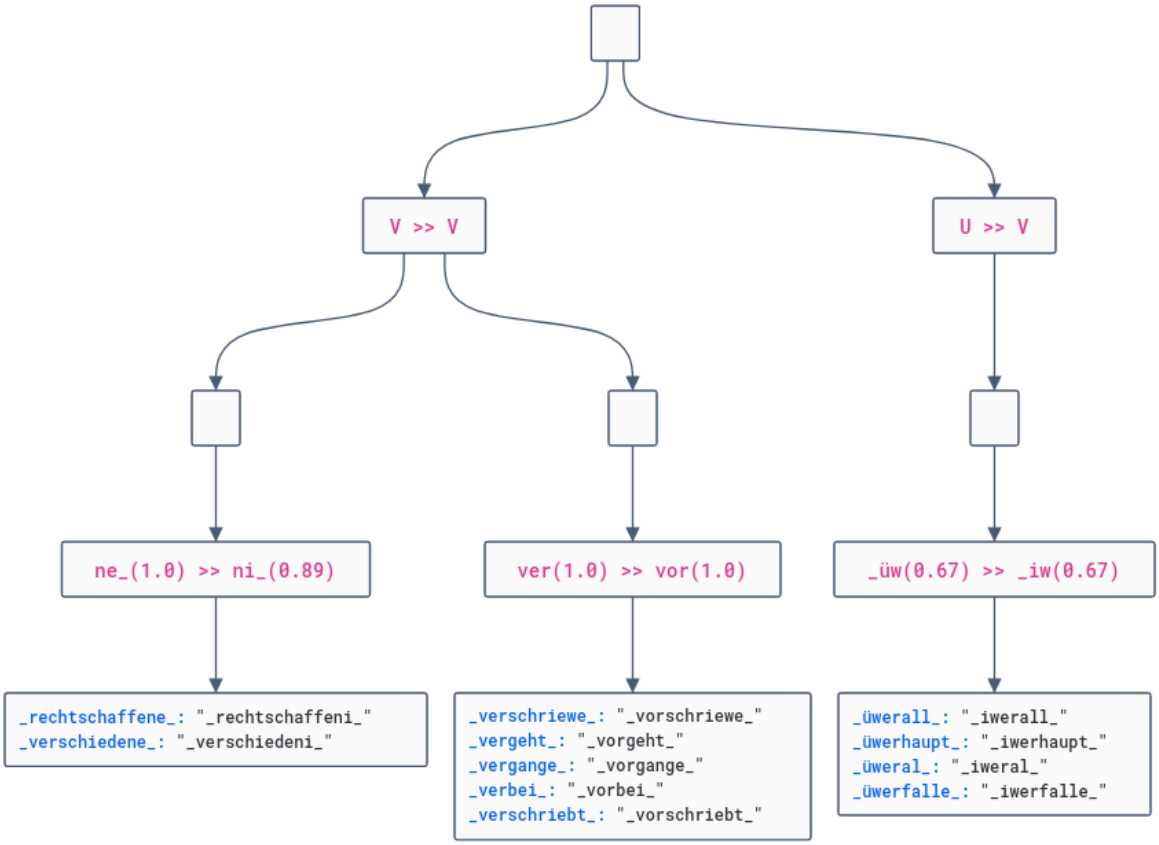


Figure 19. Extrait de la hiérarchie de schéma JSON

Finalement, à partir des 1 171 règles de substitution extraites, nous avons résumé 132 règles concernant les catégories des lettres (voir **l'annexe 2**). La partie gauche de toutes les règles représente les formes présentes en Bas-Rhin et la partie droite représente celles présentes en Haut-Rhin.

Chapitre 13. Discussion

Le tableau 7 présente les 20 premières règles. La colonne des règles représente les règles concernant les catégories des lettres, et la colonne *taille* représente le nombre d'occurrences du motif de substitution de cette catégorie de lettre parmi les 1 171 règles de substitution de formes. Par exemple, le motif de substitution, ver >> vor, dans la règle concernant des formes correspond à une règle de V >> V.

règles	taille	règles	taille
V >> V	231	CV >> CV	16
EMPTY >> V	190	EMPTY >> U	15
U >> V	108	V >> VV	14
EMPTY >> C	64	U >> VV	14
C >> C	45	EMPTY >> VV	13
V >> EMPTY	44	U >> D	12
D >> V	35	UV >> V	11
C >> CV	29	U >> EMPTY	11
C >> EMPTY	19	UC >> VC	10
EMPTY >> VC	17	VV >> V	10

Tableau 7. Extrait de règles concernant les catégories des lettres (**annexe 2**)

En analysant les résultats finaux, nous voyons que les variantes orthographiques de ce corpus se trouvent souvent sur les voyelles. En particulier dans le cas du Bas-Rhin vers le Haut-Rhin, on observe souvent une insertion de voyelle et le changement d'umlaut en voyelle sans diacritique (pour les règles *EMPTY >> V* et *U >> V*).

La variation consonantique apparaît aussi dans ce corpus (par exemple : **br/pr**, **ld/lt** et **di/ti**), et les changements vocaliques et les changements consonantiques paraissent concomitants dans certains cas. À part ce qui varie, il y a peu de variation entre la voyelle sans diacritique et la voyelle avec un diacritique (pour les règles *D >> V* et *V >> D*). Bien

que ces deux règles apparaissent également dans le tableau, nous n'avons pas recherché d'exemples de variantes correspondant à ces deux règles.

En outre, nous trouvons que cette étude présente des limitations et des points à améliorer. Tout d'abord, le sous-corpus de Haut-Rhin couvre prioritairement la production d'un dramaturge (Lustig), dont les contenus représentent 72% du total. Les habitudes de scripturalisation du Haut-Rhin reflètent donc largement les habitudes orthographiques de Lustig. Une amélioration à cela est que nous pouvons extraire les règles de variation interne de l'œuvre de l'auteur et ensuite prendre en compte le contexte de variation de l'auteur dans l'analyse des résultats. Évidemment, il ne faut pas se limiter au Haut-Rhin et au Bas-Rhin, mais aussi les analyser d'un point de vue diachronique, par exemple en comparant comment les variantes diffèrent aux trois périodes :

- Avant l'époque allemande : < 1871
- Période allemande : 1871-1918
- Période « française » : >= 1919

Pour les auteurs Clemens et Kettner, nous leur avons initialement assigné une zone dialectale basée sur la localisation de l'éditeur, c'est-à-dire le Haut-Rhin, mais le lieu de naissance des deux auteurs est le Bas-Rhin. Comme nous l'avons souligné dans la section 7.3, nous devrions leur assigner des zones dialectales en fonction de leur lieu de naissance. Nous avons corrigé leurs zones dialectales pour obtenir les n-grammes de caractères basées sur la mesure de craig.zeta, puis à partir de ces caractéristiques nous avons obtenu les résultats finaux. Toutefois, il est à noter que nous n'avons pas encore corrigé les zones dialectales de ces deux auteurs quant à l'obtention des formes caractéristiques en fonction d'autres mesures (`chisquare.zeta` et `eder.zeta`).

Ensuite, en ce qui concerne l'utilisation de la mesure zeta dans *stylo* pour extraire des formes caractéristiques de sous-corpus, nous avons choisi les résultats obtenus avec la mesure de Craig Zeta (Kinney & Craig, 2010), car elle a extrait le plus de caractéristiques. Cependant, nous avons également réservé les résultats de `eder.zeta` et de `chisquare.zeta` afin de paramétrer ces mesures dans le script. Dans les tâches futures, nous pourrons évaluer les performances de différentes mesures. De plus, selon Schöch et al. (2018), une autre variante de Zeta, la Zeta logarithmique, surpasse les autres Zeta, mais cette mesure n'est pas fournie dans *stylo*, il serait donc préférable que la Zeta logarithmique puisse être utilisée pour les études futures.

Étant donné que le corpus de cette étude est non-contemporain, nous nous intéressons aussi à la comparaison avec les règles de Millour (2020). Nous pouvons ainsi trouver les différences dans les règles de variantes obtenues entre différentes périodes de chaque corpus.

Enfin, quant à l'extraction d'exemples de variantes des règles pertinentes, notre programme demande trop de temps d'exécution et une optimisation algorithmique devrait être possible. Par ailleurs, le processus d'extraction des règles peut être intégré dans une seule application, ce qui rendra le processus plus clair et plus facile à tester.

Conclusion

Ce travail est consacré à la tâche d'extraction automatique de règles de variation orthographique au sein du corpus en alsacien. Dans ce mémoire, nous avons d'abord décrit le contexte de stage et notre problématique. Ensuite, nous avons introduit dans la partie 2 un état de l'art des méthodes pour trois tâches en rapport avec notre sujet. La première est une exploration des différentes méthodes d'identification des variantes orthographiques, qui nous permet de comprendre en détail les variantes orthographiques et nous donne une idée générale du cadre de leur étude. Deuxièmement, nous nous concentrons sur les mesures statistiques qui sont introduites et adoptées pour étudier et analyser de grandes quantités de données textuelles dans une perspective contrastive. Finalement, nous avons examiné les méthodes d'alignement les plus appropriées pour notre étude.

Dans les chapitres suivants, nous avons analysé la structure TEI du corpus source, puis extrait le contenu des pièces du corpus pour construire un jeu de données à analyser. Nous commençons par extraire des formes caractéristiques au niveau des n-grammes de caractères en utilisant la mesure de zeta. Ensuite, nous recourons au double metaphone pour alsacien dans le travail de Bernhard (2014) afin de regrouper les n-grammes de caractères. Enfin, en nous inspirant du travail de Millour (2020), nous utilisons ALPHAMALIG pour effectuer l'alignement, et nous exploitons les résultats de l'alignement pour extraire les règles de variantes.

En conclusion, cette analyse initiale montre le potentiel des résultats pour arriver à des généralisations concernant la variation graphique ; il nous reste beaucoup à découvrir dans les résultats ainsi que dans l'étude de la variation graphique. Une comparaison avec des règles de variation obtenues dans d'autres études qui reposent sur des corpus plus récents serait également informative concernant l'évolution historique des pratiques de scripturalisation en alsacien.

Bibliographie

Alonso, L., Castellon, I., Escribano, J., Messeguer, X., & Padro, L. (2004). Multiple Sequence Alignment for characterizing the linear structure of revision. *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 403-406.

Barteld, F., Biemann, C., & Zinsmeister, H. (2019). Token-based spelling variant detection in Middle Low German texts. *Lang Resources & Evaluation*, (53), 677–706. https://doi.org/10.1007/s10579-018-09441-5

Bernhard, D. (2014). Adding Dialectal Lexicalisations to Linked Open Data Resources: the Example of Alsatian. *Proceedings of the Workshop on Collaboration and Computing for Under Resourced Languages in the Linked Open Data Era (CCURL 2014)*, 23-29. https://hal.archives-ouvertes.fr/hal-00966820

Bernhard, D. (2018). *Tokeniser for the Alsatian Dialects (1.4.1)*. Zenodo. https://doi.org/10.5281/zenodo.2454993

Bollmann, M. (2019). A Large-Scale Comparison of Historical Text Normalization Systems. *Proceedings of the 2019 Conference of the North*. 10.18653/v1/n19-1389

Burrows, J. (2007). All the Way Through: Testing for Authorship in Different Frequency Strata. *Literary and Linguistic Computing*, *22.1*, 27–47. https://doi.org/10.1093/llc/fqi067

Crevenat-Werner, D., & Zeidler, E. (2008). *Orthographe alsacienne: bien écrire l'alsacien de Wissembourg à Ferrette*. J. Do Bentzinger.

Dasigi, P., & Diab, M. (2011). CODACT: Towards Identifying Orthographic Variants in Dialectal Arabic. *Proceedings of 5th International Joint Conference on Natural Language Processing*, 318–326. https://aclanthology.org/I11-1036

Doval, Y., Vilares, J., & Gómez-Rodríguez, C. (2020). Towards robust word embeddings for noisy texts. *arXiv :1911.10876 [cs]*. 10.48550/ARXIV.1911.10876

Eder, M., Rybicki, J., & Kestemont, M. (2016). Stylometry with R: a package for computational text analysis. *R Journal*, *8*(1), 107-21. https://journal.r-project.org/archive/2016/RJ-2016-007/index.html

Halliday, M. A. K. (1976). Anti-Languages. *American Anthropologist*, *78(3)*, 570–584. 10.1525 / aa.1976.78.3.02a00050

Hudlett, A. (2009). *Carte géolinguistique de l'Alsace*. Atlas historique d'Alsace, Université de Haute Alsace. Retrieved August 9, 2022, from www.atlas.historique.alsace.uha.fr

Hudlett, A., & Groupe d'Etudes et de Recherches Interdisciplinaires sur le Plurilinguisme en Alsace et en Europe. (2003). *Charte de la graphie harmonisée des parlers alsaciens : système graphique GRAPHAL - GERIPA*. Centre de Recherche sur l'Europe littéraire (C.R.E.L.), Mulhouse, France.

Kinney, A. F., & Craig, H. (Eds.). (2009). *Shakespeare, Computers, and the Mystery of Authorship*. Cambridge University Press.

Li, S. (2014). *Research on Detecting & Filtering newly Coined Profanities* [Master dissertation, Fudan University]. China Academic Journals Electronic Publishing House.

Millour, A. (2019). Getting to Know the Speakers: a Survey of a Non-Standardized Language Digital Use. *Proceedings of 9th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. https://hal.archives-ouvertes.fr/hal-02137280

Millour, A. (2020). *Myriadisation de ressources linguistiques pour le traitement automatique de langues non standardisées* [Doctoral dissertation, Sorbonne Universite]. HAL Archives Ouvertes. https://hal.archives-ouvertes.fr/tel-03083213

Millour, A., & Fort, K. (2019). Unsupervised Data Augmentation for Less-Resourced Languages with no Standardized Spelling. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 776-784. 10.26615/978-954-452-056-4_090

OLCA / EDinstitut. (2012). *Etude sur le dialecte alsacien*. Retrieved August 9, 2022, from https://www.olcalsace.org/sites/default/files/documents/etude_linguistique_olca_ed institut.pdf

Philips, L. (2000). The double metaphone search algorithm. *C/C++ Users Journal*, *18*(6), 38–43. https://dl.acm.org/doi/10.5555/349124.349132

Prokić, J., Wieling, M., & Nerbonne, J. (2009). Multiple sequence alignments in linguistics. *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cul- tural Heritage, Social Sciences, Humanities, and Education*, 18-25. https://dl.acm.org/doi/10.5555/1642049.1642052

Rafae, A., Qayyum, A., Moeenuddin, M., Karim, A., Sajjad, H., & Kamiran, F. (2015). An unsupervised method for discovering lexical variations in Roman Urdu informal text. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 823–828.

Ruiz Fabo, P., Bernhard, D., & Werner, C. (2020). Création d'un corpus FAIR de théâtre en alsacien et normalisation de variétés non-contemporaines. *2èmes journées scientifiques du Groupement de Recherche Linguistique Informatique Formelle et de Terrain (LIFT)*, 32-43. 10.5281/zenodo.4323302

Ruiz Fabo, P., Werner, C., Bernhard, D., Erhart, P., & Huck, D. (2021). MeThAL : Ressources numériques pour une relecture du théâtre en alsacien. *10 ans avec CAHIER: Des corpus d'auteurs pour les humanités numériques à leur exploitation numérique (Cahier10)*. 10.5281/zenodo.4908213

Schöch, C., Schlör, D., Zehe, A., Gebhard, H., Becker, M., & Hotho, A. (2018). Burrows' Zeta: Exploring and Evaluating Variants and Parameters. *Book of Abstracts of the Digital Humanities Conference (presented at the Digital Humanities Conference (DH2018)*. https://dh2018.adho.org/burrows-zeta-exploring-and-evaluating-variants-and-parameters/

Sood, S. O., Antin, J., & Churchill, E. F. (2012). Using crowdsourcing to improve profanity detection. *AAAI Spring Symposium - Technical Report*, *SS-12-06*, 69-74.

Steiblé, L., & Bernhard, D. (2016). Vers un lexique ouvert des formes fléchies de l'alsacien : génération de flexions pour les verbes (Towards an Open Lexicon of Inflected Word Forms for Alsatian: Generation of Verbal Inflection). *Actes de la conférence conjointe JEP-TALN-RECITAL 2016. volume 2 : TALN (Posters)*, 547–554. https://aclanthology.org/2016.jeptalnrecital-poster.30

Steiblé, L., & Bernhard, D. (2017). Putting Alsatian into words: a contrastive study of two spelling systems for the Alsatian dialects and Standard German. *8th International Contrastive Linguistics Conference*. https://hal.archives-ouvertes.fr/hal-01504537

Steiblé, L., & Bernhard, D. (2018). Pronunciation Dictionaries for the Alsatian Dialects to Analyze Spelling and Phonetic Variation. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. https://aclanthology.org/L18-1654/

Yoon, T., Park, S.-Y., & Cho, H.-G. (2010). A Smart Filtering System for Newly Coined Profanities by Using Approximate String Alignment. *2010 10th IEEE International Conference on Computer and Information Technology*, 643-650. 10.1109/CIT.2010.129

Sitographie

ALPHAMALIG http://alggen.lsi.upc.es/recerca/align/alphamalig/intro-alphamalig.html [Dernière consultation le 23/08/2022]

Beautiful Soup https://www.crummy.com/software/BeautifulSoup/ [Dernière consultation le 09/08/2022]

Lxml. https://lxml.de/ [Dernière consultation le 09/08/2022]

LiLPa. https://lilpa.unistra.fr/ [Dernière consultation le 05/08/2022]

MeThAL. https://methal.pages.unistra.fr/ [Dernière consultation le 29/08/2022]

Multiprocessing. https://docs.python.org/3/library/multiprocessing.html [Dernière consultation le 29/08/2022]

Numistral. https://numistral.fr/fr/theatre-alsacien (lien [Découvrir] pour explorer la collection) [Dernière consultation le 29/08/2022]

stylo-R Package. https://cran.r-project.org/web/packages/stylo/index.html [Dernière consultation le 13/08/2022]

Wikisource. https://als.wikipedia.org/wiki/Text:August_Lustig/A._Lustig_S%C3%A4mtliche_Werke:_Band_2 [Dernière consultation le 15/08/2022]

Glossaire

Alignement :	Une manière de représenter deux ou plusieurs séquences les unes sous les autres (https://fr.wikipedia.org/wiki/Alignement_de_s%C3%A9quences)
Corpus :	Ensemble de documents, artistiques ou non (textes, images, vidéos, etc.), regroupés dans une optique précise (https://fr.wikipedia.org/wiki/Corpus)
Partitionnement de données :	Une méthode en analyse des données (https://fr.wikipedia.org/wiki/Partitionnement_de_donn%C3%A9es)
Caractéristique :	Il peut s'agir de chaînes de caractères, de graphes ou d'autres quantités encore (https://fr.wikipedia.org/wiki/Extraction_de_caract%C3%A9ristique)
Lexème :	Une unité abstraite du vocabulaire, réalisée par des mot-formes représentant le lexème et sa morphologie flexionnelle (Bauer, 2003)
N-grammes de caractères :	Une sous-séquence de n éléments construite à partir d'une séquence donnée (https://fr.wikipedia.org/wiki/N-gramme)
Recherche d'information :	Le domaine qui étudie la manière de retrouver des informations dans un corpus (https://fr.wikipedia.org/wiki/Recherche_d'information)
Stylométrie :	un domaine de la linguistique qui utilise la statistique pour décrire les propriétés stylistiques d'un texte (https://fr.wikipedia.org/wiki/Stylom%C3%A9trie)
Token :	Désigner une entité (ou unité) lexicale, dans le cadre de l'analyse lexicale (https://fr.wikipedia.org/wiki/Token)

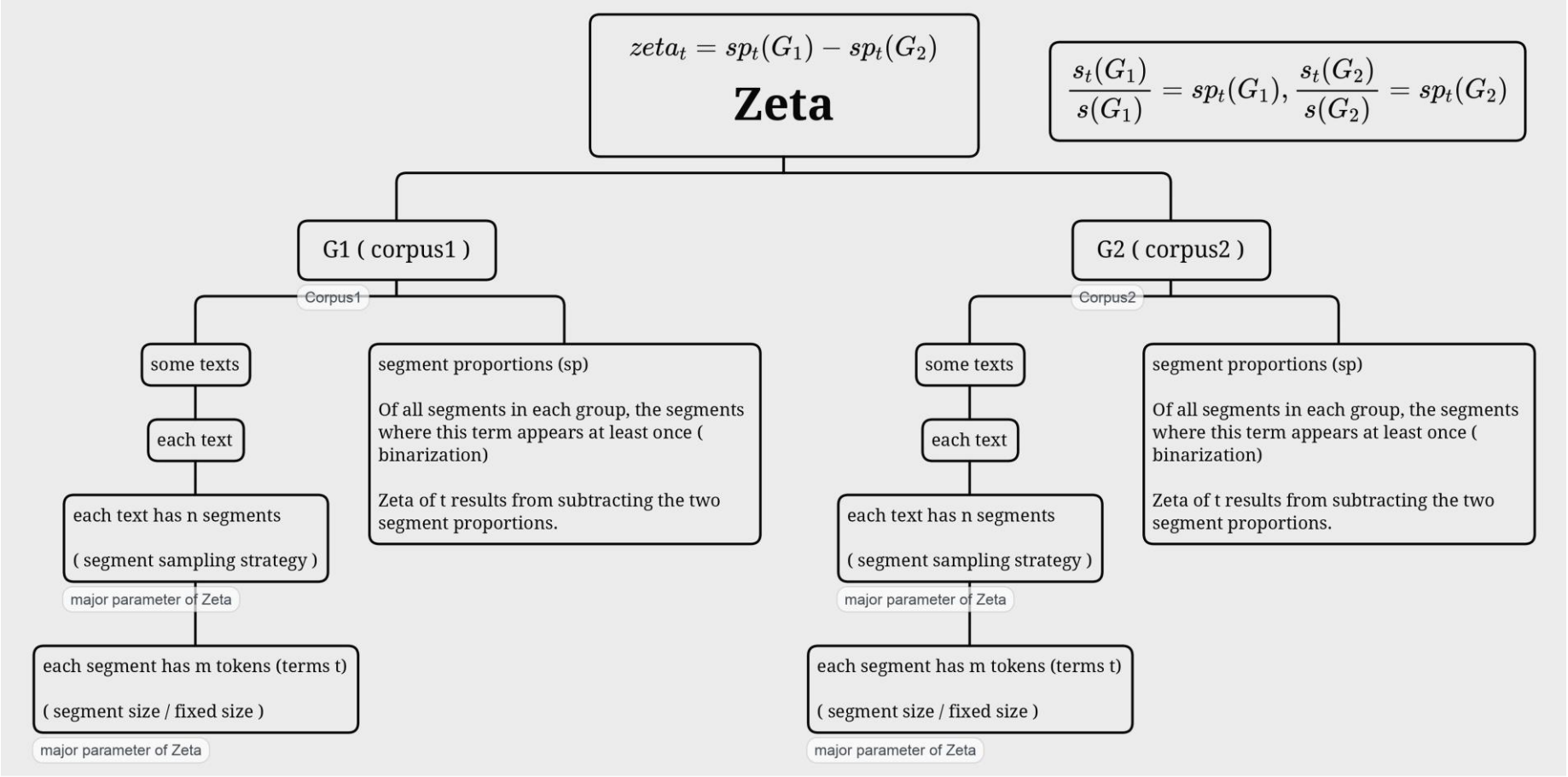
Sigles et abréviations utilisés

API :	Application Programming Interface
CSV :	Comma-separated values
CNN :	Convolutional Neural Networks
DOM :	Document Object Model
IR :	Information Retrieval
JSON :	JavaScript Object Notation
LiLPa :	Laboratoire Linguistique, langues et parole
MHV :	Mots Hors Vocabulaire
MeThAL :	Vers une Macroanalyse du Théâtre en ALsacien
MSA :	Multiple Sequence Alignment
ORTHAL :	Orthographe Alsacienne
POO :	Programmation Orientée Objet
SVM :	Support Vector Machine
TAL :	Traitement Automatique des Langues
TEI :	Text Encoding Initiative
XML :	EXtensible Markup Language

Table des illustrations

Table des annexes

Annexe 1
Informations de Zeta



Annexe 2
Règles concernant les catégories des lettres

rule	size	rule	size
V >> V	231	VCC >> VCC	2
EMPTY >> V	190	D >> C	2
U >> V	108	UCV >> VCC	2
EMPTY >> C	64	VCV >> VCV	2
C >> C	45	UV >> EMPTY	2
V >> EMPTY	44	EMPTY >> VU	2
D >> V	35	' >> V	2
C >> CV	29	CC >> CV	2
C >> EMPTY	19	DC >> C	2
EMPTY >> VC	17	DC >> VC	2
CV >> CV	16	VV >> VV	2
EMPTY >> U	15	CC >> C	2
V >> VV	14	DC >> VD	1
U >> VV	14	CVC >> VCV	1
EMPTY >> VV	13	VCD >> CC	1
U >> D	12	UC >> EMPTY	1
UV >> V	11	VC >> '	1
U >> EMPTY	11	VV >> VC	1
UC >> VC	10	UV >> VC	1
VV >> V	10	CV >> CDC	1
V >> VC	10	UV >> VU	1
CV >> C	9	C >> C'	1
VC >> V	8	VC >> VV	1
VCC >> VCV	7	D >> UVC	1
EMPTY >> EMPTY	7	U >> VC'	1
EMPTY >> '	6	CC >> CCC	1
U >> U	6	U >> UV	1
C >> VC	6	CV >> CC	1
V >> C	6	C >> DC	1
VC >> C	6	VCV >> VC	1
DC >> V	5	C >> VCV	1
EMPTY >> CV	5	D >> U	1
VCC >> VC	4	UCC >> VC	1
V >> U	4	UC >> UVC	1
C >> CC	4	UC >> UC	1
VC >> VC	4	UC >> VCV	1
U >> VC	4	CD >> CV	1
C >> V	4	D >> VU	1
VV >> EMPTY	4	UC >> DVC	1
UC >> C	4	CV >> VCV	1
VC >> CV	4	D >> VD	1
VC >> EMPTY	4	' >> EMPTY	1
V >> UV	3	CUV >> CV	1
CU >> CV	3	UCC >> DC	1
CU >> C	3	CU >> UCC	1
EMPTY >> CC	3	V  >> VCC	1
VC >> VCC	3	U >> CD	1
C >> CU	3	V >> CV	1
EMPTY >> -	3	VU >> EMPTY	1
UVC >> VC	3	D >> CV	1
U >> VU	3	UCV >> VCV	1
VU >> V	3	UCC >> C	1

Table des matières

MOTS-CLÉS : variation graphique, théâtre alsacien, textes historiques, langue non standardisée, extraction de règle

RÉSUMÉ

La variation orthographique est l'un des principaux enjeux pour le TAL des textes historiques, notamment pour les langues non standardisées où l'absence d'une convention orthographique consensuelle ne nous permet pas de se référer à une norme. Dans ce mémoire, nous avons exploré les recherches portant sur l'identification de variantes orthographiques afin de concevoir un système destiné à l'extraction de règles de variation graphique dans les dialectes alsaciens. En se basant sur le corpus de pièces de théâtre provenant du projet MeThAL, nous avons présenté la méthode permettant d'extraire des règles de variantes au niveau des n-grammes de caractères : d'abord l'extraction des n-grammes de caractères du sous-corpus par des mesures statistiques, puis le regroupement des formes caractéristiques, enfin l'alignement des différents groupes de caractéristiques et l'extraction des règles et des exemples des variantes.

KEYWORDS : spelling variation, Alsatian drama, historical texts, non-standard language, rule extraction

ABSTRACT

Spelling variation is one of the key challenges for NLP on historical texts, especially for non-standard languages where there is no consensus on the convention of spelling, it is not possible to normalize these texts. In this paper, we explored the research on the detection of spelling variation in order to develop a system for the extraction of rules of spelling variation in Alsatian dialects. Based on the corpus of dramas from the MeThAL project, we presented the method for extracting rules of variation at the level of character n-grams : first extracting the character n-grams from the sub-corpus by statistical measures, then clustering the feature forms, finally aligning the different clusters and extracting the rules of variation and examples of variants.