Apprentissage de représentations d'auteurs et d'autrices à partir de modèles de langue pour l'analyse des dynamiques d'écriture.

Enzo Terreau

N° d'ordre NNT : 2024LYO20001

THÈSE de DOCTORAT DE L'UNIVERSITÉ LUMIÈRE LYON 2

École Doctorale : ED 512

Informatique et Mathématiques

Discipline : Informatique

Soutenue publiquement le 16 janvier 2024 par :

Enzo TERREAU

Apprentissage de représentations d'auteurs et d'autrices à partir de modèles de langue pour l'analyse des dynamiques d'écriture.

Devant le jury composé de :

Christophe GRAVIER, Professeur des Universités, Université Jean Monnet Saint-Étienne, Président

Lynda TAMINE-LECHANI, Professeure des Universités, Université Paul Sabatier-Toulouse 3, Rapporteuse

Mathieu ROCHE, Directeur de recherche, CIRAD, Rapporteur

Armelle BRUN, Professeure des Universités, Université de Lorraine, Examinatrice

Damien NOUVEL, Maître de conférences, INALCO Institut National des Langues et Civilisations Orientales, Examinateur

Julien VELCIN, Professeur des Universités, Université Lumière Lyon 2, Directeur de thèse

Contrat de diffusion

UNIVERSITÉ DE LYON

UNIVERSITÉ LUMIÈRE LYON 2

Apprentissage de représentations d'auteurs et d'autrices à partir de modèles de langue pour l'analyse des dynamiques d'écriture

Auteur :

Enzo TERREAU

Laboratoire ERIC, Université Lumière Lyon 2

Jury :

Mathieu ROCHE Directeur de Recherche, CIRAD, Rapporteur
Lynda TAMINE-LECHANI Professeure des Universités, Université Paul Sabatier, Rapporteure
Armelle BRUN Professeure des Universités, Université de Lorraine, Examinatrice
Christophe GRAVIER Professeur des Universités, Université Jean Monnet, Examinateur
Damien NOUVEL Maître de Conférences, INALCO, Examinateur
Julien VELCIN Professeur des Universités, Université Lumière Lyon 2, Directeur

Discipline : Informatique

Soutenue publiquement le : 16/01/2024

Apprentissage de représentations d'auteurs et d'autrices à partir de modèles de langue pour l'analyse des dynamiques d'écriture

Auteur :
Enzo TERREAU (Laboratoire ERIC, Université Lumière Lyon 2)

Directeur :
Julien VELCIN (Professeur des Universités, Université Lumière Lyon 2, Directeur)

Summary

The recent and massive democratization of digital tools has empowered individuals to generate and share information on the web through various means such as blogs, social networks, sharing platforms, and more. The exponential growth of available information, mostly textual data, requires the development of Natural Language Processing (NLP) models to mathematically represent it and subsequently classify, sort, or recommend it. This is the essence of representation learning. It aims to construct a low-dimensional space where the distances between projected objects (words, texts) reflect real-world distances, whether semantic, stylistic, and so on.

The proliferation of available data, coupled with the rise in computing power and deep learning, has led to the creation of highly effective language models for word and document embeddings. These models incorporate complex semantic and linguistic concepts while remaining accessible to everyone and easily adaptable to specific tasks or corpora. One can use them to create author embeddings. However, it is challenging to determine the aspects on which a model will focus to bring authors closer or move them apart. In a literary context, it is preferable for similarities to primarily relate to writing style, which raises several issues. The definition of literary style is vague, assessing the stylistic difference between two texts and their embeddings is complex. In computational linguistics, approaches aiming to characterize it are mainly statistical, relying on language markers.

In light of this, our first contribution is a framework to evaluate the ability of language models to grasp writing style. We will have previously elaborated on text embedding models in machine learning and deep learning, at the word, document, and author levels. We will also have presented the treatment of the notion of literary style in Natural Language Processing, which forms the basis of our method.

Transferring knowledge between black-box large language models and these methods derived from linguistics remains a complex task. Our second contribution aims to reconcile these approaches through a representation learning model focusing on style, VADES (Variational Author and Document Embedding with Style). We compare our model to state-of-the-art ones and analyze their limitations in this context.

Finally, we delve into dynamic author and document embeddings. Temporal information is crucial, allowing for a more fine-grained representation of writing dynamics. After presenting the state of the art, we elaborate on our last contribution, BBADE (Brownian Bridge Author and Document Embedding), which models authors as trajectories. We conclude by outlining several leads for improving our methods and highlighting potential research directions for the future.

Résumé

La démocratisation récente et massive des outils numériques a donné à tous le moyen de produire de l'information et de la partager sur le web, que ce soit à travers des blogs, des réseaux sociaux, des plateformes de partage, ... La croissance exponentielle de cette masse d'information disponible, en grande partie textuelle, nécessite le développement de modèles de traitement automatique du langage naturel (TAL), afin de la représenter mathématiquement pour ensuite la classer, la trier ou la recommander. C'est l'apprentissage de représentation. Il vise à construire un espace de faible dimension où les distances entre les objets projetées (mots, textes) reflètent les distances constatées dans le monde réel, qu'elles soient sémantique, stylistique, ...

La multiplication des données disponibles, combinée à l'explosion des moyens de calculs et l'essor de l'apprentissage profond à permis de créer des modèles de langue extrêmement performant pour le plongement des mots et des documents. Ils assimilent des notions sémantiques et de langue complexes, en restant accessibles à tous et facilement spécialisables sur des tâches ou des corpus plus spécifiques. Il est possible de les utiliser pour construire des plongements d'auteurices. Seulement il est difficile de savoir sur quels aspects un modèle va se focaliser pour les rapprocher ou les éloigner. Dans un cadre littéraire, il serait préférable que les similarités se rapportent principalement au style écrit. Plusieurs problèmes se posent alors. La définition du style littéraire est floue, il est difficile d'évaluer l'écart stylistique entre deux textes et donc entre leurs plongements. En linguistique computationnelle, les approches visant à le caractériser sont principalement statistiques, s'appuyant sur des marqueurs du langage.

Fort de ces constats, notre première contribution propose une méthode d'évaluation de la capacité des modèles de langue à appréhender le style écrit. Nous aurons au préalable détaillé comment le texte est représenté en apprentissage automatique puis en apprentissage profond, au niveau du mot, du document puis des auteurices. Nous aurons aussi présenté le traitement de la notion de style littéraire en TAL, base de notre méthode.

Le transfert de connaissances entre les boîtes noires que sont les grands modèles de langue et ces méthodes issues de la linguistique n'en demeure pas moins complexe. Notre seconde contribution vise à réconcilier ces approches via un modèle d'apprentissage de représentations d'auteurices se focalisant sur le style, VADES (*Variational Author and Document Embedding with Style*). Nous nous comparons aux méthodes existantes et analysons leurs limites dans cette optique là.

Enfin, nous nous intéressons à l'apprentissage de plongements dynamiques d'auteurices et de documents. En effet, l'information temporelle est cruciale et permet une représentation plus fine des dynamiques d'écriture. Après une présentation de l'état de l'art, nous détaillons notre dernière contribution, B²ADE (*Brownian Bridge for Author and Document Embedding*), modélisant les auteurices comme des trajectoires. Nous finissons en décrivant plusieurs axes d'améliorations de nos méthodes ainsi que quelques problématiques pour de futurs travaux.

Remerciements

Si rédiger cette thèse ne fut pas une sinécure, c'est tout l'inverse de ce qui va venir. J'espère depuis un moment avoir l'occasion de remercier tous ceux qui m'ont permis d'arriver jusqu'ici dans la joie et la volupté.

Tout d'abord, un immense merci à mon directeur, Julien. Tu as su me guider et m'aiguiller avec bienveillance et patience. Tu es toujours resté positif et a fait preuve d'un optimisme sans faille tout au long de cette aventure. J'espère que ces travaux te rendre au moins une part de ce que tu m'as amené. Je te remercie aussi grandement d'avoir mis Antoine sur ma route. Il est dans mon cœur mon second directeur, à défaut de l'être officiellement. Si le doctorat était une balade à cheval, Antoine serait le cow-boy qui plus que me mettre le pied à l'étrier, aurait sellé le cheval, fait le plein de carottes et conduit ce voyage dans les grandes étendues de la recherche guitare à la main. Tu es parti heureux pour les steppes de Saint-Etienne mais je te serais éternellement reconnaissant. Julien, je ne me suis pas permis de métaphore équestre avec toi, mais c'est avec plaisir que je le ferais si jamais tu en fais la demande.

Merci aux membres du jury. Merci aux deux rapporteurs, Lynda et Mathieu d'avoir accepté de relire ce manuscrit. Merci à Christophe, également pour ton suivi en CSI avec Fadila, vos conseils et encouragements m'ont permis d'avancer sereinement dans mes travaux. Merci également à Armelle et Damien d'avoir accepté d'être examinateur·ice.

Merci à toute l'équipe LIFRANUM de Marge (et de la BnF) pour les échanges tout au long de l'année. S'ils n'ont pas mené aussi loin que nous l'espérions, ils ont toujours été très enrichissants et m'ont ouvert l'esprit sur de nombreuses idées mêlant nos domaines respectifs. Merci pour tous ces délicieux plateaux repas du vendredi midi qui maintenaient nos estomacs occupés en même temps que nos esprits. Une pensée particulière pour Fanny, mon alter ego littéraire à qui je souhaite le meilleur.

Je tiens forcément à remercier l'ensemble des membres du laboratoire ERIC pour votre aide, nos échanges et tout simplement l'ambiance chaleureuse du laboratoire : Adrien, Antoine, Jairo, Jérôme, Julien, Julien, Sabine, Stéphane, ... Un merci tout particulier à Habiba d'avoir été aux petits soins avec moi et d'avoir absorbée une partie de ma phobie administrative (Julien ayant encaissé ce qu'il restait).

Un grand merci à tous mes collègues doctorant·e·s, post-doctorant·e·s, stagiaires, notamment tous les passagers de ce grand train qu'est notre bureau. Antoine (encore), Gaël (un excellent représentant des doctorants selon moi), Martial, Eliz, Hui, Irina, Francesco (un tout aussi excellent représentant des doctorants selon moi), vous avez tous été de superbes compagnons de route sur les rails de la recherche, entre astuces SIGED et délicieux sandwichs Boogy's. Merci Redha pour les précieux conseils de fin de thèse. Je pense aussi à Arwa, Jacques, Jean, Loïc, Jean-Steve, Hugo, ..., merci à tous pour l'ambiance studieuse au bâtiment K et houblonnées en dehors.

Je n'aurais pas tenu ces 3 années sans avoir quelques soupapes de décompression. Merci à Nathan pour les badminton hebdomadaires, Djaïa pour l'ultra-trail-running plat occasionnel. Surtout, merci à l'équipe du Lyon Handball de m'avoir permis de poser mon cerveau 3 fois par semaine à l'entrée du gymnase. Tout particulièrement, merci à Guilhem, petit ange parti trop tôt, et Vincent dont nos affinités ne font que croître depuis Bourg-de-Péage. Merci l'artiste.

Merci à mes ami·e·s qui sont autant de musiciens de l'orchestre qui rythme ma vie : Antoine, mon frère d'une autre mère, le parrain des enfants que je n'ai pas, le Dino de ma Shirley, Axel, Inès, Gautier, Marc, Pauline, Marion, Mathou, Mathilde, Chrystelle, Julie, Clément, Auré, Stèph, Johann, Claire, Samir, Raphaëlle, Fanny, Thomas, Benjamin (merci d'avoir été mon lièvre), Jean, Charles, Louis, Morgane, Clémence, Leïla, Stéphane, Julie, Ben Vélo. Elise, de New York jusqu'à Lyon, que tu m'auras fait découvrir avec ses locaux, les «« moules »» (un immense merci à tous l'équipe), je te remercie du plus profond de mon cœur, pour ce voyage et pour le reste. Pour celleux que j'aurais oublié, j'en suis profondément désolé, sachez que j'ai une phobie administrative. N'hésitez pas à venir boire un café à l'appartement qu'on règle nos comptes autour d'une crêpe.

Merci enfin à ma famille, Papa d'avoir toujours proposé ton aide même quand tu ne comprenais plus les questions, de m'avoir inculqué la curiosité de tout, le goût du sport et de m'avoir maintenu les pieds sur terre. Maman de m'avoir toujours accompagné, soutenu et poussé, notamment vers la recherche en médecine. Si je suis docteur ce sera grâce à toi, en informatique je pourrais déjà soigner des machines, ça demande moins de sensibilité. Gildas merci pour ton soutien inconditionnel, nos longues discussions, ta tolérance devant mes facéties. Merci de nous avoir fait rencontrer Christelle, son bagou, son écoute, ses ciseaux et ses filles, Aglaë et Margot. Isabelle merci d'avoir été patiente avec notre fratrie, merci pour les sorties scolaires à Guerchy, merci pour tout. Lucile (et Gilles), Jonas, Léo, Paco et Etienne, merci à vous tous d'être autant de ports d'attache dans l'océan de ce monde, hâte de tous vous visiter avec mon optimiste. Merci aussi à Iety, Chupa, Hervé, Socquettes et Wallace, nous sommes tous des animaux, mais vous, vous êtes simplement les meilleurs.

Enfin, Lola, je ne t'ai citée nulle part parce que j'aurais dû te citer partout. D'abord, désolé pour le coup de poing pour la GameBoy et pour ton poster Cheval Mag. Mais surtout, merci de m'avoir (en)traîné dans toutes tes activités, tous tes voyages et toutes tes amitiés. Merci de m'avoir soutenu et accompagné dans tous mes projets, d'Issy les Moules (bisous Pauline) au fond du lac d'Aiguebelette. Je n'aurais pas assez d'une vie pour te rendre ce que tu m'as apporté, je fais du mieux que je peux et j'espère que tu me supporteras encore un peu. Du plus profond de mon âme, merci.

Table des matières

Notations

Générales

$x_{1:n}$ $x_1, x_2, ..., x_n$

$f_\theta(x)$ Fonction de la variable x et de paramètres θ

$\bigtriangledown_x f(x)$ gradient de la fonction f par rapport à la variable x (vecteur des dérivées partielles en chaque coordonnée de x)

x_i $i^{\text{ème}}$ ligne de la matrice X

$x_{i,j}$ Elément en position (i, j) de la matrice X

$\sigma(x) = \frac{1}{1+e^{-x}}$ Fonction sigmoïde

$\text{ReLU}(x) = \max(0, x)$ Fonction d'activation ReLU (*REctified Linear Unit)*

$\mathcal{L}$ Fonction de perte ou *loss*

$p_\theta(x)$ Notation générale de la probabilité de x par rapport à une loi de paramètres θ

$\mathbb{E}[x]$ ou $\mathbb{E}_{p(x)}[x]$ Espérance de x par rapport à la loi $p(x)$

$\mathcal{N}(\mu, \Sigma)$ Loi normale de moyenne μ et de variance Σ

$||x||_q = (|x_1|^p + ... + |x_n|^p)\frac{1}{p}$ Norme d'ordre q d'un vecteur x

$\text{KL}(p(x)||q(x)) = \int_{-\infty}^{+\infty} p(x) \log(\frac{p(x)}{q(x)}\, dx$ Divergence de Kullback-Leibler mesurant la dissimilarité entre les distributions de probabilités $p(x)$ et $q(x)$

$I(x, y) = \int\int p(x, y) \log \frac{p(x,y)}{p(x)p(y)} d_x d_y$ Information mutuelle entre les variables aléatoires x et y

Données textuelles

V Ensemble du vocabulaire (de taille $|V|$)

$\mathcal{C}$ Corpus de documents (de taille $|\mathcal{C}|$

w_i $i^{\text{ème}}$ mot d'une séquence de mots

d_i $i^{\text{ème}}$ document du corpus $\mathcal{C}$ de longueur $|d_i|$ et de vocabulaire $|V_{d_i}|$

a Auteurice de l'ensemble des auteurices $\mathcal{A}$

r Dimension de l'espace de représentation

z_d, z_a Représentation respectivement du document d et de l'auteurice a

$\text{sim}_{cos}(z_1, z_2) = \frac{<z_1, z_2>}{||z_1||_2||z_2||_2}$ Similarité cosinus entre les représentations z_1 et z_2

Données textuelles dynamiques

$[0, T^a]$ Intervalle temporel de publication ou d'écriture de l'auteurice a, du temps initial 0 (ou parfois t_0^a) au temps final T^a

d_t^a Document écrit ou publié par l'auteurice a au temps t (a priori unique)

z_t^a Représentation du document écrit ou publié par l'auteurice a au temps t

h_t^a Représentation de l'auteurice a au temps t

Acronymes

AD3 Attentive Deep Document Dater [RDT18].
ADAM Adaptative Moment Estimation.

BAC Blog Authorship Corpus.
BERT Bidirectional Encoder Representations from Transformers [Dev+19].
BPE Byte-Pair Encoding.
B²ADE Brownian Bridge for Author and Document Embedding.

CBOW Continuous Bag-Of-Words [Mik+13b].
CNN Convolutional Neural Network (Réseau de neurones convolutionnel).
CODE Co-Occurrence Data Embedding [Glo+04].

DAN Deep Averaging Network.
DAR Dynamic Author Representation [DLD19].
DGEA Dynamic Gaussian Embedding of Authors [Gou+22a].

GCN Graph Convolution Network.
GPT Generative Pre-trained Transformers.
GPU/TPU Graphics Processing Unit/Tensor Processing Unit.
GRU Gated Recurrent Unit ([Cho+14]).

IMDb Internet Movie Database.

LDA Latent Dirichlet Association.
LLM Large Language Model (Grand modèle de langue).
LRAP Label Ranking Average Precision.
LSTM Long Short Term Memory ([HS97]).

MLM Masked Language Modelling (Prédictions de mots masqués).
MLP Multi-Layer Perceptron (Perceptron multicouches).

NER Named Entity Recognition (Reconnaissance d'entitées nommées).

NSP Next Sentence Prediction (Prédiction de la phrase suivante).
NYT New York Times dataset.

PGD Project Gutenberg Dataset.
POS-tag Part-of-Speech Tagging (Etiquetage morpho-syntaxique).

RBF Radial Basis Function.
RNN Recurrent Neural Network (Réseau de neurones récurrents).

S2G Semantic Scholar dataset.
SGNS Skip-Gram with Negative Sampling [Mik+13a].
SLNI Stanford Natural Language Inference [Bow+15].
STEL Similarity based STyle Evaluation [WN21].
SVM Support Vector Machine.

TAL Traitement Automatique de la Langue.

UAR Universal Authorship Representation [Riv+21].
USE Universal Sentence Encoder [Cer+18].

VADE Variational Author and Document Embedding [Gou21].
VADES Variational Author and Document Embedding with Style.
VAE Variational Auto-Encoder (Auto-encodeur variationnel).
VIB Variational Information Bottleneck.

Chapitre 1

Introduction

1.1 Contexte

Le développement du web et l'accès grandissant au numérique à permis à tous de créer, produire et partager des informations en ligne. Ces données peuvent être de formats variés, notamment textuel. Leur quantité disponible est massive et ne cesse de croître via les réseaux sociaux, les médias et sites d'information, les blogs, les campagnes de numérisation, etc ... Il est indispensable d'être capable de traiter ces grandes quantités de données automatiquement. C'est l'objectif du Traitement Automatique de la Langue (ou NLP pour *Natural Language Processing*).

Le TAL est apparu avec les premiers travaux en intelligence artificielle et consistait en des modèles à base de règles logiques nécessitant d'être définies en amont. Un exemple connu est celui du premier robot conversationnel ELIZA [Wei66]. En parallèle des statistiques textuelles, le développement de l'apprentissage automatique (ou *Machine Learning*) a permis de chercher à extraire des motifs latents de corpus de textes. A la croisée des statistiques, de l'optimisation et de l'informatique, l'apprentissage automatique vise à donner la capacité aux ordinateurs de résoudre des problèmes à partir de données, des instances de ces problèmes. Une grande partie des méthodes utilisées cherche à estimer des modèles permettant de capter l'organisation des données. Cette organisation peut être observable, via des classes ou des valeurs disponibles ou annotées, il s'agit alors d'apprentissage supervisé. Elle peut aussi être non-observable, ou latente, on parle alors d'apprentissage non-supervisé. Les modèles devront d'eux mêmes découvrir la structure des données.

L'un des objectifs en TAL est de pouvoir construire des représentations des mots et/ou des textes reflétant leur organisation sémantique, syntaxique, ... Ces représentations pourront ensuite être traitées par d'autres modèles pour réaliser différentes applications ou sous-tâches, entre autres :

- La traduction automatique d'une langue vers une autre
- La génération automatique, tant pour des systèmes de questions/réponses que pour des chatbots guidant des utilisateurs
- La classification automatique de documents, selon des critères variés : thématiques, sentiments, origines, ...
- La recherche d'information, permettant d'extraire une information d'un corpus à partir d'une requête par exemple

Les approches historiques définissaient explicitement les représentations, là où les approches plus récentes s'appuient sur l'apprentissage automatique pour apprendre des représentations appréhendant l'organisation de la langue. C'est l'apprentissage de représentation (ou *Representation Learning*).

1.2 Méthodes historiques de représentation du texte

Les premières approches (encodage *one-hot*, TD-idf) construisaient les représentations des mots comme des vecteurs binaires de $\{0,1\}^{|V|}$ sur l'ensemble du vocabulaire V du corpus. Des pondérations en fonction des fréquences d'apparitions des mots dans chaque document ou dans le corpus permettaient d'apporter plus d'importance aux mots les plus rares. Ces méthodes ont deux limites très fortes.

La première est la très grande dimension de représentation. Par exemple, le Grand Robert de la langue française compte autour de 100 000 définitions. Cela va générer des vecteurs de représentations très grands, tout en étant creux donc portant peu d'information. Aux coûts de stockage et de calculs s'ajoute le problème du *curse of dimensionnality*. Plus la dimension de l'espace à caractériser est grande plus il faudra de données d'entrée pour le quadriller. Ce lien est exponentiel et c'est là un des aspects du fléau de la dimension.

La seconde limite est que ces représentations ne contiennent aucune information sémantique. Le mot "mer" sera aussi proche du mot "bateau" que du mot "bureau". La proximité des documents dans l'espace de représentation dépendra uniquement de l'utilisation de mots communs. Par exemple, pour les citations suivantes :

1. "Si vous voulez aller sur la mer, sans aucun risque de chavirer, alors, n'achetez pas un bateau : achetez une île !" *Fanny*, Marcel Pagnol
2. "Elle dit aussi que s'il n'y avait ni la mer ni l'amour personne n'écrirait des livres." *Les petits chevaux de Tarquinia*, Marguerite Duras
3. "La littérature : un coup de hache dans la mer gelée qui est en nous." *Lettre de Franz Kafka à Oskar Pollak*, Franz Kafka

Nous pourrions nous attendre à ce que les extraits de Marguerite Duras et Franz Kafka soient les plus proches, faisant intervenir à la fois la littérature et la mer. Or, avec la représentation historique Tf-idf, ça n'est pas le cas, l'extrait de Marcel Pagnol se retrouvant à proximité de celui de Franz Kafka, alors qu'ils ne mentionnent la mer que comme une métaphore.

Pour contourner les problèmes de dimension ont été proposés des modèles visant à compresser les vecteurs de documents et de mots. Les approches les plus récentes, basées sur l'apprentissage profond, visent à apprendre directement les représentations en faible dimension. C'est l'apprentissage de représentation.

1.3 Méthodes récentes de représentation du texte

L'idée derrière ces modèles est l'hypothèse distributionnelle, introduite notamment par [Har54 ; Fir57]. Cette dernière stipule que les mots apparaissant dans des contextes similaires auront des significations similaires. Les modèles en découlant sont appelées modèles de langue. Ils calculent une probabilité pour chaque mot du vocabulaire à partir de leur contexte et de leur représentation. Un des modèles fondateurs utilisant l'apprentissage profond est le modèle CBOW (*Continuous*

Bag-Of-Word)) et sa variante SkipGram [Mik+13b ; Mik+13a]. L'objectif pour l'apprentissage de représentation sémantique d'un mot est de prédire son voisinage, ou inversement de prédire le mot à partir de son voisinage. En liant la probabilité de prédire un mot en fonction de son contexte à une mesure de similarité entre leurs vecteurs (ici le produit scalaire), nous nous assurons de construire un espace capturant la sémantique.

Chaque mot possède alors une représentation unique, ce qui ne permet pas de couvrir les polysémies. La représentation associée à "mars", unique, couvrira autant les occurrences comme mois, planète, ou barre chocolatée. Pour palier cela, le modèle BERT [Dev+19] proposent des représentations contextualisées. Chaque mot aura un plongement distinct en fonction de la phrase dans laquelle il apparaît. BERT s'appuient sur la brique Transformers [Vas+17], comme tous les modèles les plus récents (LLama [Tou+23], GPT4 [Ope23]). Par un passage à l'échelle en terme de volume de données d'entrée et de taille de modèles ont été développés des modèles de langues extrêmement performants. Facilement spécialisables sur des corpus spécifiques, ils possèdent des capacités de généralisation sans précédent qui leurs permettent d'être compétitifs sur de nombreuses sous-tâches. Si ces sous-tâches sont parfois utilisées pour entraîner les modèles, l'objectif final est bien d'apprendre des représentations significatives, capturant des notions de langue complexes, tant sémantiques que syntaxiques ou grammaticales.

Ces représentations des mots peuvent ensuite être agrégées de multiples façon afin de construire des plongements à diverses échelles du texte, la phrase, le paragraphe, le document, ... Jusqu'à être étendues aux objets qui lui sont liés, comme les auteurices, les thématiques ou même les dates.

Nous allons maintenant exposer notre cadre d'application, le projet LIFRANUM, qui constitue le point de départ de cette thèse afin d'exposer les différentes limites de ces grands modèles de langue.

1.4 Le projet LIFRANUM

Le projet LIFRANUM est un projet visant à "identifier et structurer le corpus des littératures francophones nativement numériques". L'aspect nativement numérique, à savoir la production faite sur le web pour le web, implique de pouvoir traiter de grands corpus et donc l'utilisation du TAL. C'est un projet pluridisciplinaire, faisant intervenir une équipe du laboratoire en littérature MARGE, de l'Université Jean Moulin Lyon 3, une équipe de la BnF (Bibliothèque Nationale de France) et notre équipe du laboratoire en informatique ERIC, de l'Université Lumière Lyon 2. Il comporte deux aspects distincts :

— l'identification et la création du corpus de littérature francophone nativement numérique à partir de méthodes de capture du web, puis son indexation

— la structuration et l'analyse automatique de ce corpus, idéalement par des méthodes interprétables de représentation afin d'être utilisées par tous

Les travaux de notre équipe et donc en particulier les miens s'inscrivaient principalement dans le second point. Il s'agissait d'utiliser les outils du TAL et notamment d'apprentissage de représentation pour ordonner le corpus construit a priori. Il fallait pour cela adapter nos approches au cadre spécifique de la littérature et les vulgariser afin de donner des moyens d'analyse à nos collaborateurs moins familiers de ces méthodes.

Ce cadre littéraire, relativement rare en apprentissage de représentation, soulèvent différentes problématiques que nous allons détailler en section suivante avec les objectifs pour cette thèse qui en découlent.

1.5 Objectifs de la thèse

Le traitement d'un corpus littéraire implique de se focaliser sur deux concepts spécifiques. La notion d'auteurice d'une part et celle de style écrit d'autre part qui lui est liée. Le style littéraire est souvent traité en linguistique computationnelle par des approches statistiques sur des marqueurs syntaxiques. Il est en revanche moins évoqué en apprentissage de représentation, car c'est une notion difficile à identifier et mal définie. Les modèles de langue évoqués plus haut semblent être des points de départ pertinents pour capturer le style. Ils n'en restent pas moins des boîtes noires, initialement construits autour de la sémantique là où le style littéraire fait intervenir des considérations linguistiques plus variées.

L'apprentissage de représentations d'auteurs et d'autrices rencontrent les mêmes limites. Il est difficile voir impossible de déterminer quels aspects de la dynamique d'écriture des auteurices vont rapprocher ou éloigner leurs représentations dans l'espace latent, qu'ils soient sémantiques, stylistiques ou temporels. Le transfert de connaissances entre les approches usuelles en littérature et linguistique et celles de représentation en TAL est complexe. C'est un blocage dans l'objectif de représenter le style et les auteurices.

L'objectif principal de cette thèse est alors de construire des espaces de plongements des auteurices et de leurs documents capables de capturer ces dynamiques spécifiques d'écriture des auteurices. Les modèles de représentations devront autant que possible utiliser les modèles de langue récents afin de faire fructifier leur capacité de compréhension du language.

Les dynamiques en question peuvent couvrir le style écrit, la temporalité, la sémantique ... Il faudra être capable de démontrer que l'espace construit appréhende bien la notion visée. Par exemple pour le style littéraire il serait pertinent de chercher à unifier les approches linguistiques à celles des modèles de langues récentes. Les travaux autour de la sémantique étant légions, nous nous focaliserons principalement sur les autres concepts.

Nous présentons maintenant le plan de cette thèse et les contributions associées.

1.6 Plan et contributions

Dans le chapitre 2, nous développons plus en détails les méthodes existantes d'apprentissage de représentation de mots, de documents et d'auteurices. Cela nous permet d'introduire les notions et concepts indispensables à la bonne compréhension des chapitres suivants.

Dans le chapitre 3, nous évoquons le traitement du style littéraire en TAL, et la question de sa représentation notamment. Nous nous appuyons sur les travaux existants pour proposer une méthode d'évaluation de la capture du style littéraire par les modèles de représentations de documents et d'auteurices. C'est notre première contribution, publié au workshop Eval4NLP de la conférence EMNLP 2021 [TGV21].

Dans le chapitre 4, nous nous appuyons sur la méthode développée au chapitre 3 pour construire notre propre modèle d'apprentissage de représentations d'auteurs capturant le style, VADES (*Variational Author and Document Embedding with Style*). C'est notre seconde contribution. Nous comparons ensuite favorablement notre modèle aux méthodes de plongement d'auteurices ayant la même finalité.

Dans le chapitre 5, nous laissons de côté l'aspect stylistique afin de nous intéresser à la notion de temporalité dans la représentation d'auteurs, et à l'intérêt des modèles de représentation dynamique d'auteurs, que nous présenterons alors. Ensuite, nous développerons notre dernière contribution,

B²ADE (*Brownian Bridge for Author and Document Embedding*) qui propose de modéliser les auteurices comme des trajectoires continues, là où l'existant discrétise systématiquement le temps.

Enfin, nous détaillons quelques axes d'améliorations de nos travaux et des perspectives de recherches futures qui pourrait en découler. Nous ferons également un bilan de nos contributions au projet LIFRANUM.

Chapitre 2

Etat de l'art

2.1 Introduction

Au cours de ce chapitre, nous aborderons l'ensemble des concepts indispensables à la bonne compréhension du cadre dans lequel s'inscrit ce manuscrit et les contributions qui y sont développées. Nous présenterons dans un premier temps le deep learning (ou apprentissage profond), branche de l'apprentissage automatique et les notions principales mises en jeu (architecture, fonction de perte, optimisation). Ensuite, nous ferons une revue chronologique des méthodes de deep learning utilisées justement pour construire des représentations de mots. Plusieurs modèles seront présentés, des plus anciens comme le CBOW ou le Skip-Gram, au plus récents Large Language Models. Certains serviront d'étalons dans la mesure du style littéraire (Chapitre 3) ou de briques de base de nos modèles de représentation (Chapitres 4, 5). Pour finir, nous passerons de l'échelle du mot à celle du document, puis à celle de l'auteurice, en présentant différents modes d'agrégation permettant l'apprentissage de représentation pertinente. Ces techniques constituent les fondements des modèles VADES, B^{2}ADE et de leurs compétiteurs.

2.2 Deep Learning

Le contexte général de nos travaux est celui du machine learning (ou apprentissage automatique), à la croisée des statistiques, de l'optimisation et de l'informatique. Nous nous limiterons ici au cas de l'apprentissage supervisé, où chaque donnée est associée à une étiquette. Le problème générique est de déterminer un modèle qui à partir d'une base d'apprentissage $(x_i, y_i)_{1 \leq i \leq n}$ permettra de lier la donnée d'entrée x_i à son étiquette y_i. A noter que tous les types d'entrée peuvent être pris en considération en fonction du problème étudié : image, texte, son, vecteur réel... Il en va de même pour les sorties, bien que les deux cas principalement rencontrés concernent la classification ($y_i \in \{1, 2, ..., K\}$) et la régression ($y_i \in \mathbb{R}$). Pour simplifier, un modèle n'est alors qu'une classe de fonctions $f_\theta : \mathbb{R}^d \longrightarrow \Sigma$ indexée par un ensemble de paramètres $\theta \in \Theta \subset \mathbb{R}^p$, où Σ est l'espace des sorties possibles. Le but de l'apprentissage automatique est alors de trouver un ensemble θ^* tel que la fonction f_{θ^*} :

— Capture autant que possible la régularité des données permettant de lier x_i à y_i $(1 \leq i \leq n)$ à partir de la base d'apprentissage.

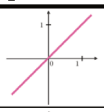
Nom	Formule	Représentation	Espaces
Linéaire	$\sigma(x) = x$		$\mathbb{R} \longrightarrow \mathbb{R}$
Tanh	$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$		$\mathbb{R} \longrightarrow [-1; 1]$
ReLU	$\sigma(x) = \max(0, x)$		$\mathbb{R} \longrightarrow \mathbb{R}^+$
Sigmoid	$\sigma(x) = \frac{1}{1 + e^{-x}}$		$\mathbb{R} \longrightarrow [0; 1]$
Softmax	$\sigma(\boldsymbol{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$	Multidimensionnelle	$\mathbb{R}^d \longrightarrow \mathbb{R}^d$

FIGURE 2.1 – Exemples de fonctions d'activation fréquentes

— Généralise efficacement sur des données nouvelles, autrement dit, pour $i_0 \notin \{1..n\}$ retourne $\hat{y}_{i_0}$, prédiction de y_{i_0} pour une nouvelle entrée x_{i_0} issue d'un ensemble test.

Il apparaît alors nécessaire d'introduire des fonctions d'évaluation permettant de quantifier la capacité des modèles à réaliser ces objectifs et à les optimiser. Ce qui va différencier les différentes approches d'apprentissage automatique sont les types de fonctions f_θ utilisées. Dans le cas du deep learning, ces dernières s'appuient sur les réseaux de neurones comme brique élémentaire. C'est ce que nous détaillons ci-après.

2.2.1 Les réseaux de neurones

Un neurone est une fonction f très simple ayant pour seuls paramètres un vecteur de poids $w \in \mathbb{R}^d$ et un biais $b \in \mathbb{R}$ à laquelle on associe une fonction d'activation σ. Ainsi pour une entrée $x \in \mathbb{R}^d$ un neurone associera la sortie (où le produit scalaire est noté $\langle ., . \rangle$) :

$$y = f_{w,b}(x) = \sigma(\langle w, x \rangle + b) \tag{2.2.1}$$

Les fonctions d'activation sont nombreuses (2.1) et permettent d'insérer des non-linéarités dans les modèles. Une couche est alors simplement un ensemble de neurones, paramétrés donc par une matrice W et un vecteur de biais $\mathbf{b}$. Un réseau de neurones est une structure constituée de plusieurs couches de neurones où la sortie d'une couche devient l'entrée de la suivante et ainsi de suite. Il est aussi possible que le résultat issu d'un neurone soit utilisé par des neurones de la même couche ou d'une couche précédente (cas réseaux de neurones récurrents, qui seront abordés plus loin). L'architecture la plus basique est le Multilayer Perceptron (MLP). Chaque neurone est connecté à tous les neurones de la couche suivante. En général, la dernière fonction d'activation est différente de celles des couches cachées. Par exemple, dans le cas d'un problème de classification binaire

($y \in \{0,1\}$), la dernière couche aurait une sigmoïde pour fonction d'activation afin de retourner $\mathbb{P}(y=1|x)$. Un MLP à L couches se résume à la fonction suivante :

$$f_\theta^{MLP}(x) = (f_1 \circ f_2 \circ ... \circ f_L)(x) \tag{2.2.2}$$

Les principaux facteurs sur lesquelles il est possible de jouer sont le nombre de couches L et le nombre de neurones contenus par chacune. C'est ce qui fixera le nombre de paramètres totaux du réseau, la force des réseaux de neurones résidant dans leur profondeur. Une fois l'architecture choisie, il convient d'estimer les meilleurs paramètres possibles à partir de la base d'entraînement : c'est la phase d'optimisation.

2.2.2 Entraînement et optimisation des modèles

Dans un premier temps, il est nécessaire de définir une fonction de perte $\mathcal{L}(\theta)$ (ou loss function). Celle-ci permettra de quantifier l'erreur réalisée dans la prédiction de y par notre réseau de neurones et nous cherchons donc à la minimiser. En d'autres termes, à estimer un minimum global θ^* tel que :

$$\forall \theta, \mathcal{L}_{(x,y)}(\theta) \geq \mathcal{L}_{(x,y)}(\theta^*) \tag{2.2.3}$$

Cependant, les fonctions entrant en jeu en deep learning n'étant en règle générale pas convexe, ce sont des minima locaux qui sont atteints. La propriété précédente n'est alors vérifiée que dans un voisinage du minimum. Dans le cas d'un problème de régression, c'est généralement l'erreur quadratique moyenne qui est minimisée :

$$\mathcal{L}_{(x,y)}(\theta) = \mathbb{E}_{(x,y)}[\|y - f_\theta(x)\|_2^2] = \frac{1}{n}\sum_{i=1}^{n} \|y_i - f_\theta(x_i)\|_2^2 \tag{2.2.4}$$

Pour notre exemple de classification binaire est préférée l'entropie croisée binaire car la minimiser correspond à optimiser le modèle en maximisant la vraisemblance. Cela correspond à minimiser l'écart entre la distribution des données d'entraînement et celle induite par le modèle :

$$\mathcal{L}_{(x,y)}(\theta) = \sum_{i=1}^{n} -y_i \log(f_\theta(x_i)) - (1-y_i)\log(1 - f_\theta(x_i) \tag{2.2.5}$$

Il est fréquent de vouloir contrôler la taille des paramètres, pour éviter des problèmes de sur-apprentissage notamment et limiter la complexité du modèle. Cela passe généralement par l'ajout d'un terme de régularisation à la fonction de perte (où la norme L_q est notée $\|.\|_q$) :

$$\Omega(\theta) = \sum_{j=1}^{p} \|\theta_j\|_q \tag{2.2.6}$$

L'utilisation de la norme 1 amènera des solutions parcimonieuses, tandis que la norme 2 réduira la norme des paramètres. Le nombre élevé de paramètres et la complexité des fonctions en jeu ne permet pas de trouver une formule analytique du minimiseur θ^* de $L(\theta)$, ce qui implique d'utiliser une approche numérique : la descente de gradient. C'est une méthode itérative qui s'appuie sur une propriété des minima assurant que, si la fonction est dérivable, sa dérivée sera nulle en ces points. En partant d'un point initial θ_0, la règle de mise à jour est la suivante :

$$\theta_{k+1} = \theta_k - \eta \bigtriangledown_\theta \mathcal{L}_{(x,y)}(\theta) \tag{2.2.7}$$

Dans la mesure où η, le pas d'apprentissage (ou learning rate), est suffisamment petit, nous sommes assurés que la fonction de perte converge. C'est à dire que sa variation d'une étape à une autre soit inférieure à un seuil fixé. En réalité, dans le cadre du deep learning et des grands ensemble de données, il est impossible de traiter l'ensemble des entrées d'un coup. C'est pourquoi nous utilisons plutôt la descente de gradient stochastique (Voir algorithme 1).

Algorithm 1 Descente de gradient stochastique

Fix parameters η (learning rate), b (batch size), N number of epochs.
Random initialization $\theta = \theta_0$.

for N iterations **do**
  for each subset B of training data (x_i, y_i) of size b **do**
    $\theta = \theta - \eta \frac{1}{b} \sum_{i \in B} \bigtriangledown_\theta \mathcal{L}_{(x_i, y_i)}(\theta)$
  end for
end for

Chaque sous-ensemble B du jeu d'entraînement est appelé un batch. Une itération sur l'ensemble des données est appelée une epoch. La taille des batchs, le nombre d'epochs et la learning rate sont trois paramètres importants d'entraînement des modèles de deep learning. Le gradient n'est alors plus calculé que pour un batch. Grâce à la méthode de la dérivation en chaîne et de la rétropropagation du gradient (non détaillées ici), il est très simple d'utiliser cet algorithme pour l'entraînement des réseaux de neurones. Plusieurs variantes ont été proposées au cours des dernières années pour le rendre moins sensible au choix du pas d'apprentissage. L'idée est d'ajouter une correction lors de la mise à jour du gradient. La méthode la plus connue est **Adam** pour Adaptive Moments estimation.

Enfin, d'autres méthodes de régularisation existent pour aider les modèles à généraliser. L'early stopping, qui vise à arrêter l'entraînement lorsque la loss sur un ensemble de validation arrête de décroître, ou le dropout, consistant à éteindre aléatoirement certains neurones lors d'une itération. L'ensemble de ces techniques combiné à l'essor des moyens de calculs, notamment grâce au TPU et GPU permettant de paralléliser efficacement les calculs matriciels, explique l'explosion du deep learning ces dernières années. En permettant de produire des réseaux de plus en plus profonds, traitant des ensemble de données de plus en plus grands, les modèles sont de plus en plus performants dans de nombreuses tâches. C'est d'autant plus vrai en Traitement Automatique de la Langue, c'est ce que nous allons détailler dans la section suivante.

2.3 Deep Learning pour le Traitement Automatique de la Langue

Au cours de cette section, nous allons détailler une brique essentielle du TAL qu'est l'apprentissage de représentation de mots. En effet, l'apprentissage de représentation en général a pour but de construire des représentations vectorielles des entrées, de faible dimension, capturant les caractéristiques des données et donc significatives pour la réalisation d'un ensemble de sous-tâches dans un second temps. Dans le cadre du TAL, nous souhaitons apprendre notamment des plongements de mots (ou *embeddings*) possédant un sens sémantique. Nous allons évoquer les principaux modèles et les architectures associées, d'abord avec les travaux références de [Mik+13b ; Mik+13a], puis avec les représentations contextualisées jusqu'à la révolution BERT de [Dev+19]. Enfin nous

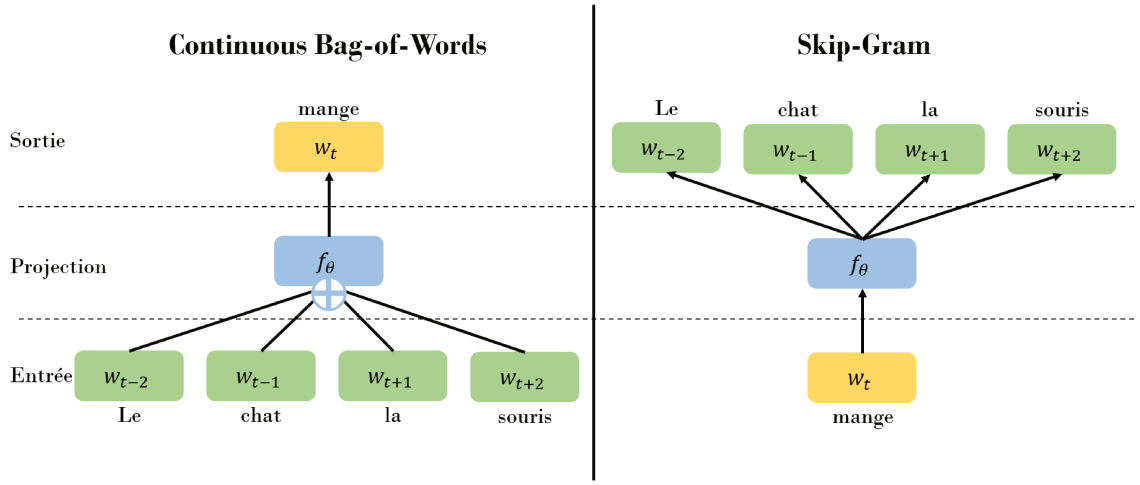


FIGURE 2.2 – Schéma des modélisations CBOW et Skip-Gram

terminerons en évoquant le récent changement d'échelle apporté par les Large Language Models (ou LLMs), sous-ensemble fondateur des Foundation Models.

2.3.1 Sac de mots, Skip-Gram et leurs variantes

Si Word2Vec et ses variantes [Mik+13b ; Mik+13a] sont des modèles fondateurs pour l'apprentissage de plongements de mots, des travaux antérieurs proposaient déjà d'utiliser des réseaux de neurones avec une approche similaire [BDV00]. Le succès de Mikolov peut s'expliquer en partie par l'essor du deep learning, dû à l'augmentation des données disponibles et des capacités de calcul notamment. Il profite de ces données d'entrée continues, de faibles dimensions, pour des applications dans de multiples domaines (psychologie, sciences sociales, sciences cognitives, ...).

Le modèle en lui-même s'appuie sur l'hypothèse distributionnelle, formulée par Harris (également évoqué à la même période dans les travaux de Firth et Wittgenstein) en 1954 [Har54] : "*words are characterised by the company that they keep*". Autrement dit, il est possible d'apprendre des représentations de mots en essayant de prédire les mots qui l'entoure dans une phrase. En effet, deux mots au sens proche seront souvent observés dans des contextes sémantiques proches. Word2Vec propose alors deux modélisations distinctes : CBOW (pour Continuous Bag-of-Words) et Skip-Gram (voir figure 2.2). Chaque mot est tout d'abord associé à un vecteur de taille r via une matrice de paramètres $W \in \mathbb{R}^{|V| \times r}$, où V est l'ensemble du vocabulaire et r correspond à la dimension de l'espace latent (l'ordre de grandeur usuel est de 10^2). A chaque mot w_i sont associées deux représentation. La première vers la couche cachée, notée u_i. La seconde vers la couche de sortie, $v_i = f_\theta(w_i)$, obtenue après projection. Les paramètres à apprendre sont la matrice W et la matrice de projection θ. Nous allons d'abord introduire le modèle Skip-Gram. Le but est, à partir d'une séquence de mot $w_1, w_2, ..., w_T$, de maximiser la log-vraisemblance suivante :

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c}^{j\neq 0}\log p(w_{t+j}|w_t) \tag{2.3.1}$$

où c est la taille du contexte choisie (généralement 2 ou 3) et T la taille du document. La probabilité de trouver le mot w_j dans le contexte de taille c centré en w_i en utilisant la fonction softmax est alors définie par :

$$p(w_j|w_i) = \frac{e^{v_j \cdot u_i}}{\sum_{j=1}^{|V|} e^{v_j \cdot u_i}} \tag{2.3.2}$$

En pratique, le coût de calcul du gradient dans ce cas est trop élevé car proportionnel à la taille du vocabulaire $|V|$, aux alentours de 32 000 mots pour le français courant. Comme alternative, ils proposent d'utiliser le Noise Contrastive Estimation (NCE). Le problème devient un simple cas de classification binaire, où le modèle doit être capable de distinguer à partir d'une paire de mots s'ils cooccurent (exemple positif) ou non (exemple négatif). Pour chaque occurence effectivement observée dans le corpus, les auteurs tirent k exemples négatifs (choisi entre 5 et 20 en général) avec la loi catégorielle suivante (loi multinomiale avec un tirage unique) $p(w) \sim Multi((f_j^{\frac{3}{4}})_{1\leq j\leq |V|})$ avec f_j fréquence d'occurence du mot w_j dans le corpus d'entraînement. Ainsi, un mot courant aura plus de chance d'être tiré comme exemple négatif. Le nouvel objectif à optimiser est alors :

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c}^{j\neq 0}(\log\sigma(v_{t+j}\cdot u_t) + \sum_{j'=1}^{k}\mathbb{E}_{w_{j'}\sim p(w)}[\log\sigma(v_{j'}\cdot u_i)]) \tag{2.3.3}$$

où σ est la fonction sigmoïde (voir Figure 2.1). Ce modèle est appelé Skip-Gram with Negative Sampling (SGNS). Il tend à rapprocher les représentations des mots qui cooccurent et éloigner ceux qui ne cooccurent pas. Dans le cas du modèle CBOW, l'objectif est de prédire le mot central de la fenêtre en utilisant ses voisins. Comme pour Skig-Gram, les auteurs définissent la probabilité d'observer le mot w_i en fonction de son contexte comme :

$$p(w_i|(w_{t+j})_{-c\leq j\leq c}) = \frac{e^{u_i\cdot\sum_{-c\leq j\leq c}^{j\neq 0} v_j}}{\sum_{i'=1}^{|V|} e^{u_{i'}\cdot\sum_{-c\leq j\leq c}^{j\neq 0} v_j}} \tag{2.3.4}$$

Le développement est identique ensuite à celui du modèle SGNS. Sur un même corpus, il sera beaucoup plus long d'entraîner SNGS que CBOW. Skip-gram construit des plongements qui auront un meilleur sens sémantique (« chat » et « chien »), tandis que CBOW rapproche plutôt des mots à la morphologie proche « chat » et « chats »). Enfin, CBOW aura tendance a se focaliser sur les mots les plus fréquents, quand SGNS représentera mieux les mots rares et donc la diversité du vocabulaire. Enfin, nous allons rapidement présenter le modèle Glove [PSM14]. Il a surpassé Word2Vec en étant à la fois plus rapide et plus efficace sur un ensemble de sous-tâches d'évaluation. Ces tâches permettent de quantifier la qualité des plongements appris. Elles consistent en ce cas en la détection d'entités nommées, la résolution d'analogie et de similarité de mots. Les auteurs corrigent le problème principal de Word2Vec qui considère de la même façon toutes les cooccurences, peu importe leur rareté. Pour cela, on introduit la matrice de cooccurrence X, tel que X_{ij} dénombre

les cooccurrences entre w_i et w_j. Ils cherchent à minimiser l'objectif suivant :

$$\sum_{i=1}^{|V|}\sum_{j=1}^{|V|}(f(X_{ij})(u_i \cdot v_j + a_i + b_j - \log X_{ij})^2$$
$$f(x) = \begin{cases} \frac{x}{100}^{\alpha} & \text{if } x < 100 \\ 1 & \text{otherwise} \end{cases} \tag{2.3.5}$$

Les nouveaux paramètres b, a et α visent à corriger les différences de cooccurrences. Empiriquement, $\alpha = \frac{3}{4}$ donne les meilleurs résultats. Glove est encore énormément utilisé, de nombreuses représentations pré-entrainées étant disponibles dans plusieurs langues. Il est alors très simple et peu coûteux de les fine-tuner (ou affiner) sur un corpus spécifique en les réentraînant sur quelques epochs. Cependant, deux reproches principaux peuvent être fait à ces méthodes :

— Elles ne prennent pas en compte l'ordre dans lequel les mots apparaissent pour construire les représentations (ce sont des "sacs de mots").

— Elles ne peuvent pas gérer les polysémies, le mot avocat, qu'il soit un légume ou un métier n'aura qu'une seule et unique représentation.

Ainsi, dans la prochaine section nous allons introduire des modèles plus complexes, apprenant des représentations contextualisés des mots.

2.3.2 Plongements contextualisés de mots : CNN, RNN et Attention

Apprendre des plongements contextualisés implique de pouvoir produire une nouvelle représentation à chaque nouveau texte que l'on souhaite encoder. Ceci nécessite une architecture pouvant traiter une séquence de mots, on en distingue trois grandes catégories : les réseaux de neurones convolutifs (ou CNN), les réseaux de neurones récurrents (ou RNN) et les Transformers. Nous allons les détailler successivement ici.

Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs [Lec+98] proviennent du traitement de l'image. En effet, leur structure permet de traiter des données en 2D ou 3D sans que le nombre de paramètres explose, comme ce serait le cas avec les couches denses présentées plus haut. Chaque couche possède un ensemble de filtre de convolution qui seront appliqués sur un tenseur. En transformant un texte en une séquence de représentation vectorielle avec les méthodes présentées au cours de la section précédente, il est possible d'appliquer des filtres de dimension $c \times r$, où c est la taille du contexte qui sera considérée et r la dimension des plongements. En appliquant une fonction d'agrégation (dite de pooling), comme la moyenne ou le maximum, nous obtenons une unique représentation contextualisée de la phrase. A noter qu'il est possible d'appliquer plusieurs de ces étapes successivement. Le principal inconvénient des CNN pour le texte est la taille des filtres, en effet, $r \propto 10^2$ implique beaucoup de paramètres. Egalement, la taille de la fenêtre utilisée limite aussi la capacité à capturer des contextes plus longs.

Réseaux de neurones récurrents

Les réseaux de neurones récurrents introduisent une notion de mémoire qui permet de traiter efficacement des séries temporelles de longueurs variées et dans notre cas, des phrases. Les deux architectures principales sont GRU (Gated Recurrent Unit) [Cho+14] et LSTM (Long Short Term Memory) [HS97] (Figure 2.3). Soit $(x_t)_{0 \leq t \leq T}$ une série de plongements de mots, le réseau fonctionne

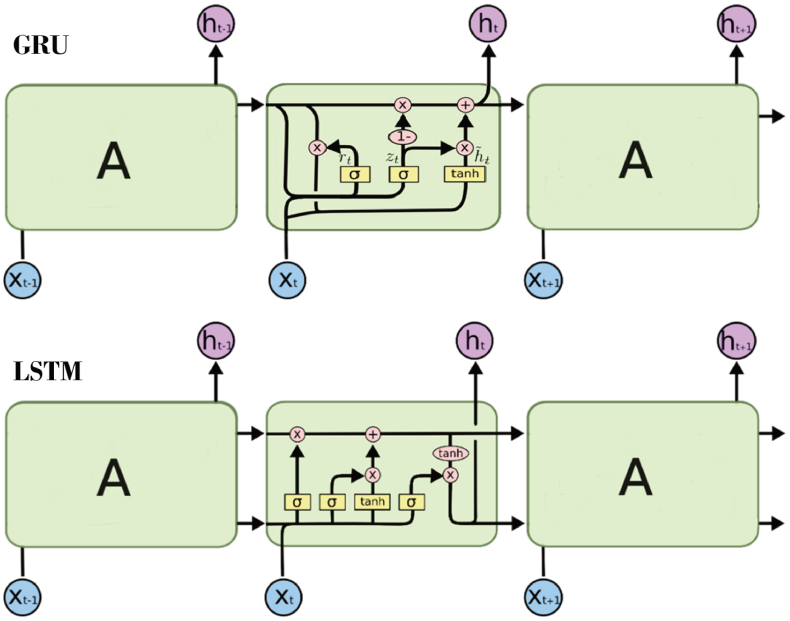


FIGURE 2.3 – Schéma des modèles GRU et LSTM, source *Understanding LSTMs*

en boucle fermée, prenant en entrée x_t et un état caché h_{t-1} fonction des entrées précédentes. Par exemple pour le GRU :

$$\begin{array}{ll} z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z & \text{Vecteur de mise à jour} \\ r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r & \text{Vecteur de remise à zéro} \\ \hat{h}_t = \phi(W_h x_t + U_h(r_t \dot{h}_{t-1}) + b_h & \text{Vecteur d'activation candidat} \\ h_t = (1 - z_t)\dot{h}_{t-1} + z_t \cdot \hat{h}_t & \text{Vecteur de sortie} \end{array} \tag{2.3.6}$$

L'état caché initial h_0 est initialisé à zéro et les fonctions d'activation utilisées sont souvent Tanh ou ReLU. Le LSTM suit le même fonctionnement avec un second état caché supplémentaire, représentant l'état de la cellule. L'introduction des notions de "porte" (z_t et r_t), qui laisse passer ou non le signal, permette d'éviter les problèmes de vanishing gradient des premiers types de RNN. Chaque état caché permet d'obtenir une représentation contextualisé des mots correspondants. Cependant, l'aspect séquentiel implique que la représentation du mot x_t ne dépendra que des mots précédents, ce qui est problématique pour certaines applications en TAL, comme la traduction, ou l'apprentissage de représentation. Pour palier cela, on utilise un bi-RNN. Le premier lira la séquence dans l'ordre et le second dans l'ordre inverse. La concaténation des états cachés issus des deux RNN permet des plongements contextualisés prenant en compte la totalité de la phrase. Le modèle ELMO [Pet+18] est basé sur un Bi-LSTM à deux couches prenant comme entrée les représentations de Glove.

Là où les modèles précédents se concentrent essentiellement sur des tâches de représentation ou de classification, il est aussi possible d'utiliser deux RNN pour réaliser de la génération de texte

(Question-Réponse, traduction automatique, ...). Dans ce cas-là, le premier servira d'encodeur. Il prendra en entrée une phrase, par exemple une question et l'état caché final sera l'entrée du second RNN, le décodeur. En sortie du décodeur, un MLP et une fonction softmax sur le vocabulaire permettent d'obtenir la probabilité que chaque mot débute la réponse. Des symboles spéciaux sont définis pour initialiser le décodeur (<START>) et terminer la phrase (<EOS>).

$$\begin{aligned} &h_T^{enc} = f_{enc}(x_T, h_{T-1}^{enc}) \\ &x_0^{dec} = w_{\text{START}} \\ &y_1^{dec}, h_1^{dec} = f_{dec}(x_0^{dec}, h_T^{enc}) \\ &y_2^{dec}, h_2^{dec} = f_{dec}(y_1^{dec}, h_2^{dec}) \\ &... \end{aligned} \tag{2.3.7}$$

En itérant, il est possible de générer une réponse complète à la question fournie à l'encodeur. De nombreuses approches, que nous ne détaillerons pas, permettent de générer des phrases plus variées et vraisemblables que de simplement choisir le mot avec la probabilité la plus élevée. Il reste des limitations à ces modèles, appelés Seq2seq (pour Sequence to Sequence), qui vont être corrigé par les Transformers [Vas+17]

D'abord, la phrase entière est représentée comme un unique vecteur, alors qu'elle peut présenter différents concepts. Ensuite, le traitement se fait séquentiellement, ce qui peut être coûteux d'un point de vue computationnel. Une nouvelle architecture contourne ces problèmes et a révolutionné la modélisation du langage : les Transformers [Vas+17].

Transformers

Les modèles Seq2seq ont deux principaux défauts, d'abord, le traitement séquentielle de la phrase peut devenir vite coûteux d'un point de vue computationnel. Ensuite, ils produisent une représentation unique pour une phrase entière pouvant présenter différents concepts. Les Transformers proposent d'utiliser une architecture se basant sur le mécanisme d'attention [BCB14] entièrement parallélisable et associant à chaque token une représentation dépendant de l'ensemble de la phrase. Les auteurs calculent tout d'abord un score d'alignement, déterminant la proximité entre l'ensemble des états cachés et la sortie du décoder précédente s_{t-1} :

$$e_{t,i} = f_{att}((h_i)_{1 \leq i \leq T}, s_{t-1}) \tag{2.3.8}$$

f_{att} peut être n'importe quelle fonction mesurant la similarité entre deux vecteurs, même un réseau de neurones. En appliquant une fonction softmax, on peut alors donner un vecteur de contexte unique au décodeur à chaque pas de temps :

$$c_t = \sum_{i=1}^{T} \alpha_{t,i} h_i \text{ où } \alpha_{t,i} = \text{softmax}(e_{t,i}) \tag{2.3.9}$$

Les auteurs l'applique à un Seq2seq constitué de 2 RNN. On peut considérer ce problème comme l'exécution d'une requête s_{t-1} sur une base de données de paires clé-valeur identiques (h_i, h_i). Il est alors possible de généraliser ce mécanisme à n'importe qu'elle architecture Seq2seq, en se passant du côté séquentiel de l'information. Le mécanisme général d'attention se base sur 3 vecteurs de paramètres, les Queries q (correspondant à la requête s_{t-1} et les Keys k_i les Values v_i, qu'on associe à chaque élément de la séquence d'entrée x_i. Le mécanisme d'auto-attention fournit la sortie suivante :

$$\text{attention}(Q, K, V) = \sum_{i=1}^{T} \text{softmax}(q_i \cdot k_i) v_i \tag{2.3.10}$$

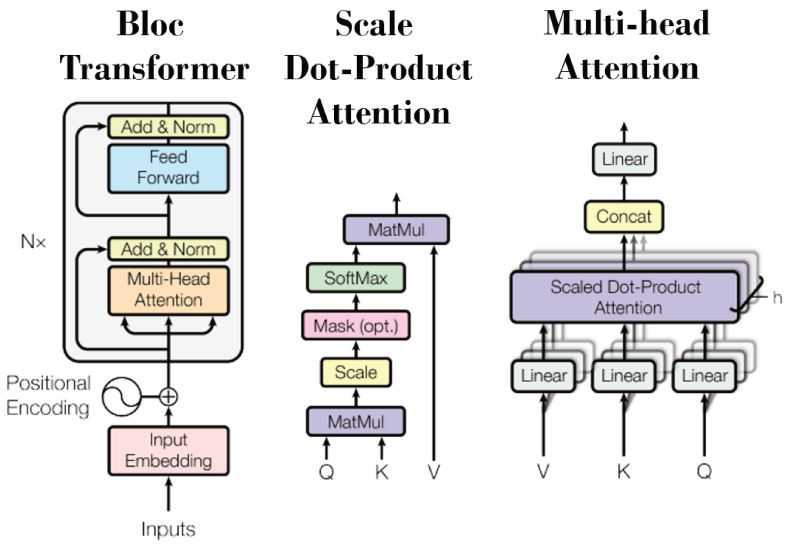


FIGURE 2.4 – Schéma du bloc Transformer, source : Attention Is All You Need [Vas+17]

En particulier, on parle de self-attention quand la requête q est obtenu à partir de l'entrée x_i, par exemple $q_i = W_q x_i$, où W_q est une matrice de paramètres. De la même manière, les Keys k_i et les Values v_i sont paramétrisés par des matrices qui leurs sont propres.

L'architecture Transformer [Vas+17] s'appuie sur ce mécanisme d'attention, où chaque couche possède non pas une, mais plusieurs tête d'attention (W_Q, W_K, W_V). C'est l'attention multi-tête. Les sorties de chaque tête sont additionnées, normalisées puis passées par un MLP avec des connexions résiduelles. Ce bloc Tranformer est répété plusieurs fois, dans l'encodeur et dans le décodeur (voir Figure 2.4). Cette architecture permet de traiter tous types d'entrées matricielles. L'aspect séquentiel des données textuelles est perdu. Pour palier à cela, les auteurs et autrices ajoutent à la matrice initiale des plongements de mots une matrice de position définie comme suit :

$$PE \in \mathbb{R}^{T \times r} = \begin{cases} PE_{i,2j} = \sin(\frac{i}{10000^{\frac{2j}{r}}}) \\ PE_{i,2j+1} = \cos(\frac{i}{10000^{\frac{2j}{r}}}) \end{cases} \tag{2.3.11}$$

Les auteurs et autrices suggèrent que le choix de cette longueur d'onde permet au modèle d'apprendre facilement à gérer les positions relatives des mots devant être liés. Le modèle Transformer a été développé pour la traduction automatique. En 2019, [Dev+19] introduisent BERT, véritable révolution dans l'apprentissage de plongements de mots. Ils utilisent l'encodeur du Transformer, 12 couches contenant chacune 12 têtes d'attention et l'entraînent sur deux tâches de modélisation du langage :

— La prédiction de mots masqués (ou MLM pour Masked Language Modeling).

— La prédiction de paire de phrases (ou NSP pour Next Sentence Prediction).

La première consiste à masquer aléatoirement des mots dans les données d'entraînement, avec une probabilité de 15%. Ensuite, les représentations contextualisées des mots masqués obtenues

en sortie de l'encodeur sont passées dans un MLP et une softmax pour obtenir une probabilité sur l'ensemble du vocabulaire en sortie. La fonction de perte utilisée est l'entropie croisée qui permettra d'entraîner le modèle. A noter que les modèles comme BERT ne tokenize pas les phrases mot à mot, mais utilise à la place le Byte Pair Encoding (BPE) ou sa variante, le WordPiece, qui permet de réduire la taille du vocabulaire et donc les coûts de calcul des fonctions softmax notamment. Très brièvemment, l'idée est de découper les mots en fonction des préfixes et suffixes déjà rencontrés. Par exemple, "Natural Language Processing" sera tokenizé usuellement ("Natural", "Language", "Processing") et ("Natural", "Language", "Process", "##ing") en WordPiece. La seconde tâche d'entraînement vise à faire apprendre à BERT à construire des représentations de phrase capturant la sémantique. Pour cela, il prendra en entrée des paires de phrases et devra prédire si elles sont effectivement consécutives dans le corpus ou non. Un token spécial les sépare ([SEP]) et un autre est ajouté au début ([CLS]), dont la représentation servira, après un MLP et une sigmoïde à évaluer si la paire est positive et négative. La fonction de perte est une simple entropie croisée binaire. Ainsi, le token [CLS] contient une représentation sémantique forte de la phrase entière.

Il est déjà possible de citer de nombreuses améliorations apportées à BERT, avec les modèles RoBERTa [Liu+20b], AlBERT [Lan+20], ou encore des extensions à d'autres langues comme le français [Le+20 ; Mar+20], voir des modèles multilingues. D'autres modèles proposent de corriger l'une des principales limitations de BERT, à savoir la taille limitée des entrées (512 tokens). Notamment, les modèles BigBird [Zah+20] et Longformer [BPC20] réduise la dépendance quadratique de la self-attention en la remplaçant par une combinaison d'attention aléatoires et de fenêtres glissantes plus ou moins dilatées. Ils augmentent ainsi jusqu'à 8 fois la limite de tokens de BERT. Plusieurs modèles proposent des tâches d'entraînement supplémentaire, comme le parsing ou la prédiction d'arbre syntaxique pour une meilleure compréhension du langage [Bai+21 ; Liu+20a]. Egalement, le modèle GPT [Rad+18] qui s'appuie sur le décodeur uni-directionnel du Transformer pour faire de la génération de texte. Les tranformers et BERT ont constitué une véritable révolution pour la modélisation du langage, en produisant des modèles bidirectionnels très performants dans un grand nombre de sous-tâches, grâce à des plongements de mots et de phrases contextualisés. Ils signalent aussi le début d'un changement d'échelle, en permettant la parallélisation du traitement du texte et donc l'accroissement de la taille des modèles et des jeux de données. C'est l'ère des Large Language Models que nous allons présenter dans la sous-section suivante.

2.3.3 Large Language Models et Foundation Models

L'arrivée de BERT a changé le paradigme du TAL. Là où l'objectif était principalement d'entraîner des modèles supervisés sur des tâches spécifiques, il est maintenant de construire des modèles auto-supervisés les plus généralistes possibles. Le crawling du web a permis la création de datasets énormes, comme le corpus Wikipedia ou encore Common Crawl, ne nécessitant pas d'annotations que ce soit pour le MLM ou la NSP. Ont également été créés de multiples benchmarks pour des évaluations de plus en plus variées (compréhension du langage, arithmétique, Q&A, ...). C'est véritablement la taille des modèles et des jeux de données qui permet à ces modèles d'apprendre implicitement la sémantique et la syntaxe du langage humain. Il est alors simple et peu coûteux de fine-tuner ces modèles pré-entraînés sur des corpus annotés et spécialisés de taille plus réduite.

Le fine-tuning consiste à entraîner sur quelques epochs et avec un pas d'apprentissage assez faible un modèle pré-entraîné, afin de bénéficier des connaissances généralistes qu'il a pu accumuler tout en le spécialisant sur une application donnée. L'augmentation des performances des modèles a augmenté avec leur taille, aussi permise par l'amélioration des ressources computationnelles dispo-

nibles et des moyens financiers mis en jeu. Ainsi, on parle de Large Language Models à partir d'un milliard de paramètres, là où BERT en posséde 340M.

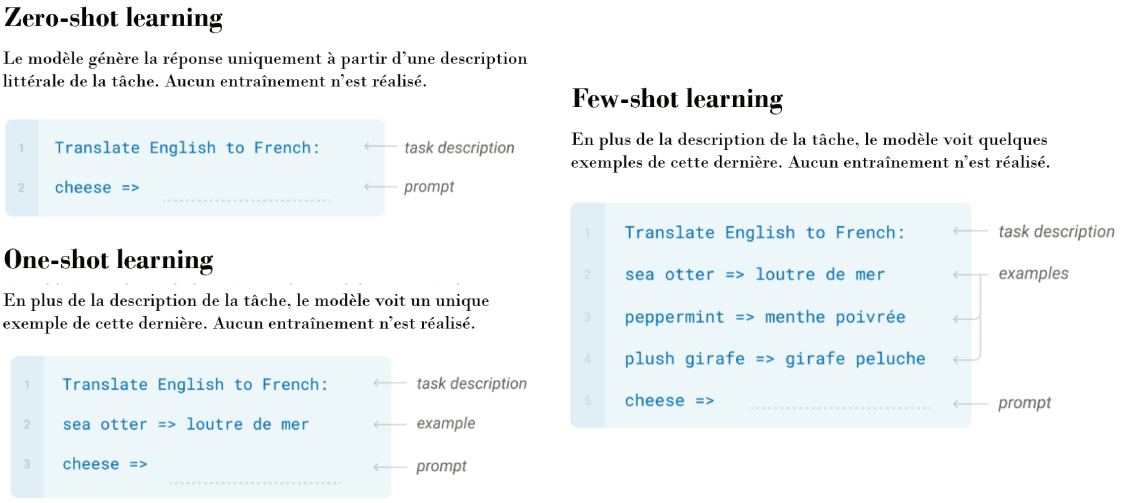


FIGURE 2.5 – Illustration des différentes méthodes d'évaluation des LLM génératifs.

En 2019, l'article *Language Models are Unsupervised Multitask Learners* [Rad+19] correspond à la publication du modèle GPT-2. Grand frère du modèle génératif GPT (plus de 10 fois le nombre de paramètres), les auteurs montrent que même sans étape de fine-tuning, les modèles de langue obtiennent de bons résultats sur des benchmarks complexes de TAL, comme la traduction automatique, le résumé de texte ou la compréhension, uniquement à partir d'une description littérale de la tâche attendue. L'idée est d'écarter la phase d'affinage et de faire passer la spécialisation au moment de l'inférence, en fournissant comme entrée une explication de la tâche à accomplir. C'est le zero-shot learning. GPT3 [Bro+20] introduit également le one-shot learning et le few-shot learning (voir Figure 2.5).

Cette prolifération de modèles à la très forte capacité de généralisation a fait naître la notion de *Foundation Models* [Bom+21], définit par un grand nombre de chercheurs en intelligence artificielle. Ce sont des modèles de très grandes tailles :

— Pré-entraînés sur divers datasets (textes, images, ...)

— auto-supervisés (entraînés sans annotation quelconque)

— Créant des représentations des données généralistes pour de nombreuses sous-tâches

Ils constituent la première brique de multiples applications, ce qui posent plusieurs questions. Le côté auto-supervisé implique la reproduction des biais (sexiste, raciste, ...) naturellement présents dans les jeux de données d'entraînement non filtrés. Ils posent aussi la question des droits d'auteurices, dont les productions sont massivement utilisées par les LLMs sans véritable transparence. L'accès aux modèles et aux ressources permettant de les utiliser devient également de plus en plus complexes. Ce sont de nouveaux axes de recherche, vers plus de sobriété et de fairness qui s'ouvrent.

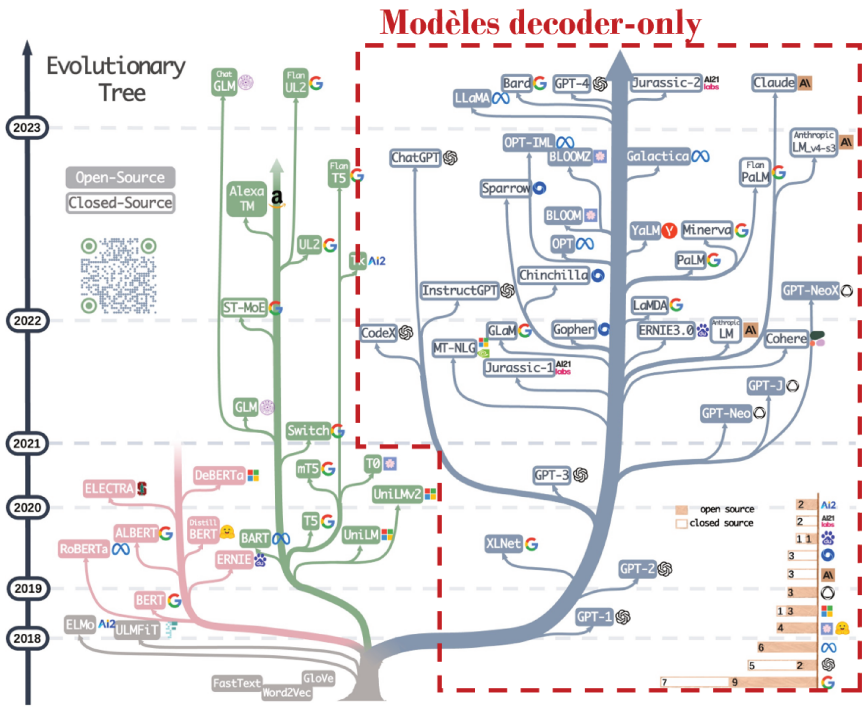


FIGURE 2.6 – Cartographie des différents LLMs publiés. Source https://github.com/Mooler0410/LLMsPracticalGuide

Une cartographie des LLMs est montrée Figure 2.6. Le bloc Transformer en constitue toujours la brique essentielle. Quelques modifications interviennent sur les fonctions de normalisation et leur position, l'encoding des positions ou encore le choix des fonctions d'activation. Très récemment, l'ajout de méthodes de Reinforcement Learning à partir de feedback humain à permis une grande avancée. Les principales innovations sont techniques et dans le passage à l'échelle [Zha+23]. BLOOM a par exemple été entraîné pendant plus de 100 jours. Une très grande partie des modèles les plus récents est consacré à la génération et est donc constitué uniquement d'un décodeur. Les LLMs décodeurs les plus récents n'utilisent plus uniquement le MLM comme tâche d'entraînement. L'Autoregressive Blank Infilling par exemple vise à recréer des pans entiers de phrase masqués. A cela s'ajoute des tâches supervisées de détection de similarité, de parsing de dépendances, ...

A l'opposée, les encodeurs sont entraînés pour produire des représentations significatives et pouvant être utilisées ensuite pour de nombreuses tâches, notamment la génération. Ils arrivent à capter des notions de linguistique et de sémantique complexe afin de modéliser le langage au mieux. Pour cela, dans notre cadre d'application, nous nous limiterons aux encodeurs seuls. En effet, nous avons présenté un ensemble de méthodes permettant de représenter les mots comme des vecteurs (contextualisés ou non), jusqu'aux grand modèles de langues les plus récents.

Et c'est sur ces représentations puissantes que les principales méthodes d'apprentissage de plongements de documents et d'auteurices que nous allons introduire en section suivante, s'appuient.

2.4 Apprentissage de plongements de documents et auteurices

Nous avons vu jusqu'à maintenant une cartographie des méthodes d'apprentissage de plongements de mots, voir de phrase. Dans le cadre de traitement de grands corpus, il est intéressant d'agréger ces représentations à l'échelle du document, puis de l'auteurice, afin de pouvoir les utiliser dans des applications comme la classification, la recommandation ou encore l'attribution d'auteurices. Ainsi, nous allons détailler ici les différentes méthodes d'agrégation proposée dans la littérature. Nous nous limiterons ici aux approches à base de réseaux de neurones visant à obtenir des représentations en faible dimension. Nous présenterons les méthodes découlant de Word2Vec, puis celles s'appuyant sur des fonctions d'agrégation de plongements de mots pré-entrainés. Enfin, nous nous intéresserons au modèles apprenant des plongements d'auteurices.

2.4.1 Méthodes de Word2Vec à Doc2Vec

Dans la continuité de Word2Vec, les auteurs ont proposé Doc2Vec [LM14], modèle s'appuyant sur un fonctionnement similaire tout en projetant en plus les documents dans l'espace de plongements des mots. Encore une fois, ils proposent deux modèles distincts. A la matrice de plongements des mots à apprendre $W_e \in \mathbb{R}^{|V|\times r}$, s'ajoute celle de plongement des documents $W_d \in \mathbb{R}^{|\mathcal{C}|\times r}$, où $\mathcal{C}$ est le corpus de documents. La première méthode, nommée PV-DM pour Paragraph Vector-Distributed Memory étend le framework CBOW. Chaque document est associé à un identifiant unique, considéré comme un token qui cooccure avec tous les mots qui le composent. L'objectif est alors de prédire le mot au suivant à partir du token de son document et d'une fenêtre de 3 mots. La seconde méthode, nommée PV-DBOW, étend le framework Skig-Gram de la même façon. A partir du token du document, l'objectif est de prédire les mots formant une fenêtre contextuelle. Les deux approches sont illustrées Figure 2.7. Doc2Vec est plus performant que les approches historiques,

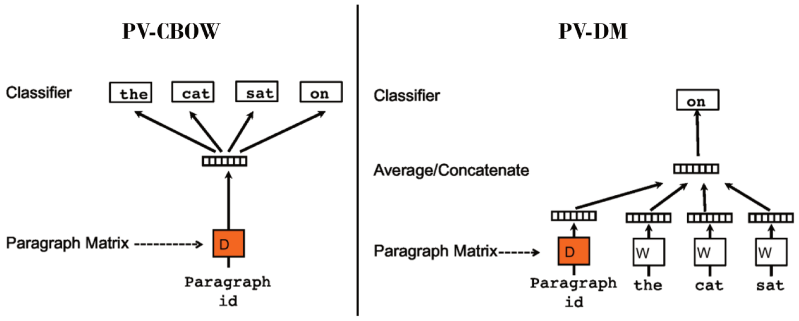


FIGURE 2.7 – Schéma des méthodes Doc2Vec. Source [LM14]

mais non inductif. Cela signifie qu'il est impossible de produire une représentation d'un nouveau document sans devoir réentraîner le modèle de zéro : c'est un problème majeur. C'est pourquoi, dans la section suivante, nous présenterons des modèles d'agrégation de plongements de mots plus complexes.

2.4.2 Méthodes récentes d'agrégation

Pour formaliser le problème, on note $\mathcal{C}$ un corpus de documents $(d_i)1 \leq i \leq |\mathcal{C}|$. Chaque document i est une suite de longueur $|d_i|$ mots $(w_1^i, w_2^i, ..., w_{|d_i|}^i)$. Ici, pour plus de simplicité, on notera de la même manière les mots et leurs représentations vectorielles. Notre objectif est de déterminer une fonction d'agrégation f_{agg}^{θ} tel que le document d_i ait pour représentation $z_{d_i} = f_{agg}^{\theta}(w_1^i, w_2^i, ..., w_{|d_i|}^i)$. Des documents proches sémantiquement ou syntactiquement doivent alors être proches dans l'espace de représentation. Par exemple, la fonction d'agrégation la plus simple revient à moyenner les représentations des mots. Nous allons présenter ici essentiellement des réseaux de neurones et les objectifs qui leurs sont associés. La méthode Skip-Thought [Kir+15] est une des rares méthodes auto-supervisée. Elle s'appuie sur une architecture Seq2seq à base de GRU avec deux décodeurs distincts. En créant des triplets de phrase successives à partir du corpus, l'objectif est que chaque décodeur génère, respectivement, la phrase précédant et la phrase suivant celle qui se trouve au centre, donnée en entrée de l'encodeur. Après la phase d'entraînement, les représentations des documents sont obtenues en sorties de l'encodeur. Un simple classifieur linéaire sur les représentations de documents issues de Skip-Thought suffit à être efficace sur l'ensemble des métriques d'évaluation de compréhension du langage. Il est uniquement dépassé par AdaSent [ZLP15], spécifiquement entraîné sur chaque tâche et à l'architecture plus complexe. Cette dernière est un ensemble de couches convolutionnelles avec des connexions résiduelles à chaque niveau qui permettent de conserver les représentations intermédiaires.

On peut noter que BERT, via la représentation du token [CLS] propose déjà une représentation des documents qu'il traite, de façon également auto-supervisée. Il en va de même pour les LLMs plus récents (GLM, T5, UniLM, XLNet), qui s'appuient tous sur l'architecture Transformer et varient surtout par les tâches et les corpus d'entraînement utilisés. Cependant la plupart des méthodes de représentation de documents récentes associent l'apprentissage supervisé à l'auto-supervisé, sur des datasets annotés, notamment le corpus Stanford Natural Language Inference (SNLI) [Bow+15]. Ce jeu de données contient environ 600 000 paires de phrases ainsi qu'un label parmi 'implication', 'contradiction' ou 'neutre'. Les auteurs du modèle InferSENT [Con+17] testent différentes

fonctions d'agrégation sur SNLI (BiLSTM, type Adasent, self-attention) pour l'apprentissage de représentation de phrase. Ils les fine-tunent ensuite sur les benchmarks usuels pour prouver l'intérêt de l'entraînement supervisé. La meilleure architecture étant dans leur cas le BiLSTM avec max-pooling.

Le modèle USE (Universal Sentence Encodeur) [Cer+18] procède de la même façon en proposant deux architectures, une à base de Transformer et l'autre utilisant un Deep Averaging Network (DAN) [Iyy+15a]. Le DAN consiste simplement en un MLP auquel on passe la moyenne des représentations de la phrase d'entrée. Il a l'avantage d'être rapide à entraîner et de n'avoir aucune limite de taille d'entrée, là où BERT est contraint à 512 tokens par exemple. Pour autant, il obtient des résultats très proches du modèle plus complexe à base de Transformers sans capturer l'aspect séquentiel des données.

Toutes les méthodes évoquées ici utilisent comme base les plongements de mots non-contextualisés de Glove (voir de Word2Vec) qu'ils fine-tunent au cours de l'entraînement de leur modèle. Le modèle SentenceBERT [RG19] lui spécialise directement le modèle BERT sur SNLI comme InferSENT en montrant qu'il améliore ainsi la qualité sémantique des représentations [CLS].

Nous avons détaillé au cours de cette section un ensemble non-exhaustif de méthodes d'apprentissage de représentations de documents, bien souvent limité à une phrase, rarement plus. Ce sont principalement des architectures usuelles (RNN, CNN, Transformer, DAN) combinés à un ensemble de datasets annotés qui permettent d'obtenir les meilleurs résultats en terme de modélisation de la sémantique. Se pose maintenant la question de l'apprentissage de représentations d'auteurices, qui va être détaillé dans la section suivante

2.4.3 Ajout de plongements d'auteurices

En reprenant les notations de la section précédente, l'objectif est, pour l'auteurice a de notre corpus ayant écrit un ensemble de documents $(d_i^a)_{1\leq i\leq n_a}$ de taille n_a, de déterminer la fonction $z_a = f_{agg}^{\theta}((d_i^a)_{1\leq i\leq n_a})$, qui lui associera la représentation $z_a \in \mathbb{R}^r$ en faible dimension. En réalité, d'autres données que la seule production textuelle peuvent intervenir (métadonnées, graphe de liens...), mais nous nous limiterons à ce cadre dans cette section. De même, nous n'évoquerons que les méthodes d'apprentissage profond bien qu'ayant connaissance des approches de type Author Topic Model [Ros+04] par exemple.

Dans la continuité des modèles Word2Vec et Doc2Vec est proposé le modèle Author2Vec [Jaw+16]. Il est constitué de deux modèles : l'un se focalise sur la représentation du graphe de coauteurice et l'autre sur la production textuelle. Nous présenterons ici le second, le Content-Info Model. A partir du plongement z_a associé à l'auteurice a et d'un document d et son plongement z_d, le modèle doit prédire si a a effectivement écrit d ou non. Pour chaque paire auteurice-document existante, il en sélectionne une négative. A l'aide d'un réseau de neurones comme suit :

$$\begin{aligned} h_c^{\times} &= z_a \cdot z_d \\ h_c^{+} &= z_a + z_d \\ h_c &= \tanh(W_c^{\times} h_c^{\times} + W_c^{+} h_c^{+} + b_c) \end{aligned} \tag{2.4.1}$$

Il optimise ensuite la fonction de perte suivante :

$$\mathcal{L} = \text{softmax}(U h_c + b) \tag{2.4.2}$$

Les paramètres du modèle consistent en l'ensemble des matrices et vecteurs de biais, ainsi que les plongements des auteurices et des documents. Ces derniers sont initialisés avec Doc2Vec. Les

représentations des documents et des mots sont dans le même espace, mais cette méthode est entièrement non inductive. Il est ainsi impossible d'obtenir la représentation d'un nouveau document ou auteurice. Un autre modèle dérivé de Doc2Vec est proposé, nommé Usr2Vec [Ami+17]. C'est un dérivé de travaux précédents des mêmes auteurs et autrices [Ami+16]. Ils souhaitent maximiser la probabilité conditionnelle suivante :

$$p((d_i^a)_{1\leq i\leq n_a}|a) \propto \sum_{d\in\mathcal{C}} \sum_{w_j\in d} \log p(w_j|a_i) \tag{2.4.3}$$

Comme dans le cas de Word2Vec, estimer cette quantité est coûteux. Il est plus simple de passer alors par du negative sampling et de minimiser l'objectif (de type Hinge-loss) suivant, où j' correspond à l'exemple négatif (à savoir un mot non-écrit par l'auteurice a) :

$$\mathcal{L}(w_j, a) = \sum_{w_{j'}\in V} \max(0, 1 - w_j \cdot a + w_{j'} \cdot a) \tag{2.4.4}$$

Ce modèle est équivalent à la variante PV-DBOW de Doc2Vec où les auteurices sont vus comme des documents. Une des différences est qu'ici les plongements des mots sont pré-entrainés. Le modèle est évalué sur la prédiction de la santé mentale d'utilisateurs de Twitter. Ici, aucune représentation de documents n'est apprise et le modèle est encore une fois non inductif. Le modèle VADE (Variational Author and Document Embedding) [Gou21] corrige cela en proposant un modèle apprenant à la fois des représentations d'auteurices et de documents et inductif pour ces dernières. Un avantage supplémentaire est qu'il propose des représentations gaussiennes, la variance permettant d'évaluer la diversité dans la production de chaque auteurice par exemple. VADE s'appuie sur le framework VIB (Variational Information Bottleneck) qui sera détaillé dans une section suivante et propose différents réseaux de neurones comme encodeur de documents (DAN, Attention) sur SentenceBERT et USE. Les tâches d'évaluation sont l'identification d'auteurices à partir d'un document et le clustering d'auteurices et de documents. C'est l'un des rares modèles à utiliser les modèles pré-entrainés récents, sans les fine-tuner cependant.

Le modèle Author2Vec$_2$ [Wu+20] propose lui d'affiner un modèle de langue. Les auteurs et autrices agrègent l'ensemble des productions de chaque auteurice issues de BERT à l'aide d'un Bi-GRU. Ils obtiennent ensuite les représentations des auteurices après une couche de K-sparse encoding (couche linéaire où seuls les K neurones les plus actifs sont conservés) et d'un MLP. Le modèle est entraîné et évalué sur une tâche de classification de genre, une de détection de dépression chez des utilisateurs de Reddit et une de classification de personnalité de la nomenclature MBTI (Myers Briggs Type Indicator, outil d'évaluation psychologique à la validité très discuté dans la communauté scientifique). A noter qu'il est à la fois inductif pour les auteurices et les documents. Cependant, rien ne garantit que chaque auteurice soit bien séparé de documents qu'il n'a pas écrit, ce qui peut-être rédhibitoire pour des applications de type attribution d'auteurices. De la même façon, le modèle UAR (Universal Authorship Representation) [Riv+21] agrège un ensemble de représentations de phrases issus de SentenceBERT par auto-attention et pooling afin de construire la représentation de l'auteurice. Ce modèle est fine-tuné sur une tâche d'attribution d'auteurices.

Enfin, certains modèles proposent d'apprendre des représentations d'auteurices dynamiques, suivant l'évolution des auteurices dans le temps. Le modèle DAR (Dynamic Author Representation) [DLD19] et le modèle [Gou+22a]. Ils seront détaillés dans un chapitre ultérieur.

Une comparaison des méthodes abordées ici est proposée Table 2.1. Nous avons présenté ici un ensemble de méthodes de plongement d'auteurices. Nous pouvons remarqué qu'assez peu sont

Modèle	Tâche d'entraînement	Architecture	Avantages	Inconvénients
Content-Info	Attribution d'auteurices	MLP	Rapide	Non-inductif pour les documents
User2Vec	Classification d'auteurices	Doc2Vec	Rapide	Pas de représentation de documents Entraînement supervisé
VADE	Attribution d'auteurices	encodeur + MLP	Modèles pré-entraînés récents Modélisation gaussienne	Ne finetune pas l'encodeur
Author2Vec$_2$	Classification d'auteurices	BERT + GRU	Fine-tune un LLM Entièremment inductif	Entraînement supervisé
UAR	Attribution d'auteurices	SBERT + Attention	Fine-tune un LLM Entièremment inductif	Documents et auteurices dans des espaces différents
DAR	Génération de textes	LSTM + MLP	Représentation dynamique	Coûteux pour les documents longs
DGEA	Attribution d'auteurices	RNN + MLP	Représentation dynamique Modélisation gaussienne	Ne finetune pas l'encodeur

TABLE 2.1 – Tableau récapitulatif des méthodes de représentation d'auteurices abordées dans ce chapitre.

inductives, au moins pour les documents. Seules deux choisissent de spécialiser un modèle de langue récents pour bénéficier de ses capacités de compréhension du langage. D'un point de vue plus général, il est difficile de savoir ce sur quoi se focalise principalement les modèles de représentation pour construire les plongements. Les méthodes d'évaluation se limitent souvent à l'attribution d'auteurices et à la classification ou au clustering sur des jeux de données très thématiques (santé mentale, domaine de publication, ...). Ces tâches peuvent être réalisées en se basant essentiellement sur la sémantique. Bien que certains modèles disent capturer le style littéraire, c'est quelque chose qui est rarement évalué, ce qui peut poser problème dans un cadre d'analyse littéraire, comme nous le verrons à l'aide de nos contributions dans les chapitres 3 et 4.

2.5 Conclusion

Ce chapitre a permis d'évoquer brièvemment les bases théoriques indispensables à la bonne compréhension des modèles de deep learning dont nous aurons besoin dans la suite du manuscrit. Nous avons pu revoir les fondements de ce sous-domaine de l'intelligence artificielle, du neurone élémentaire aux architectures les plus complexes en passant par leur entraînement et optimisation. Nous avons pu dans un second temps évoquer l'utilisation de ces réseaux de neurones pour l'apprentissage de représentation à différentes niveaux du langage : de l'échelle la plus simple, le mot, avec des méthodes pionnières comme Word2Vec et Glove, jusqu'au plongement contextualisé des grands modèles de langues les plus récents. Enfin, nous avons changé d'échelle en évoquant différents modèles permettant d'apréhender des corpus en représentant successivement des documents, puis des auteurices.

Dans le prochain chapitre nous nous intéresserons à la définition du style littéraire en TAL. Nous détaillerons un ensemble de marqueurs qui tentent de le définir et si les récentes avancées en apprentissage profond permettent de mieux l'appréhender. Nous présenterons pour cela une première contribution qui vise à mesurer la capture du style littéraire par des modèles de représentation.

Chapitre 3

Le style littéraire en Traitement Automatique de la Langue

3.1 Introduction

Dans cette section, nous allons nous focaliser sur le style littéraire, en présentant sa définition usuelle en informatique et en linguistique computationnelle et les méthodes permettant de l'appréhender, des plus anciennes au plus récentes. Nous ferons ensuite le lien avec les modèles de représentations de la langue utilisés en TAL et présentés en première partie. Enfin, nous proposerons notre première contribution visant à quantifier la capture du style par les modèles de plongements de documents et d'auteurices [TGV21].

3.2 Style littéraire et attribution d'auteurices

En littérature, le style littéraire est souvent défini comme les particularités d'écriture d'un auteur ou d'une autrice, l'écart dans sa production par rapport à une norme linguistique. La complexité et la perpétuelle évolution du langage fait qu'il est difficile de quantifier cet écart. Cependant, dès le 19ème siècle Mendenhall [Men87] a essayé de caractériser les spécificités de l'écriture de certains auteurs ou certaines autrices en étudiant les fréquences de certains mots et leurs longueurs. Ce sont les prémices de la stylométrie, branche de l'informatique cherchant à identifier le style écrit propre à leur auteurice. La définition retenue en linguistique computationnelle est le plus souvent celle proposée par Karlgren [Kar04] :

« *A style is a consistent and distinguishable tendency to make some [of these] linguistic choices. Style is, on a surface level, very obviously detectable as the choice between items in a vocabulary, between types of syntactical constructions, between the various ways a text can be woven from the material it is made of.* » [1]

1. Le style est la tendance cohérente et distincte à prendre certains décisions linguistiques. À première vue, le style est très clairement détectable comme l'ensemble des choix faits entre les éléments d'un vocabulaire, entre les types de constructions syntaxiques, entre les différentes façons dont un texte peut être érigé à partir du matériau dont il est bati.

Style écrit	**Extrait**
Lipogramme	*Voici. Au stop, l'autobus stoppa. Y monta un zazou au cou trop long, qui avait sur son caillou un galurin au ruban mou. Il s'attaqua aux panards d'un quidam dont arpions, cors, durillons sont avachis du coup ; puis il bondit sur un banc et s'assoit sur un strapontin où nul n'y figurait.*
Litotes	*Nous étions quelques-uns à nous déplacer de conserve. Un jeune homme, qui n'avait pas l'air très intelligent, parla quelques instants avec un monsieur qui se trouvait à côté de lui, puis il alla s'asseoir.*
Métaphoriquement	*Au centre du jour, jeté dans le tas des sardines voyageuses d'un coléoptère à grosse carapace blanche, un poulet au grand cou déplumé harangua soudain l'une, paisible, d'entre elles et son langage se déploya dans les airs, humide d'une protestation. Puis attiré par un vide, l'oisillon s'y précipita.*
Surprises	*Ce que nous étions serrés sur cette plate-forme d'autobus ! Et ce que ce garçon pouvait avoir l'air bête et ridicule ! Et que fait-il ? Ne le voilà-t-il pas qui se met à vouloir se quereller avec un bonhomme qui - prétendait-il ! ce damoiseau ! - le bousculait ! Et ensuite il ne trouve rien de mieux à faire que d'aller vite occuper une place laissée libre ! Au lieu de la laisser à une dame !*

TABLE 3.1 – Extrait d'*Exercices de style* de Raymond Queneau.

On résume souvent cela à l'ensemble des choix d'écriture faits ne contenant pas d'information sémantique. En linguistique, la sémantique représente le fond, le sens et le signification des mots, par opposition à la syntaxe, qui représentera la forme. Un très bon exemple illustratif est l'ouvrage *Exercices de style*, où Raymond Queneau va écrire de 99 façons différentes la même histoire. Le tableau 3.1 en propose des extraits, chaque paragraphe correspondant au même passage de l'histoire. La distinction entre sémantique et style littéraire est claire et saute encore plus aux yeux, notamment pour l'extrait métaphorique. L'hypothèse selon laquelle les marqueurs du style littéraire contiennent peu voir pas d'information sémantique est assez forte, mais très utilisée en linguistique computationnelle. C'est ce qui pourrait la distinguer de l'approche purement littéraire.

Une seconde hypothèse très importante est celle de la persistance du style. Un auteur ou une autrice, au cours de sa vie et dans ses écrits aura tendance à conserver une cohérence dans sa manière d'écrire. Ainsi, l'objectif premier de la stylométrie est l'attribution d'auteurices, à savoir prédire l'auteurice d'un document parmi un ensemble d'auteurices connus. Une méthode utilisant uniquement des marqueurs non-sémantiques et pouvant attribuer à coup sûr un document à son auteurice capturerait théoriquement parfaitement le style littéraire. Les applications qui découlent de l'attribution d'auteurices sont nombreuses : la vérification de paternité, l'attribution de paragraphe dans le cas d'oeuvre composite, l'analyse de messages malveillants, le profiling d'auteurices, ... Les jeux de données PAN rassemblent un ensemble de tâches d'évaluation de méthode de stylométrie. Dans un cadre littéraire, on peut citer les nombreux travaux visant à confirmer que Molière avait bel et bien écrit ses pièces (que des rumeurs attribuaient à Corneille). Nous allons maintenant nous focaliser sur les marqueurs qui sont fréquemment utilisés en stylométrie, des plus simples au plus complexes, présentés dans de nombreux surveys [Hol94 ; Sta09 ; Nea+17].

3.2.1 Marqueurs de fréquences simples

Historiquement, les premiers marqueurs utilisés étaient aussi les plus simples à extraire, que ce soit avec ou sans ordinateur. Ce sont, par exemple, la longueur moyenne des mots, des phrases, la fréquence de chaque lettre, nombre, majuscule, ponctuation, le nombre total de mots ... Ils sont applicables à tout type de langage ou de corpus, ce qui en fait un avantage non-négligeable. Ces marqueurs sont souvent agrégés en fréquence par phrase, afin de pouvoir traiter également des textes de longueur variés. Il est également possible de compter la moyenne de syllabes par mots, ou la fréquence des mots courts ou longs, mais ces indicateurs ne sont plus indépendant de la langue du corpus. De nombreuses méthodes en stylométrie utilisent également les fréquences de n-grams de caractères (généralement $n = 3$), car cela permet de capturer à la fois la sémantique et le style [Sta09]. En contre-partie, la dimension des vecteurs à considérer augmente fortement et il est nécessaire souvent d'y combiner des outils de sélection de features.

Des marqueurs statistiques plus complexes ont ensuite été proposé, permettant d'évaluer notamment la richesse et la diversité du vocabulaire, entre autres :

— L'entropie de Shannon ($H = -\sum_{k=1}^{|V_d|} f_k \log_2(f_k)$ où $|V_d|$ est la taille du vocabulaire dans le document i et f_k la fréquence du mot k

— L'hapax legomena (et le dislegomena), la proportion de mots n'apparaissant qu'une fois (respectivement deux fois) dans un texte

— L'indice de diversité de Simpson $D_s = 1 - \frac{\sum_{k=1}^{|V_d|} f_k(1-f_k)}{|d|(|d|-1)}$, où $|d|$ est la longueur du document i.

Des indicateurs de lisibilité et de complexité existent aussi, souvent spécifiques à l'anglais. Ils sont en général calculés à partir de formules issues de la linguistique, ou d'une base de mots spécifiques. Par exemple, le test de lisibilité de Flesch-Kincaid fournit un score correspondant en théorie au niveau scolaire américain nécessaire pour comprendre un texte :

$$FK_{score} = 0.39 \times \frac{\text{nb mots totaux}}{\text{nb de phrase}} + 11.80 \times \frac{\text{nb syllabes total}}{\text{nb mots totaux}} - 15.59 \tag{3.2.1}$$

En a été dérivé le score de lisibilité de Dale-Chall variant entre 0 (simple) et 10 (complexe) :

$$DC_{score} = 15.79 \times \text{fréquence de mots difficiles} + 0.0496 \times \frac{\text{nb mots totaux}}{\text{nb phrases}} \tag{3.2.2}$$

Ici, une liste de 763 mots courants, compréhensibles par des élèves de 10 ans, a été définie en amont pour évaluer les mots difficiles. De la même manière, les fréquences d'utilisation de mots outils sont aussi très souvent utilisés. Pour l'ensemble des types de marqueurs présentés jusqu'ici, il est uniquement nécessaire d'avoir un outil permettant de séparer les mots (tokenizer) et les phrases (sentence-tokenizer). En général, ce sont des modèles à base de règles.

De nombreux travaux utilisent également des caractéristiques sémantiques, comme des bi-grams ou tri-grams de mots, ou encore des fréquences de mots issus d'un vocabulaire spécifique pour des applications à des domaines précis. Nous ne détaillerons pas ces marqueurs car ils relèvent plus de l'attribution d'auteurices que de l'étude du style littéraire à proprement parler, d'où la prise en compte de la sémantique. De même, nous nous focalisons uniquement sur le texte et donc ne traiterons pas les marqueurs structurels pour des applications à du code ou à du html par exemple.

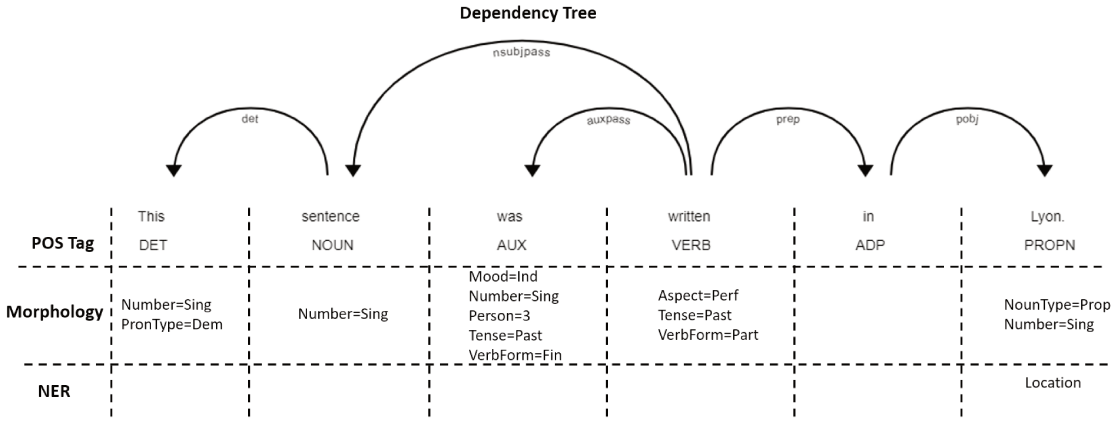


FIGURE 3.1 – Exemple de traitements possible du texte avec Spacy

3.2.2 Avancées permises par le machine learning

Le développement de l'apprentissage automatique et notamment dans le traitement de la langue a permis une analyse plus poussée des structures syntaxiques des documents. L'essentiel des marqueurs cités plus haut sont lexicaux ou des fréquences de symboles. L'idée ici est d'étiqueter les mots par leur fonction syntaxique. On distingue plusieurs type de tâches. Le Part-of-Speech Tagging (POS Tagging) consiste simplement à déterminer la fonction grammaticale d'un mot (verbe, nom, déterminant, ...). L'analyse morphosyntaxique vise à associer à un mot son genre, son nombre, son temps s'il s'agit d'un verbe, sa forme ... L'analyse des dépendances a pour but de reconstruire l'arbre syntaxique d'une phrase et les liens entre chaque mot. Enfin, la détection d'entité nommée (NER pour Named Entity Recognition) consiste à catégoriser des groupes nominaux comme des noms de personnes, d'organisation, de lieux ou des quantités (temps, monnaie, ...). Le fait qu'un mot, en fonction de son contexte puisse avoir des fonctions différentes explique la difficulté de ces tâches. Les premiers modèles de POS tagging, au milieu des années 60, étaient essentiellement à base de règles de co-occurence. Puis, les modèles de Markov cachés, jusqu'à l'utilisation d'arbres de décision et de SVM. Les modèles les plus récents sont basés sur des transformers ou des réseaux de neurones récurrents. De nombreux benchmarks permettent de suivre l'évolution et les progrés des modèles proposés (voir Stanford NLP Taggers ou POS Tagging benchmarks par exemple). La figure 3.1 montre un exemple de ce qu'il est possible d'obtenir comme analyse à partir de la librairie python Spacy.

Ces modèles sont utilisés en stylométrie soit comme de nouveaux marqueurs de fréquence syntaxique, ou comme outils pour modifier le texte en amont de l'analyse. Par exemple, [SW18] remplace différentes proportions des mots d'un corpus par leur tag avant de s'évaluer en attribution d'auteurices. De la même façon [Sta17 ; Sta18] masque les mots considérés comme sémantiquement déterminants avec un token spécial pour améliorer les performances lorsque le corpus est multi-domaines. Cette méthode a ensuité été amélioré en utilisant les marqueurs morphologiques des mots en question. Combinés principalement à des arbres de décision ou des SVM, performant avec les données parcimonieuses, parfois aves des réseaux de neurones, ces marqueurs obtiennent

de bons résultats en attribution d'auteurices sur les différentes compétitions PAN et permettent a priori de caractériser en partie le style littéraire. Seuls quelques modèles tentent de traiter directement le texte brut avec des réseaux de neurones profonds pour extraire le style et en apprendre une représentation, comme nous allons le voir dans la section suivante.

3.2.3 Apprentissage de représentation et stylométrie

Il est possible de distinguer deux types d'approche en apprentissage de représentation du style littéraire. La première est celle habituelle utilisant la tâche d'attribution d'auteurices pour entraîner le modèle. Par exemple [GSR19] utilise des RNNs à cette fin là. [BZM18] s'appuie sur des CNN pour de l'attribution multi-auteurices cette-fois ci. D'autres variations consistent à prédire si deux (ou trois) documents ont été écrits par le même auteur ou la même autrice [Din+16 ; Jas+18], c'est la vérification d'auteurices. Si ces modèles obtiennent de bon résultats sur de nombreux jeux de données, issus des compétitions PAN ou non, ils souffrent d'une perte d'information dûe à l'utilisation de réseaux de neurones profonds. Il est très difficile de savoir si le modèle s'appuie plus sur l'information sémantique ou stylistique pour produire ses résultats, là où l'utilisation de marqueurs permet de s'affranchir de l'une ou de l'autre.

Ainsi, la seconde approche souhaitant s'extraire de cette problématique propose un apprentissage par un processus de généralisation. Les auteurices des jeux de données d'entraînement et de test sont alors totalement différents et parfois provenant de domaines également différents. [Bou+19] utilise une combinaison des jeux de données PAN, Amazon Reviews et MLPA. [Hay+20] affine DistilBERT sur un grand corpus issu du web pour ensuite s'évaluer sur un jeu de données neuf et surpasser de nombreuses baselines en attribution d'auteurices. Mais de la même façon, les seules méthodes d'évaluation sont l'attribution d'auteurices et dans certains cas la classification d'auteurices sur des classes thématiques. La capacité de généralisation à de nouveaux auteurices ou jeux de données n'assure pas que l'essentiel de l'information utilisée par les modèles soit de nature stylistique.

3.3 Problématique de l'évaluation de la capture du style littéraire

L'utilisation de réseaux de neurones et de grands modèles de langues permet d'améliorer grandement les résultats en stylométrie tout en produisant des représentations des documents et/ou d'auteurices utilisables pour de nombreuses sous-tâches. Cependant, ces modèles sont des boîtes noires, très peu interprétables et il est difficile d'évaluer ce sur quoi ils portent leur attention. L'attribution d'auteurices peut se faire par la sémantique autant que par la stylistique. Les grands benchmarks sur lesquels se comparent les grands modèles de langues sont plutôt orientés sur des tâches sémantiques. Quelques travaux proposent des évaluations orientées sur le style. [Aka+18] propose un jeu de données de paires de mots japonais annotés en fonction de leur proximité stylistique. Ils l'utilisent ensuite pour produire des plongements de mots stylistiques. Cette notion est difficilement transposable aux langues à flexion (français, anglais, ...) où le style d'un mot seul a peu de sens. [NC17] propose également de classer des paires de mots en fonction de leur niveau de formalité. [KH21] étend cette idée en agrégeant de nombreux jeux de données pour créer XS-LUE, un ensemble de benchmarks de classification de phrases, chacun portant sur un axe donné (sarcasme, métaphore, genre, âge, ...). C'est au total 15 styles à évaluer pour plus d'un million de phrases. Chaque style est regroupé dans une grande famille parmi : interpersonnelle, figurative,

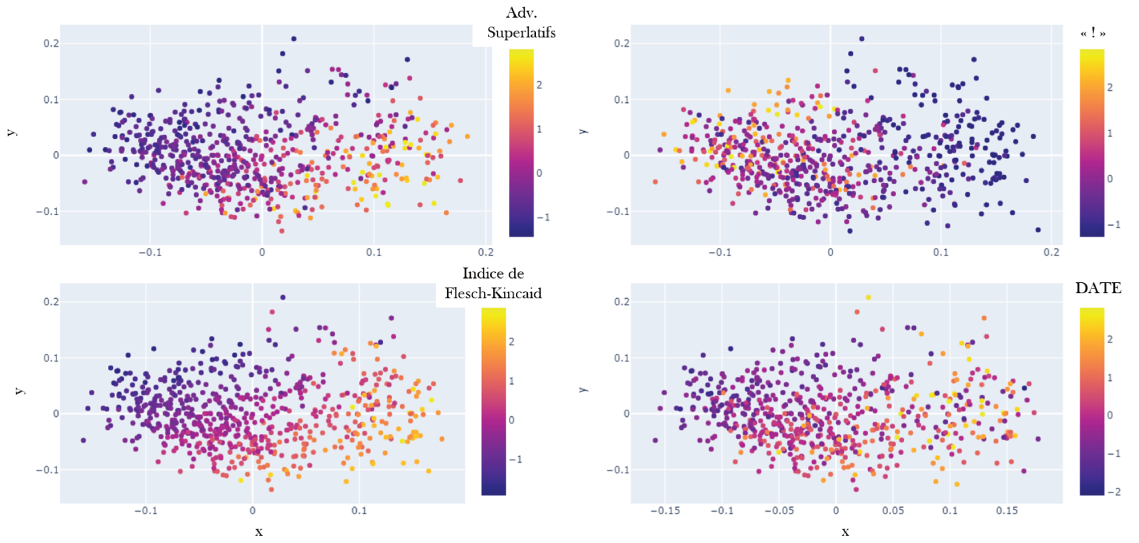


FIGURE 3.2 – Projection des plongements d'auteurices du projet Gutenberg issus de USE avec le gradient de 4 marqueurs stylistiques centrés réduits : fréquence de dates, de points d'exclamation, d'adverbes superlatifs et indice de lisibilité de Flesch-Kincaid.

affective, personnelle. Enfin, [WN21] propose STEL, pour *similarity based STyle EvaLuation*. La tâche consiste à ordonner deux phrases tests S_1 et S_2 pour qu'elles correspondent à l'ordre stylistique des phrases ancres A_1 et A_2. Le fait que les phrases ancres et tests traitent des thématiques totalement différentes permet de se focaliser sur le style selon les auteurices. Ils décomposent ce framework selon deux axes généraux, formel/informel et simple/complexe et deux axes spécifiques, contraction et substitution de nombre. C'est BERT qui performe le mieux parmi les modèles de langues testés.

Si ces méthodes ont l'avantage d'être moins spécifique à une application particulière, comparativement à l'attribution d'auteurices notamment, elles montrent quelques limitations. Tout d'abord, il est difficile de limiter le style littéraire à un nombre d'axe prédéfini et relativement faible. De plus, si il existe forcément des corrélations entre style et sémantique, c'est assurément le cas pour certains axes proposés. Les phrases issues par exemple d'un entretien d'embauche, autour du champ lexical du travail, seront forcément plus formelles. Enfin, elles sont figées dans le temps car dépendante d'un ensemble de phrases annotées à un instant T. C'est un facteur limitant devant l'évolution perpétuelle de la langue et donc des styles d'écriture qui lui sont associés. Pour répondre à ces problématiques, nous présentons dans la prochaine section notre première contribution.

3.4 Contribution 1 : Méthode d'évaluation

Dans cette section, nous présentons une méthode d'évaluation de la capture du style par les modèles de plongements d'auteurices [TGV21]. Si cette dernière a été introduite pour les représen-

tations d'auteurices, elle s'applique également au cas des représentations de documents.

3.4.1 Présentation du framework d'évaluation

Comme évoqué dans la section précédente, agréger un ensemble de marqueurs stylistiques assez simple permet a priori de construire une bonne approximation du style littéraire d'un auteur ou d'une autrice en obtenant de bons résultats en attribution d'auteurices. Ces marqueurs peuvent facilement être extraits d'un corpus et agrégés par auteurices. En entraînant un simple modèle de régression à prédire la valeur de ces marqueurs à partir du plongement des auteurices correspondant permettrait de quantifier dans quelle mesure le modèle de représentation capture la stylistique. La figure 3.2 montre l'intuition derrière notre méthode. En projetant en 2 dimensions les représentations d'auteurices du projet Gutenberg obtenus par le modèle USE (le plongement de l'auteurice correspondant à la moyenne des représentations de ses documents) avec le gradient de quelques marqueurs stylistiques, une tendance semble apparaître. Ce modèle de langue arriverait donc à déceler certaines notions grammaticales et syntaxiques.

Avec l'émergence des grands modèles de langue, de plus en plus opaque, de nombreux articles proposent d'appliquer des classifieurs aux représentations produites afin de capter des informations qu'elles contiennent, ou au contraire ne contiennent pas. Ces frameworks sont appelés des *probing tasks*, en général construites de sorte à isoler un phénomène linguistique de sorte à mesurer à quel point il est assimilé par le modèle. Par exemple, [SPK16] utilise les représentations intermédiaires d'un modèle de traduction pour prédire des POS-tag et évaluer si ce dernier capture la syntaxe de la langue source. [Con+18] propose un benchmark de 10 probing tasks, évaluant entre autres, la capture de la temporalité, de la syntaxe en inversant certains mots de phrase, des longueurs des phrases, etc... C'est dans ce cadre que nous souhaitons placer notre framework.

Malheureusement, aucune métrique de régression ne donne de mesure absolue de la performance. Nous proposons dans notre cadre d'utiliser l'erreur quadratique moyenne (MSE), dans un schéma de cross-validation à 10 répétitions. Cependant, puisque chaque marqueur est centré et réduit avant régression, prédire pour chaque marqueur la valeur moyenne obtenue sur le corpus donnera une MSE de 1.0, qui peut être vu comme un maximum. Nous proposons d'utiliser comme modèle une régression à support de vecteurs avec un noyau RBF (Radial Basis Function). Cela permet d'obtenir à la fois des résultats rapides, tout en performant le mieux vis à vis de modèles plus complexes. Un schéma récapitulatif du framework proposé est montré figure 3.3. Il faut maintenant sélectionner les marqueurs de style les plus pertinents.

3.4.2 Choix de marqueurs stylistiques pertinents

La bonne sélection des descripteurs stylistiques à extraire du corpus est primordiale. Elle doit pouvoir caractériser autant que possible les variations stylistiques que peut proposer un corpus multi-auteurices donné, qu'elles soient phonétiques, syntaxiques ou structurelles. Elles ne doivent contenir aucune information sémantique. Ici nous proposerons un ensemble de descripteurs pour l'anglais. A partir des travaux de [SSV18 ; ZZ05 ; EM18], nous avons sélectionné 242 marqueurs pertinents. Nous y ajoutons les 43 POS-tag pour l'anglais ainsi que les 18 types d'entités nommées également disponibles en nous basant sur les travaux de [Szw17]. Ces derniers les utilisent pour l'attribution d'auteurices avec de bons résultats. Au total, nous obtenons 303 caractéristiques de style distinctes, que nous regroupons en catégorie et résumons dans le tableau 3.2. Un tableau détaillé est disponible en appendice A. Pour l'extraction, nous utilisons les librairies Spacy et nltk. En par-

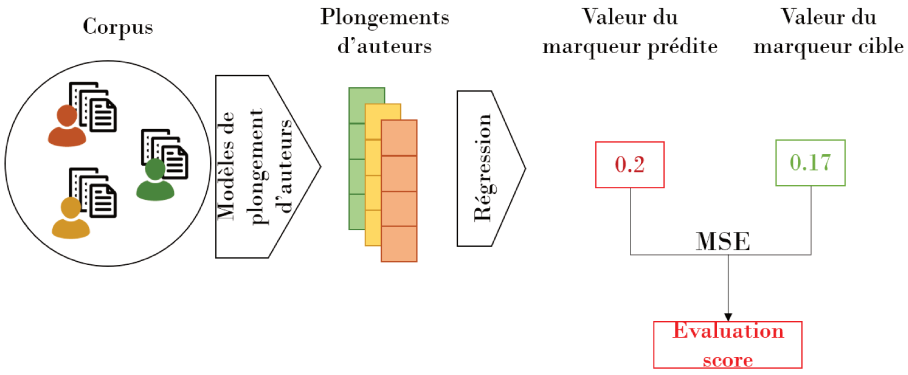


FIGURE 3.3 – Schéma du cadre d'évaluation proposé.

Catégories	Exemples	Nombre de marqueurs
Lettres	Fréquences de lettre	26
Nombre	Fréquences de nombre	11
Structurel	Longueur moyenne des mots, Hapax Legomena, ...	9
Ponctuation	Fréquences des signes de ponctuation	36
Mots outils	Fréquences des mots outils (does, once, doing, ...)	153
Tag	Fréquences des POS-tag	43
Ner	Fréquences des entités nommées	18
Index	Index de lisibilité et de complexité	7

TABLE 3.2 – Listes des marqueurs stylistiques et de leurs catégories. Les fréquences sont calculées par phrase.

ticulier, pour spacy, le tokenizer de mots et de phrases ainsi que le POS-tagger et la reconnaissance d'entités nommées. Pour nltk, le dictionnaire de mots outils et le dictionnaire de prononciation pour le décompte des syllabes.

3.5 Evaluation

3.5.1 Jeux de données et langues

Projet Gutenberg

Le projet Gutenberg est une bibliothèque multilingue de plus de 60 000 livres électroniques tombés dans le domaine public au Etats-Unis. Ils sont disponibles gratuitement depuis 1971 et est encore mis à jour. Nous avons pu récupéré et pré-traité le corpus en utilisant [GF18]. Les oeuvres couvrent un ensemble de genres variés, romans, essais, théâtres, ... des auteurs et autrices de l'Antiquité à nos jours. Ce jeu de données est pertinent dans notre cadre d'étude car il s'inscrit dans la littérature classique. Nous nous limitons aux oeuvres en anglais (qui peuvent être des traductions),

Datasets statistics			
Dataset	**Authors**	**Avg. Tokens**	**Avg. Texts**
BAC10	10	91(±184)	2350(±639)
BAC50	50	98(±167)	1466(±562)
R-BAC	500	243(±342)	50(±0)
R-PGD	664	2315(±961)	10(±0)

TABLE 3.3 – Statistiques descriptives des jeu de données utilisés BAC : Blog Authorship Corpus, PGD : Project Gutenberg Dataset.

ce qui recoupent environ 40 000 livres et sélectionnons aléatoirement 10 oeuvres par auteurices ayant au moins ce niveau de production dans le corpus. Ainsi nous créons un corpus de 664 auteurices et 6640 textes. Afin de limiter la taille des textes pour pouvoir les traiter efficacement, nous nous limitons aux 200 premières phrases. Nous notons ce jeu de données R-PGD pour Reduced Project Gutenberg Dataset.

Blog Authorship Corpus

Le Blog Authorship Corpus est constitué 681 288 posts de blogs, de 19 320 auteurices distincts assemblés au début des années 2000 par [Sch+06]. Ce jeu de données est utilisé dans de nombreux benchmarks d'attribution d'auteurices, en ne conservant que les 10 ou 50 auteurices les plus prolifiques. Nous noterons ces variations respectivement BAC10 et BAC50. Comme pour le projet Gutenberg, nous créons notre propre réduction du corpus en extrayant 500 auteurices et 50 de leurs posts tirés aléatoirement, nous le notons R-BAC. Les évaluations habituelles en stylométrie se focalisent souvent sur de petits nombres d'auteurices (CCAT50, IMDb62, ...). En conserver un plus grand nombre vise à augmenter le spectre de style possiblement observable. R-BAC et R-PGD nous permettent d'étudier deux domaines bien spécifiques d'écriture, le web et la littérature, tout deux ancrés dans notre cadre d'étude. Un récapitulatif des statistiques de ces différents corpus est proposé tableau 3.5.1.

3.5.2 Compétiteurs

Nous souhaitons désormais appliquer notre métrique d'évaluation de la capture du style à différent modèles de représentation de documents afin de déterminer lesquels sont les plus efficaces dans la réalisation de cette tâche. Nous les évaluerons également sur d'autres tâches pertinentes qui seront détaillées plus bas. Pour cela, nous sélectionnons deux types de modèles.

Tout d'abord, nous choisissons quelques grands modèles de langue afin d'évaluer leur capacité à capturer des notions de langues relativement complexes. Notamment, nous utiliserons Sentence-BERT (SBERT) [RG19], la version DAN du Universal Sentence Encoder (USE) [Cer+18] et l'encodeur de phrase du modèle T5 afin d'évaluer un LLM récent [Raf+19]. Nous utilisons la version 4 de USE disponible sur TensorflowHub (USEv4. Pour SBERT et Sentence-T5 nous utilisons les modèles disponibles sur huggingface. Les représentations de documents longs pour SBERT et Sentence-T5 sont les moyennes des représentations des paragraphes qui les constituent tout en respectant leur limite de tokens. Les représentations d'auteurices sont obtenues en moyennant les représentations de leurs documents dans le jeu d'entraînement.

Le reste des modèles testés sont des modèles de représentation pré-entrainés sur la tâche d'attribution d'auteurices. Nous réimplementons le Content-Info Model [Jaw+16](voir section 2.4.3), qui apprend des représentations des auteurices et des documents via un objectif similaire à Word2Vec, mais est non-inductif. Ainsi, les documents de la phase de test auront été vus pendant l'entraînement. Nous réimplémentons le modèle en initialisant les plongements des documents et des auteurices avec USE, car cela donne de meilleurs résultats.

Nous testons aussi le modèle Deepstyle [2] [Hay+20], qui fine-tune DistilBERT sur un grand corpus afin d'apprendre à représenter le style par généralisation. Nous utilisons également sa version fine-tuné sur nos jeux de données en particulier, que l'on notera Deepstyle-ft. Nous utilisons les hyperparamètres recommandés par les auteurices pour cela (pas d'apprentissage de $5e-5$ pendant 5 epochs et des batchs de taille 64). Ces seconds modèles permettent de déterminer si l'attribution d'auteurices permet effectivement de se focaliser sur le style littéraire de ces derniers, que ce soit directement, c'est à dire en étant fine-tuné sur le corpus d'évaluation, ou par généralisation, en transférant les connaissances obtenues d'un entraînement sur un autre grand corpus.

3.5.3 Tâches d'évaluation

Attribution d'auteurices

Tout d'abord, afin de mesurer la capacité des marqueurs stylistiques à capter le style littéraire, nous les évaluons en attribution d'auteurices sur les deux jeux de données avec des nombres d'auteurices variables. Pour cela, nous utilisons un SVM dont nous optimisons les paramètres par grid-search et validation croisée pour chaque jeu de données.

Les modèles sont également évaluer sur l'attribution d'auteurices sur les jeux de données R-BAC et R-PGD. Pour cette tâche nous utilisons deux métriques, l'accuracy, qui correspond à la proportion de documents effectivement attribués à leur auteurice, comprise entre 0 et 1. C'est la métrique habituelle dans l'état de l'art. Nous utilisons également l'erreur de couverture, ou coverage error, qui correspond au rang moyen à prendre en compte pour effectivement obtenir l'auteurice du document dans la prédiction. Elle est comprise entre 1 et le nombre d'auteurices considérés. Ici, nous la normalisons afin qu'elle soit sous forme du pourcentage d'auteurices à prendre en compte pour couvrir le bon. Nous associons chaque document avec l'auteurice dont la similarité cosinus est la plus élevée. Pour rappel, la similarité cosinus permet d'évaluer la proximité entre deux représentations z_d et z_a :

$$\text{sim}_{cos} = \frac{< z_a, z_d >}{\|z_a\|_2 \|z_d\|_2} \tag{3.5.1}$$

Classification de thématiques

La seconde évaluation consistera en la classification de thématiques sur R-PGD à partir des plongements de documents. Nous avons extrait 31 thématiques et avons associé à chaque document sa thématique majoritaire via une LDA (Latent Dirichlet Association de la librairie gensim). Nous avons choisi le nombre de thématiques qui maximisait la valeur de cohérence sur notre corpus et avons utilisé la libraire Gensim. Nous appliquons un SVM optimisé par validation croisée sur les plongements d'auteurices afin de classer chaque auteurice.

Métrique de capture du style

2. Modèle et code disponible ici : https://github.com/hayj/DeepStyle

Attribution d'auteurices avec marqueurs stylistiques et SVM			
Dataset	**Nombre d'auteurices**	**Accuracy ↑**	**Erreur de couverture ↓**
Projet Gutenberg Réduit	10	82.1 (0.9)	1.04 (0.28)
	50	64.5 (1.7)	1.80 (0.46)
	100	53.4 (1.3)	2.28 (0.85)
Blog Authorship Corpus	10	41.8 (1.2)	18.5 (1.4)
	50	29.1 (0.3)	10.7 (3.5)

TABLE 3.4 – Résultats en attribution d'auteurices avec un SVM en utilisant uniquement les marqueurs stylistiques. Les résultats sont les moyennes obtenus sur 10 répétitions, l'écart type est entre parenthèses.

Enfin, nous évaluons également ces modèles sur notre métrique de capture du style littéraire détaillée plus haut.

3.6 Résultats

Nous détaillons ici l'ensemble des résultats obtenus sur chacune des tâches d'évaluation. Après avoir analysé la capacité des marqueurs stylistiques à associer un document à son auteurice sans aucune information sémantique et donc à approximer le style littéraire, nous nous pencherons sur les résultats obtenus par les modèles sélectionnés.

3.6.1 Attribution d'auteurices par marqueurs stylistiques

Les résultats pour l'attribution d'auteurices par marqueurs stylistiques sont disponible dans le tableau 3.4. Pour le Projet Gutenberg Réduit, nous avons sélectionné aléatoirement un nombre donné d'auteurices (10, 50, 100) puis avons répété l'évaluation suffisamment de fois afin de balayer l'ensemble des auteurices (respectivement 67, 14 et 7 fois). En utilisant uniquement des marqueurs basés sur le style, nous arrivons à obtenir plus de 80% d'accuracy avec 10 auteurices. Peu importe le nombre d'auteurices, l'erreur de couverture moyenne est au plus égale à 2.28, ce qui signifie que ce modèle très simple, sans aucune information thématique, arrive effectivement à capturer des informations syntaxiques et grammaticales propre aux auteurices.

Pour le Blog Authorship Corpus, nous avons utilisé les variations BAC10 et BAC50 de la littérature. On constate que les résultats sont moins bons, mais les textes sont beaucoup plus courts (seulement une centaine de mots, contre plus de 2000 pour le Projet Gutenberg) et donc il est plus difficile d'en dégager des styles forts. A titre de comparaison, les meilleurs modèles atteignent autour de 60% d'accuracy sur BAC10 et BAC50. Malgré tout, sur BAC10 la bonne prédiction est classée en moyenne entre le rang 1 et le rang 2 (erreur de couverture de 18.5%) et dans les 5 premiers sur BAC50. Encore une fois, ces marqueurs très simples permettent d'obtenir une bonne approximation du style littéraire.

Nous avons également réalisé une étude d'ablation par catégories afin d'évaluer quels marqueurs portent le plus d'information (voir Figure 3.4). Pour cela, nous avons répété notre processus d'évaluation avec 50 auteurices en ôtant successivement chaque catégorie. Les résultats montrent qu'elles participent toute autant à la caractérisation des auteurices tout en portant des informations dis-

Prédiction de thématiques à partir des plongements de documents		
Modèles	**Accuracy ↑**	**Erreur de couverture ↓**
Content-Info	52.6 (3.3)	10.36 (0.5)
Deepstyle	53.5 (2.9)	10.5 (0.2)
Deepstyle-ft	57.1 (3.3)	11.6 (1.5)
SBERT	60.4 (2.3)	9.7 (0.9)
Sentence-T5	60.0 (1.2)	10.5 (0.6)
USE	60.1 (1.4)	10.4 (1.0)

TABLE 3.5 – Résultats en classification de thématiques sur R-PGD à partir des plongements de documents et d'un SVM (31 thématiques). Les résultats sont la moyenne sur 10 répétitions, entre parenthèses l'écart type.

	R-PGD 664 auteurices		**R-BAC 500 auteurices**	
Modèles	**Accuracy ↑**	**Erreur de couverture ↓**	**Accuracy ↑**	**Erreur de couverture ↓**
SBERT	7.0 (0.6)	22.4 (0.8)	4.9 (0.4)	22.6 (0.7)
Sentence-T5	6.7 (0.8)	16.0 (0.6)	12.4 (1.0)	14.8 (0.4)
USE	26.3 (0.5)	6.9 (0.4)	17.2 (0.5)	11.1 (0.7)
Deepstyle	26.8 (0.9)	9.4 (0.7)	20.1 (1.1)	12.3 (0.7)
Deepstyle-ft	52.4 (0.8)	**2.0** (0.6)	50.1 (1.0)	**4.3 (0.4)**
Content-Info	**92.6 (0.9)**	6.3 (0.5)	**81.0 (0.8)**	8.9 (0.7)

TABLE 3.6 – Résultats en attribution d'auteurices sur R-BAC et R-PGD pour l'ensemble de nos compétiteurs. Le meilleur modèle est en gras, le second est souligné, l'écart type entre parenthèse. Content-Info est à prendre avec précaution car il n'est pas inductif.

tinctes. En effet, c'est la combinaison de l'ensemble des catégories qui obtient les meilleurs résultats de manière significative. Les résultats d'ablation au niveau des marqueurs est disponible en appendice A. Enfin, nous avons également testé l'extraction des descripteurs avec la librairie Stanza, dont les résultats en tokenization, POS-tagging et reconnaissance d'entitées nommées sont meilleurs que les modèles de spacy. Cependant, cela n'a produit aucune amélioration significative des résultats, en multipliant par 5 le temps de calculs, d'où notre choix de conserver spacy.

3.6.2 Pour l'attribution d'auteurices

Les résultats sont présentés Tableau 3.6. Sans surprise, ce sont les modèles entraînés spécifiquement sur les deux jeux de données d'évaluation qui performent le mieux. Si les résultats de Content-Info sont peu informatifs de par son caractère transductif, il est intéressant de voir qu'il est dépassé en terme d'erreur de couverture par Deepstyle-ft. Cela signifie que lors d'une erreur de prédiction, Content-Info classe assez loin le bon auteur ou la bonne autrice. On peut émettre l'hypothèse que comme modèle très simple, il se focalise essentiellement sur le contenu et ces erreurs sont essentiellement des oeuvres où l'auteurice s'éloigne de ses thématiques habituelles. Deepstyle est le meilleur des modèles de langue non spécifiquement entraîné durant l'évaluation, suivi de près par USE. Pour rappel, Deepstyle est une version de DistilBERT fine-tuné sur un grand corpus d'at-

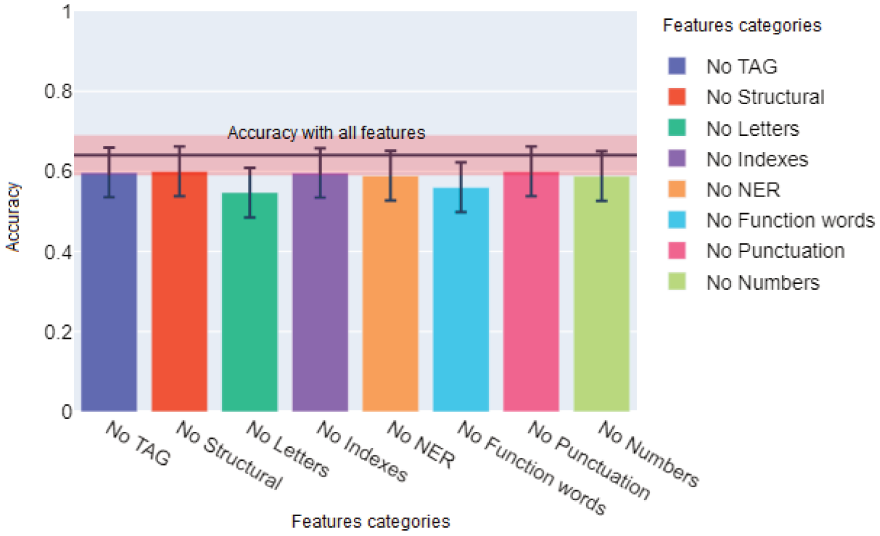


FIGURE 3.4 – Etude d'ablation sur les catégories de marqueurs pour l'attribution d'auteurices sur le Projet Gutenberg avec 50 auteurices. Les barres d'erreur correspondent à l'écart type.

tribution d'auteurices, comprenant notamment le Blog Authorship Corpus, ce qui peut expliquer sa légère supériorité sur R-BAC. Conformément à ce que disent ses auteurices, cette tâche lui permet de généraliser suffisamment pour être performant sur de nouveaux jeux de données, même très différents du set d'entraînement comme R-PGD. USE, bénéficie également de son entraînement sur un ensemble de tâches très variées (similarité sémantique, analyse de sentiments, de polarité, ...) et un grand ensemble de données. De manière plus surprenante, SBERT et Sentence-T5 sous-performent. SBERT bénéficie d'un plus petit corpus d'entraînement et sur une unique tâche (SNLI [Bow+15]). De manière plus surprenante, SBERT et Sentence-T5 sous-performent. SBERT est bénéficie d'un plus petit corpus d'entraînement et sur une unique tâche (SNLI [Bow+15]). Il est plus difficile d'expliquer les résultats de Sentence-T5. En effet, c'est un grand modèle de langue pré-entraîné sur un benchmark proche de celui de USE, sous un format de génération de texte cependant. L'utilisation de l'encodeur uniquement ne permet peut-être pas d'exploiter pleinement le modèle. Ou l'objectif de génération de texte amène le modèle à se focaliser sur des éléments moins pertinents pour l'attribution d'auteurices. Cela rejoindrait les scores de SBERT, BERT étant initialement entraîné sur la prédiction de mots masqués.

Si ces résultats sont intéressants en tant que tels, c'est leur mise en perspective avec l'évaluation stylistique qui permettra de déterminer ce que peut apporter l'attribution d'auteurices, par exemple.

3.6.3 Pour la classification de thématiques

Les résultats en classification de thématiques sont disponible Tableau 3.5. Cette tâche permet de déterminer à quel point les plongements des modèles sont focalisés sur l'information sémantique. Les résultats sont assez homogènes et seul Deepstyle performe significativement moins bien que les autres modèles. Sa version fine-tuné sur le Projet Gutenberg le dépasse de 3.6 points en accuracy,

ce qui semble indiquer que l'attribution d'auteurices amène les modèles de langue à se focaliser plutôt sur le contenu sémantique. SBERT, Sentence-T5 et USE, bien qu'uniquement pré-entraîné sur des tâches très générale de modélisation du langage performe aussi bien que les autres modèles. L'information sémantique semble là aussi assez aisément captée.

3.6.4 Pour la capture du style

Les résultats sur la capture du style sont présentés Tableau 3.7. Ils sont également résumé sous forme de radar Figure 3.5. Bien que très différents, l'un purement littéraire (R-PGD), l'autre issu des réseaux sociaux (R-BAC), les deux jeux de données donnent des tendances similaires. Les modèles Content-Info et Deepstyle-ft, tous deux entraînés sur les corpus d'évaluation performent le moins bien sur tous les axes ou presque, bien que conçus pour construire des représentations de documents et/ou d'auteurices cohérentes. Sentence-T5 obtient des scores très moyen, malgré que ce soit un des modèles de langue les plus récents. Il semblerait que les tâches d'entraînement text-to-text ne permettent pas à l'encodeur seul de capturer ces notions de langue.

Deepstyle obtient de bons résultats sur R-BAC, se plaçant premier ou deuxième sur la moitié des axes. Il bénéficie de son pré-entraînement sur un grand corpus issu du web, très proche du Blog Authorship Corpus. Il a en revanche plus de mal à généraliser sur le Projet Gutenberg, où il peut être limiter par la longueur des textes. La comparaison entre Deepstyle et Deepstyle-ft, le premier ayant des résultats globablement légèrement meilleurs sur 9 axes sur 16, permet d'affirmer que l'attribution d'auteurices seule n'est pas une tâche suffisante pour pousser les modèles de langue à se focaliser sur le style littéraire. Dans la mesure où on prend comme définition du style tous les choix d'écriture hors de la sémantique.

USE est le modèle qui obtient les meilleurs scores, sur les deux corpus et sur 13 axes sur 16. Suivi de près par SBERT, ils semblent être capable de capturer des notions linguistiques et grammaticales complexes, que ce soient les tags ou les entités nommées par exemple. On pouvait s'y attendre pour SBERT, [Cla+19a] a notamment montré que chaque têtes d'attention se focalise naturellement sur certaines fonctions des mots (sujet, complément, ...). Notre évaluation semble montrer que ce phénomène se transmet aux plongements d'auteurices.

C'est en revanche plus surprenant pour USE, dont nous utilisons la version DAN (Deep Averaging Network), qui moyenne les représentations des mots avant d'utiliser un MLP. Cette modélisation sac-de-mots perd la vision séquentielle du langage. Nous montrons ici que cette hypothèse n'est pas nécessaire pour détecter et résoudre des tâches basées sur la syntaxe, ce qui confirme les travaux de [Iyy+15b]. Pouvoir se passer d'un traitement syntaxique permet des gains élevés en terme de temps de calculs et de ressources, comparativement aux modèles à base de Transformer.

3.7 Conclusion et perspectives

Nous avons développé dans ce chapitre la définition du style écrit en linguistique computationnelle et comment il est appréhendé par le biais de la tâche d'attribution d'auteurices. Nous avons évoqué les différentes méthodes pour l'appréhender, des marqueurs stylistiques au plus récent modèles de représentation. Ces derniers étant de véritables boîtes noires, se pose la question de l'évaluation de leur capacité à mesurer le style littéraire. Les méthodes actuelles sont limitées, notamment car dépendantes d'un certains nombre de catégories figées. Pour cela, nous avons présenté notre première contribution, une mesure d'évaluation par régression de marqueurs stylistiques. Cette méthode a permis de montrer les limites de l'attribution d'auteurices dans le but de

Erreur quadratique moyenne et écart type sur la régression de marqueurs pour R-PGD								
Modèles	**Lettres**	**Nombre**	**Structurel**	**Ponct.**	**Mots outils**	**TAG**	**NER**	**Index**
Content-Info	0.67 (0.17)	0.88 (0.12)	0.55 (0.19)	0.68 (0.16)	0.72 (0.19)	0.65 (0.17)	0.74 (0.14)	0.50 (0.16)
SBERT	0.66 (0.27)	0.89 (0.07)	0.42 (0.15)	0.78 (0.23)	0.71 (0.21)	0.58 (0.22)	0.72 (0.15)	0.37 (0.14)
USE	**0.61 (0.27)**	0.86 (0.09)	**0.34** (0.18)	**0.59 (0.26)**	**0.65 (0.24)**	**0.45 (0.29)**	**0.65 (0.17)**	**0.27 (0.15)**
Deepstyle-ft	0.79 (0.16)	0.92 (0.09)	0.65 (0.15)	0.82 (0.17)	0.84 (0.13)	0.74 (0.14)	0.84 (0.08)	0.60 (0.14)
Deepstyle	0.68 (0.22)	**0.78 (0.08)**	0.41 (0.28)	0.86 (0.16)	0.85 (0.15)	0.81 (0.11)	0.89 (0.13)	0.46 (0.31)
Sentence-T5	0.71 (0.26)	0.91 (0.06)	0.51 (0.20)	0.81 (0.22)	0.76 (0.19)	0.61 (0.22)	0.76 (0.15)	0.41 (0.18)
Erreur quadratique moyenne et écart type sur la régression de marqueurs pour R-BAC								
Modèles	**Lettres**	**Nombre**	**Structurel**	**Ponct.**	**Mots outils**	**TAG**	**NER**	**Index**
Content-Info	0.80 (0.15)	0.85 (0.07)	0.62 (0.23)	0.92 (0.09)	0.87 (0.12)	0.90 (0.05)	0.93 (0.07)	0.70 (0.29)
SBERT	0.68 (0.28)	**0.78 (0.05)**	0.49 (0.23)	0.90 (0.11)	0.85 (0.16)	0.86 (0.08)	0.87 (0.13)	0.54 (0.28)
USE	**0.67 (0.25)**	0.83 (0.05)	0.45 (0.20)	**0.78 (0.17)**	**0.81 (0.17)**	**0.63 (0.21)**	**0.80 (0.17)**	**0.38 (0.18)**
Deepstyle-ft	0.73 (0.18)	0.82 (0.06)	0.53 (0.20)	0.84 (0.15)	0.87 (0.12)	0.77 (0.12)	0.86 (0.11)	0.53 (0.18)
Deepstyle	0.68 (0.22)	0.78 (0.08)	**0.41 (0.28)**	0.86 (0.16)	0.85 (0.15)	0.81 (0.11)	0.89 (0.13)	0.46 (0.31)
Sentence-T5	0.73 (0.26)	0.80 (0.06)	0.50 (0.26)	0.93 (0.08)	0.87 (0.14)	0.87 (0.09)	0.90 (0.11)	0.57 (0.29)

TABLE 3.7 – Prédiction de marqueurs stylistiques sur R-BAC et R-PGD L'erreur quadratique moyenne (écart type entre parenthèses) sur la régression de descripteurs stylistiques à partir des plongements d'auteurices avec un SVR. Les 303 marqueurs stylistiques sont regroupés par catégories. En gras le meilleur score pour chaque axe, souligné la seconde meilleure valeur.

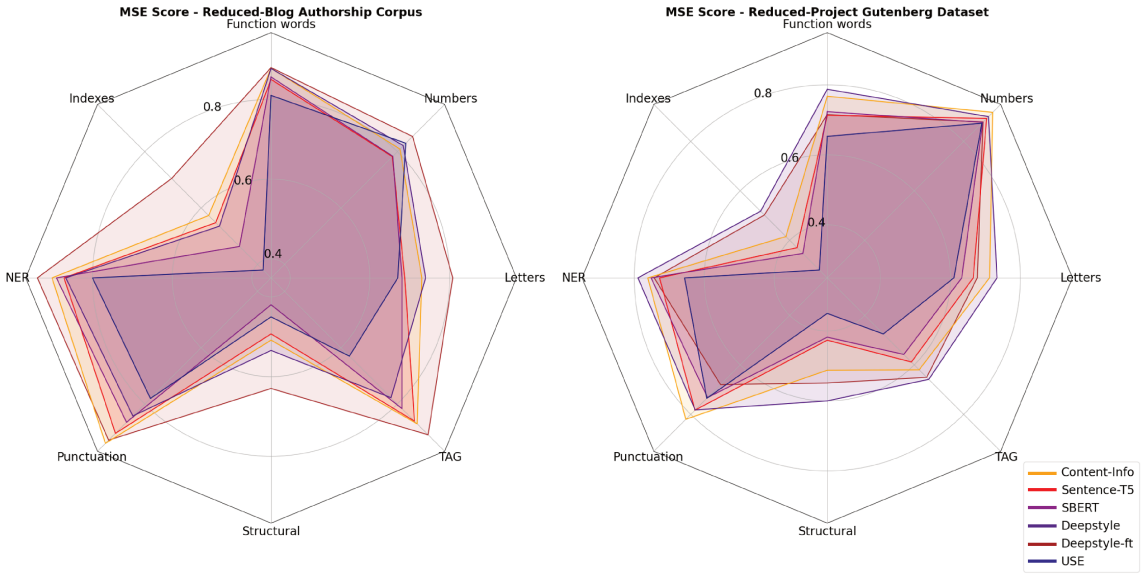


FIGURE 3.5 – Représentation des résultats de régression de marqueurs stylistiques sous la forme de graphique radar pour faciliter la lecture des résultats

représenter le style littéraire, ainsi que la forte capacité des modèles de langue, notamment USE, à mesurer des notions de langue complexe. Ces modèles sont donc des briques essentielles pour construire des modèles de représentation d'auteurices et du style littéraire, mais nécessitent sûrement des contraintes plus fortes que l'attribution d'auteurices pour cela. Il pourrait être intéressant de comparer la version Transformer et la version DAN de USE afin de déterminer l'importance de l'architecture comparativement à celle des tâches et des corpus d'entraînement. Nous pouvons nous attendre à ce que la version Transformer de USE soit plus performante bien que plus coûteuse, mais cela reste à démontrer.

Il est également possible d'améliorer notre métrique d'évaluation selon plusieurs axes. Tout d'abord, la liste des marqueurs n'est pas figée et il est probablement possible de l'enrichir notamment avec l'appui de chercheurs en linguistiques et en littérature. Il serait pertinent de pouvoir associer chaque variation de descripteur à un auteur, une autrice ou un document type agissant comme exemple (un extrait de Proust pour le marqueur traitant de la longueur des phrases, etc). Le second défaut de cette méthode est la perte d'une partie de l'information syntaxique à cause de l'utilisation de fréquence et de moyenne. Si il est déjà possible d'ajouter des mesures de variance pour contrer cela, s'inspirer de la méthode des motifs [DL14] pourrait être intéressant. Plus un document sera long et plus l'auteurice aura le temps de mettre son style en place et de le définir, mais c'est aussi ce qui fait la difficulté de l'évaluation des modèles de langue qui agrège cette information. La plupart des probing tasks s'applique uniquement à des phrases. Pouvoir les étendre à des documents longs sans perte d'information dû à l'agrégation serait une véritable avancée. Enfin, le transfert de style d'écriture d'auteurices en génération de texte est de plus en plus envisagé [TG21]. Les métriques d'évaluation sur le transfert du style reposent essentiellement sur des classifieurs qui peuvent être focalisés sur la sémantique. Il pourrait être intéressant d'essayer d'étendre notre framework à ce genre d'application.

Chapitre 4

Apprentissage de représentations de documents et d'auteurs se concentrant sur le style

4.1 Introduction

Les sections précédentes nous ont permis de démontrer l'intérêt de l'apprentissage de représentation de documents et d'auteurices en TAL. Elles permettent de résoudre de nombreuses sous-tâches, de la classification à la génération de texte en passant par l'identification d'auteurices ou la recommandation. Nous avons également pu montrer en proposant une méthode d'évaluation dédiée au style littéraire que ces plongements se focalisent principalement sur la sémantique, même dans le cas de modèles utilisant l'attribution d'auteurices comme objectif. En effet, les méthodes s'intéressant à la stylistique à proprement parler, par exemple dans le cadre des tâches PAN, sont rarement des modèles de représentation mais plutôt une combinaison de descripteurs linguistiques et de modèles de classification. De plus, peu de modèles associent à l'apprentissage de plongements de documents celui des auteurices du corpus. Se focaliser sur le style écrit est pertinent dans notre cadre d'application qu'est la littérature. Ça l'est tout autant sur des corpus plus traditionnel, que ce soit pour la vérification ou le profiling d'auteurices, par exemple. Nous allons donc nous intéresser dans cette section aux modèles de plongements d'auteurices et/ou de documents capturant le style littéraire. Après avoir présenté les méthodes existantes ainsi que leur limite, nous détaillerons notre seconde contribution, le modèle VADES (Variational Author and Document Embedding with Style). Nous présenterons le framework du Variational Information Bottleneck (VIB) sur lequel il s'appuie, ainsi que l'architecture retenue. Enfin, nous analyserons les résultats expérimentaux sur différentes tâches : l'attribution d'auteurices et la prédiction de descripteurs du style. Ainsi, nous montrons que notre modèle contrairement aux méthodes existantes capturent efficacement le style littéraire tout en étant capable de rivaliser quand il faut associer les auteurices à leur production.

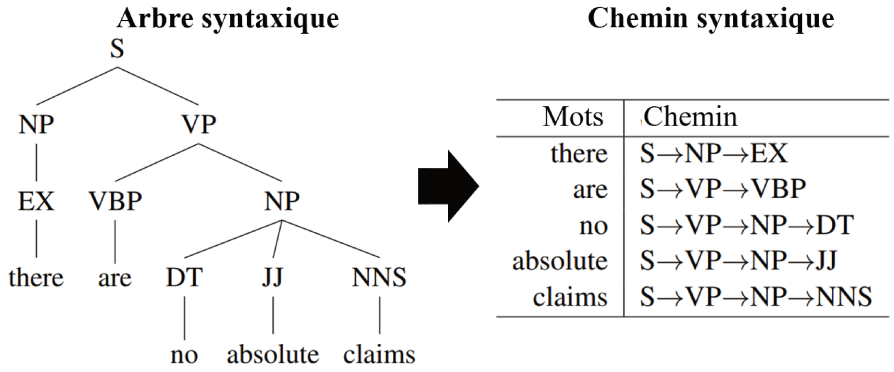


FIGURE 4.1 – Exemple d'arbre syntaxique et de chemins syntaxiques associés

4.2 Etat de l'art

S'il existe de nombreux modèles de représentations d'auteurices (voir section 2.4.3), très peu tentent de se focaliser sur la capture du style littéraire. Pour rappel, en TAL le style écrit sous-entend tous les choix d'écriture fait par l'auteurice en dehors de la sémantique. A notre connaissance, seul [Mah+19] dit explicitement vouloir capturer le style de chaque auteurice dans leurs plongements. Pour cela, ils s'appuient sur le modèle Doc2Vec en remplaçant l'identifiant du document par l'identifiant de l'auteurice. Toutefois, en lieu et place des mots, ils utilisent des trigrams de caractère annotés en fonction de leur position dans chaque mot (préfixe, suffixe, interne) pour représenter le texte. Selon plusieurs travaux en stylistique, ces trigrams de caractère permettent à la fois de capturer le style et le contenu des textes. Leur modèle apprend donc des représentations de chaque auteurice et du vocabulaire constitué de chaque trigram. Ils obtiennent la représentation d'un document ensuite en moyennant les représentations des trigrams qui le composent.

Les autres modèles s'intéressant à la représentation du style littéraire se limitent tous au plongement de documents, en s'évaluant cependant sur la tâche d'attribution d'auteurices. Par exemple, [SSV18] utilise un CNN sur un ensemble de descripteurs mixtes (longueur moyenne des mots, fréquences de caractères, n-grams de mots et de caractères, ...). [JH19] apprend des représentations de mots et de POS tag. Ils passent en entrée d'un CNN les phrases et leurs traductions syntaxiques en POS tag, avant de les agréger par un LSTM et une couche d'attention pour obtenir la représentation du document. De la même façon, [Zha+18] combine la représentation de la phrase sous forme de trigrams de caractères, avec celle utilisant l'arbre de dépendances. Ce dernier permet d'associer un chemin syntaxique $c(w)$ à chaque mot w (voir Figure 4.1). En apprenant un plongement pour chaque type de dépendance, que l'on note $q_i \in \mathbb{R}^r$, ainsi qu'un plongement $p_i \in \mathbb{R}^d$ pour chaque rang i possible dans un chemin syntaxique, il est possible d'obtenir la représentation de chaque chemin comme suit :

$$q_{c(w)} = \sum_{i \in c(w)} q_i \circ p_i \tag{4.2.1}$$

Les représentations syntaxiques des trigrams sont ensuite passées dans deux CNNs parallèles, puis après max-pooling sont concaténées pour générer une réprésentation finale.

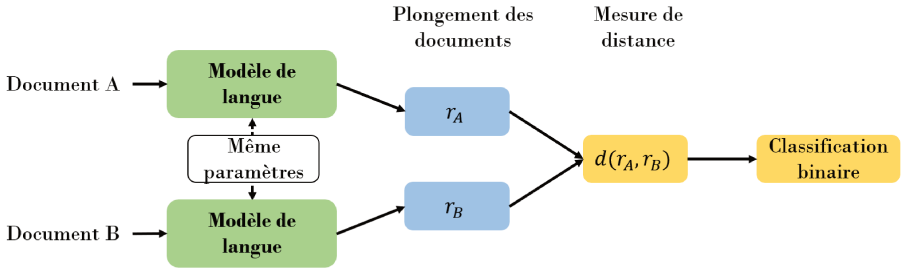


FIGURE 4.2 – Illustration de l'architecture d'un réseau de neurones siamois.

Les méthodes plus récentes se basent essentiellement sur la spécialisation de grands modèles de langue en attribution d'auteurices ou sur des variations de cette tâche. Par exemple, [Hay+20] entraîne DistilBERT (version distillée de BERT) sur un grand corpus de référence, provenant de l'agrégation de plusieurs jeux de données. Ce qu'ils appellent la *cohérence intra-auteurice*, à savoir le fait que l'auteurice aura un style cohérent dans toute sa production, leur permet d'affirmer qu'ils apprennent effectivement des représentations capturant le style, d'autant plus que le corpus d'entraînement sera grand et diversifié. Le modèle est ensuite évalué en attribution d'auteurices sur un nouvel ensemble d'auteurices. Afin de s'assurer que le modèle spécialisé se focalise effectivement sur le style et non sur le contenu, une variation de l'attribution d'auteurices est utilisée : la vérification d'auteurices. Pour rappel, l'attribution d'auteurices consiste à prédire l'auteurice d'un document donné. La vérification d'auteurices vise à comparer deux documents afin de prédire si ils ont été écrits par la même personne. Elle s'appuie en général sur l'utilisation de réseaux de neurones siamois (voir Figure 4.2) durant l'entraînement, similaire à ce qui est fait pour SBERT (pour Sentence-BERT) [RG19]. [Boe+19] utilisent cette architecture siamoise avec des plongements de mots pré-entraînés et un LSTM pour déterminer si deux documents proviennent du même auteur ou de la même autrice. Contrairement à SBERT qui se contente de prendre comme fonction de perte la distance euclidienne entre les représentations, ils utilisent la fonction de perte suivante, où $d(z_d, z_{d'})$ correspond à la distance entre les représentations des documents d et d' et y est le label indiquant si l'auteurice des documents est identique ($y = 1$) ou non ($y = 0$) :

$$\mathcal{L}(z_d, z_{d'}) = \frac{y}{2} \max(d(z_d, z_{d'}) - \tau_1; 0)^2 + \frac{1-y}{2} \max(d(z_d, z_{d'}) - \tau_1; 0)^2 \tag{4.2.2}$$

Cette fonction de perte, appelée *max-margin loss* permet de faire intervenir des seuils de tolérance τ_1 et τ_2 sur la prédiction. Dans des travaux suivants [BN19], ils modifient l'architecture de leur modèle nommé AdHominem, en conservant les métriques et fonction de perte. Pour chaque mot dans une phrase, AdHominem apprend un plongement, ainsi qu'une agrégation de plongements de caractères le composant obtenus après convolution et max-pooling. Ces deux représentations sont ensuite concaténées et passées dans un LSTM bidirectionnel. Puis, une couche d'attention permet d'obtenir un plongement de phrase. Chaque plongement des phrases d'un document est à son tour donné à un LSTM puis une couche d'attention. Les couches d'attention permet d'interpréter l'importance des mots et des phrases d'un document pour la construction de la représentation finale du document.

[ZJ21] utilise l'architecture de SBERT et de SRoBERTa. Ils se basent cependant sur l'approximation continue de la *max-margin loss*, où P^+ constitue l'ensemble des paires positives et P^- des

paires négatives :

$$\mathcal{L}(P^+, P^-) = \frac{1}{|P^+|} \sum_{(i,j)\in P^+} \text{Softplus}(\text{LogSumExp}(\alpha[d(z_{d_i}, z_{d_j}) - \tau_1])) + \frac{1}{|P^-|} \sum_{(i,j)\in P^-} \text{Softplus}(\text{LogSumExp}(\alpha[d(z_{d_i}, z_{d_j}) - \tau_2])) \tag{4.2.3}$$

La fonction $\text{Softplus}(z) = \log(1 + e^z)$ est une approximation lissée et non-nulle de la fonction d'activation ReLU ($\text{ReLU}(z) = \max(0, z)$). Ces deux propriétés supplémentaires lui permettent de rendre l'apprentissage plus stable, notamment quand la profondeur des réseaux grandit. Ce modèle est simplement une extension de SBERT et SRoBERTa à la tâche de vérification d'auteurices.

Enfin, [WSN22] construit une nouvelle variation de la vérification d'auteurices, appelée *Contrastive Authorship Verification* (vérification d'auteurices contrastive). A partir d'une phrase de référence d_1 d'un autrice ou d'un auteur donnée, appelée ancre et de deux autres phrases d_2 et d_3, l'objectif est de déterminer laquelle de ces deux dernières a été écrite par la même main que l'ancre d_1. De nouveau, l'architecture siamoise s'applique bien dans ce cadre, cette fois avec la *triplet loss*, également utilisée dans SBERT :

$$\mathcal{L}(z_{d_1}, z_{d_2}, z_{d_3}) = \max(d(z_{d_1} - z_{d_2}) - d(z_{d_1} - z_{d_3}) + \tau; 0) \tag{4.2.4}$$

La similarité cosinus est utilisée comme fonction d dans ces travaux, permettant d'évaluer la similarité entre les plongements. L'idée est pour chaque auteurice de rapprocher ses documents et d'éloigner ceux des autres auteurices dans l'espace latent. Le modèle est testé sur un jeu de données de conversations Reddit, ce qui permet de tester différentes configurations de triplet. Lorsque A_1 et B proviennent de la même conversation (donc sur des contenus proches), de la même thématique Reddit, ou d'une autre thématique aléatoire. Le modèle de langue spécialisé dans l'article est RoBERTa. Cette variation de la vérification d'auteurices peut être compliquée à mettre en place sur des jeux de données autre que conversationnels.

Nous avons présenté ici une revue exhaustive des méthodes de plongements d'auteurices, mais surtout de documents essayant de se focaliser sur la capture du style écrit. Dans la prochaine section, nous allons évoquer les forces et faiblesses qu'elles peuvent présenter et qui nous ont poussé à développer le modèle VADES.

4.3 Forces et faiblesses des modèles existants

Le premier manque évident est l'absence d'apprentissage de représentation d'auteurices. Plusieurs modèles existent (voir section 2.4.3) mais un seul porte son attention sur le style. La plupart des modèles s'orientant vers la représentation du style littéraire sont des modèles de plongements de documents. Les approches récentes comme [Hay+20 ; WSN22] proposent d'entraîner les modèles sur des corpus de très grande taille afin d'obtenir des modèles apprenant le style par généralisation et donc efficace sans nouvel entraînement sur d'autres corpus. Si l'on veut idéalement balayer toutes les variations possibles de la langue et donc du style, ce genre d'approche nécessite énormément de données, issues de sources variées (blogs, littérature, médias, ...).

L'essentiel des modèles présentés précédemment utilisent deux tâches d'entraînement : l'attribution d'auteurices et la vérification d'auteurices. Nous avons pu montrer dans le chapitre précédent

que l'attribution d'auteurices ne garantit en rien que le modèle va se focaliser sur le style plutôt que sur le contenu. Il en va de même pour la vérification d'auteurices. Certains travaux justifient cependant une capacité à représenter le style avec des expérimentations un peu plus poussées. Par exemple, [WSN22] s'évalue sur STEL (*similarity based STyle EvaLuation*. Il s'agit, à partir de deux phrases ancres A_1 et A_2 traitant du même contenu mais au style différent, d'associer les phrases correspondantes en terme de style S_1 et S_2. Quatre styles sont testés ici, le niveau de formalité, de contraction, de complexité et la présence de nombre. A chaque fois les modèles SRoBERTa spécialisé sur la vérification d'auteurices sont battus par la baseline SRoBERTa.

De même, [ZJ21] testent leurs modèles en permutant l'ordre des mots, des phrases, ou encore en ne conservant que les mots outils ou que les mots de contenu, avec de très faible réduction de performance. Cela confirme seulement le grand nombre de paramètres entrant en jeu dans la prise de décision des modèles et leur faible niveau d'interprétabilité. Par des variations sur la ponctuation, la présence ou non de sujet, entre autres, ils montrent cependant la capacité des modèles de langue à capturer certaines notions de langue.

Le modèle AdHominem de [BN19] a l'avantage d'être interprétable, en proposant des poids d'attention pour chaque token et chaque phrase d'un document. Si il est toujours difficile pour les mots de contenu de savoir si c'est leur fonction ou leur signification qui est mise en avant, c'est une véritable avancée. A travers différents exemples, ils montrent que leur modèle se focalise effectivement sur certaines notions de style, allant de la capitalisation des lettres, à l'utilisation de syntaxe particulière ou de faute délibérée.

Enfin, une bonne partie des méthodes présentées s'appuient sur des modèles de langue pouvant traiter des textes d'une longueur assez réduite (typiquement 512 tokens pour BERT). C'est un aspect limitant dans un cadre littéraire ou les textes rencontrés peuvent être relativement long, le style des auteurs et autrices se déroulant tout au long d'un ouvrage. En s'appuyant sur les limites mais aussi les bonnes orientations constatées dans les travaux existant, nous allons présenter notre seconde contribution, le modèle VADES et le framework VIB sur lequel il est construit.

4.4 Contribution 2 : VADES

Notre objectif est de construire un modèle de représentation des auteurices et des documents dans le même espace $\mathbb{R}^r$ capturant le style littéraire. Plus précisément, un espace de représentation rapprochant les auteurices de leur production et des documents et auteurices stylistiquement proches. Si l'attribution d'auteurices semblent être un objectif d'entraînement pertinent, il faut aussi pouvoir guider l'apprentissage afin de forcer le modèle à "oublier" l'information sémantique et à se focaliser sur le style écrit.

Le chapitre précédent, qui détaille une métrique d'évaluation de la capacité des modèles de langue à capter le style littéraire, montre que les récents grands modèles de langue (BERT, USE, ...) appréhendent des notions grammaticales et syntaxiques complexes. C'est également confirmé par les analyses faites dans les travaux évoqués plus haut. Ainsi, nous souhaiterions utiliser comme brique élémentaire un LLM, que nous spécialiserions.

Nous disposons pour cela d'un corpus de textes, issu de la littérature ou du web. Bien qu'un document puisse avoir été écrit par de multiples auteurices, nous nous limiterons dans les expérimentations au corpus classiques mono-auteurices de l'attribution d'auteurices. Cependant, notre cadre doit pouvoir s'appliquer indifféremment aux deux cas évoqués ci-dessus.

Enfin, pouvoir traiter des documents sans limite de taille est un objectif. De la même façon, pouvoir obtenir une mesure de la variabilité du style des auteurices dans leurs productions et pour-

quoi pas au sein même d'un document serait d'un intérêt certain. C'est pourquoi nous nous sommes intéressés au cadre variationnel et notamment au framework VIB pour *Variational Information Bottleneck*

4.4.1 Le framework VIB

Le framework VIB est une extension variationnelle du principe de l'Information Bottleneck [TPB99] proposé par [Ale+17]. L'objectif général est, pour un ensemble d'observations x, associées à des étiquettes y, de construire des représentations latentes z telles que :

$$\arg\max_z I(z, y) - \beta I(z, x), \tag{4.4.1}$$

où I est l'Information Mutuelle, définie par :

$$I(x, y) = \int\int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} d_x d_y. \tag{4.4.2}$$

L'Information Bottleneck vise à compresser au maximum l'information contenue dans z de sorte à ce qu'elle contienne l'essentiel de l'information nécessaire à la prédiction de l'étiquette y. Dans notre cadre d'application, nous souhaiterions typiquement maximiser l'information stylistique capturée, tout en minimisant celle sémantique. L'équation 4.4.1 montre la présence de l'hyperparamètre β permettant de contrôler l'équilibre entre ces deux sous-objectifs.

Dans cette approche, $p(x, y)$, appelée, la loi d'encodage est un choix de modélisation. En général, le calcul explicite de l'Information Mutuelle n'est pas possible. [Ale+17] propose une borne inférieure à l'équation 4.4.1 en utilisant une approximation variationnelle :

$$-\mathcal{L}_{vib} = \mathbb{E}[\log q(y|z)] - \beta KL(p(z|x)||q(z)) \tag{4.4.3}$$

où $q(y|z)$ est l'approximation variationnelle de $p(y|z)$ et $q(z)$ approxime $p(z)$. Maximiser l'équation 4.4.1 revient à minimiser L_{vib}.

[Oh+19] propose d'utiliser ce framework pour construire des plongements probabilistes d'images. Avec une architecture de réseaux de neurones siamois et une fonction de perte contrastive, ils séparent les paires d'images positives ($y = 1$) et négatives ($y = 0$). Nous pouvons étendre ce modèle aux plongements de documents et d'auteurices, déjà pour l'attribution d'auteurices. Chaque auteurice a (respectivement document d) est associé à une représentation z_a (respectivement z_d) latente (non-observée). Nous construisons alors un ensemble de paires (a, d) avec pour étiquette $y_a = 1$ si a a effectivement écrit d. Dans le même temps, nous tirons aléatoirement k paires négatives (a', d) avec pour étiquette $y_a = 0$, où a' n'a pas écrit d. Les lois d'encodage $p(z|x)$ pour les auteurices et les documents sont des lois normales, par choix de modélisation. Ces paires permettent d'entraîner le modèle en suivant l'objectif de rapprocher l'auteurice de sa production. L'attribution d'auteurices seule ne suffisant pas à garantir la capture du style littéraire, il faut guider cette apprentissage en ajoutant une contrainte stylistique.

4.4.2 Contrainte stylistique

En s'appuyant sur les travaux réalisés sur la stylistique en linguistique computationnelle, présentés notamment dans le chapitre précédent, il est possible de construire un proxy du style littéraire

à partir d'un ensemble de descripteurs grammaticaux et syntaxiques simples. Notamment, la section 3.4.2 détaille un ensemble de 303 marqueurs pertinents du style permettant d'obtenir de bons résultats en attribution d'auteurices sans contenir d'information sémantique directe. Nous traitons chaque document d pour extraire un vecteur de $r = 300$ descripteurs du style noté z_d^f. Ces marqueurs peuvent être obtenus simplement à partir des librairies usuelles de TAL (nltk, spacy, ...), permettant la tokenisation, le pos-tagging et la reconnaissance d'entité nommée notamment. En supposant que ces vecteurs capturent une approximation du style littéraire, ils peuvent servir à guider l'apprentissage de notre modèle dans cette direction.

De la même façon que pour les paires auteurices-documents, nous allons créer pour chaque document des paires document-descripteurs. Nous avons une paire (d, z_d^f) positive ($y_f = 1$) et k paires $(d, z_{d'}^f)$ négatives ($y_f = 0$), où d' est un document du corpus tiré aléatoirement, différent de d. Ces paires vont permettre d'entraîner la contrainte stylistique, à savoir le plongement z_d doit être proche de son vecteur de descripteur z_d^f.

Une représentation schématique de notre modèle, apppelé VADES (Variational Author and Document Embedding with Style) est proposé figure 4.3. Il contient alors l'ensemble de paramètres suivants. Pour chaque auteurice a, modélisé par une gaussienne, nous apprenons sa moyenne $\mu_a \in \mathbb{R}^r$ et sa variance sous la forme d'une matrice diagonale $\sigma_a^2 \in \mathbb{R}^r$. Ce sont deux couches de plongements (respectivement *Mean Embedding Layer* et *Variance Embedding Layer* Figure 4.3). Pour chaque document d, nous faisons appel à un encodeur de document entraînable qui à partir du texte fournira un vecteur $d_0 \in \mathbb{R}^{r_0}$ (le bloc encodeur sur la Figure 4.3). Les documents étant également modélisés par des gaussiennes, deux fonctions f et g nous permettent d'obtenir la moyenne $\mu_d = f(d) \in \mathbb{R}^r$ et la variance diagonale $\sigma_d^2 = g(d) \in \mathbb{R}^r$ (ce sont les deux blocs MLP de la Figure 4.3).

En se basant sur [Oh+19], il est possible d'exprimer la probabilité d'obtenir une étiquette positive ou négative en fonction des représentations des paires auteurices-documents (z_a, z_d) et des paires documents-descripteurs (z_d, z_d^f). Cette probabilité est la perte contrastive faible suivante :

$$\begin{aligned} q(y_a = 1|z_a, z_d) &= \sigma(-c_a||z_a - z_d||_2 + e_a) \\ q(y_f = 1|z_d, z_d^f) &= \sigma(-c_f||z_d - z_d^f||_2 + e_f), \end{aligned} \tag{4.4.4}$$

Où σ est la fonction sigmoid, $c_a, c_f > 0$ et $e_a, e_f \in \mathbb{R}$ sont des paramètres. Afin de contrôler l'importance apportée à la contrainte stylistique par rapport à l'objectif d'attribution d'auteurices, nous introduisons un hyper-paramètre $\alpha \in [0, 1]$. Ainsi, nous obtenons la fonction de perte issu du VIB suivante :

$$\begin{aligned} \mathcal{L} = &- (1-\alpha)\mathbb{E}_{p(z_a|x_a),p(z_d|x_d)}[\log q(y_a|z_a, z_d)] \\ &- \alpha\mathbb{E}_{p(z_d|x_d)}[\log q(y_f|z_d, z_d^f)] \\ &+ \beta\left(KL(p(z_a|x_a)||q(z_a)) + KL(p(z_d|x_d)||q(z_d))\right) \end{aligned} \tag{4.4.5}$$

Ici, $\alpha = 0$ produira des plongements qui prédiront bien la relation entre l'auteurice et sa production mais sans se baser sur les descripteurs du style. Au contraire, $\alpha = 1$ rapprochera les documents de leurs descripteurs mais pas de leurs auteurices. Ainsi, le choix de cet hyper-paramètre est crucial et doit être fait minutieusement, en fonction de l'application et du type de jeu de données utilisés, très orienté sur le style écrit ou non par exemple. Malgré les choix de modélisation, les espérances de l'équation 4.4.5 ne sont toujours pas calculables pour les encodeurs que nous souhaitons utiliser.

Ce problème est contourné en les estimant à partir de L tirages de Monte Carlo par triplet d'observations (un document, un ou une auteurice, un vecteur de marqueurs), suivant la loi $p(z|x)$ comme fait dans [Oh+19]. En notant $z^{(l)}$ la représentation correspondant au $l^{\text{ème}}$ tirages de Monte Carlo, nous obtenons :

$$\begin{aligned}\mathbb{E}[\log q\left(y_a|z_a, z_d\right)] &\approx \frac{1}{L}\sum_{l=1}^{L}\log q(y_a|z_a^{(l)}, z_d^{(l)}) \\ \mathbb{E}[\log q\left(y_d|z_d, z_d^f\right)] &\approx \frac{1}{L}\sum_{l=1}^{L}\log q(y_d|z_d^{(l)}, z_d^{f\,(l)})\end{aligned} \tag{4.4.6}$$

Nous utilisons alors l'astuce de reparamétrisation, similaire à ce qui est fait pour les auto-encodeurs variationnelles [KW14] :

$$z_a^{(l)} = \mu_a + \sigma_a \odot \epsilon,\ z_d^{(l)} = \mu_d + \sigma_d \odot \epsilon \text{ with } \epsilon \sim \mathcal{N}(0,1)$$

Cette fonction de perte peut alors être minimisée par backpropagation. Pour des raisons que nous détaillerons plus tard, notamment d'interprétabilité de l'espace latent, il est important que la dimension de ce dernier soit la même que celle du vecteur de descripteurs stylistiques. L'encodeur de texte peut, lui, produire des représentations sans contrainte sur leurs dimensions. Bien que son choix soit important, c'est une brique interchangeable de notre modèle et du framework que nous proposons. Nous allons détailler maintenant le choix d'encodeur fait, ainsi que celui des fonctions f et g.

4.4.3 Encodeur de documents

Le bloc d'entrée de notre modèle est un encodeur de texte, plongeant un document en langage naturel dans un vecteur de $\mathbb{R}^r$. De nombreuses architectures peuvent être proposées et entrainées. Cependant, nous choisissons ici d'utiliser un modèle de langue pré-entrainé.

Les grands modèles de langue (LLM), entrainés sur des jeux de données massifs, sont maintenant facilement accessibles en ligne (notamment via Huggingface). Ils sont terriblement efficaces sur de nombreuses tâches de TAL après seulement quelques epochs de spécialisation. Le framework VIB permet d'introduire facilement un encodeur de texte pré-entrainé [MBH21]. Plusieurs travaux, [MBH21 ; Gou+22b], extraient les moyennes et variances des sorties de l'encodeur avec des MLP. Cette approche est simple et rapide. Nous faisons le même choix, en construisant f et g comme deux MLP à deux couches avec activation tanh et linéaire, prenant la même entrée et ayant pour dimension cachée r_0. La dimension de sortie est elle identique à la taille du vecteur de descripteurs ($r = 300$).

Concernant l'encodeur à proprement parler, plusieurs contraintes émergent. Nous souhaitons que notre modèle puisse capturer efficacement l'information stylistique d'un document. [TGV21 ; Cla+19b] montrent que les LLM s'appuient déjà sur des notions complexes de syntaxe et de grammaire pour construire leurs représentations. Ils ont donc le pouvoir explicatif requis pour notre objectif.

La seconde contrainte concerne la longueur des documents. Notre modèle doit pouvoir traiter des textes longs, fréquents dans un contexte littéraire. Des pièces de théâtre, des romans ou des essais, là où le style écrit affleure le plus. C'est une contrainte forte, par exemple BERT est limité

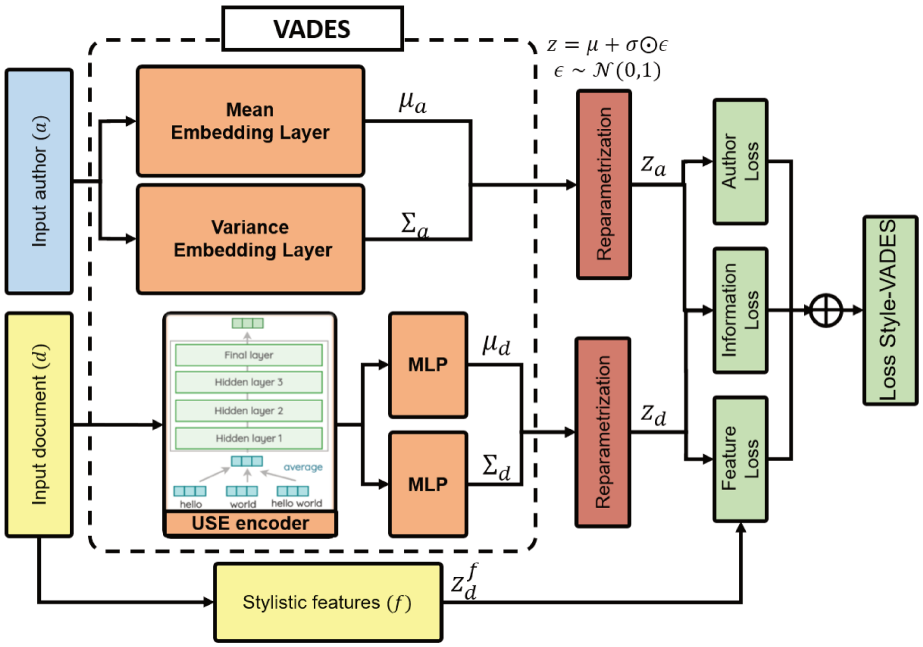


FIGURE 4.3 – Schéma du modèle VADES (Variational Author and Document Embedding with Style)

à 512 tokens. Les modèles les plus récents vont jusqu'à 4096. Certaines alternatives modifient le type d'attention appliquer par les Transformers à des documents longs [Zah+20 ; BPC20]. Afin de contourner ce problème, nous faisons le choix d'utiliser la version Deep Averaging Network (DAN) de l'Universal Sentence Encoder (USE). En plus de ne présenter aucune contrainte de taille, ce modèle est plus rapide que les méthodes à base de Transformers et surpasse SBERT sur la prédiction de marqueurs stylistiques [TGV21] (voir section 3.6.4). Enfin, nous pouvons noter que notre modèle est transposable dans toutes les langues (car dépendant d'un encodeur pris sur l'étagère) et peut produire des représentations de nouveaux documents. Il n'est pas possible d'inférer des représentations d'auteurices hors du jeu d'entraînement puisqu'elles proviennent non pas d'un encodeur mais d'une couche de plongements. Cela peut être limitant dans le cadre de la littérature numérique par exemple où des auteurices peuvent émerger continuellement.

Il faudrait alors ré-entraîner tout le modèle, même si des solutions plus simples peuvent être envisagées, comme geler les plongements des auteurs de l'entraînement et ne fine-tuner que la partie traitant les documents et les nouveaux auteurs et nouvelles autrices. A priori, l'approximation la plus simple serait de les représenter par la moyenne des plongements de leurs documents. D'autres fonctions d'agrégation peuvent être envisagées, jusqu'à des MLP, des LSTM ou des Transformers, nous laissons cela à des travaux futurs.

Datasets statistics			
Dataset	**Authors**	**Avg. Tokens**	**Avg. Texts**
IMDb62	62	341(±223)	1000(±0)
BAC10	10	91(±184)	2350(±639)
BAC50	50	98(±167)	1466(±562)
R-BAC	500	243(±342)	50(±0)
R-PGD	664	2315(±961)	10(±0)

TABLE 4.1 – Statistiques descriptives des jeux de données utilisés. BAC : Blog Authorship Corpus, PGD : Project Gutenberg Dataset.

4.5 Evaluation

4.5.1 Jeux de données

Nous allons manipuler différents jeux de données en fonction des tâches d'évaluation. Ils sont tous très fréquent dans le domaine de l'attribution d'auteurices et de l'analyse du style écrit. Comme au chapitre précédent, nous utilisons le Projet Gutenberg et le Blog Authorship Corpus. Pour rappel, le Projet Gutenberg est une bibliothèque multilingue de productions variées (romans, essais, théâtre, ...) dans le domaine public américain. Notre extraction ici se limite à l'anglais et aux 664 auteurices les plus prolifiques (au moins 10 oeuvres) dont nous sélectionnons aléatoirement 10 textes. Cette valeur permet à la fois de conserver un assez grand nombre d'auteurices tout en ayant suffisamment d'oeuvres pour chacun. Ce corpus sera noté R-PGD. Le Blog Authorship Corpus est lui un ensemble de posts de blogs de 19 320 auteurices assemblés dans les années 2000. De la même façon, nous créons notre corpus de 500 auteurices avec 50 posts chacun, noté R-BAC. Ces corpus seront utilisés pour évaluer la capacité à capturer différents styles d'écriture, ainsi nous le souhaitons équilibré en terme d'oeuvres par auteurices. Pour évaluer les capacités de notre modèle dans un cadre déséquilibré plus proche du réel, nous utiliserons également les sous-ensembles fréquents dans la littérature, BAC50 et BAC10. Ce sont respectivement les 50 et les 10 auteurices les plus prolifiques avec l'ensemble de leur production. Ces deux corpus reflètent chacun un aspect des données sur lesquelles nos travaux devraient être appliqués, le côté littéraire pour le Project Gutenberg et nativement numérique pour le Blog Authorship Corpus.

IMDb

Egalement, nous introduisons le jeu de données IMDb (Internet Movie Database), qui est l'un des plus utilisé dans la littérature en attribution d'auteurices. Il a été introduit en 2014 par [SZB14] et est composé de 271 000 critiques de films provenant de 22 116 utilisateurs du site web IMDb. Cependant, la plupart des travaux s'évaluent sur sa réduction à seulement 62 auteurices et 1000 critiques chacuns, noté IMDb62. Nous nous limiterons donc à cette réduction également. Comme nous le montrerons par la suite, les scores d'attribution d'auteurices sont très élevés sur ce corpus, proche des 96 %, notamment à cause du faible nombre d'auteurices qu'il contient et sur une catégorie de texte très spécifique. Un récapitulatif des statistiques descriptives des différents jeux de données utilisés est fourni table 4.5.1.

4.5.2 Compétiteurs

Nous allons évaluer le modèle VADES et le confronter à plusieurs modèles existants. Comme baseline simple, nous utilisons la version DAN de l'USE [Cer+18]. Comme modèle de représentations d'auteurices, nous évaluons tout d'abord la partie Content-Info du modèle Aut2Vec [Jaw+16], présenté en section 2.4.3. Pour rappel, à partir de la distance entre les représentations d'auteurices et de documents puis d'un MLP, Content-Info produit la probabilité que ces associations soient effectivement observées dans le corpus. Ce modèle s'appuie sur des plongements de mots pré-entrainés type Word2Vec et n'est pas inductif. Nous évaluons également le modèle Ngram Doc2Vec de [Mah+19] présenté plus haut.

Enfin, nous nous comparons à deux méthodes de plongements de documents capturant le style. Le premier est le modèle DeepStyle [Hay+20], DistilBERT spécialisé sur la tâche d'attribution d'auteurices sur un large corpus. Ici, nous le spécialisons en plus sur nos jeux de données sur quelques epochs.

Le second le modèle SRoBERTa que nous spécialisons sur la tâche de vérification d'auteurices sur nos corpus comme évoqué dans [ZJ21]. Chaque texte est découpé en extraits de 256 tokens afin de ne pas dépasser la taille limite du modèle RoBERTa. Ensuite, pour chaque extrait, un exemple positif (même auteurice) et négatif (autre auteurice) est tiré aléatoirement. Le modèle est ensuite entraîné en utilisant la *max-margin loss* présenté en équation 4.2.3.

Dans le cas où les modèles ne produisent pas de plongements d'auteurices (USE, DeepStyle et SRoBERTa), leurs représentations sont obtenues en moyennant la représentation de leurs documents dans les données d'entraînement.

4.5.3 Tâches d'évaluation

VADES sera évalué sur la tâche d'attribution d'auteurices sur les trois corpus de référence pour cette tâche, à savoir IMDb62, BAC10 et BAC50. En effet, ces corpus ont été créés spécifiquement pour cette tâche là et reviennent très souvent dans la littérature ce qui permet de se comparer à de nombreuses méthodes de l'état de l'art. La métrique utilisée est l'accuracy et pour chaque jeux de données 20% des données sont utilisées en guise de test, 70% pour l'entraînement et 10% pour la validation croisée avec groupe.

La seconde tâche d'évaluation est la régression de marqueurs stylistiques, présentée en section 3.6.4. Nous suivons le protocole développé lors du chapitre précédent, pour chaque document un ensemble de 303 descripteurs du style sont extraites avec les librairies habituelles en TAL, puis agrégés par auteurice. Ainsi, nous obtenons pour chacun une approximation de leur style littéraire. Le but est alors de prédire la valeur de ces marqueurs à partir du plongement de l'auteurice correspondant. Nous utilisons comme modèle de régression une régression à vecteur de support (SVR) avec un noyau de fonction de base radiale, car c'est ce qui offre les meilleurs résultats tout en restant suffisamment rapide. Nous appliquons un schéma de cross-validation à 10 groupes. La métrique correspondante est l'erreur des moindres carrés (MSE). A noter qu'un modèle prédisant pour chaque marqueur sa valeur moyenne sur l'ensemble du corpus obtiendra une MSE de 1.0, car chaque marqueur est centré et réduit, c'est donc une valeur référence pour les modèles ne performant pas bien, l'idéal étant bien évidemment une MSE de 0.0.

4.5.4 Paramètres

VADES

Concernant le choix de l'encodeur, malgré les nombreux avantages offerts par USE, nous avons malgré tout voulu tester BERT et sa variante plus sobre DistilBERT. Les limites de taille d'entrée de ces modèles imposent de séparer les textes longs, notamment du projet Gutenberg et d'extraire pour chacun de ces extraits les marqueurs correspondants. Si cela permet d'augmenter le nombre d'exemples d'entraînement, le temps de calcul s'en trouve démultiplié. Nous avons très vite été confrontés aux limites de temps de calcul imposées sur l'HPC Jean Zay, malgré l'utilisation d'entraînement distribué et des méthodes de précision mixte et ce sans atteindre des résultats probants. Nous laissons donc ces tests à des travaux ultérieurs.

Nous avons testé une variation de la fonction de perte liées aux marqueurs stylistiques, à savoir une simple distance euclidienne entre la représentation du document et son vecteur de marqueurs, à opposer à l'entropie croisée présentée plus haut. Les résultats pour les deux variations sont présentées plus bas, mais l'entropie croisée, plus souple, permet de mieux gérer l'ensemble des descripteurs dont les vecteurs peuvent être parcimonieux. Les architectures des fonctions f et g ont été présentées en section précédentes. Ce sont des MLP à deux couches avec BatchNormalization, dropout de 0.2 et régularisation L_2 ($1e^{-5}$). Pour le reste des hyperparamètres, la grille de recherche utilisée est présentée table 4.2. Pour L, nous obtenons un bon compromis entre vitesse et accuracy en faisant 10 tirages. Nous atteignons rapidement un plateau en augmentant sa valeur. De la même façon, nous tirons 10 exemples négatifs pour chaque document. Nous utilisons l'algorithme d'optimisation Adam avec un pas d'apprentissage de $1e^{-3}$. Le nombre d'epochs optimal pour l'entraînement varie grandement en fonction du nombre d'auteurices à considérer dans le corpus. Pour éviter ce problème, nous pratiquons une méthode proche de l'early stopping : lorsque les résultats sur le jeu de validation diminuent, nous divisons par 2 le pas d'apprentissage jusqu'à atteindre un plateau (en général après trois epochs). De cette façon, l'entraînement sur R-PGD (664 auteurices) nécessite 18 epochs, quand il n'en faut que 3 sur BAC10 (10 auteurices).

A noter que de nombreux travaux continuent à être publiés sur la meilleure manière de spécialiser des grands modèles de langue et notamment sur le choix du schéma de pas d'apprentissage le plus performant. C'est un axe d'amélioration possible de notre modèle.

Comme évoqué plus haut, le choix de l'hyperparamètre α est très important et dépend du type de jeu de données et de la tâche d'évaluation. Ici, pour chaque tâche nous l'avons sélectionné par gridsearch. Nous présentons également les résultats pour d'autres valeurs, ainsi qu'une analyse de son influence sur les performances de VADES.

Enfin, nous proposons aussi une variante de VADES sans le framework VIB et donc sans la partie variationnelle, afin de justifier de son intérêt dans notre cadre. Ce modèle sera noté VADES no-VIB. Les représentations sont alors statiques (sans variance) et il n'y a plus d'échantillonnage.

Content-Info

Nous choisissons d'initialiser les plongements des documents et des auteurices avec leurs représentations via USE (la moyenne des documents pour les auteurices), plutôt qu'avec Word2Vec car cette approche fournit de meilleurs résultats. L'espace latent a pour dimension 512 et la couche intermédiaire possède 256 neurones. Une nouvelle fois, le réseau de neurones est optimisé avec Adam et un pas d'apprentissage de $1e^{-3}$, avec 10 exemples négatifs tirés pour chaque paire auteurice-document. Ces paramètres ont été obtenus après une gridsearch.

Ngram Doc2Vec

Nous utilisons le code fournit dans l'article [1] pour l'extraction des trigrams de caractères anno-

1. Github Ngram Doc2Vec

Grille de recherche des hyperparamètres de VADES	
Hyperparamètres	**Grille**
Nb de paires négatives	{1, 5, **10**, 20}
Monte Carlo sampling	{1, 5, **10**, 20}
Pas d'apprentissage	{1e-2, **1e-3**, 1e-4, 1e-5}
β	{1e-1, 1e-2, ..., **1e-12**}
Type de loss de marqueurs	{L2, **Cross-Entropy**}

TABLE 4.2 – Grille de recherche utilisée pour la sélection des hyperparamètres de VADES. Les valeurs retenues sont en gras.

tés. Les meilleurs résultats sont obtenus avec un chevauchement partiel de ces derniers. L'espace latent est de dimension 300, nous filtrons les trigrams ayant une fréquence d'apparition inférieure à 2% et les trigrams les plus fréquents sont sous-échantillonnés aléatoirement avec une probabilité de $1e^{-5}$. Le meilleur modèle pour chacun des jeux de données est la variante SGNS de Word2Vec. La représentation d'un nouveau document est obtenu en pondérant chacun des plongements des trigrams qui le composent à l'inverse de sa fréquence d'apparition dans les documents. Nous tirons 10 exemples négatifs pour chaque paire et entraînons le modèle sur 100 epochs avec un pas d'apprentissage initial de 0.25.

DeepStyle

Ici, nous utilisons également le modèle pré-entraîné fournit avec l'article [2]. Pour le spécialiser sur nos corpus, nous pratiquons l'early stopping avec un pas d'apprentissage de $5e^{-5}$ et des batchs de taille 64. Le nombre d'epochs d'entraînement ne dépasse jamais 5, quelque soit le jeu de données utilisé.

SRoBERTa

Une nouvelle fois, nous nous appuyons sur le code proposé dans l'article [3]. Nous utilisons comme modèle de langue de base RoBERTa, car il fournit de meilleurs résultats que BERT. Les représentations issues de la dernière couche d'attention sont agrégées dans une représentation unique via une couche d'attention pooling. Les textes trop longs sont découpés en portion de taille maximale 128 tokens. Chaque extrait est associé à un extrait de son auteurice (de la même oeuvre ou non) et 6 exemples négatifs (d'autres auteurices) sont tirés. Les seuils de la *max-margin loss* sont sélectionnés par gridsearch et valent 0.4 pour le plus bas et 0.6 pour le plus haut. L'optimisation est faite avec Adam, un pas d'apprentissage de $1e^{-5}$ avec des batchs de taille 64 et de l'early stopping sur 10 epochs maximum. La probabilité de masquer un token est laissé à 0.1 comme dans l'article, de même que le paramètre $\alpha = 30$ de la fonction de perte.

2. https ://github.com/hayj/deepstyle
3. https ://github.com/lingjzhu/idiolect

4.6 Résultats

4.6.1 Pour l'attribution d'auteurices

Les résultats pour l'attribution d'auteurices sur chacun des trois jeux de données de référence IMDb62, BAC10 et BAC50 sont proposés table 4.3. Nous comparons ici notre modèle (la valeur d'α est précisé à chaque fois) avec plusieurs méthodes de l'état de l'art en attribution d'auteurices. A noté que ces méthodes ne sont pas nécessairement des modèles d'apprentissage de représentations. Par exemple, BertAA [Fab+20] performant le mieux sur BAC10 et BAC50 spécialise BERT sur l'attribution d'auteurices avant de combiner ses représentations avec un ensemble de descripteurs du style et de n-gram de caractères dans une régression logistique. Les bons résultats obtenus avec cette méthode confirme l'intérêt d'intégrer des marqueurs du style pour l'apprentissage de représentations d'auteurices. Cette observation est aussi appuyée par le fait que VADES performe mieux avec $\alpha = 0.1$ qu'avec $\alpha = 0.0$, donc en intégrant un minimum de marqueurs stylistiques plutôt que de se focaliser uniquement sur l'attribution d'auteurices.

Notre modèle est significativement dépassé uniquement par Syntax CNN, DeepStyle-ft et BertAA. Comme montré dans [Fab+20], BERT et ses variantes sont vraiment adaptés pour traiter des textes courts et des corpus équilibrés, mais difficilement utilisables dans notre framework sans augmenter les temps d'entraînement de manière critique. Le modèle Syntax CNN [Zha+18] encode chaque phrase d'un document séparément avec sa syntaxe grâce à l'arbre des dépendances associé. Malheureusement, ce modèle était difficile à reproduire. Il nécessite un pré-traitement des données dans un format très spécifique, difficilement applicable à des textes longs comme ceux du Projet Gutenberg car très chronophage.

4.6.2 Pour la capture du style

Comme précisé plus tôt, les plongements d'auteurices sont utilisés pour prédire la valeur de différents marqueurs stylistiques correspondant à chacun des auteurices. Comme montré table 4.3, utiliser ces descripteurs et une simple régression logistique permet d'atteindre des scores décents en attribution d'auteurices, proches de ceux d'USE, méthode de l'état de l'art en représentation de phrases. Cela confirme qu'ils sont une bonne approximation du style littéraire, étant donné qu'ils ne contiennent aucune information sémantique directe. Ainsi, un modèle capable de les capturer est capable de représenter le style écrit.

Les résultats concernant l'évaluation de la capture du style par les différents modèles de plongements sont présentés table 4.4. Comme attendu, notre modèle surpasse aisément tous les compétiteurs sur tous les axes. DeepStyle, entraîné uniquement sur l'attribution d'auteurices est le moins bon modèle. Bien que cette approche se base sur la spécialisation d'un modèle de langue déjà apte à capturer des notions syntaxiques, ce n'est pas l'information qui semble être retenue par le réseau in fine. Cette idée est confirmée avec le modèle SRoBERTa. En effet, au cours de l'entraînement sur la tâche de vérification d'auteurices, à mesure que la fonction de perte diminue, les scores en régression de marqueurs régressent. Ces modèles semblent se focaliser principalement sur l'information sémantique pour prédire les relations auteurices-documents.

VADES de son côté est guidé par la fonction de perte stylistique pour se focaliser sur la capture de notions linguistiques complexes. USE obtient sur la plupart des axes les seconds meilleurs scores, ce qui confirme l'intérêt d'utiliser ce modèle comme encodeur dans notre framework. Il surpasse même les modèles spécifiques d'apprentissages de représentation d'auteurices comme Ngram Doc2Vec et Content-Info.

Méthodes	IMDb62 62 auteurices	Blog Authorship Corpus 10 auteurices	Blog Authorship Corpus 50 auteurices
USE	60.2 (0.5)	40.7 (0.4)	24.7 (0.4)
Descripteurs stylistiques + LR	88.2 (0.6)	40.9 (0.5)	28.4 (0.6)
LDA+Hellinger* [EK13]	82	52.5	18.3
Impostors* [KY14]	x	35.4	22.6
Word Level TF-IDF*	91.4	x	x
CNN-Char* [RGB16]	91.7	61.2	49.4
C.Att + Sep.Rec.* [SZL19]	91.8	x	x
Token-SVM* [SZB14]	92.5	x	x
SCAP* [Fra+06]	94.8	48.6	41.6
Cont. N-gram* [SVS17]	94.8	61.3	52.8
(C+W+POS)/LM* [Kam+17]	95.9	x	x
N-gram + Style* [Mah+19]	95.9	x	x
N-gram CNN* [Zha+18]	x	63.7	53.1
Syntax CNN* [Zha+18]	96.2	64.1	56.7
DeepStyle [Hay+20]	**96.7 (0.8)**	64.3 (1.0)	58.5 (0.9)
BertAA* [Fab+20]	93.0	**65.4**	**59.7**
VADES no-VIB (0.5)	91.3 (1.1)	60.9 (1.2)	50.2 (1.2)
VADES (0.0)	94.9 (1.2)	62.6 (1.2)	52.4 (1.2)
VADES (0.1)	95.6 (1.2)	63.8 (1.2)	53.8 (1.2)
VADES (0.5)	91.6 (1.3)	61.0 (1.3)	50.5 (1.2)

TABLE 4.3 – Attribution d'auteurices sur IMDb62 et le Blog Authorship Corpus. Les résultats avec une * sont rassemblés de différents papiers, x correspond à un résultat manquant pour un corpus donné. Le meilleur modèle est en gras, le second est souligné, l'écart type est entre parenthèses.

	Score de régression moyen et standard déviation (modèle SVR) sur le corpus R-PGD							
Méthodes	**Lettres**	**Nombres**	**Structurel**	**Ponctuation**	**Mots outils**	**TAG**	**NER**	**Index**
Content-Info	0.67 (0.17)	0.88 (0.12)	0.55 (0.19)	0.68 (0.16)	0.72 (0.19)	0.65 (0.17)	0.74 (0.14)	0.50 (0.16)
Ngram Doc2Vec	0.63 (0.20)	0.88 (0.12)	0.51 (0.20)	0.58 (0.21)	0.68 (0.19)	0.59 (0.19)	0.71 (0.14)	0.45 (0.15)
SRoBERTa	0.69 (0.24)	0.91 (0.10)	0.50 (0.18)	0.79 (0.21)	0.76 (0.20)	0.64 (0.21)	0.79 (0.13)	0.44 (0.18)
USE	0.61 (0.27)	0.86 (0.09)	0.34 (0.18)	0.59 (0.26)	0.65 (0.24)	0.45 (0.29)	0.65 (0.17)	0.27 (0.15)
DeepStyle	0.79 (0.16)	0.92 (0.09)	0.65 (0.15)	0.82 (0.17)	0.84 (0.13)	0.74 (0.14)	0.84 (0.08)	0.60 (0.14)
VADES L_2-loss (0.5)	0.53 (0.22)	0.65 (0.09)	0.30 (0.14)	0.63 (0.26)	0.54 (0.20)	0.41 (0.25)	0.60 (0.15)	0.23 (0.09)
VADES no-VIB (0.5)	0.55 (0.23)	0.67 (0.11)	0.32 (0.14)	0.66 (0.27)	0.58 (0.21)	0.44 (0.27)	0.62 (0.16)	0.24 (0.14)
VADES (0.0)	0.84 (0.24)	0.91 (0.12)	0.66 (0.13)	0.85 (0.18)	0.91 (0.15)	0.71 (0.23)	0.88 (0.09)	0.61 (0.16)
VADES (0.5)	0.50 (0.22)	0.60 (0.11)	0.28 (0.14)	0.62 (0.27)	0.53 (0.21)	0.40 (0.27)	0.58 (0.15)	0.20 (0.11)
VADES (0.9)	**0.47 (0.22)**	**0.53 (0.10)**	**0.26 (0.13)**	**0.59 (0.28)**	**0.50 (0.21)**	**0.39 (0.26)**	**0.56 (0.15)**	**0.19 (0.10)**
	Score de régression moyen et standard déviation (modèle SVR) sur le corpus R-PGD							
Méthodes	**Lettres**	**Nombres**	**Structurel**	**Ponctuation**	**Mots outils**	**TAG**	**NER**	**Index**
Content-Info	0.80 (0.15)	0.85 (0.07)	0.62 (0.23)	0.92 (0.09)	0.87 (0.12)	0.90 (0.05)	0.93 (0.07)	0.70 (0.29)
Ngram Doc2Vec	0.77 (0.16)	0.88 (0.05)	0.67 (0.16)	0.78 (0.13)	0.84 (0.12)	0.82 (0.09)	0.86 (0.11)	0.67 (0.13)
SRoBERTa	0.69 (0.27)	0.81 (0.06)	0.53 (0.24)	0.86 (0.17)	0.88 (0.15)	0.88 (0.07)	0.92 (0.11)	0.54 (0.28)
USE	0.67 (0.25)	0.83 (0.05)	0.45 (0.20)	0.78 (0.17)	0.81 (0.17)	0.63 (0.21)	0.80 (0.17)	0.38 (0.18)
DeepStyle	1.05 (0.09)	1.05 (0.07)	1.01 (0.05)	0.98 (0.22)	1.05 (0.09)	0.95 (0.19)	0.91 (0.20)	1.03 (0.07)
VADES (0.9)	**0.52 (0.23)**	**0.55 (0.09)**	**0.31 (0.17)**	**0.76 (0.22)**	**0.67 (0.20)**	**0.57 (0.20)**	**0.73 (0.18)**	**0.32 (0.20)**

TABLE 4.4 – Régression de descripteurs stylistiques sur R-PGD et R-BAC. MSE (écart type entre parenthèses) sur la prédiction de marqueurs du style à partir des plongements d'auteurices via SVR. Les 303 marqueurs sont regroupés par familles. En gras, le meilleur score pour chaque axe, le second est souligné. Notre modèle (α entre parenthèse) obtient les meilleurs résultats avec $\alpha = 0.9$.

4.6.3 Etude de la sensibilité des paramètres de VADES

Nous analysons ici les résultats obtenus par différentes configuration de VADES.

Perte stylistique L_2 et entropie croisée

En table 4.4 nous pouvons comparer l'utilisation de l'erreur quadratique moyenne pour la fonction de perte stylistique (VADES L_2-loss) à l'utilisation de l'entropie croisée (VADES). La première impose clairement une contrainte plus forte et on pourrait s'attendre donc à obtenir de meilleurs résultats. Ça n'est pas le cas, l'entropie croisée semble offrir plus de souplesse dans l'apprentissage de représentation ce qui justifie notre choix de fonction de perte.

Avec VIB et sans VIB

Les tables 4.3, 4.4 permettent de comparer l'intérêt du framework VIB et de la modélisation variationnelle sur les deux tâches d'évaluation que sont l'attribution d'auteurices et la régression de descripteurs stylistiques. Si l'apport sur la première semble faible, avec des écarts non-significatifs sur IMDb62 et BAC50, ce n'est pas le cas sur la seconde. Les vecteurs de descripteurs peuvent être parcimonieux, notamment sur des textes assez courts, ou certains POS-tag, entités nommées ou mots outils ne sont pas présents. La modélisation variationnelle permet de mieux gérer cette aspect là des vecteurs de descripteurs et donc d'obtenir de bien meilleurs scores de MSE, d'où l'intérêt de l'utilisation du VIB. Il semble offrir plus de flexibilité ce qui est clé quand il s'agit de capturer une notion aussi complexe que le style écrit.

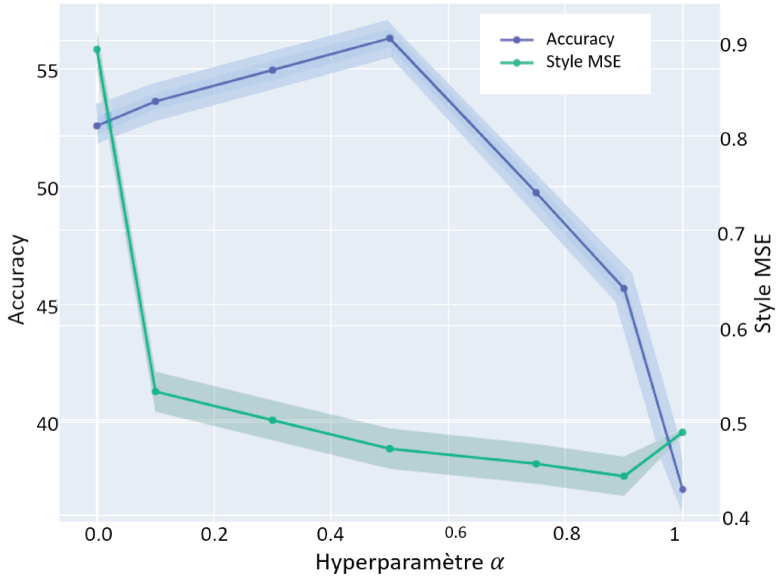


FIGURE 4.4 – Impact de l'hyperparamètre α. Nous affichons l'évolution de la métrique de style (score MSE moyen) et de l'accuracy en fonction d'α sur le corpus R-PGD

Choix de l'hyperparamètre α Sur la figure 4.4 nous affichons l'évolution de nos métriques d'évaluation pour nos deux tâches en fonction des différentes de l'hyperparamètre α sur le corpus R-PGD. Ce dernier permet de pondérer l'importance accordée à l'attribution d'auteurices ou au style écrit. Ajouter de l'information stylistique permet d'augmenter les résultats en attribution d'auteurices en apportant une information supplémentaire. Cela force le modèle à extraire des informations stylistiques discriminantes des textes. Cependant, là où $\alpha = 0.1$ offre la meilleure accuracy sur IMDb62, BAC10 et BAC50 (voir table 4.3), l'équivalent est obtenu pour $\alpha = 0.5$ sur R-PGD. Les textes plus longs et plus littéraire du projet Gutenberg semblent porter plus d'information stylistique. Il est donc important de bien connaître son corpus pour choisir la valeur d'α la plus pertinente.

De la même manière, éteindre complètement l'attribution d'auteurices ($\alpha = 0.0$) entraîne une détérioration de la capture du style. En effet, les auteurices tendent à avoir un style cohérent dans l'ensemble de leur production. Ainsi, rapprocher l'auteurice de ses documents aide aussi à capter ses habitudes d'écriture. Ce phénomène est appelé la *cohérence intra-auteurice* dans la littérature [ZJ21 ; Hay+20].

4.6.4 Visualisation et analyse de la variance

Nous proposons ici une analyse qualitative des représentations d'auteurices et de documents produites par VADES sur le Projet Gutenberg. Une première représentation d'un ensemble réduit d'auteurices est proposée figure 4.5. Nous pouvons observer des clusters correspondant à chaque auteurice. Cependant, comme attendu, les oeuvres qui diffèrent de celles de leurs auteurices se

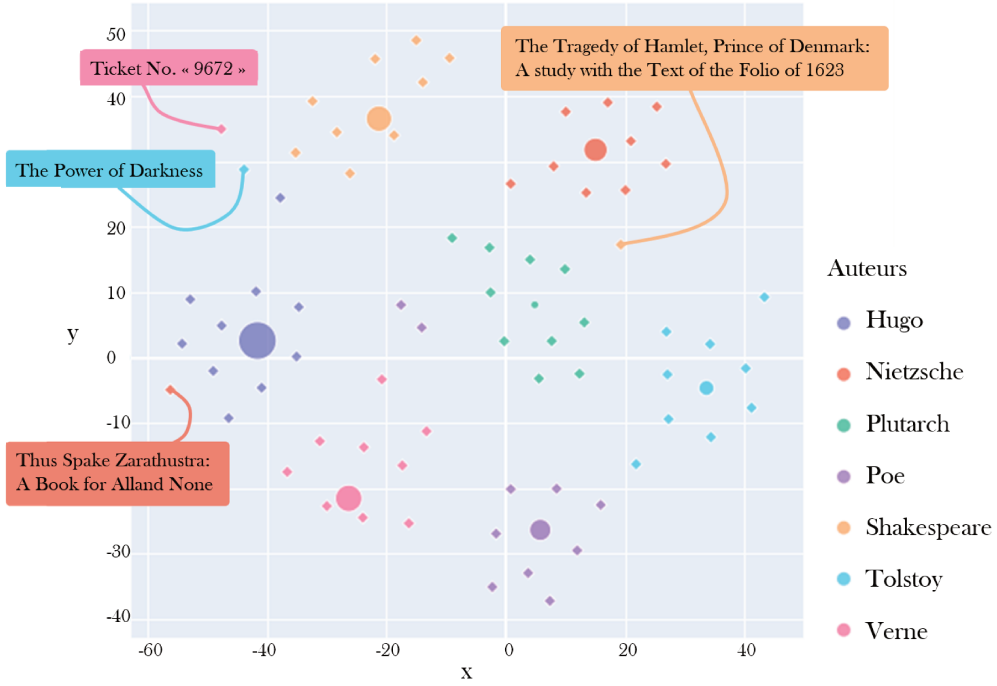


FIGURE 4.5 – Plongements d'auteurices et de documents issus du corpus R-PGD. Nous proposons ici une projection 2D via T-SNE des plongements d'auteurices et de documents produites par VADES ($\alpha = 0.5$). Les diamants correspondent aux oeuvres, les points aux auteurices. La taille du point correspond à la variance de l'auteurice apprise.

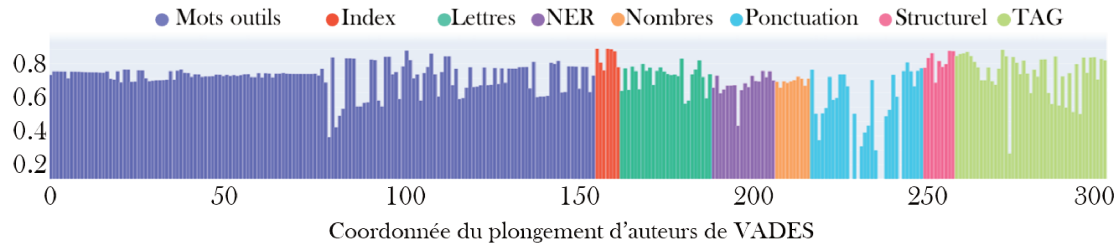


FIGURE 4.6 – Score de corrélation entre la $i^{\text{ème}}$ coordonnée du plongement d'auteurices de VADES et le $i^{\text{ème}}$ marqueur stylistique sur le corpus R-PGD. Les quelques valeurs nulles de la catégories Ponctuation correspondent à des éléments non présents dans ce corpus.

retrouvent proches de productions qui leur ressemblent en terme de style. Par exemple, *Ainsi parlait Zarathoustra* de Nietzsche, oeuvre entre la poésie, le roman et la philosophie est plus proche des travaux d'Hugo, tandis que le reste de ses textes sont principalement des essais. Il en va de même avec *La puissance des ténèbres* de Tolstoï, un drame en 5 actes se rapprochant plus de Shakespeare que de ses autres romans. Enfin, la version d'Hamlet présente dans le corpus est entièremment commentée et se retrouve donc plus proche des travaux analytiques et philosophiques de Nietzsche et Plutarch. La variance apprise des auteurices est également représentée. Hugo, qui a écrit tant des romans que de la poésie et des drames a une plus grande variance que le reste des auteurices.

La table 4.5 montre les 10 auteurices ayant la plus forte variance apprise par VADES. Nous y retrouvons Casanova, ainsi qu'Hugo ou Bertrand Russell par exemple. L'ensemble des oeuvres du premier sont ses mémoires et autobiographies, datant de la fin du 18ème siècle. Elles ont été retravaillées et réécrites par Casanova plusieurs fois au cours de sa vie, avant de traverser de longs épisodes de censures et de nombreuses traductions et versions jusqu'à nos jours, du français vers l'allemand puis vers l'anglais. Cela peut expliquer ses grandes variations dans le style. Les textes de Betrand Russell, mathématicien britannique du 19ème, sont eux aussi extrêmement variés, allant d'essais historiques, politiques, à des ouvrages fondateurs en mathématiques. Enfin, l'auteurice correspondant à "Library of Congress. Copyright Office" est en réalité un ensemble de documents listant par période les titres et les auteurices tombant dans le domaine public. La variété des titres proposés suffit à expliquer la grande variance ici.

A l'opposé, l'American Thread Company publie comme document des magazines décrivant des patrons et des modèles de crochets et de coutures, dans un cadre très spécifique et formaté. Autre exemple, J.W. Duffield est un auteur jeunesse canadien du début du 20ème siècle, qui a écrit notamment une série d'oeuvres autour du personnage de Bert Wilson (*Bert Wilson in the Rockies, Bert Wilson at Panama, Bert Wilson at the Wheel, ...* d'où les faibles variations dans sa plume. Il en va de même de Ralph Connor, auteur canadien de la même période, dont les romans très descriptifs suivaient très souvent des schémas narratifs très similaire.

4.6.5 Interprétabilité de l'espace de plongements

Puisque nous utilisons la distance L_2 comme mesure de l'écart entre les représentations des documents et les vecteurs de descripteurs stylistiques (voir équation 4.4.2), chacun des 300 axes du

auteurices avec la plus faible variance	**auteurices avec la plus grande variance**
American Thread Company	Vaknin, Samuel
Stevenson, Burton Egbert	Library of Congress. Copyright Office
Zangwill, Israel	Vasari, Giorgio
Connor, Ralph	Cannon, Richard
Erckmann-Chatrian	Hugo, Victor
Herrick, Robert	Calvert, Albert Frederick
Burroughs, Edgar Rice	Hall, E. Raymond (Eugene Raymond)
Duffield, J. W	Kerr, Robert
Mitchell, S. Weir (Silas Weir)	Russell, Bertrand
A. L. O. E	Casanova, Giacomo

TABLE 4.5 – auteurices avec les plus faibles et plus grandes variances (en terme de norme euclidienne) apprises par VADES sur le corpus R-PGD

plongement obtenu correspond à un marqueur du style donné. Comme vu précédemment, la fonction de perte contrastive faible permet de garantir cette contrainte L_2 tout en étant plus flexible qu'une simple perte de régression L_2. Sur la figure 4.6 nous montrons le score de corrélation de Pearson entre l'$i^{\text{ème}}$ marqueur de style et la coordonnée correspondante dans les plongements d'auteurices issus de VADES. Ces scores de corrélation sont toujours maximum pour chaque marqueur par rapport à toutes les autres coordonnées possibles. Afin d'illustrer un peu plus cet aspect de notre modèle, nous affichons figure 4.7 pour une sélection de 3 axes leurs valeurs en fonction du marqueur correspondant. Une nouvelle fois les corrélations apparaissent clairement. L'espace latent construit par VADES est donc entièrement interprétable en cela que chaque axe correspond à une notion linguistique claire. Dans le contexte d'un projet pluridisciplinaire c'est une véritable valeur ajoutée. A noter que c'est la raison pour laquelle la dimension de l'espace latent doit être identique à celle des vecteurs de descripteurs. Nos représentations visant à être utilisables par des chercheurs et chercheuses en linguistiques, en littérature et par le grand public, cet aspect interprétable par construction est un vrai plus.

4.7 Conclusion et perspectives

Dans ce chapitre, nous avons fait une revue des modèles d'apprentissage de représentation d'auteurices se focalisant sur la capture du style. N'étant que très peu nombreux, nous avons également présenté les méthodes de plongements de documents avec le même objectif. Devant le peu de modèle proposant d'apprendre à la fois des plongements d'auteurices et de documents et devant les limites des méthodes existantes, nous avons proposé notre contribution, VADES (Variational Author and Document Embedding with Style).

Elle présente l'avantage de guider l'attribution d'auteurices en utilisant des marqueurs du style afin de forcer l'encodeur à se focaliser sur la capture du style écrit. Ainsi, notre modèle surpasse aisément tous les compétiteurs évalués en capture du style et rivalise avec l'état de l'art en attribution d'auteurices. Bien que s'appuyant sur l'encodeur USE [Cer+18] afin de traiter des documents de longueurs variées, notre framework peut s'utiliser avec n'importe quel encodeur de texte et donc dans n'importe quelle langue et peut proposer des représentations de nouveaux documents. Par

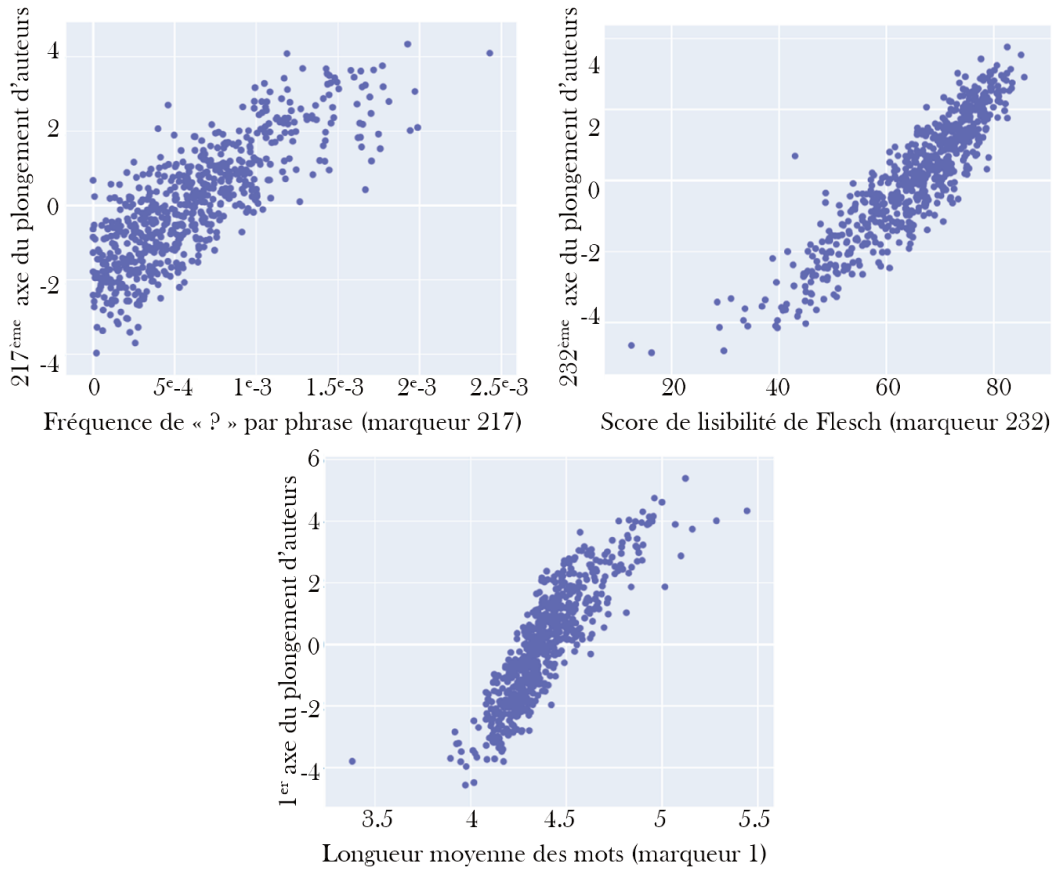


FIGURE 4.7 – $i^{\text{ème}}$ coordonnée du plongement d'auteurices de VADES en fonction du $i^{\text{ème}}$ marqueur stylistique sur le corpus R-PGD, pour une sélection de 3 marqueurs données. La corrélation entre chaque marqueur et son axe correspondant transparaît clairement.

construction, l'espace de représentation est interprétable et chaque axe des plongements correspond à un descripteur linguistique. Enfin, le cadre variationnel du VIB permet d'utiliser les variances des auteurices comme une approche de la variété stylistique de sa production.

Cependant, plusieurs améliorations de notre modèle sont envisageables. Si les encodeurs comme BERT semblent trop souffrir de la limite de taille qu'ils imposent en entrée, il pourrait être intéressant de tester les approches à base de Transformer pour document long commme LongFormer [BPC20] et BigBird [Zah+20]. Ou encore d'y incorporer les grands modèles de langue très récents comme LLaMA. Bien que déjà évoqué lors du chapitre précédent, un autre axe d'amélioration est la sélection de descripteurs pertinents. Que ce soit avec l'aide de chercheurs et chercheuses en littérature ou par l'utilisation de modèles d'extraction plus performants, la liste des marqueurs du style peut toujours être enrichie. Enfin, dans le but de séparer autant que possible sémantique et stylistique dans un espace latent interprétable, il pourrait être intéressant d'essayer de transposer les méthodes de désentrelacement utilisées pour l'apprentissage de représentation de mots à l'apprentissage de représentations de documents et/ou d'auteurices. Par exemple, [Lia+20] propose d'associer à chaque mot une étiquette qui servira au désentrelacement. Ou encore [Jai+18] utilise une variation de la *max-margin loss* se déclinant selon plusieurs aspects afin de créer une représentation par aspect. Dans notre cadre, il est possible d'utiliser certains descripteurs pouvant servir d'étiquette ou d'aspect (adverbe, entités nommées, ...), la difficulté de ces approches étant de réussir à se passer d'annotation manuelle pour le désentrelacement.

Chapitre 5

Apprentissage de représentations temporelles de documents et d'auteurs

5.1 Introduction

5.1.1 Les représentations à l'épreuve du temps

Au cours du chapitre précédent, nous avons développé un modèle d'apprentissage de représentations d'auteurices et de documents, VADES visant à capturer le style, défini comme l'ensemble des choix d'écriture ne faisant pas entrer la sémantique en jeu. Nous souhaitons l'utiliser ici pour faire une analyse temporelle en déterminant à quel point le style d'un auteur ou d'une autrice est sujet à évoluer au cours de sa vie. Pour cela, nous avons sélectionné un ensemble de 11 auteurices pour lesquels nous pouvions obtenir facilement les dates de production d'une majorité de leurs oeuvres, avec notamment certains dont le style littéraire est connu pour avoir fortement évolué au cours de leur vie (Gustave Flaubert, F. Scott Fitzgerald). Nous notons alors pour chaque auteurice a T^a sa période de production (du livre le plus ancien au plus récent). Nous séparons chronologiquement chacune de ces périodes en 3 parties (T_1^a, T_2^a, T_3^a) et utilisons T_1^a pour entraîner VADES ($\alpha = 0.5$). Nous représentons les deux tiers restants, T_2^a et T_3^a ainsi que les plongements de chaque écrivain Figure 5.1. L'emplacement de chaque chiffre correspond au plongement du document correspondant et le chiffre en lui même à sa période d'écriture (1 pour T_1^a, 2 pour T_2^a, 3 pour T_3^a). Nous pouvons observer que les oeuvres les plus tardives ont tendance à s'éloigner parfois fortement des représentations de leurs auteurices, soulignant une fluctuation non négligeable du style littéraire avec le temps. Elles vont même parfois jusqu'à créer des groupes franchement à part du reste de la production (notamment pour Oscar Wilde, où Victor Hugo). Tout de même, quelques auteurices (Winston Churchill notamment), conservent un style cohérent au cours de leur vie, bien qu'alternant les récits autobiographiques et les fictions.

Afin de confirmer de manière quantitative cette observation, nous représentons chaque période de productions (T_1^a, T_2^a, T_3^a) de chaque auteurice par la moyenne des plongements des documents correspondant. Nous évaluons alors la similarité entre les productions de chaque auteurice pour

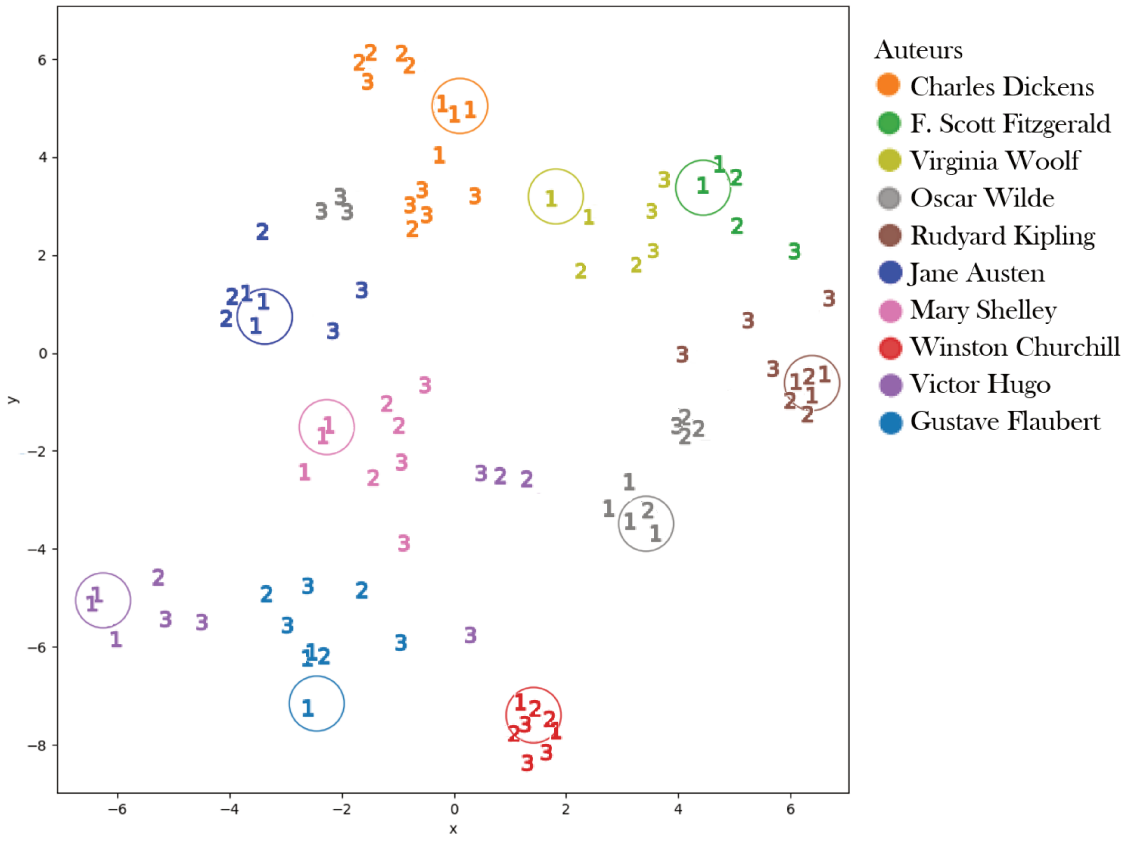


FIGURE 5.1 – Projection 2D des plongements d'auteurices et de documents issus du Projet Gutenberg à partir de VADES ($\alpha = 0.5$). Nous avons sélectionné un ensemble d'auteurices dont nous avons les dates d'écriture de certaines de leurs oeuvres. Nous avons ensuite entraîné VADES sur le tiers le plus ancien des productions de chacun. Enfin, nous montrons ici les documents faisant parties des deux tiers les plus récents. Chaque document est représenté par le chiffre du tiers auquel il appartient (1 pour celui d'entraînement, 2 et 3 pour ceux de tests, dans l'ordre chronologique). Chaque cercle correspond à un écrivain et sa variance.

Auteurices	**Similarité cosinus entre périodes**			**Durée $T_3^a - T_1^a$ en année**
	$T_1^a - T_2^a$	$T_1^a - T_3^a$	$T_2^a - T_3^a$	
Tous (11)	0.83	0.81	0.86	28.9
F. Scott Fitzgerald	0.81	0.79	0.74	21
Charles Dickens	0.75	0.89	0.96	33
Virginia Woolf	0.74	0.67	0.69	10
Jane Austen	0.96	0.95	0.96	27

TABLE 5.1 – Similarité cosinus entre les périodes d'écriture d'un ensemble de 11 auteurices du Projet Gutenberg. Nous présentons la moyenne sur tous les auteurices et quelques exemples d'auteurices.

chaque période à l'aide la similarité cosinus et présentons les résultats agrégés Table 5.1. Ces résultats confirment notre hypothèse selon laquelle le style évolue avec le temps, même sur des périodes relativement courtes comme pour Virginia Woolf. Seule Jane Austen a conservé un style littéraire très identifiable et constant sur l'ensemble de sa vie. L'information temporelle est non-négligeable lorsque l'on souhaite représenter un auteur ou une autrice. Il serait donc pertinent de chercher à apprendre des plongements d'auteurices dynamiques, c'est à dire capturant cette notion de temporalité en faisant évoluer la représentation en fonction de la période considérée.

5.1.2 Vers des représentations dynamiques des auteurices

L'objectif est de créer un espace de représentation contenant à la fois les auteurices et leurs documents, mais où les auteurices seraient modélisés comme des trajectoires paramétrées par le temps, le long desquelles les documents se placeraient en fonction de leur période de création. Comme dans le chapitre précédent, nous souhaitons pouvoir nous appuyer sur des grands modèles de langues comme encodeurs afin de bénéficier au maximum de leur capacité à représenter le langage et son évolution et de pouvoir évoluer aisément avec les progrès constants faits dans le domaine. Pour cela, nous ferons dans un premier temps un état de l'art des méthodes de représentations temporelles de documents, dont les plus récentes s'articulent notamment avec ce type d'encodeur.

Ensuite, nous évoquerons les méthodes d'apprentissage de représentations dynamiques d'auteurices. Elles sont en réalité très peu nombreuses et une seule plonge également les documents [Gou+22a]. Nous présenterons alors dans une seconde partie notre cadre, qui s'appuie sur la modélisation des trajectoires d'auteurices sous forme d'un pont brownien, ainsi que le modèle que nous avons développé autour de ce cadre, appelé B^{2}ADE. C'est le premier modèle à apprendre des représentations temporelles globales et continues, là où l'état de l'art doit discrétiser l'espace temporel et associer un plongement par pas de temps. Elle permet également d'apprendre des représentations pour des documents à plusieurs auteurices, aspect que nous évaluerons qualitativement.

Nous choisissons ici de nous concentrer indifféremment sur tous les marqueurs caractérisant la production d'un auteur ou d'une autrice, thématique autant que stylistique. Ces deux aspects peuvent tous deux évoluer indépendamment au cours de leurs vies. Nous nous limiterons donc dans cette partie à des jeux de données et des applications classiques dans ce domaine, sans évaluer spécifiquement la capture du style littéraire. Nous évoquerons dans les perspectives la possibilité de combiner justement ces deux axes et laissons ce travail spécifique à des travaux ultérieurs.

Bien que très intuitive, l'hypothèse faite offre en pratique de bons résultats en comparaison des baselines et modèles existants auquels nous nous comparerons. Pour cela nous utiliserons deux

jeux de données fréquemment utilisés en représentation dynamiques d'auteurices. Enfin, nous montrerons qualitativement (Section 5.5.5) que notre méthode à l'avantage de produire un espace de représentation intéressant en tant que tel .

5.2 Etat de l'art

Nous allons d'abord nous intéresser aux modèles de représentations de documents incorporant la notion de temps et de datation dans leur cadre d'apprentissage, avant de se focaliser sur les plongements dynamiques d'auteurices. Comme pour le chapitre précédent, nous nous limiterons aux méthodes d'apprentissage profond.

5.2.1 Représentation temporelle de documents

La prise en compte du temps par les modèles de langues et les modèles de représentation est un problème important, à la base de beaucoup de sous-tâches, de la simple datation à la recherche d'information, mais aussi la détection d'évènements ou le résumé de texte. En effet, la langue évolue rapidement et les modèles entraînés à un instant T ne sont qu'une photo de ce qu'elle était à cet instant. La représentation du mot souris changera drastiquement si entraîné sur un corpus de 1950 ou de 2020. Incorporer des notions de temporalité permet d'enrichir les plongements et de garantir une cohérence sémantique entre les données traitées d'un corpus qui peut être bien plus récent que le corpus d'entraînement. Sur le web, la date de création et/ou de publication d'un document est une métadonnée souvent facilement accessible. De nombreux travaux illustrent cet impact des changements sémantiques temporels sur la qualité des plongements de mots et donc sur la capacité à être efficace sur un ensemble de sous-tâches [Laz+21 ; HLJ16]. [Amb+21 ; Ken+15] montrent par exemple que le vocabulaire majoritairement utilisé change drastiquement même sur des périodes de temps très courtes de seulement quelques années.

Dans la continuité du modèle Word2Vec [Mik+13b], plusieurs modèles d'apprentissage de plongements de mots prenant en compte la dynamique temporelle ont été proposés. Par exemple, [BM17] propose une version dynamique du modèle Skip-Gram, [RB18] des plongements avec une probabilité d'évolution exponentielle ou encore [EM16], une dépendance temporelle linéaire entre les représentations. Seulement, chaque mot ne possède qu'une représentation par tranche temporelle, ce qui rend impossible l'inférence de représentation de documents dont on ignore la date. [HP19] va même jusqu'à entraîner des plongements et un bi-LSTMs distinct pour chaque intervalle de temps, avant d'agréger les représentations en sortie pour créer une représentation unique d'un document. Sur des grands corpus s'étalant sur de longues périodes cela devient rapidement insoutenable en terme de temps de calcul.

L'arrivée des transformers et leur capacité à contextualiser les plongements de mots permet de proposer des représentations dynamiques en évitant justement ces multiplications de modèles. De même, se focaliser directement sur la représentation de documents permet de créer un espace encodant directement la dynamique temporelle sans nécessiter différentes tranches de représentation. La tâche d'évaluation associée est souvent la datation de documents, vu comme un problème de classification multi-classe. Elle est parfois accompagnée de la détection de changement sémantique quand des plongements de mots sont également appris. La plupart des méthodes historiques se basent sur des analyses statistiques de caractéristiques et de n-grams de mots. L'utilisation de modèles d'apprentissage profond est assez récente, le premier étant le modèle NeuralDater [Vas+18] proposé en 2018, à notre connaissance.

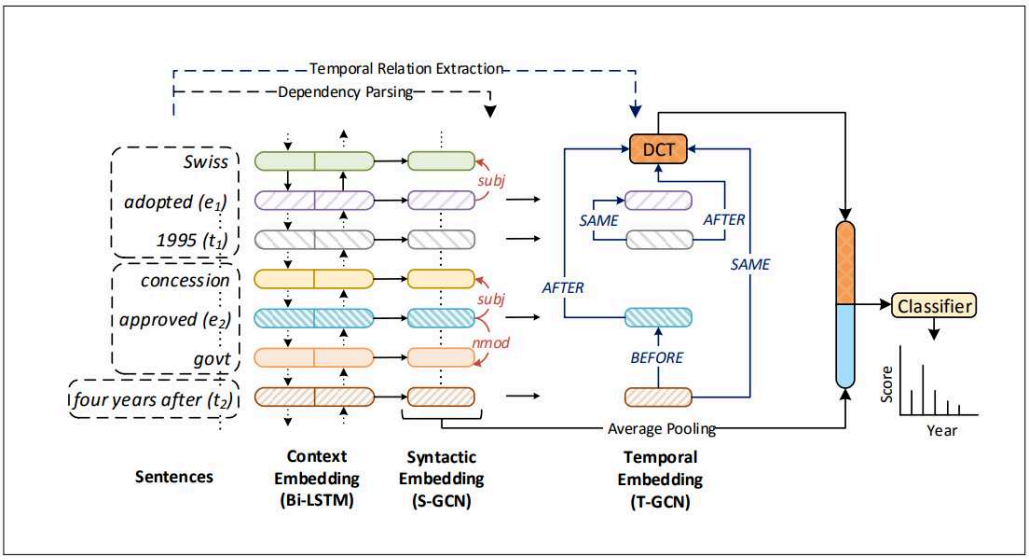


FIGURE 5.2 – Illustration du modèle Neural Dater issu de l'article *Dating Documents using Graph Convolution Networks* [Vas+18]. Le plongement final appris est indiqué par "DCT" pour *Document Creation Time*.

Par exemple, Neural Dater [Vas+18] utilise un GCN (*Graph Convolution Networks*) pour prendre en compte à la fois l'information syntaxique mais aussi les relations entre les différents évènements temporels présents dans les documents. Cela nécessite le prétraitement des textes bruts avec des modèles d'extraction d'évènements et de tri d'évènements CAEVO (*Cascading EVent Ordering System*). Les fichiers obtenus sont alors des XML au format TimeML, qui sont ensuite donnés en entrée d'un modèle créant un graphe de relation entre les évènements temporels et causaux CATENA (*CAusal and TEmporal relation extraction from NAtural language texts*). Chaque plongement de mots est contextualisé grâce à un bi-LSTM. Les représentations obtenues sont alors agrégées par un GCN s'appuyant sur le graphe des dépendances syntaxiques. Enfin, un second GCN basé sur le graphe des évènements permet d'obtenir une représentation finale après pooling, servant à prédire la date après un softmax. Une illustration du modèle issue de l'article est proposée Figure 5.2.

Ce modèle est ensuite amélioré en AD3 (*Attentive Deep Document Dater*) dans un second article [RDT18]. Deux modèles distincts sont en réalité proposés, où la couche finale de pooling de Neural Dater est remplacée par un GCN avec attention sur le graphe temporel pour OE-GCN (*Ordered Event-GCN*) et par une couche d'attention sur le contexte pour AC-GCN (*AC-GCN*). Les poids d'attention d'AC-GCN permettent d'interpréter quels mots ont pesé le plus dans la datation d'un document donné. Le modèle combinant AC-GCN et OE-GCN constitue l'état de l'art actuel en datation de documents, dans une approche relevant plus de la recherche d'information et au prix d'un long travail de pré-traitement des données. Nous justifions en section 5.4.2 pourquoi nous ne les avons pas utilisés.

Plus récemment, [SJN21] fine-tune un ensemble de modèle BERT à la régression ordinale pour dater chaque phrase d'articles scientifiques. Les résultats sont ensuite agrégés au niveau de l'article pour produire une date de création unique. Ces travaux montrent que l'approche au niveau de la

phrase offre les meilleurs résultats.

Enfin, le modèle TempoBERT [RGR22] spécialise BERT sur la tâche de prédiction de mots masqués en ajoutant des tokens spécifiques indiquant la date de création de la phrase. Ces tokens ont une probabilité plus élevée d'être masqués durant l'entraînement (90%, contre 15% pour les tokens usuels). Ce modèle s'évalue en détection de changement sémantique et en datation de documents. Il obtient dans les deux tâches de très bons résultats et rivalise même sur la seconde avec une version fine-tunée de BERT sur la datation, alors que son entraînement ne porte que partiellement sur la datation.

L'étude de l'impact de la diachronie sur la qualité des modèles de représentations de mots est un problème très étudié et traité. Il en résulte plusieurs modèles temporels de plongements de mots, proposant souvent une représentation par tranche temporelle. Il est alors assez difficile d'extraire une représentation d'un document dont la date de création serait inconnue. Seuls quelques solutions utilisant l'apprentissage profond et des modèles de langue contextualisés proposent un espace de représentation unique capturant la dynamique du langage. De la même façon, il n'existe à notre connaissance que 3 modèles permettant d'apprendre des représentations dynamiques d'auteurices, que nous allons présenter dans la section suivante.

5.2.2 Représentation dynamique d'auteurices

La plus ancienne méthode de représentation dynamique d'auteurices date de 2006 et étend l'approche du modèle CODE (*Co-occurrence Data Embedding*) à des données dynamiques. C'est un modèle de langue se basant sur les distances entre les représentations des auteurices et des mots. Typiquement, en notant u_i la représentation du mot w_i et h_a celle de l'auteurice a, le modèle CODE propose la probabilité suivante d'observer le mot w_i donné l'auteurice a :

$$p(w_i|h_a) = \frac{p(w_i)e^{-\|u_i - h_a\|_2^2}}{\sum_{v=1}|V|p(w_v)e^{-\|u_v - h_a\|_2^2}} \tag{5.2.1}$$

Où V est l'ensemble du vocabulaire. [SSG07a] incorpore l'aspect dynamique simplement en découpant le temps en tranche temporelle et en associant une représentation distincte pour chaque tranche par mots et par auteurices, notée respectivement w_t^i et h_t^a. La modélisation d'un intervalle seul est similaire à l'équation ci-dessus, où $p(w_i)$ est remplacé par $p(w_i|t)p(h_a|t)$, soit les probabilités d'observer le mot w_i et l'auteurice a au temps t. Le modèle est alors définit par les probabilités de transition choisies, qui sont ici des lois normales partageant toutes la même variance Γ, fixée a priori :

$$\begin{aligned} p(h_t^a|h_{t-1}^a) &= \mathcal{N}(h_{t-1}^a, \Gamma) \\ p(u_t^i|u_{t-1}^i) &= \mathcal{N}(u_{t-1}^i, \Gamma) \end{aligned} \tag{5.2.2}$$

Après un ensemble d'approximations afin d'obtenir un modèle gaussien linéaire, il est possible d'appliquer les équations du filtre de Kalman (que nous ne détaillerons pas) pour obtenir les paramètres u et a maximisant la log-vraisemblance de leur modèle.

La seconde méthode de représentation dynamique d'auteurices est le modèle DAR (*Dynamic Author Representation*) [DLD19]. Il entraîne un modèle de langue simple, constitué uniquement d'un LSTM comme décodeur, dont l'entrée est paramétrée par les représentations des auteurices. Plus précisément, notons $\mathcal{A}$ l'ensemble des auteurices considérés, ayant publiés sur un ensemble d'intervalles temporels $\{1, 2, ..., T\}$ discrétisé. L'objectif est de prédire la séquence de mots d'un

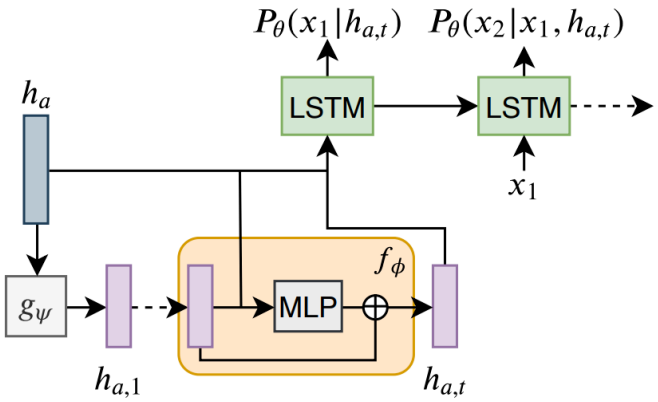


FIGURE 5.3 – Illustration du modèle DAR (*Dynamic Author Representation*) issu de l'article *Learning Dynamic Author Representations with Temporal Language Models* [DLD19]

document d_t^a connaissant son auteurice a et sa période d'écriture t. Autrement dit, en notant $(w_1, ..., w_{|d_t^a|})$ cette séquence, de maximiser la vraisemblance suivante :

$$p(d_t^a|a,t) = \prod_{k=0}^{|d_t^a|} p(w_{k+1}|w_{0:k}, a, t) \tag{5.2.3}$$

Chacune des probabilités du produit est obtenue en sortie d'un LSTM à deux couches. Celui-ci est initialisé avec la concaténation du token spécifique de début de phrase w_0 avec la représentation statique h^a et dynamique h_t^a au temps t de l'auteurice a. Les plongements statiques proviennent d'une couche de plongements entraînables et la dynamique est modélisée comme suit :

$$h_t^a = h_{t-1}^a + f_\theta(h_{t-1}^a, h_t^a) \tag{5.2.4}$$

Où f_θ est un MLP. Ainsi, la représentation de l'auteurice a au temps $t+1$ dépend uniquement de celle au temps t et de sa représentation statique. A noter que la première représentation temporelle h_1^a est initialisée à partir de h^a avec un MLP à part, noté g_ϕ. Le modèle DAR est évalué sur la tâche de modélisation du langage uniquement. En effet, bien que capable de générer des représentations de mots à partir du LSTM, ces dernières ne sont pas dans le même espace que celles des auteurices. S'il est possible d'obtenir des représentations de documents par agrégation, elles n'auront aucune signification si l'auteurice et la tranche temporelle ne sont pas fournis en entrée du LSTM. C'est une forte limitation de ce modèle.

Le dernier modèle de représentation dynamiques d'auteurices, DGEA (*Dynamic Gaussian Embedding of Authors*) [Gou+22a], n'a pas ce problème. DGEA modélise les auteurices comme des gaussiennes dont les documents sont issus. La représentation $d_{i,t}^a$ du $i^{\text{ème}}$ document de l'auteurice a au temps t suit la loi normale :

$$d_{i,t}^a \sim \mathcal{N}(\phi_t^a, {\sigma_t^a}^2 I) \text{ où } \phi_t^a, {\sigma_t^a}^2 \in \mathbb{R}^r \tag{5.2.5}$$

r étant la dimension de l'espace de représentation. Il s'agit alors de déterminer les lois de transition de la tranche t à la tranche $t+1$. Deux modèles distincts sont proposés.

Le premier, K-DGEA, fait évoluer les moyennes des plongements d'auteurices selon un modèle de Markov du premier ordre, proche de ce qui est fait par DAR :

$$\begin{aligned} \phi_t^a &\sim \mathcal{N}(\phi_{t-1}^a, {\delta^a}^2 I) \\ \phi_1^a &\sim \mathcal{N}(\phi_0^a, {\delta_0^a}^2 I) \end{aligned} \tag{5.2.6}$$

Une suite de simplification et de factorisation de la vraisemblance permet d'obtenir un modèle plus simple, en notant n_t^a le nombre de textes écrits par l'auteurice a au temps t :

$$\begin{aligned} \phi_t^a &\sim \mathcal{N}(\phi_{t-1}^a, {\delta^a}^2 I) \\ \bar{d_t^a} &\sim \mathcal{N}(\phi_t^a, \frac{{\sigma_t^a}^2 I}{n_t^a}) \end{aligned} \tag{5.2.7}$$

où $\bar{d_t^a}$ est la moyenne des documents écrits par l'auteurice a durant l'intervalle de temps t. C'est un modèle gaussien linéaire simple sur lequel, comme pour [SSG07b], il est possible d'appliquer les calculs du filtre de Kalman avec un modèle EM pour estimer les paramètres optimaux.

La seconde variante du modèle DGEA, R-DGEA modélise la variance et la moyenne de l'auteurice a au temps t comme une fonction de l'ensemble de ses publications passées (notées $d_{1:t-1}^a$) :

$$\begin{aligned} \phi_t^a &= f(d_{1:t-1}^a) \\ \sigma_t^a &= h(d_{1:t-1}^a) \end{aligned} \tag{5.2.8}$$

f est un LSTM prenant en entrée la moyenne des représentations des documents des périodes successives et la variance est obtenue à partir d'un MLP prenant en entrée la moyenne de l'auteurice correspondant ϕ_t^a. Les deux modèles DGEA s'appuient sur des modèles de représentations de documents pré-entraînés pour initialiser les différentes représentations (USE, SBERT, InferSent). Les modèles sont évalués sur les tâches d'attribution d'auteurices, de prédiction de liens et de classification d'auteurices. Si R-DGEA semble être légèrement plus performant, K-DGEA le surpasse sur certaines tâches. L'un des grand avantage de R-DGEA est sa capacité à inférer des représentations d'auteurices non vus pendant l'entraînement.

Des trois méthodes précédentes, une seule plonge à la fois les auteurices et les documents dans le même espace. Cependant, elles découpent toutes le temps en intervalle donné, bien souvent d'une année sur les jeux de données considérés, à savoir de presse et de publications scientifiques. En faisant cela, l'aspect continu du temps se perd, avec les informations temporelles résultant de l'agrégation des documents. En introduisant la modélisation par le pont brownien, nous allons présenter notre troisième contribution, le modèle B²ADE qui vise à produire des plongements dynamiques de documents et d'auteurices, ces derniers étant continus.

5.3 Contribution 3 : B²ADE

Nous disposons d'un corpus de textes et pour chaque texte nous possédons sa date d'écriture ou de publication, en fonction du contexte, ainsi que son ou ses auteurices. Notre objectif est de construire un espace où les plongements des auteurices seraient des trajectoires paramétrées par le temps. De sorte que les documents se placeront le long de leurs auteurices en fonction de la période à laquelle ils ont été produits. De la même façon, si des auteurices collaborent pendant une période, leurs trajectoires doivent être proches. Comme lors du chapitre précédent, nous souhaitons

pouvoir nous appuyer sur n'importe quel encodeur de documents récents de sorte à pouvoir profiter au maximum des avancées dans la modélisation de la langue. Afin de modéliser la dynamique temporelle, nous nous tournons vers les processus gaussiens, comme c'est souvent le cas dans la littérature [SSG07a ; Gou+22a]. Certains travaux récents ont introduit le pont brownien dans ce cadre.

5.3.1 Pont Brownien et application en apprentissage profond

Un pont brownien [Ioa91] de $X_0 = a \in \mathbb{R}$ à $X_T = b \in \mathbb{R}$ sur le domaine $[0, T]$ est le processus gaussien définit par :

$$X_t = (1 - \frac{t}{T})X_0 + \frac{t}{T}X_T + W_t + \frac{t}{T}W_T \tag{5.3.1}$$

Où W_t est un mouvement brownien standard [Ioa91]. La particularité du pont brownien par rapport au mouvement brownien est que ses points de départ et d'arrivée sont fixés. Notamment, la densité du processus du pont brownien entre X_0 à $t = 0$ et X_T à $t = T$ est assez directe :

$$p(X_t|X_0, X_T) = \mathcal{N}((1 - \frac{t}{T})X_0 + \frac{t}{T}X_T, \frac{t(T-t)}{T}) \tag{5.3.2}$$

Cela correspond à une interpolation linéaire entre départ et arrivée, avec une incertitude grandissante à mesure que l'on s'éloigne temporellement de ces derniers. Le pont brownien est utilisé par exemple pour désentrelacer les mouvements d'objets donnés dans des vidéos assez simples [Bha+20], dans un cadre d'auto-encodeurs variationnels. Plus récemment, ce processus gaussien a été utilisé pour entraîner un modèle de langue à générer des textes longs plus cohérents [Wan+22], dans une approche intitulée Time Control, schématisée Figure 5.4.

Leurs jeux de données sont des ensembles de textes structurés et cohérents (transciption d'une hotline, recettes de cuisine, ...). Leur objectif est de représenter chaque phrase successive d'un document comme suivant un pont brownien. Ils utilisent la fonction de perte contrastive suivante, en notant (x_1, x_2, x_3) un triplet de phrases observées et f_θ l'encodeur :

$$\begin{aligned} L_N &= \mathbb{E}_X[-\log \frac{e^{d(x_0, x_t, x_T; f_\theta)}}{\sum_{x_{t'} \in B} e^{d(x_0, x_{t'}, x_T; f_\theta)}}] \\ \text{avec } d(x_0, x_t, x_T; f_\theta) &= -\frac{1}{2\sigma^2}||f_\theta(x_t) - (1 - \frac{t}{T})f_\theta(x_0) - \frac{t}{T}f_\theta(x_T)||_2^2 \end{aligned} \tag{5.3.3}$$

Ici, $\sigma^2 = \frac{t(T-t)}{T}$. B est un ensemble d'exemples négatifs tirés aléatoirement. La distance correspond à la norme euclidienne entre la représentation du document issu de l'encodeur $f_\theta(x_t)$ et l'interpolation linéaire entre $f_\theta(x_0)$ et $f_\theta(x_T)$ correspondant au temps t. Elles visent donc à rapprocher $f_\theta(x_t)$ de cette interpolation.

Les phrases commençant et terminant chaque document étant très proches sur leurs corpus, $z_0 = f_\theta(x_0)$ est fixé à l'origine de l'espace de représentation et z_T à 1 par régularisation L_2. Le cadre utilisé ici est équivalent à la fonction de perte usuelle InfoNCE introduite dans [OLV18] utilisée pour l'apprentissage contrastif. Ainsi, l'objectif peut être vu comme une borne inférieure sur l'information mutuelle $I(x_t, \{x_0, x_T\}) \geq \log(N) - L_N$. Minimiser la fonction de perte revient à chercher à maximiser la quantité d'information entre la trajectoire effective et l'interpolation linéaire entre ses extrémités. L'encodeur utilisé est GPT-2 (non entraînable) sur lequel est placé un MLP. Une fois l'entraînement réalisé, l'encodeur est gelé et un décodeur (GPT-2 également) est entraîné

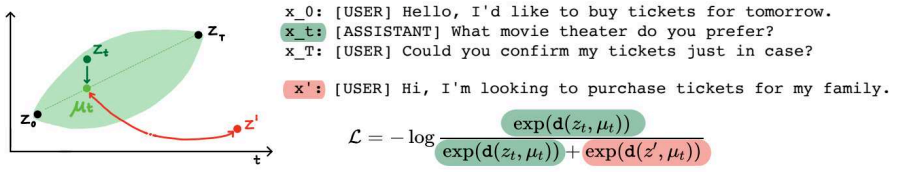


FIGURE 5.4 – Illustration du modèle Time Control issu de l'article *Language Modeling via Stochastic Processes* [Wan+22]. La représentation z_t de la phrase x_t tend à être rapprochée de l'interpolation au temps t entre z_0 et z_T. A contrario, celle de l'exemple négatif z' issu d'une autre conversation en sera éloignée.

à la génération de textes, conditionné par les plongements issus de Time Control. Cela permet de créer des textes longs qui restent cohérents comparativement aux méthodes usuelles. Nous avons présenté ici le seul modèle d'apprentissage profond s'appuyant sur une modélisation de processus temporel par le pont brownien en TAL [Wan+22]. Dans la section suivante, nous présentons notre application au cadre de la représentation d'auteurices et de documents.

5.3.2 Modélisation par le pont brownien

Dans notre cadre, nous souhaiterions ajouter en plus la dimension de l'auteurice. Notons d_t^a le document de l'auteurice a écrit au temps t et $z_t^a = f_\theta(d_t^a) \in \mathbb{R}^r$ sa représentation. Notre modélisation considérant le temps comme continu, nous supposons pour simplifier les notations qu'à chaque temps t, chaque auteurice ne peut avoir écrit qu'un seul document. Pour chaque auteurice, son plongement à l'instant t est noté $h_t^a \in \mathbb{R}^r, 0 \leq t \leq T$. Pour simplifier nous notons indépendamment $[0, T]$ l'intervalle de production de tous les auteurices, mais il peut être différent pour chacun. Nous souhaitons avoir z_t^a proche de h_t^a avec h_t^a suivant le pont brownien :

$$p(h_t^a | h_0^a, h_T^a) = \mathcal{N}((1 - \frac{t}{T})h_0^a + \frac{t}{T}h_T^a, \frac{t(T-t)}{T}) \tag{5.3.4}$$

Il est alors nécessaire de fixer pour chaque auteurice des points de départs et d'arrivées spécifiques (voir Figure 5.5), ce qui est difficilement compatible avec la fonction de perte de Time Control (nous le justifierons par l'expérimentation plus loin). Pour contourner ce problème, nous utilisons le cadre du Variational Information Bottleneck [TPB99 ; Ale+17] détaillé au chapitre précédent (4.4.1) et que nous allons présenter de nouveau ici plus succintement. L'objectif général du VIB, introduit par [Ale+17] est le suivant :

$$\arg\max_z I(z, y) - \beta I(z, x), \tag{5.3.5}$$

A savoir compresser au maximum l'information que les représentations latentes z sortent des observations x pour conserver uniquement le nécessaire afin de prédire l'étiquette y. Cela revient à minimiser la fonction de perte suivante :

$$L_{vib} = -\mathbb{E}[\log q(y|z)] + \beta KL(p(z|x)||q(z)) \tag{5.3.6}$$

Où $q(y|z)$ est l'approximation variationnelle de $p(y|z)$ et $q(z)$ approxime $p(z)$. Maximiser l'équation 4.4.1 revient à minimiser l'équation 4.4.3. $p(z|x)$ est un choix de modélisation.

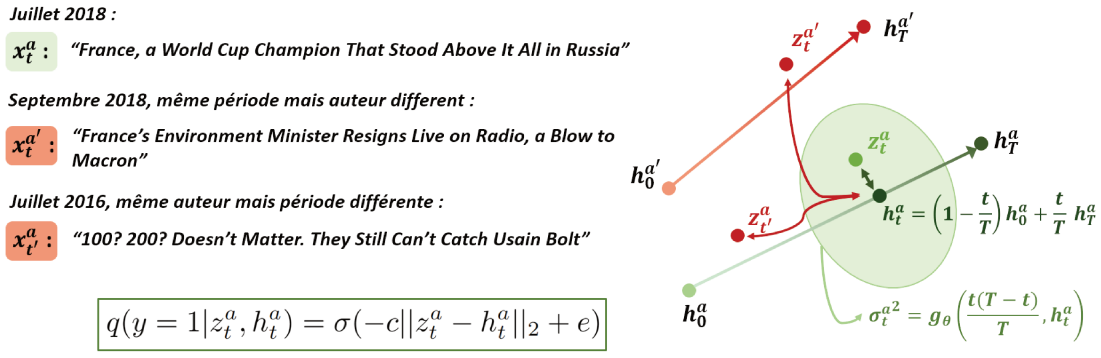


FIGURE 5.5 – Schéma de la modélisation du modèle B²ADE. La représentation z_t^a du document écrit par l'auteurice a au temps t tend à être rapproché de l'interpolation au temps t entre h_0^a et h_T^a, qui correspondent aux plongements initial et final de a. A contrario, les exemples négatifs, provenant d'auteurices différents ($z_t^{a'}$), de temps différents ($z_{t'}^a$) ou des deux, doivent en être éloignés.

En particulier, nous avons pour chaque document des paires positives (z_t^a, h_t^a) et pouvons en créer des négatives $(z_t^a, h_{t'}^{a'})$, en prenant une autrice ou un auteur différent, une date de création différente, ou les deux. Les paires positives sont associées à l'étiquette $y = 1$ et négative à l'étiquette $y = 0$. La représentation de l'auteurice h_t^a suit alors la loi normale de l'équation 5.3.4, où h_0^a et h_T^a sont deux couches d'embeddings entraînables. z_t^a est également modélisé comme une loi normale, de moyenne $\mu_t^a = f_\theta(d_t^a)$ et de variance diagonale $\eta_t^{a\,2} = g_\phi(d_t^a)$. En s'appuyant sur [Oh+19], on peut obtenir la perte contrastive faible suivante :

$$q(y = 1|z_t^a, h_t^a) = \sigma(-c||z_t^a - h_t^a||_2 + e) \tag{5.3.7}$$

Où $c > 0$ et $e \in \mathbb{R}$ sont des paramètres, σ la fonction sigmoïde. Notre fonction de perte devient alors :

$$\begin{aligned}\mathcal{L} = &- \mathbb{E}_{p(z_t^a|d_t^a),p(h_t^a|a_t)}[\log q(y|z_t^a, h_t^a)] \\ &+ \beta\left(KL(p(h_t^a|a_t)||q(h_t^a)) + KL(p(z_t^a|d_t^a)||q(z_t^a))\right)\end{aligned} \tag{5.3.8}$$

La seconde partie de l'équation 5.3.8, faisant intervenir les divergences de Kullback-Leibler est en réalité équivalente à une régularisation sur les moyennes et variances de notre modélisation. Nous montrerons par l'expérimentation qu'elle peut être supprimée sans impact sur le modèle. L'espérance de l'équation 5.3.8 n'étant pas calculable analytiquement malgré les choix de modélisation, nous l'obtenons en moyennant L tirages de Monte Carlo par paires d'entraînement :

$$\mathbb{E}[\log q\,(y|z_t^a, h_t^a)] \approx \frac{1}{L}\sum_{l=1}^{L} \log q(y|z_t^{a\,(l)}, h_t^{a\,(l)}) \tag{5.3.9}$$

Nous utilisons alors l'astuce de reparamétrisation [KW14] :

$$z_t^{a(l)} = \mu_t^a + \eta_t^a \odot \epsilon,\ h_t^{a(l)} = (1 - \frac{t}{T})h_0^a + \frac{t}{T}h_T^a + \frac{t(T-t)}{T}\epsilon \text{ avec } \epsilon \sim \mathcal{N}(0,1) \tag{5.3.10}$$

où l indique l'indice du tirage. Cette fonction peut finalement être minimisée par rétro-propagation du gradient. Le cadre VIB permet de conserver la modélisation gaussienne du pont brownien autrement qu'en considérant la variance comme un simple facteur de régularisation comme dans Time Control. Il permet à la fois de rapprocher les documents des trajectoires d'auteurices tout en les plaçant au bon niveau temporel. Contrairement aux modèles de l'état de l'art, il n'y a pas besoin de discrétiser l'espace temporel. Cette modélisation permet aussi de créer des représentations temporelles continues sous forme de trajectoire et non discrétisée par pas de temps comme l'état de l'art. Nous détaillons dans la section suivante les choix d'architecture réalisée, notamment pour l'encodeur de documents.

5.3.3 Architecture du modèle

Une représentation schématique de notre modèle est proposée Figure 5.6.

Plongement initiaux et finaux des auteurices

Comme précisé plus haut, les extrémités de chaque trajectoire d'auteurices h_0 et h_T sont obtenues via deux couches de plongements entraînables. A priori correspondant au temps $t = 0$ et $t = T$. Cependant, chaque auteurice peut avoir des périodes d'écriture différentes, il est donc possible également de fixer des temps initiaux et finaux distincts pour chacun, afin de mieux modéliser leurs dynamiques propres. Nous faisons le choix de prendre comme date initiale le premier document publié de chaque auteurice et comme date finale le dernier (pour le jeu d'entraînement), il est possible de faire d'autres choix. Par exemple, en choisissant T strictement supérieur à la date maximale d'écriture d'un document, le modèle extrapolera les trajectoires d'auteurices pour des dates non vues durant l'entraînement. Nous testerons cette configuration pendant les expérimentations.

Calcul de la variance

La variance de la représentation d'auteurices vaut simplement $\sigma_t^{a\,2} = \frac{t(T-t)}{T}$, c'est celle définie par le pont brownien. Cependant, cette dépendance est uniquement temporelle et les évolutions de chaque auteurice peuvent leurs être propres. Ce peut être des changements drastiques de style, de sujet ou de genre avec le temps. Nous calculons également une log-variance propre à chaque représentation d'auteurice à un temps donné en passant h_t^a à travers un MLP. La variance finale est obtenue en sommant l'exponentielle de cette log-variance à la variance du pont brownien, pondéré par un paramètre apprenable α. Ainsi : $(\sigma_t^a)^2 = e^{\text{MLP}(h_t^a)} + \alpha\frac{t(T-t)}{T}$. L'idée d'apprendre une variance à partir de la moyenne de notre gaussienne est similaire à ce qui est fait dans R-DGEA [Gou+22a]. Le MLP possède 2 couches à activation tangeante hyperbolique et LeakyReLU et dropout de 0.2. Plusieurs travaux modélisant des séquences apprennent également une représentation latente du temps [LDB18 ; Som+23], similaire au codage positionel des tokens dans BERT [Dev+19]. Nous n'avons pas trouvé cela pertinent dans notre cadre mais le laissons à de futurs travaux.

Choix de l'encodeur

Concernant l'encodeur de documents, le véritable bloc d'entrée de notre modèle, plusieurs modèles sont envisageables. Nous nous orientons vers un grand modèle de langue, mais qui devrait

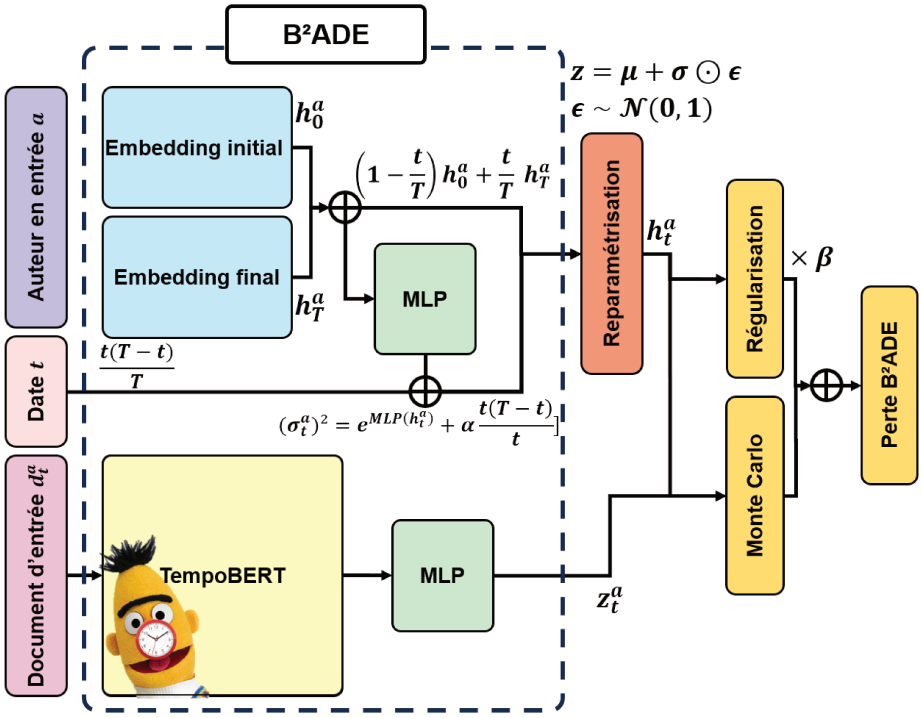


FIGURE 5.6 – Schéma du model B²ADE.

idéalement posséder une capacité à capter la dynamique temporelle dans les documents. Pour cela, nous utilisons TempoBERT [RGR22], que nous avons détaillé plus haut. TempoBERT est une version de BERT fine-tunée sur la tâche de prédiction de mots masqués avec des tokens spécifiques de date. Nous le fine-tunons sur chaque corpus étudié séparément puis gelons ses poids durant l'entraînement de B²ADE. La moyenne des documents est obtenues ensuite via un MLP distinct, respectivement à 3 couches, avec activation LeakyReLU et dropout de 0.2. L'expérimentation montre que les résultats sont meilleurs en annulant la variance des documents, nous la prenons donc nulle pour tous les documents. Si d'autres encodeurs temporels plus performants existent (Neural Dater [Vas+18] et AD3 [RDT18]), ils nécessitent des formats d'entrée spécifiques. Nous laissons leur incorporation à notre modèle pour des travaux futurs. Notre modèle est transposable dans toutes les langues dès lors qu'un encodeur y existe.

Enfin, il peut traiter indifféremment des documents avec des auteurices uniques ou multiples. Nous n'avons pas évalué cet aspect là de VADES en nous limitant à des corpus mono-auteurices, ici nous en proposons une analyse qualitative. Un document à plusieurs auteurices génèrera autant de paires d'entraînement qu'il a d'auteurices. C'est également le cas de DAR et DGEA, autres modèles de l'état de l'art.

Nous allons maintenant évaluer notre modèle sur différents corpus et sur plusieurs tâches.

5.4 Evaluation

5.4.1 Jeux de données

Nous allons nous appuyer sur deux jeux de données en anglais. En suivant ce qui est fait par [DLD19], chacun permettra d'évaluer un aspect différent d'utilisation de notre modèle, à savoir l'imputation et la prédiction, que nous détaillons plus bas. Ce sont les deux jeux de données utilisés dans la littérature.

NYT

Le premier est introduit par [Yao+18] et a été préparé par [DLD19] avec le modèle DAR. Il est constitué d'un ensemble de titres d'article de presse du New York Times s'étalant de 1990 à 2015 inclus. Ce sont donc des textes très courts sur des thématiques assez variées, allant du sport à la politique en passant par des faits divers, des rubriques nécrologiques, etc ... Il est constitué de 41 446 documents pour 546 auteurices et est séparé en données d'entraînement, test et validation selon un schéma 70/20/10, avec un échantillonnage stratifié par auteurices. Nous l'utiliserons pour de l'imputation. C'est à dire que les données d'entraînement, de test et de validation proviendront de toute la période temporelle disponible. Cela permet d'évaluer la capacité d'interpolation des modèles.

S2G

Le second jeu de données est a également été préparé par [DLD19]. C'est un ensemble de titres d'articles scientifiques sur le machine learning publiés dans 22 conférences entre 1985 et 2017 inclus. Il a été introduit par [Amm+18]. Une nouvelle fois, ce sont des documents extrêmement courts. Cependant, contrairement au jeu de données NYT, nous disposons ici uniquement de l'année de publication et non de la date complète. Le corpus contient 1117 auteurices pour 45496 titres. Chaque article peut avoir plusieurs auteurices ici. Nous utiliserons ce jeu de données en imputation, mais aussi pour faire de la prédiction. Seuls 70% des premiers documents (dans l'ordre chronologique) de chaque auteurice seront disponibles durant l'entraînement, l'évaluation et la validation seront réalisées sur les pas de temps restants. Nous retrouvons ainsi la séparation stratifiée par auteurices 70/20/10. A noter que chaque auteurice ayant un rythme de publication spécifique, les données d'entraînement s'arrêtent plus ou moins tôt pour chacun. C'est le cadre d'évaluation le plus difficile, il permet d'évaluer la capacité des modèles à capturer la dynamique des auteurices et à extrapoler.

Un tableau récapitulatif des statistiques basiques de nos deux jeux de données est disponible Table 5.4.1. Bien que ces deux jeux de données ne proposent que des textes très courts, rien n'empêche d'utiliser notre modèle sur des textes plus longs soit en les découpant pour s'assurer de respecter la limite de tokens de l'encodeur, soit en s'appuyant sur un encodeur adapté, comme BigBird [Zah+20] ou encore LongFormer [BPC20].

5.4.2 Compétiteurs

Nous allons évaluer le modèle B²ADE et le comparer à plusieurs modèles existants. Comme baseline simple, nous utilisons une nouvelle fois la version DAN de l'encodeur de phrases USE [Cer+18]. Bien que statique, c'est un modèle très performant en attribution d'auteurices sur des documents courts. Nous allons également nous comparer à des versions fine-tunées de BERT sur différentes tâches de classification. La première sur l'attribution d'auteurices (notée BERT-ft$_A$),

Statistiques des jeux de données				
Dataset	**Auteurices**	**Longueur**	**Textes par auteurice**	**Période**
NYT	546	8.4(±2.5)	76(±51)	[1990, 2015]
S2G	1117	8.7(±2.9)	48(±27)	[1985, 2017]

TABLE 5.2 – Statistiques descriptives des jeux de données utilisés. NYT : titres d'article de presse du New York Times, S2G : titre d'articles scientifiques en lien avec le machine learning.

la seconde sur la prédiction de date (notée BERT-ft$_T$) et enfin la dernière sur ces deux tâches simultanément (notée BERT-ft$_{T+A}$).

Pour évaluer l'aspect dynamique de B²ADE nous évaluons également le modèle TempoBERT [RGR22], qui entraîne BERT sur la tâche de prédiction de mots masqués en ajoutant à chaque texte un token unique correspondant à son année d'écriture. Pour chacun de ces modèles ne produisant pas de représentation d'auteurices, nous la créons en moyennant les représentations de leurs documents du jeu d'entraînement. Les modèles NeuralDater [Vas+18] ou AD3 [RDT18], bien que plus performants en datation, ne sont pas pertinents sur nos jeux de données. En effet, ils nécessitent un pré-traitement avec des outils d'extraction de relation entre évènements qui n'ont pas lieu d'être sur nos données, constituées uniquement de titres très courts. Leur utilisation n'est donc pas pertinente dans ce cadre. Néanmoins, pour des applications sur des textes plus longs il serait tout à fait judicieux de les intégrer comme encodeurs.

Pour finir, nous nous comparons à deux modèles de représentations dynamiques d'auteurices à proprement parler. Tout d'abord, le modèle DAR [DLD19], qui apprend une représentation statique pour chaque auteurice. Elle sera ensuite utilisée pour créer des représentations dynamiques à chaque pas de temps, le tout en faible dimension. Ces représentations sont fournies en entrée d'un RNN pour les entraîner à la génération de texte. A noter que si DAR peut produire des représentations de documents, elles sont dans un espace différent de celui des auteurices et sont de faibles qualités si l'auteurice et la tranche temporelle ne sont pas fournis en entrée. Nous l'illustrerons dans la partie résultat.

Enfin, nous choisissons le modèle K-DGEA [Gou+22a], variante du modèle DGEA la plus rapide. Elle fait évoluer les représentations des auteurices dans le temps comme un modèle de Markov du premier ordre. Avec l'hypothèse d'une variance propre à chaque auteurice et indépendante du temps, le modèle est simplifié et optimisé par un algorithme Espérance-Maximisation (EM).

5.4.3 Tâches d'évaluation

Attribution d'auteurices

Chacun des modèles présentés ci-dessus ainsi que B²ADE seront évalués sur la tâche d'attribution d'auteurices sur les corpus NYT et S2G, en imputation et en prédiction. A noter que cette tâche en question ne fait pas intervenir spécifiquement de notion temporelle mais permet d'évaluer la capacité des modèles à rapprocher les documents de leurs auteurices. Pour chaque modèle, nous utilisons la similarité cosinus entre les documents tests et les représentations d'auteurices pour déterminer l'auteurice. Pour les baselines BERT-ft$_{T+A}$ et BERT-ft$_A$ nous utilisons directement la prédiction en sortie du modèle. Les métriques utilisées pour NYT seront l'accuracy et l'erreur de couverture. Etant donné que le jeu de données S2G offre la possibilité d'avoir plusieurs auteurices

pour un seul document, nous utiliserons le score de précision moyen de classement des étiquettes ou LRAP (*Label Ranking Average Precision*). Cette dernière calcule pour chaque vrai auteurice d'un texte la part de voisin de rang supérieur également auteurices. Elle est contenue entre 0 et 1, 1 étant le meilleur score possible. Nous utilisons la similarité cosinus entre les représentations d'auteurices et de documents pour attribuer l'auteurice à chaque document.

Datation de documents

La seconde tâche est la datation de documents. Cette tâche permet d'évaluer la capacité des modèles à capturer la dynamique temporelle des corpus étudiés. Pour chaque document il faut prédire l'année à laquelle il a été écrit. Les métriques associées sont l'accuracy et l'erreur absolue moyenne (MAE). Cette tâche est évaluée sur NYT et S2G mais uniquement en imputation. Pour la prédiction nous ne disposons pas des pas de temps des données tests durant l'entraînement. Un modèle prédisant toujours l'année centrale des intervalles de temps obtiendra une MAE de 13.5 sur NYT et de 17.5 sur S2G. Cette dernière mesure l'écart absolu entre l'année prédite et l'année effective de rédaction. Pour chaque modèle nous entraînons un algorithme des k-plus-proches-voisins à la classification de date sur les documents d'entraînements (les paramètres sont optimisés par GridSearch et validation croisée). Pour les baselines BERT-ft$_{T+A}$, BERT-ft$_T$ et TempoBERT nous utilisons directement les prédictions en sortie des documents.

Classification d'auteurices

Enfin, sur le jeu de données S2G nous évaluerons les modèles sur une tâche de classification, en imputation et en prédiction. En effet, nous disposons pour chaque article de la conférence dans laquelle il a été publié (IJCAI, ACL, EMNLP, CVPR, ...). Nous associons à chaque auteurice la conférence dans laquelle il a le plus publié pour chaque année $(t_0^a, t_1^a, ..., t_k^a)$ du jeu d'entraînement et voulons prédire la conférence associée à leurs représentations aux temps $(t_{k+1}^a, ..., T^a)$ où a correspond à chaque auteurice. En effet, chaque auteurice n'a pas le même rythme de publication et donc des pas de temps d'entraînement et d'évaluation propres. Nous évaluons ici surtout les modèles de représentations d'auteurices DAR, K-DGEA et B²ADE car ce sont les seuls capables d'extrapoler la dynamique des représentations qu'ils apprennent. Pour DAR, nous affichons les résultats avec les représentations statiques seules (DAR (statique)), dynamiques seules (DAR (dynamiques)) et les deux concaténées (DAR (concat)). Pour K-DGEA, nous utilisons la représentation de l'auteurice à T pour modéliser l'auteurice à T+1, T+2, ... Enfin, à titre de comparaison, nous utilisons les modèles statiques TempoBERT et USE. La représentation de l'auteurice sur l'intervalle t est la moyenne de ses documents écrits à cette période. De la même façon, pour T+1, T+2, ... nous utilisons sa représentation à T. Le corpus S2G liste les articles de 22 conférences distinctes. Nous utilisons comme métrique l'accuracy et comme classifieur un SVM linéaire optimisé par gridsearch sur les pas de temps d'entraînement. Nous affichons les résultats en utilisant différents pourcentages des pas de temps d'entraînement afin d'évaluer la capacité d'extrapolation des modèles.

5.4.4 Paramètres

B²ADE

Nous avons testé plusieurs variations des fonctions de perte, notamment une similaire au modèle Time Control (voir équation 5.3.3). Seulement l'entraînement ne progressait que très peu et le modèle semblait n'apprendre que de la partie régulation qui fixent les extrémités des représentations d'auteurices, jusqu'à exploser. C'est cette observation qui nous a orientée vers le cadre variationnel

Grille de recherche des hyperparamètres de B²ADE	
Hyperparamètres	**Grille**
Nb de paires négatives	{1, 5, **10**, 20}
Monte Carlo sampling	{1, 5, **10**, 20}
Pas d'apprentissage	{1e-2, 1e-3, **5e-4**, 1e-4, 5e-5, 1e-5}
β	{1e-1, 1e-2, ..., **1e-12**}

TABLE 5.3 – Grille de recherche utilisée pour la sélection des hyperparamètres de B²ADE. Les valeurs retenues sont en gras.

du VIB. En effet, l'aspect temporel dans le modèle Time Control est continu, chaque phrase correspondant à un pas de temps de 1, là où il est beaucoup plus irrégulier dans notre cadre. Le VIB semble apporter plus de stabilité. A titre de comparaison, nous présenterons également les résultats avec une simple perte L_2 entre h_t^a et z_t^a, noté B²ADEL_2, sans aspect variationnel. Cela permet de justifier l'apport du cadre VIB. Les architectures détaillées des MLP sont présentées dans la description du modèle.

Sur le corpus NYT, la date de publication est disponible. Le temps est alors modélisé comme le nombre de jour depuis le 1er janvier 1990. Sur le corpus S2G en revanche, seule l'année est disponbile. Nous plaçons donc tous les documents d'une année au premier janvier et comptons le nombre de jour depuis le 1er janvier 1985.

Nous avons également testé pour les auteurices une variance uniquement temporelle, ainsi que sans aucune variance, donc sans modélisation gaussienne, comparativement au modèle global où la variance est temporelle et dépend de l'auteurice. Nous les noterons respectivement B²ADE-t, B²ADE-no-var et B²ADE. Nous détaillerons également ces résultats plus bas. Pour le reste des hyperparamètres, nous utilisons une grille de recherche sur un ensemble de validation, cette dernière est détaillée Table 5.3. Similairement à VADES, nous tirons 10 exemples négatifs et exécutons 10 tirages de Monte-Carlo, ce qui offre les meilleurs résultats dans un temps raisonnable. Nous utilisons l'algorithme d'optimisation Adam avec un pas d'apprentissage de $5e^{-4}$ et une réduction linéaire jusqu'à 0 du pas d'apprentissage à chaque étape (passage d'un batch). Nous lançons l'entraînement sur 100 epochs et pratiquons l'early stopping sur l'ensemble de validation. A noter que nous ajoutons le token correspondant à l'année d'écriture du document pendant l'entraînement et le masquons avec une probabilité croissante de 0.2 à 1 pendant les 5 premières epochs. Cela permet de progresser rapidement au début de l'entraînement. Enfin, nous utilisons $\beta = 1e^{-}12$, la deuxième partie de la fonction de perte correspondant à une régularisation sur les moyennes et variance. C'est la valeur qui offre les meilleurs résultats, mais elle correspond à une régularisation nulle, il est donc préférable simplement de ne pas ajouter ce second terme à la fonction de perte VIB.

TempoBERT

Nous utilisons le code fourni avec l'article [RGR22] [1]. Nous masquons le token de date avec une probabilité 0.90 comme recommandé et optimisons le pas d'apprentissage par gridsearch. Nous entraînons le modèle sur 15 epochs avec early stopping sur la perplexité sur l'ensemble de validation sur chacun des corpus. Pour prédire la date d'un document nous masquons simplement le token

1. https://github.com/guyrosin/tempobert

de date et utilisons la sortie du modèle comme prédiction. Nous utilisons ce modèle gelé comme encodeur de B²ADE.

DAR

Les corpus S2G et NYT proviennent tous deux du modèle DAR. Nous utilisons donc les paramètres recommandés dans l'article sur chacun des deux corpus, ainsi que le code fourni avec [DLD19] [2].

K-DGEA

Nous utilisons le code gracieusement fourni par l'auteurice [Gou+22a]. Une nouvelle fois, K-DGEA a été évalué sur les corpus S2G et NYT, nous utilisons donc les paramètres recommandés dans l'article. Nous faisons le choix d'utiliser l'encodeur de phrase USE [Cer+18], car il offre les meilleurs résultats en attribution d'auteurices.

5.5 Résultats

Pour rappel, le jeu de données NYT est utilisé en imputation. Autrement dit, les données tests et d'entraînement peuvent provenir de tous les pas de temps et la séparation est uniquement stratifiée par auteurices. Le jeu de données S2G est lui utilisé en prédiction. Pour chaque auteurice, 70% de leurs documents les plus anciens sont utilisés pour l'entraînement. Les 30% les plus récents sont ensuite séparés en validation et tests. Cela permet d'évaluer la capacité des modèles à extrapoler et à capturer la dynamique des auteurices.

5.5.1 Pour l'attribution d'auteurices

Les résultats pour l'attribution d'auteurices sur le jeu de données NYT sont présentés Table 5.4 et Table 5.5 pour le jeu de données S2G. Sur le corpus NYT, ce sont les modèles USE et BERT-ft$_A$ qui obtiennent les meilleurs résultats en fonction de la métrique étudiée. Si ce n'est pas étonnant pour BERT-ft$_A$, fine-tuné sur cette tâche, USE confirme sa forte capacité de modélisation même en étant pris sur l'étagère. B²ADE se place troisième tant que l'erreur de couverture que pour l'accuracy suivi de près par TempoBERT. Il arrive facilement premier des modèles de représentations dynamiques, plus de deux points devant K-DGEA en erreur de couverture. DAR quand à lui confirme qu'il n'est pas capable de produire des plongements de documents pertinents sans être conditionné par l'auteurice, ce qui est fatal sur cette tâche d'évaluation.

Il faut aussi noter que l'ajout de l'information temporelle de manière brute chez BERT-ft$_{T+A}$ et BERT-ft$_T$ les pénalise fortement. BERT-ft$_{T+A}$ régresse de 10 points en accuracy, quand BERT-ft$_T$ fait moins bien que la prédiction aléatoire. Pourtant, TempoBERT est aussi uniquement entraîné sur l'information temporelle. Mais le masquage s'appliquant aux tokens de date ET aux tokens usuels lui permet de conserver ses capacités à modéliser le langage. Plusieurs travaux présentent la prédiction de mots masqués comme une alternative robuste au fine-tuning [Zha+20].

Des conclusions légèrement différentes peuvent être tirées des résultats sur S2G. Si BERT-ft$_A$ obtient bien les meilleurs scores en imputation et en prédiction, l'aspect temporel semble plus important ici. En effet, l'information temporelle dans un cadre de publication scientifique semble permettre d'attribuer plus facilement les documents, en témoigne le score de TempoBERT et de

2. https://github.com/edouardelasalles/dar

Méthodes	NYT (546 auteurices) Erreur de couverture ↓	Accuracy ↑
USE	**11.1 (0.0)**	12.7 (0.0)
BERT-ft$_A$	11.4 (0.8)	**14.3 (0.4)**
BERT-ft$_T$	88.2 (1.0)	0.5 (0.2)
BERT-ft$_{T+A}$	33.4 (0.8)	3.7 (0.6)
TempoBERT	12.1 (1.1)	10.2 (0.9)
DAR	47.1 (1.1)	1.1 (0.2)
K-DGEA	14.5 (1.2)	9.8 (0.8)
B²ADE	11.9 (0.8)	12.2 (1.0)
B²ADE-L_2	27.2 (1.1)	10.1 (0.9)
B²ADE-t	12.5 (0.7)	11.9 (1.1)
B²ADE-no-var	13.1 (0.4)	11.3 (0.6)

TABLE 5.4 – Attribution d'auteurices sur le corpus NYT en imputation. En imputation, les jeux d'entraînement et de test couvrent tout l'espace temporel. Le meilleur modèle est en gras, le second souligné. L'écart type entre parenthèses. Pour l'erreur de couverture, le plus petit score est le meilleur, le plus grand pour l'accuracy.

	S2G (1117 auteurices)			
	Imputation		Prédiction	
Méthodes	**Erreur de couverture ↓**	**LRAP ↑**	**Erreur de couverture ↓**	**LRAP ↑**
USE	19.9 (0.0)	10.4 (0.0)	22.3 (0.0)	7.6 (0.0)
BERT-ft$_A$	**12.8 (0.8)**	**13.4 (0.7)**	**17.9 (1.3)**	**8.9 (0.8)**
BERT-ft$_T$	90.2 (1.9)	0.8 (0.2)	94.0 (2.3)	1.1 (0.2)
BERT-ft$_{T+A}$	15.3 (1.2)	10.6 (0.7)	25.9 (1.2)	6.7 (0.5)
TempoBERT	15.9 (1.5)	10.5 (1.1)	21.6 (0.7)	6.9 (0.4)
DAR	38.7 (2.0)	1.2 (0.4)	41.8 (1.2)	0.8 (0.4)
K-DGEA	20.4 (1.3)	10.1 (0.8)	28.2 (1.1)	6.6 (0.5)
B²ADE	15.5 (1.4)	10.4 (1.0)	20.6 (1.1)	7.3 (0.8)
B²ADE-L_2	24.2 (1.2)	9.7 (1.0)	35.5 (1.3)	4.3 (0.9)
B²ADE-t	15.9 (1.6)	10.3 (1.2)	21.1 (1.2)	7.1 (0.8)
B²ADE-no-var	19.9 (1.1)	10.1 (0.6)	22.3 (1.1)	7.0 (0.9)

TABLE 5.5 – Attribution d'auteurices sur le corpus S2G en imputation et en prédiction. En imputation, les jeux d'entraînement et de test couvrent tout l'espace temporel. En prédiction, le jeu d'entraînement concerne les premiers pas de temps et le jeu de test les derniers. Le meilleur modèle est en gras, le second souligné. L'écart type entre parenthèses. Pour l'erreur de couverture, le plus petit score est le meilleur, le plus grand pour la LRAP.

BERT-ft$_{T+A}$. En effet, le vocabulaire y est plus restreint et USE semble avoir plus de mal à distinguer les auteurices sans un entraînement spécifique. Pour cette raison, l'attribution d'auteurices sur ce corpus est plus complexe que sur NYT, en plus d'avoir plus d'auteurices d'où les scores comparativement moins bons.

Notre modèle se place troisième en imputation, mais second en prédiction, ce qui confirme sa capacité à extrapoler la dynamique de chaque auteurice. Il reste compétitif avec des approches fine-tunées sur une tâche unique ce qui est rassurant quand notre objectif est double : capturer les liens entre l'auteurice et sa production, mais aussi son évolution dans le temps. C'est l'ensemble des résultats en attribution et en datation qui permettront de juger de la qualité de cette modélisation.

5.5.2 Pour la datation de documents

Les résultats concernant la datation de documents sur les corpus NYT et S2G sont proposés Table 5.6, pour rappel il s'agit uniquement d'imputation ici. Sur NYT et pour les deux métriques, l'accuracy et l'erreur absolue moyenne c'est B^{2}ADE qui obtient les meilleurs scores. Sur une période de 26 années, il arrive à prédire la date d'un titre du New York Times avec une erreur moyenne de moins de 5 ans. Il dépasse même les modèles spécifiquement entraînés à la prédiction de la date, comme TempoBERT ou BERT$_T$.

Pour S2G, il se place premier en accuracy et second en erreur absolue moyenne. Bien que la période de temps sur S2G soit bien plus longue (35 ans contre 26), les erreurs absolues moyennes sont comparables et autour de 5 ans. Cela montre à quel point le vocabulaire utilisé en machine learning est très marqué temporellement, là où certains articles du New York Times sont plus intemporels (sur la mode, la maison, le business, ...). De plus, l'information supplémentaire de l'auteurice du document, présente pendant l'entraînement et encodé dans l'espace de représentation est une information cruciale pour la bonne datation d'un document, de par les thématiques spécifiques abordées ou simplement la période d'écriture. Cela est confirmé par les bons résultats de BERT$_T$ + A, ou même ceux corrects de BERT$_A$, qui n'a pourtant aucune information temporelle directe.

Les autres modèles de représentation d'auteurices sont assez loin derrière. Respectivement 2.8 points sur NYT et 6.4 sur S2G pour DAR, qui souffre toujours de ne pas manipuler de documents pendant l'entraînement et 2.0 points sur NYT et 2.5 points sur S2G pour K-DGEA. Il faut signaler également que ces deux modèles adoptent une approche discrète du temps, quand celle de B^{2}ADE est continue, ce qui lui permet plus de finesse dans la représentation dynamique des documents.

La combinaison des deux tâches d'évaluation précédentes, à savoir l'attribution d'auteurices et la datation de documents, confirme que l'objectif premier de notre modèle est bien atteint. Celui de construire des trajectoires des auteurices reflétant l'évolution de leur écriture, ici orientée plutôt sur les thématiques, mais qui selon l'encodeur ou le jeu de données peut être orientée vers le style littéraire. Le temps est bien encodé globalement dans l'espace de plongement et notre modèle dépasse des grands modèles de langue fine-tunés spécifiquement sur certaines de ces tâches.

5.5.3 Pour la classification d'auteurices

Les résultats pour la classification d'auteurices sont disponibles Table 5.7 pour la prédiction et Table 5.8 pour l'imputation. En prédiction, ils visent à confirmer la capacité d'extrapolation des modèles dynamiques, puisque l'entraînement a lieu sur les représentations d'auteurices de t_0^a à T^a et les tests sur les plongements à $T^a + 1$, $T^a + 2$, ..., $T^a_{\max}$. Nous rappelons que pour chaque année, la classe de l'auteurice est la conférence dans laquelle il publie le plus, parmis 22 conférences.

Méthodes	NYT (26 années) Accuracy ↑	NYT (26 années) MAE ↓	S2G (33 années) Accuracy ↑	S2G (33 années) MAE ↓
USE	10.4 (0.0)	6.8 (0.0)	9.81 (0.0)	5.1 (0.0)
BERT-ft$_A$	10.2 (0.1)	6.3 (0.1)	8.9 (0.5)	5.5 (0.3)
BERT-ft$_T$	13.0 (0.1)	5.5 (0.2)	12.2 (0.4)	4.4 (0.2)
BERT-ft$_{T+A}$	12.0 (0.1)	5.4 (0.2)	12.3 (0.3)	**4.2 (0.2)**
TempoBERT	13.1 (0.2)	5.2 (0.1)	12.1 (0.3)	4.7 (0.5)
DAR	5.6 (0.2)	7.5 (0.3)	4.8 (0.4)	10.0 (1.1)
K-DGEA	10.6 (0.4)	6.7 (0.3)	7.4 (0.3)	6.8 (0.5)
B²ADE	**13.3 (0.3)**	**4.7 (0.3)**	**12.6 (0.5)**	4.3 (0.4)
B²ADE-L_2	10.5 (0.2)	5.3 (0.3)	8.9 (0.2)	5.4 (0.3)
B²ADE-t	12.8 (0.4)	5.0 (0.3)	12.2 (0.4)	4.6 (0.2)
B²ADE-no-var	11.7 (0.3)	5.1 (0.2)	11.8 (0.3)	4.8 (0.3)

TABLE 5.6 – Datation de documents sur les corpus NYT et S2G en imputation. Le meilleur modèle est en gras, le second souligné. L'écart type entre parenthèses. Pour la MAE (erreur absolue moyenne), le plus petit score est le meilleur, le plus grand pour l'accuracy.

Nous affichons également les résultats en imputation, bien que les séparations entre entraînement et test ne soient pas faites spécifiquement par pas de temps. Ainsi, les modèles peuvent avoir vu quelques documents correspondant à une année dont ils devront prédire la classe de l'auteurice. Cela n'impacte pas vraiment les conclusions de cette section.

Sur cette tâche d'évaluation, les modèles de représentations d'auteurices sont de loin les plus performants devant les simples modèles de langue. En effet, même la représentation statique de DAR dépasse USE et TempoBERT. Chacun des trois modèles DAR, K-DGEA et B²ADE encode relativement bien les dynamiques d'évolution en arrivant à prédire sur plusieurs pas de temps les axes de recherche de plus d'un tiers des auteurices.

Même si le corpus S2G n'offre qu'une année de publication et pas une date précise, ce qui empêche la modélisation temporelle la plus fine pour B²ADE il partage la première place avec DAR, autant en imputation qu'en prédiction. La modélisation par une interpolation linéaire, bien que très simple, confirme ici sa robustesse. Nous pouvons aussi constater que l'essentiel de l'information du modèle DAR est contenu dans les représentations statiques des auteurices, l'aspect dynamique important peu. Notre modèle arrive autant à interpoler qu'à extrapoler la dynamique des auteurices tout en produisant des représentations continues. Nous allons maintenant analyser les résultats entre les différentes configurations de B²ADE.

5.5.4 Etudes d'ablation de B²ADE

Nous comparons ici les différentes configurations de B²ADE évoqués plus haut sur les deux premières tâches, d'attribution et de datation.

B²ADE-L_2 Puisque nous voulons contraindre à chaque instant t la représentation du document z_t^a à être proche du point h_t^a de la trajectoire de l'auteurice a, il pourrait sembler suffisant d'utiliser une simple fonction de perte de type erreur quadratique, avec tirage d'exemples négatifs.

	S2G Prédiction Accuracy (22 classes)			
Méthodes	**100 %**	**75%**	**50%**	**25%**
USE	28.8 (1.5)	28.3 (1.7)	25.8 (2.4)	24.8 (1.6)
TempoBERT	29.2 (2.3)	28.6 (2.5)	27.5 (1.6)	26.1 (1.6)
DAR (dynamique)	17.5 (1.3)	17.4 (1.4)	17.4 (2.3)	17.2 (1.5)
DAR (statique)	34.5 (1.1)	34.2 (1.8)	34.2 (2.0)	33.4 (1.6)
DAR (concat)	35.3 (1.6)	35.0 (1.4)	**34.9 (0.9)**	**34.7 (1.2)**
K-DGEA	35.0 (2.1)	34.7 (2.1)	34.2 (1.8)	33.0 (1.9)
B²ADE	**35.7 (2.0)**	**35.5 (2.0)**	34.8 (1.8)	34.5 (1.9)

TABLE 5.7 – Classification d'auteurices sur le corpus S2G en prédiction. Il s'agit de prédire la conférence dans laquelle les auteurices ont publié le plus pour chaque nouveau pas de temps à partir de leurs représentations temporelles. Les pourcentages indiquent la proportion de pas de temps du jeu d'entraînement utilisé pour entraîner le classifieur. Le meilleur modèle est en gras, le second souligné. L'écart type entre parenthèses. L'objectif est d'obtenir la plus grande accuracy.

	S2G Imputation Accuracy (22 classes)			
Méthodes	**100 %**	**75%**	**50%**	**25%**
USE	34.2(1.4)	33.5 (1.3)	32.9 (1.4)	30.8 (1.5)
TempoBERT	28.9 (1.6)	28.5 (1.6)	26.7 (1.5)	25.1 (1.4)
DAR (dynamique)	18.2 (1.3)	18.0 (1.2)	17.7 (1.7)	17.2 (1.5)
DAR (statique)	40.5 (1.3)	40.1 (1.1)	39.6 (0.9)	38.5 (1.3)
DAR (concat)	41.6 (1.7)	**41.3 (1.5)**	40.2 (0.8)	**39.5 (1.2)**
K-DGEA	40.3 (1.9)	40.1 (2.0)	39.6 (1.8)	38.8 (1.8)
B²ADE	**41.7 (1.6)**	41.1 (1.7)	**40.4 (1.6)**	39.2 (1.7)

TABLE 5.8 – Classification d'auteurices sur le corpus S2G en imputation. Il s'agit de prédire la conférence dans laquelle les auteurices ont publié le plus pour chaque nouveau pas de temps à partir de leurs représentations temporelles. Les pourcentages indiquent la proportion de pas de temps du jeu d'entraînement utilisé pour entraîner le classifieur. Le meilleur modèle est en gras, le second souligné. L'écart type entre parenthèses. L'objectif est d'obtenir la plus grande accuracy.

Seulement, comme observé Table 5.4 et 5.5, cette contrainte ne semble pas assez souple. Et si les résultats en datation sont très satisfaisants, le modèle ne parvient pas à séparer assez les trajectoires pour rivaliser en attribution.

B²ADE-no-var Cette variante se passe de la modélisation variationnelle du VIB. Sur tous les axes d'évaluation elle est relativement loin derrière B²ADE. Si les écarts sont plus faibles en datation (Table 5.6, c'est aussi que la tâche est plus difficile, ce qui confirme la pertinence du cadre variationnelle dans notre cadre et notamment l'utilisation du pont brownien comme hypothèse de modélisation.

B²ADE-t Cette variante utilise uniquement la variance temporelle du pont brownien, valant $\frac{t(T-t)}{T}$, là où B²ADE concatène la variance temporelle avec une propre à chaque auteurice. C'est un choix similaire à ce qui est fait dans VADES, mais aussi R-DGEA [Gou+22a]. L'ajout de cette variance propre à chaque auteurice permet de modéliser une dynamique d'évolution qui leur serait propre, certains restant sur les mêmes thématiques ou dans le même style, d'autres explorant un peu plus. Les résultats confirment le bien fondé de cette hypothèse de modélisation. En effet, B²ADE-t obtient de moins bons résultats, surtout en datation. Nous allons maintenant proposer une analyse plus qualitative de B²ADE à travers quelques visualisations.

5.5.5 Analyse qualitative de l'espace de représentation

Nous proposons une première représentation sur le corpus NYT des 10 auteurices les plus prolifiques et de leur production Figure 5.7. Les documents sont représentés par des points dont le gradient de couleur indique la période de publication de l'article (clair pour les plus anciens en 1990, foncé pour les plus récents en 2015). Les auteurices sont représentés par leur interpolation linéaire entre leur point de départ h_0^a et d'arrivée h_T^a.

Analyse des dynamiques sur NYT

La dynamique temporelle apparaît clairement le long des trajectoires, s'étalant du début de la production de chaque auteurice jusqu'à la fin. Si tous les auteurices représentés ont écrit sur la quasi totalité de la période du corpus NYT, Tyler Kepner n'est arrivé qu'en 2000, ce qui se traduit par une trajectoire plus courte sur la projection. B²ADE encode bien localement et pour chaque auteurice une dynamique propre. Tyler Kepner et Jack Curry ont deux trajectoires parallèles, très proches, car tous deux spécialistes de la section sports. De la même façon, Stephen Holden et Janet Maslin publient majoritairement dans les sections arts et films, d'où la proximité de leurs représentations. De son côté, William Grimes est l'auteur avec la plus grande diversité de sujets traités tout au long de sa carrière, ce qui se traduit par des documents plutôt étalés autour de sa trajectoire.

Ces projections confirment de manière qualitative la capacité de B²ADE à rapprocher chaque auteurice de sa production et à capturer leur dynamique, mais aussi à rapprocher deux auteurices prochent dans leur production dans l'espace de représentation. Nous allons le confirmer en nous intéressant aux plongements sur le corpus S2G, où les cas de co-écriture sont possibles.

Analyse des dynamiques sur S2G

Une représentation identique à celle du corpus NYT sur le corpus S2G est proposée Figure 5.8. Nous avons sélectionné les 10 auteurices les plus prolifiques, ainsi que quelques-uns de leurs co-auteurices. Les même observations que sur NYT sont possibles. Tout d'abord, les auteurices aux

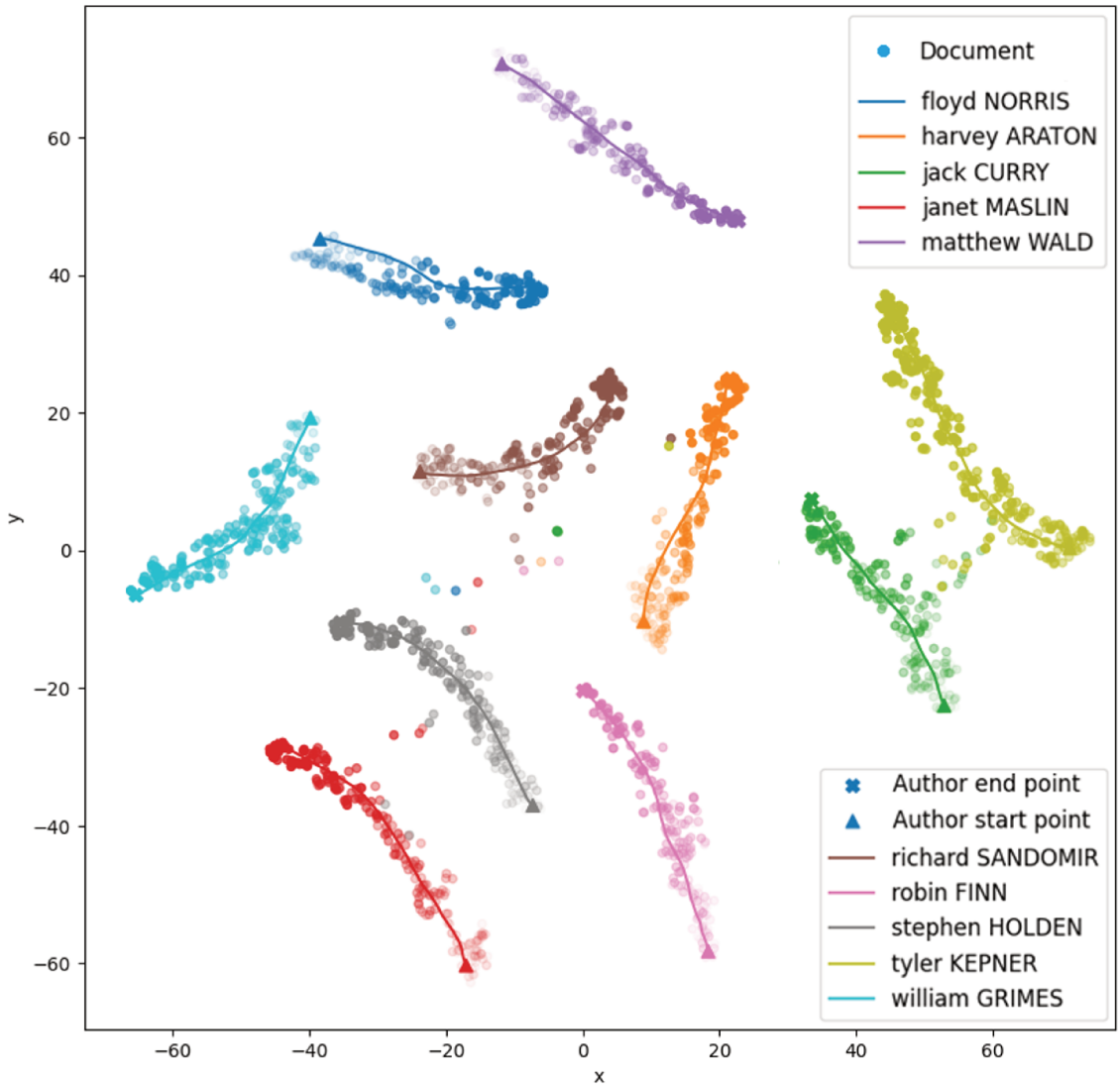


FIGURE 5.7 – Plongements des trajectoires d'auteurices et de documents issus du corpus NYT par B²ADE. Nous proposons ici une projection 2-D via T-SNE des 10 auteurices les plus prolifiques du jeu de données NYT et de leur production. Le gradient de couleur correspond à la période de publication du document.

périodes de production plus courtes (Kyunghyun Cho seulement 4 ans, Shuicheng Yan, 13 ans, ...) ont des trajectoires plus courtes.

En prenant le score de corrélation de Pearson entre la distance $||h_T^a - h_0^a||_2$ et la période de publication sur chacun des corpus, nous obtenons dans les deux cas des corrélations significatives, de 0.63 pour NYT et de 0.55 pour S2G. Ce lien entre période de production et longueur de la trajectoire est donc confirmé quantitativement, notre modèle parvient à représenter les dynamiques de chaque auteurice localement.

Là où les travaux des jeunes chercheurs et chercheuses sont au début de leur carrière centrés sur quelques thématiques très spécifiques, ils vont avoir tendance à se diversifier avec le temps. C'est ce qu'on constate globalement sur quasiment toutes les trajectoires, avec des nuages de points plus dispersés autour des dernières années. Nous constatons expérimentalement que les variances sont plus grandes en norme autour de h_T^a qu'autour de h_0^a. L'aspect variationnel et l'ajout d'une variance propre à chaque auteurice et dépendant de l'instant permet d'ajouter de la finesse de modélisation à l'interpolation linéaire brute du pont brownien. Si les trajectoires semblent assez rectilignes pour NYT, notamment car les thématiques principales traitées par les auteurices évoluent assez peu au cours de leur carrière, elles sont plus courbées sur S2G. C'est d'autant plus observable pour les auteurices publiant dans de nombreuses conférences différentes (Yoshua Bengio, Philip Yu, Bernard Schölkopf). Cela pourrait témoigner de l'évolution de leurs centres d'intérêt.

Le cas particulier des co-auteurices

Enfin, intéressons-nous maintenant au cas des auteurices multiples. Par exemple, Christopher Manning et Dan Klein ont partagé 14 articles entre 2001 et 2004, avant de s'orienter chacun dans des directions différentes, très orientée TAL pour Manning et un peu plus généraliste pour Klein. Leurs trajectoires sont parallèles et assez proches, avant de s'éloigner pour les années les plus récentes.

Les articles de Kyunghyun Cho présent dans le corpus S2G ont été publiés entre 2013 et 2017 et la majeure partie ont été co-écrits avec Yoshu Bengio. La représentation de Cho rejoint donc la fin de celle de Bengio comme nous pouvons le voir. Le même type d'observation est possible entre Jiawei Han et Philip Yu ou Thomas Huang, avec des collaborations plus ponctuelles cependant.

Bien que forcément limité par la projection T-SNE, ces deux visualisations nous ont permis de valider également qualitativement les qualités de notre modèle B²ADE. La capacité à capturer les dynamiques d'évolution des auteurices par des représentations continues. A un niveau local, où chaque auteurice conservera sa temporalité dans l'espace de représentation. Mais aussi à un niveau global, avec la possibilité de travailler sur des corpus avec des cas de co-écritures, qui constituent autant d'informations supplémentaires pour enrichir la qualité des plongements de B²ADE.

5.6 Conclusion et perspectives

Au cours de ce chapitre, nous nous sommes d'abord intéressés à la prise en compte du temps dans les représentations de documents et d'auteurices. Complexe avec des représentations de mots statiques, l'essor des plongements de mots contextualisés a permis quelques solutions [RGR22 ; Vas+18]. Dans le cadre des plongements dynamiques d'auteurices, seules deux solutions récentes existent : DAR [DLD19] et DGEA [Gou+22a], modélisant chacune le temps de manière discrète. C'est pourquoi nous avons proposé B²ADE (Brownian Bridge for Author and Date Embedding).

En utilisant le pont brownien pour appréhender le plongement d'auteurice comme une trajectoire continue, à savoir une interpolation linéaire. Ce modèle permet de créer un espace de représentation encodant directement la dynamique temporelle en plaçant chaque document sur la flèche du temps

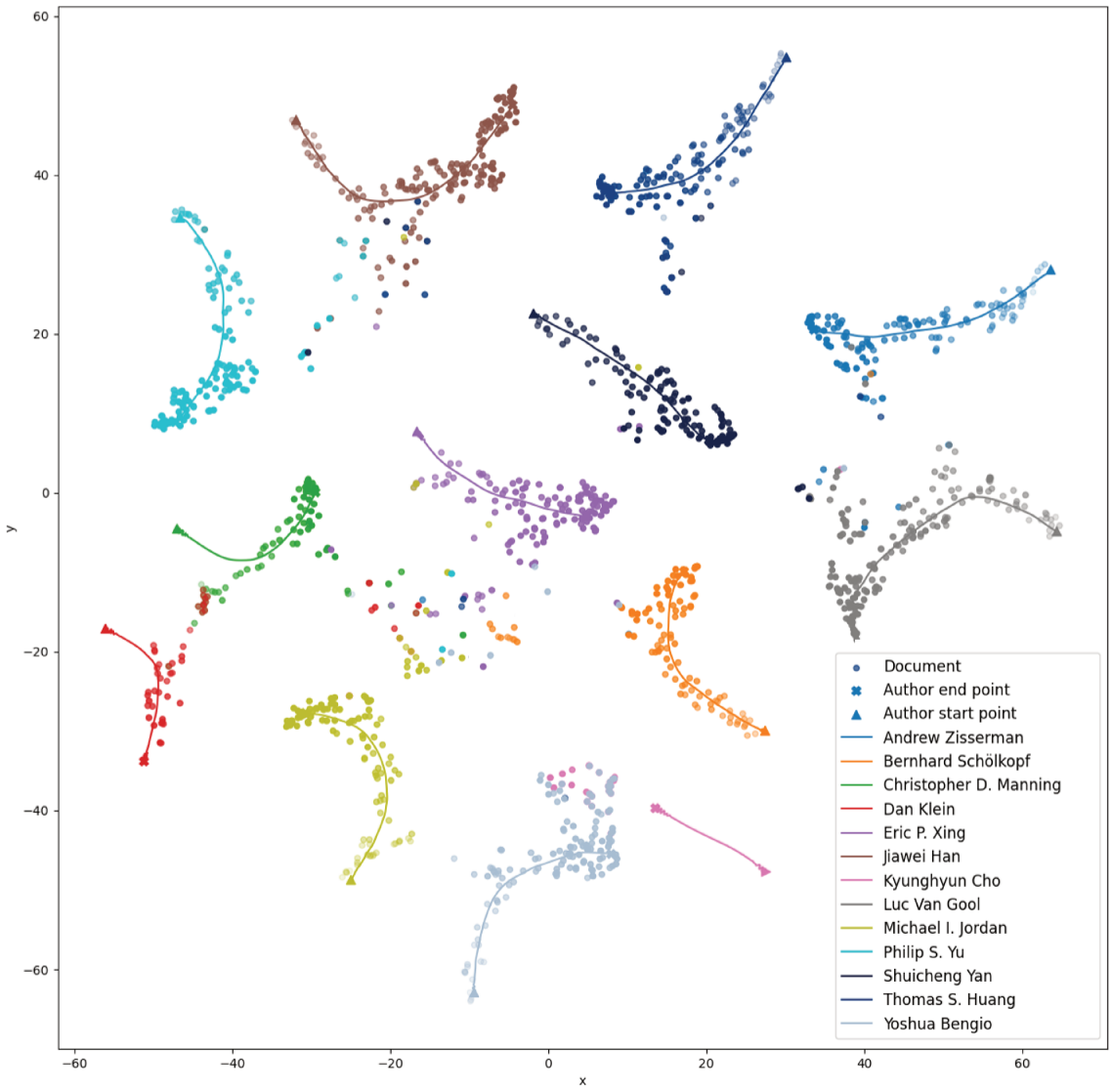


FIGURE 5.8 – Plongements des trajectoires d'auteurices et de documents issus du corpus S2G par B²ADE. Nous proposons ici une projection 2-D via T-SNE d'une sélection d'auteurices parmi les plus prolifiques du jeu de données S2G et de leur production. Le gradient de couleur correspond à la période de publication du document.

de son auteurice. Le cadre variationnel du VIB permet d'apporter plus de richesse à la modélisation et de donner à chaque auteurice une dynamique d'évolution qui lui est propre. Si notre modèle peut s'utiliser avec tous les encodeurs de documents, TempoBERT [RGR22] l'aide à capturer cette dynamique.

Dans le cadre d'amélioration, il pourrait être intéressant d'utiliser VADES où tout modèle se focalisant sur le style, comme encodeur, afin d'analyser les évolutions de ce dernier dans le temps. De plus, les échelles de temps considérées restent relativement courtes (quelques dizaines d'années), il faut encore pouvoir analyser le comportement de B²ADE sur des temps plus longs, que l'on croise par exemple en littérature.

Une variation intéressante dans l'analyse du style serait d'utiliser une approche similaire à celle de [Wan+22]. Nous évoquions des échelles de temps longs, passer à une échelle de temps très courte et modéliser chaque document comme une trajectoire. En effet, les textes longs présentent une dynamique dans l'articulation des phrases qu'il est intéressant d'analyser pour l'étude du style littéraire [Rea+16]. Est-ce que les documents issus de la même plume suivront des trajectoires communes ?

Enfin, la modélisation du temps peut-être enrichie. Comme le codage de position de BERT, plusieurs solutions plongent également la date ou le temps au sein même de leur modèle afin de lui permettre plus d'expressivité [Som+23 ; LDB18].

Chapitre 6

Conclusions

6.1 Conclusion

Nous avons commencé ce manuscrit en présentant les notions clefs de l'apprentissage profond, les architectures usuelles et leurs techniques d'entraînement et d'optimisation. Nous nous sommes ensuite focalisés sur son application à l'apprentissage de représentation, de mots dans un premier temps, statiques, contextualisés, jusqu'aux grands modèles de langue. Dans un second temps, nous avons détaillé les principaux modèles de représentations de documents et enfin d'auteurices, par le truchement de différentes fonctions d'agrégation.

Puis nous avons évoqué le traitement du style littéraire en Traitement Automatique de la Langue. Les méthodes de linguistique computationnelle sont purement statistiques et s'appuient sur des marqueurs plus ou moins complexes. Certains modèles de représentation de documents cherchent à le capturer. Il est apparu impossible d'évaluer si un modèle de plongements d'auteurs arrivent effectivement à appréhender le style. Nous avons donc proposé notre première contribution, un framework d'évaluation s'appuyant sur les marqueurs du style de la linguistique pour évaluer à quel point les modèles de plongements capturent le style. Ce framework a permis notamment de montrer les limites de l'attribution d'auteurs comme seule tâche d'entraînement, et le potentiel des modèles de langue récents à appréhender des notions de langue complexes.

Dans la continuité de ces travaux, nous avons proposé un modèle de plongements d'auteurs et de documents unifiant les approches à base de marqueurs et les récents encodeurs de textes afin de représenter le style littéraire. Le modèle VADES (*Variational Author and Document Embedding with Style*) utilise le cadre variationnel du VIB (*Variational Information Bottleneck*) pour représenter les auteurices et leurs documents comme des gaussiennes dans un seul et même espace latent. Nous nous comparons à l'état de l'art en représentation d'auteurs orientée vers le style, présentée en amont, en attribution d'auteurices et sur notre métrique. VADES rivalise avec les compétiteurs en attribution d'auteurs et les surpassent largement en capture du style littéraire.

Enfin, nous avons fait le constat suivant : l'écriture d'un auteur ou d'une autrice fluctue grandement dans le temps et il serait plus judicieux de chercher à le représenter en prenant en compte cet aspect temporel. Nous nous sommes donc intéressés aux modèles de plongements dynamiques de documents et d'auteurs. Ils sont peu nombreux, et discrétisent tous le temps pour traiter les corpus. En nous appuyant sur le processus gaussien du pont brownien (*Brownian Bridge*), nous proposons le modèle B^{2}ADE (*Brownian Bridge for Author and Document Embedding*), qui représentent les

auteurices comme des trajectoires continues. Cette modélisation variationnelle se marie très bien avec le cadre VIB, et permet à B²ADE d'être efficace tant en attribution d'auteurices qu'en datation de documents ou en classification d'auteurices, surpassant même l'existant sur cette dernière tâche.

6.2 Perspectives dans le cadre de LIFRANUM

Ici, nous présentons les perspectives d'application de nos travaux dans le cadre du projet LIFRANUM, nous exposerons ensuite nos perspectives de recherche plus générales.

Cette thèse s'inscrivait dans le cadre du projet LIFRANUM (voir section 1.4). Les deux premières contributions proposées sont d'ailleurs directement construites dans l'optique de ce projet pour y être appliquées. Néanmoins, cela n'a pas pu se faire, pour plusieurs raisons.

La première est la difficulté à élaborer un corpus sur lequel travailler. Deux solutions ont été envisagées :

— Utiliser la solution d'archivage de la BnF pour crawler massivement le web

— Utiliser les APIs des différentes plateformes de production utilisées par les auteurs numériques (Blogger, Wordpress, Twitter, ...)

Si la première a l'avantage d'être automatique et de pouvoir passer à l'échelle facilement, elle possède deux inconvénients majeurs. Le premier est de récupérer énormément de pages parasites dans le corpus, ne relevant pas de la littérature francophone nativement numérique. Le second est de produire des fichiers au format WARC, des archives difficiles à traiter car contenant le contenu HTML entier de plusieurs pages, dont il faudrait parvenir à extraire le texte, l'auteurice, ... La seconde solution a l'avantage de fournir des données propres, le texte, le titre, l'auteurice. Par contre, il faut définir en amont quels comptes ou blogs sont pertinents à intégrer au corpus, soit manuellement, soit automatiquement au risque d'intégrer des contenus parasites.

Bien que ne faisant pas partie des missions du projet associées à la thèse, nous avons créé via les APIs Blogger et Wordpress un corpus test pour le projet, mais qui n'a pas pu être exploité pour l'instant. L'objectif in fine est bien d'appliquer VADES à ces données, avec un encodeur en langue française (CamemBERT [Mar+20], FlauBERT [Le+20]) ou multilingue (USE [Cer+18]), et un ensemble de descripteurs définis avec les chercheurs en littérature du laboratoire MARGE.

En prenant du recul, la collaboration entre littéraire et informaticien n'a pas été aussi fructueuse qu'espérée, justement à cause de l'absence de ce corpus autour duquel échanger et expérimenter. Si les échanges ont été de qualité et intéressants, il a manqué cette base de travail qui aurait permis de les mettre en application et d'avancer. Les différentes tentatives de vulgarisation pouvaient sembler creuses sans réelle application derrière. Enfin, quelques linguistes computationnelles auraient pu permettre de raccorder les deux mondes que sont la littérature et le TAL afin d'amener plus de fluidité dans les échanges.

Nous espérons encore pouvoir appliquer nos modèles VADES et B²ADE à un hypothétique corpus LIFRANUM et de pouvoir discuter des résultats et des analyses avec nos collaborateurs du laboratoire MARGE.

Avant d'évoquer les perspectives qui découlent de nos recherches, nous détaillons dans la section suivante les infrastructures de calculs utilisées pour l'entraînement des différents modèles développés dans cette thèse ainsi que le coût énergétique engendré.

6.3 Infrastructure de calculs et empreinte carbone

Ces travaux ont bénéficié d'un accès aux ressources de calcul et de stockage au IDRIS au travers de l'allocation de ressources 2021-AD011012369 attribuée par GENCI sur la partition V100 du calculateur Jean Zay.

Cette allocation a été renouvelée 2 fois, pour les années 2022 et 2023. Elles ont bénéficié principalement aux développements des modèles détaillés dans les chapitre 4 et 5. Le nombre d'heures GPU consommées totales est d'environ 19 500 (selon le découpage chronologique suivant : $5000 + 7000 + 7500$). A cela s'ajoute environ 3000 heures GPU via Google Colab (nous avons estimé environ 1000 h.GPU par an).

A titre informatif, nous détaillons dans la Table 6.3 ce que cela représente en termes de consommation électrique et d'émission de gaz à effets de serre. Pour la consommation des heures GPU de Jean Zay, nous avons utilisé les ressources disponibles sur leur site [1]. Concernant Google Colab, nous nous sommes appuyés sur les travaux de Lucia Bouza Heguerte, Aurélie Bugeau et Loïc Lannelongue [HBL23]. Pour l'équivalence entre kWh et émission de kCO_2 équivalente, nous avons utilisé les données disponibles sur le site de RTE pour la consommation en France [2] et celles de l'Agence Européenne de l'Environnement [3] pour l'Europe de l'Ouest. En effet, les clusters GPU Google Colab que nous avons utilisé y sont situés pour la plupart.

	h.GPU		Conso. Elec. (MWh)			Emission (kCO_2eq)		
Années	**Jean Zay**	**Colab**	**Jean Zay**	**Colab**	**Total**	**Jean Zay**	**Colab**	**Total**
2021	5000	1000	1.30	1.65	2.95	72.8	267.3	340.1
2022	7000	1000	1.81	1.65	3.46	123.1	277.2	400.3
2023	7500	1000	1.94	1.65	3.59	131.9	277.2	409.1
Total	19500	3000	5.05	4.95	10.00	327.8	821.7	1149.5

TABLE 6.1 – Répartition des heures GPU de calculs utilisées au cours de cette thèse, par année et par ressources. Sont associées la consommation électrique estimée ainsi que les émissions en kCO_2eq en découlant.

Bien que ces résultats ne prennent en compte que nos calculs réalisés sur des clusters GPUs, nous avons consommé en 3 ans environ 10 MWh. Pour comparaison, la consommation électrique moyenne d'une personne en France en 2022 est de 2.2 MWh. En termes d'émissions, nous arrivons à un peu plus d'une tonne de CO_2 équivalente. Cela correspond à 1 aller-retour Paris-New York en avion, ou encore à la production d'une tonne de viande bovine.

1. http://www.idris.fr/jean-zay/calcul-empreinte-carbone.html
2. https://www.rte-france.com/eco2mix/les-emissions-de-co2-par-kwh-produit-en-france
3. https://www.eea.europa.eu/en/analysis/indicators/greenhouse-gas-emission-intensity-of-1

6.4 Perspectives générales

6.4.1 Choix des marqueurs les plus pertinents selon l'application et la langue

La liste des marqueurs que nous avons proposée chapitre 3 et utilisée chapitre 4, disponible en Appendice A, est extraite de plusieurs travaux d'analyse du style littéraire. Nous l'avons également étendue au français [4]. Seulement, c'est une transcription telle quelle des descripteurs choisis pour l'anglais vers le français. Ces deux langues sont pourtant extrêmement différentes, l'une anglo-saxonne l'autre latine, c'est déjà une limite des marqueurs du français que nous proposons.

Les échanges que nous avons pu avoir au cours de cette thèse avec des chercheurs en linguistique nous ont orientés vers d'autres descripteurs. Par exemple, nous pourrions ajouter des mesures de variance, dans la taille des phrases, des mots ou dans les fréquences d'apparition de certains descripteurs. Le choix de ces marqueurs peut évoluer avec la langue étudiée ainsi qu'avec le corpus sur lequel ils seront utilisés. Ces travaux de construction du meilleur proxy du style écrit sont à mener en s'appuyant autant que possible sur les connaissances des chercheurs en linguistique et littérature.

6.4.2 Le cas des documents longs

Plusieurs fois au cours des chapitres précédents et principalement du chapitre 4 nous avons mentionné l'une des grandes limites des modèles de langue comme BERT ou RoBERTa : ils ne peuvent traiter que des documents de 512 tokens au plus. Les modèles les plus récents atteignent des limites de taille de contexte autour des 4096 tokens, GPT4 [Ope23] proposant même une variante entraînée sur des contextes de 32000 tokens. Cela correspond à environ 50 pages de roman. Ces tailles de contexte dépassent même celles des modèles spécifiquement construits pour les documents longs, comme BigBird [Zah+20] ou LongFormer [BPC20]. Ces derniers remplacent l'auto-attention par une combinaison de fenêtre d'attention et d'attention aléatoire qui permet de réduire la complexité des calculs.

Habituellement, les modèles de représentation d'auteurices et de documents traitent les documents longs en les coupant en autant de textes d'entraînement respectant la taille limite. Sur le Projet Gutenberg, le nombre d'exemples devient rapidement insurmontable. Il serait intéressant d'évaluer et de comparer l'ensemble des encodeurs capables de représenter des documents plus longs, que ce soit pour capturer le style ou la temporalité. En effet, sur ces deux aspects l'information pertinente peut s'articuler tout au long du texte.

6.4.3 Désentrelacement des représentations

L'idéal en apprentissage de représentation d'auteurices serait de réussir à désentrelacer l'aspect sémantique de l'aspect stylistique. C'est à dire construire une représentation interprétable dans laquelle ces deux notions seraient séparées. Même si certains travaux semblent indiquer que cette séparation n'est pas strictement possible [Jaf+20 ; Lam+19], l'objectif est de pouvoir les désentrelacer autant que possible.

Les méthodes utilisées nécessitent souvent des données annotées. Dans [Lia+20] il faut une étiquette par mot. Dans [Jai+18] chaque document doit être décrit selon plusieurs aspects. Dans

4. https://github.com/EnzoFleur/style_embedding_evaluation

le cadre du style littéraire, l'annotation est difficile car c'est une notion mal définie. Néanmoins, certains descripteurs clés peuvent être utilisés afin de créer des catégories tout en se passant d'annotation manuelle. A notre connaissance l'essentiel des méthodes visant à désentrelacer le style du contenu sémantique concernent la génération de texte, et se focalisent sur des styles définis (positif/négatif, familier/formel, ...).

6.4.4 Modéliser les documents comme des trajectoires

Nous avons déjà évoqué dans le chapitre 5 l'intérêt d'utiliser un encodeur se focalisant sur le style littéraire dans le modèle B²ADE afin de déterminer si l'évolution du style des auteurices étaient visibles.

Dans les documents longs, les romans, les essais, le style s'articule et se construit tout au long du texte, dans l'enchaînement des phrases et le déroulement de la pensée. Les méthodes de représentation de documents concentrent tout cela dans un seul vecteur en perdant nécessairement un peu de cette information.

En utilisant le modèle B²ADE nous pouvons représenter chaque document comme une trajectoire. Comme dans l'article [Wan+22], chaque phrase correspond à un pas de temps. Chaque plongement d'auteurices serait le point de départ de chacun des documents de leurs productions. Analyser ces trajectoires permettrait peut-être ensuite d'extraire des groupes d'auteurices au style similaire dans l'agencement de leur discours. De la même façon, d'autres processus gaussiens peuvent être utilisés, comme c'est fait pour les vidéos dans [Bha+20]. En particulier, le mouvement brownien fractionnel (*fractional brownian motion*) permet d'accorder plus ou moins d'importance à l'aspect temporel en fonction d'un hyperparamètre. Voir si les trajectoires se focalisant sur l'évolution temporelle capturent plus le style que celle visant à lisser la représentation au cours du temps est une question que l'on peut se poser.

6.4.5 Modélisation plus complexe du temps

Enfin, dans la construction du modèle B²ADE nous utilisons le temps en jours tel que nous l'obtenons dans le jeu de données. Or, dans la littérature il est courant d'en apprendre une représentation également afin d'obtenir une modélisation temporelle plus fine. Par exemple, [LDB18] plonge le temps en dimension 32 par un MLP avant de concaténer sa représentation à celle du document. [Som+23] utilise une modélisation du temps identique à celle de l'encodage de position utilisée dans BERT (voir Section 2.3.2) pour plus d'expressivité. Nous pensons qu'ajouter ce type de modélisation temporelle à notre modèle B²ADE permettrait d'améliorer ses capacités à capturer les dynamiques des auteurices.

Annexe A

Appendice A : Détails des descripteurs du style

Nous présentons dans le tableau A la liste détaillée des descripteurs du style présenté en section 3.4.2. Pour chacun, nous fournissons une description ainsi que la famille de descripteurs dans lequel nous le plaçons. Enfin, nous faisons une étude d'ablation sur la tâche d'attribution d'auteurices sur le Projet Gutenberg réduit à 10 auteurs. Nous affichons les scores obtenus en accuracy et en erreur de couverture avec une régression logistique en retirant chacun des descripteurs. Pour rappel, l'accuracy en utilisant tous les marqueurs est de 82.1 (écart type de 0.9) et l'erreur de couverture de 1.04 (écart type de 0.28). Chaque marqueur a son importance dans la caractérisation du style écrit puisque ces scores dépassent assez largement les scores obtenus en ablation. Pour les POS-TAG (*Part-of-speech TAG*) et NER (entitées nommées) la liste de leur signification est disponible ici : glossaire spacy.

Marqueur	Description	Famille	Accuracy	Erreur de couverture
avg_w_len	Longueur moyenne des mots	Structural	78.3 (0.8)	1.66 (0.34)
tot_short_w	Fréq. de mots de moins de 4 lettres	Structural	79.3 (0.9)	1.61 (0.36)
tot_digit	Fréq. de chiffre	Structural	75.0 (1.1)	1.79 (0.45)
tot_upper	Fréq. de lettres majuscules	Structural	79.2 (0.9)	1.63 (0.29)
avg_s_len	Longueur moyenne des phrases	Structural	79.0 (1.0)	1.61 (0.38)
hapax	Hapax Legomena	Structural	77.5 (1.0)	1.65 (0.35)
dis	Dislegomena	Structural	78.5 (1.0)	1.67 (0.49)
syllable_count	Syllabes moyennes par mots	Structural	78.8 (1.0)	1.64 (0.4)
avg_w_freqc	Fréq. moyenne des mots	Structural	81.3 (1.0)	1.55 (0.31)
a_letter	Fréq. de la lettre a	Lettres	77.8 (1.1)	1.61 (0.43)
b_letter	Fréq. de la lettre b	Lettres	76.5 (0.9)	1.55 (0.31)
c_letter	Fréq. de la lettre c	Lettres	79.7 (0.8)	1.63 (0.28)
d_letter	Fréq. de la lettre d	Lettres	79.8 (0.8)	1.58 (0.31)
e_letter	Fréq. de la lettre e	Lettres	79.7 (0.9)	1.63 (0.37)
f_letter	Fréq. de la lettre f	Lettres	78.7 (1.0)	1.59 (0.31)
g_letter	Fréq. de la lettre g	Lettres	80.0 (0.9)	1.57 (0.43)
h_letter	Fréq. de la lettre h	Lettres	76.2 (0.8)	1.63 (0.28)
i_letter	Fréq. de la lettre i	Lettres	80.8 (0.9)	1.46 (0.28)

j_letter	Fréq. de la lettre j	Lettres	79.7 (1.0)	1.58 (0.36)
k_letter	Fréq. de la lettre k	Lettres	78.7 (1.2)	1.59 (0.38)
l_letter	Fréq. de la lettre l	Lettres	78.3 (0.8)	1.69 (0.4)
m_letter	Fréq. de la lettre m	Lettres	74.0 (1.3)	1.73 (0.48)
n_letter	Fréq. de la lettre n	Lettres	78.2 (0.8)	1.6 (0.35)
o_letter	Fréq. de la lettre o	Lettres	75.5 (0.8)	1.7 (0.27)
p_letter	Fréq. de la lettre p	Lettres	77.8 (1.1)	1.63 (0.37)
q_letter	Fréq. de la lettre q	Lettres	80.3 (0.9)	1.57 (0.28)
r_letter	Fréq. de la lettre r	Lettres	78.5 (0.8)	1.55 (0.31)
s_letter	Fréq. de la lettre s	Lettres	76.8 (1.0)	1.63 (0.41)
t_letter	Fréq. de la lettre t	Lettres	77.8 (0.9)	1.52 (0.25)
u_letter	Fréq. de la lettre u	Lettres	79.7 (1.1)	1.56 (0.33)
v_letter	Fréq. de la lettre v	Lettres	77.5 (0.8)	1.7 (0.4)
w_letter	Fréq. de la lettre w	Lettres	79.0 (0.9)	1.58 (0.3)
x_letter	Fréq. de la lettre x	Lettres	79.2 (0.7)	1.62 (0.28)
y_letter	Fréq. de la lettre y	Lettres	76.3 (1.1)	1.66 (0.39)
z_letter	Fréq. de la lettre z	Lettres	84.2 (0.9)	1.36 (0.23)
digit_0	Fréq. du nombre 0	Nombres	81.2 (0.9)	1.47 (0.32)
digit_1	Fréq. du nombre 1	Nombres	78.8 (0.7)	1.63 (0.26)
digit_2	Fréq. du nombre 2	Nombres	78.5 (0.9)	1.6 (0.31)
digit_3	Fréq. du nombre 3	Nombres	76.7 (0.9)	1.65 (0.35)
digit_4	Fréq. du nombre 4	Nombres	78.0 (1.3)	1.57 (0.41)
digit_5	Fréq. du nombre 5	Nombres	78.8 (0.8)	1.59 (0.3)
digit_6	Fréq. du nombre 6	Nombres	78.2 (0.7)	1.59 (0.28)
digit_7	Fréq. du nombre 7	Nombres	77.3 (0.8)	1.6 (0.32)
digit_8	Fréq. du nombre 8	Nombres	78.7 (0.8)	1.57 (0.31)
digit_9	Fréq. du nombre 9	Nombres	79.7 (0.8)	1.54 (0.3)
func_w_freq	Fréq. de mots outils	Mots outils	81.3 (0.9)	1.47 (0.29)
am	Fréq. du mot am	Mots outils	78.8 (1.0)	1.57 (0.33)
an	Fréq. du mot an	Mots outils	77.5 (0.8)	1.66 (0.31)
same	Fréq. du mot same	Mots outils	79.0 (1.0)	1.6 (0.38)
until	Fréq. du mot until	Mots outils	75.8 (1.2)	1.7 (0.38)
whom	Fréq. du mot whom	Mots outils	75.3 (1.3)	1.74 (0.44)
myself	Fréq. du mot myself	Mots outils	76.5 (1.0)	1.59 (0.33)
then	Fréq. du mot then	Mots outils	75.2 (0.9)	1.71 (0.45)
needn	Fréq. du mot needn	Mots outils	77.7 (0.9)	1.64 (0.33)
her	Fréq. du mot her	Mots outils	79.5 (1.1)	1.55 (0.39)
we	Fréq. du mot we	Mots outils	76.2 (1.0)	1.67 (0.4)
mightn	Fréq. du mot mightn	Mots outils	77.3 (0.9)	1.67 (0.35)
they	Fréq. du mot they	Mots outils	73.7 (0.8)	1.7 (0.33)
yourselves	Fréq. du mot yourselves	Mots outils	80.3 (1.2)	1.56 (0.39)
ma	Fréq. du mot ma	Mots outils	80.8 (0.8)	1.5 (0.28)
or	Fréq. du mot or	Mots outils	80.7 (0.8)	1.5 (0.27)
when	Fréq. du mot when	Mots outils	78.7 (1.1)	1.64 (0.37)
did	Fréq. du mot did	Mots outils	78.8 (1.1)	1.58 (0.36)
some	Fréq. du mot some	Mots outils	79.0 (1.1)	1.61 (0.38)
the	Fréq. du mot the	Mots outils	77.8 (1.0)	1.57 (0.34)
by	Fréq. du mot by	Mots outils	76.7 (1.1)	1.61 (0.35)
so	Fréq. du mot so	Mots outils	78.3 (0.8)	1.7 (0.31)

up	Fréq. du mot up	Mots outils	79.3 (0.8)	1.51 (0.26)
his	Fréq. du mot his	Mots outils	79.3 (0.9)	1.59 (0.3)
are	Fréq. du mot are	Mots outils	76.7 (0.9)	1.61 (0.28)
not	Fréq. du mot not	Mots outils	77.7 (1.1)	1.6 (0.41)
him	Fréq. du mot him	Mots outils	80.8 (1.1)	1.61 (0.32)
from	Fréq. du mot from	Mots outils	77.0 (0.9)	1.62 (0.35)
while	Fréq. du mot while	Mots outils	79.8 (1.0)	1.69 (0.47)
at	Fréq. du mot at	Mots outils	77.7 (1.0)	1.64 (0.5)
o	Fréq. du mot o	Mots outils	78.8 (1.0)	1.66 (0.44)
couldn	Fréq. du mot couldn	Mots outils	79.0 (0.8)	1.55 (0.26)
few	Fréq. du mot few	Mots outils	77.5 (0.9)	1.66 (0.33)
she	Fréq. du mot she	Mots outils	78.7 (0.9)	1.57 (0.26)
re	Fréq. du mot re	Mots outils	75.0 (1.1)	1.71 (0.38)
out	Fréq. du mot out	Mots outils	76.3 (1.1)	1.65 (0.43)
hers	Fréq. du mot hers	Mots outils	76.3 (1.0)	1.64 (0.31)
me	Fréq. du mot me	Mots outils	76.5 (0.9)	1.64 (0.31)
s	Fréq. du mot s	Mots outils	79.5 (0.9)	1.59 (0.37)
but	Fréq. du mot but	Mots outils	79.7 (1.0)	1.56 (0.32)
what	Fréq. du mot what	Mots outils	78.7 (0.8)	1.55 (0.31)
off	Fréq. du mot off	Mots outils	79.2 (11.5)	1.61 (0.4)
isn	Fréq. du mot isn	Mots outils	76.8 (0.9)	1.66 (0.37)
yourself	Fréq. du mot yourself	Mots outils	78.2 (0.9)	1.62 (0.32)
after	Fréq. du mot after	Mots outils	76.3 (12.9)	1.71 (0.43)
t	Fréq. du mot t	Mots outils	80.2 (0.9)	1.55 (0.38)
who	Fréq. du mot who	Mots outils	78.8 (10.6)	1.53 (0.33)
each	Fréq. du mot each	Mots outils	75.8 (11.3)	1.68 (0.36)
under	Fréq. du mot under	Mots outils	77.0 (11.3)	1.64 (0.37)
been	Fréq. du mot been	Mots outils	81.0 (0.9)	1.5 (0.33)
shouldn	Fréq. du mot shouldn	Mots outils	78.8 (0.9)	1.6 (0.35)
y	Fréq. du mot y	Mots outils	82.3 (0.8)	1.52 (0.35)
doing	Fréq. du mot doing	Mots outils	77.2 (12.2)	1.68 (0.39)
yours	Fréq. du mot yours	Mots outils	79.2 (0.9)	1.52 (0.29)
for	Fréq. du mot for	Mots outils	80.5 (0.9)	1.52 (0.3)
in	Fréq. du mot in	Mots outils	77.7 (0.6)	1.66 (0.31)
ve	Fréq. du mot ve	Mots outils	80.2 (0.8)	1.52 (0.36)
them	Fréq. du mot them	Mots outils	81.2 (0.9)	1.54 (0.32)
herself	Fréq. du mot herself	Mots outils	77.7 (10.2)	1.64 (0.31)
as	Fréq. du mot as	Mots outils	78.8 (0.9)	1.62 (0.36)
more	Fréq. du mot more	Mots outils	79.5 (0.8)	1.58 (0.33)
does	Fréq. du mot does	Mots outils	79.7 (0.8)	1.6 (0.35)
most	Fréq. du mot most	Mots outils	76.8 (1.2)	1.63 (0.38)
if	Fréq. du mot if	Mots outils	76.7 (0.7)	1.57 (0.27)
both	Fréq. du mot both	Mots outils	76.7 (1.1)	1.58 (0.33)
there	Fréq. du mot there	Mots outils	78.0 (1.1)	1.57 (0.4)
own	Fréq. du mot own	Mots outils	79.3 (1.1)	1.62 (0.43)
don	Fréq. du mot don	Mots outils	77.8 (1.1)	1.61 (0.35)
m	Fréq. du mot m	Mots outils	79.8 (0.8)	1.54 (0.36)
now	Fréq. du mot now	Mots outils	77.5 (1.0)	1.61 (0.34)
himself	Fréq. du mot himself	Mots outils	79.8 (0.8)	1.65 (0.28)
hadn	Fréq. du mot hadn	Mots outils	76.0 (0.9)	1.67 (0.36)

about	Fréq. du mot about	Mots outils	76.8 (0.9)	1.56 (0.3)
with	Fréq. du mot with	Mots outils	78.7 (1.0)	1.58 (0.32)
down	Fréq. du mot down	Mots outils	79.7 (0.8)	1.64 (0.37)
you	Fréq. du mot you	Mots outils	76.5 (1.1)	1.67 (0.33)
he	Fréq. du mot he	Mots outils	77.7 (0.8)	1.62 (0.26)
which	Fréq. du mot which	Mots outils	78.3 (1.0)	1.52 (0.29)
were	Fréq. du mot were	Mots outils	78.7 (1.0)	1.65 (0.39)
our	Fréq. du mot our	Mots outils	78.2 (1.0)	1.63 (0.37)
against	Fréq. du mot against	Mots outils	77.7 (1.0)	1.64 (0.45)
here	Fréq. du mot here	Mots outils	79.3 (0.9)	1.64 (0.37)
their	Fréq. du mot their	Mots outils	78.3 (0.80)	1.71 (0.31)
do	Fréq. du mot do	Mots outils	78.0 (1.1)	1.66 (0.41)
doesn	Fréq. du mot doesn	Mots outils	78.7 (1.0)	1.62 (0.36)
will	Fréq. du mot will	Mots outils	82.8 (0.8)	1.5 (0.37)
and	Fréq. du mot and	Mots outils	79.8 (1.1)	1.61 (0.42)
it	Fréq. du mot it	Mots outils	79.2 (0.90)	1.57 (0.33)
that	Fréq. du mot that	Mots outils	78.7 (0.8)	1.61 (0.28)
can	Fréq. du mot can	Mots outils	73.8 (1.2)	1.71 (0.41)
than	Fréq. du mot than	Mots outils	78.5 (1.0)	1.59 (0.3)
won	Fréq. du mot won	Mots outils	80.3 (1.1)	1.56 (0.42)
this	Fréq. du mot this	Mots outils	77.0 (1.1)	1.68 (0.38)
mustn	Fréq. du mot mustn	Mots outils	78.2 (0.9)	1.6 (0.29)
why	Fréq. du mot why	Mots outils	77.0 (1.1)	1.63 (0.39)
your	Fréq. du mot your	Mots outils	79.8 (0.8)	1.66 (0.3)
such	Fréq. du mot such	Mots outils	80.7 (0.9)	1.45 (0.3)
d	Fréq. du mot d	Mots outils	77.0 (0.8)	1.62 (0.32)
themselves	Fréq. du mot themselves	Mots outils	81.0 (0.9)	1.5 (0.32)
had	Fréq. du mot had	Mots outils	76.0 (1.0)	1.67 (0.35)
wasn	Fréq. du mot wasn	Mots outils	79.8 (0.7)	1.53 (0.28)
through	Fréq. du mot through	Mots outils	76.8 (0.8)	1.7 (0.35)
before	Fréq. du mot before	Mots outils	78.5 (0.7)	1.63 (0.34)
having	Fréq. du mot having	Mots outils	80.2 (0.8)	1.54 (0.39)
my	Fréq. du mot my	Mots outils	79.0 (0.9)	1.57 (0.35)
is	Fréq. du mot is	Mots outils	78.7 (1.0)	1.59 (0.34)
over	Fréq. du mot over	Mots outils	77.7 (0.8)	1.61 (0.35)
only	Fréq. du mot only	Mots outils	76.5 (0.7)	1.72 (0.33)
aren	Fréq. du mot aren	Mots outils	80.0 (1.2)	1.58 (0.45)
these	Fréq. du mot these	Mots outils	78.5 (0.8)	1.58 (0.36)
into	Fréq. du mot into	Mots outils	77.3 (0.7)	1.6 (0.3)
was	Fréq. du mot was	Mots outils	78.0 (0.80)	1.78 (0.52)
i	Fréq. du mot i	Mots outils	81.0 (1.2)	1.48 (0.35)
those	Fréq. du mot those	Mots outils	79.7 (1.1)	1.52 (0.34)
should	Fréq. du mot should	Mots outils	80.3 (1.1)	1.57 (0.37)
ours	Fréq. du mot ours	Mots outils	77.2 (1.3)	1.58 (0.35)
below	Fréq. du mot below	Mots outils	77.8 (1.2)	1.65 (0.48)
didn	Fréq. du mot didn	Mots outils	81.6 (0.7)	1.46 (0.28)
above	Fréq. du mot above	Mots outils	73.3 (1.2)	1.78 (0.48)
being	Fréq. du mot being	Mots outils	80.3 (0.7)	1.5 (0.28)
theirs	Fréq. du mot theirs	Mots outils	80.3 (0.7)	1.55 (0.28)
ourselves	Fréq. du mot ourselves	Mots outils	80.7 (1.1)	1.56 (0.37)

be	Fréq. du mot be	Mots outils	78.8 (0.9)	1.63 (0.35)
a	Fréq. du mot a	Mots outils	75.8 (1.2)	1.64 (0.43)
ain	Fréq. du mot ain	Mots outils	77.8 (0.9)	1.56 (0.29)
between	Fréq. du mot between	Mots outils	77.0 (1.2)	1.67 (0.35)
all	Fréq. du mot all	Mots outils	79.5 (0.9)	1.57 (0.34)
during	Fréq. du mot during	Mots outils	75.3 (1.1)	1.69 (0.39)
how	Fréq. du mot how	Mots outils	80.0 (0.8)	1.57 (0.29)
too	Fréq. du mot too	Mots outils	78.5 (0.9)	1.58 (0.31)
nor	Fréq. du mot nor	Mots outils	80.3 (1.2)	1.52 (0.41)
again	Fréq. du mot again	Mots outils	80.2 (0.8)	1.52 (0.32)
to	Fréq. du mot to	Mots outils	79.5 (0.8)	1.6 (0.32)
no	Fréq. du mot no	Mots outils	76.5 (0.9)	1.71 (0.42)
hasn	Fréq. du mot hasn	Mots outils	75.3 (0.9)	1.79 (0.34)
haven	Fréq. du mot haven	Mots outils	80.2 (0.8)	1.59 (0.36)
because	Fréq. du mot because	Mots outils	79.0 (0.8)	1.57 (0.35)
weren	Fréq. du mot weren	Mots outils	78.5 (1.0)	1.47 (0.27)
further	Fréq. du mot further	Mots outils	75.3 (1.0)	1.7 (0.38)
shan	Fréq. du mot shan	Mots outils	75.8 (0.9)	1.71 (0.36)
have	Fréq. du mot have	Mots outils	75.7 (1.1)	1.73 (0.35)
of	Fréq. du mot of	Mots outils	79.0 (1.2)	1.56 (0.35)
on	Fréq. du mot on	Mots outils	75.5 (1.0)	1.66 (0.35)
ll	Fréq. du mot ll	Mots outils	78.8 (0.8)	1.65 (0.43)
wouldn	Fréq. du mot wouldn	Mots outils	79.3 (0.9)	1.55 (0.35)
other	Fréq. du mot other	Mots outils	80.2 (0.8)	1.53 (0.26)
any	Fréq. du mot any	Mots outils	76.7 (1.1)	1.67 (0.43)
its	Fréq. du mot its	Mots outils	75.3 (1.0)	1.61 (0.37)
very	Fréq. du mot very	Mots outils	72.0 (0.9)	1.78 (0.35)
once	Fréq. du mot once	Mots outils	78.8 (1.1)	1.61 (0.43)
where	Fréq. du mot where	Mots outils	76.2 0.(0)	1.65 (0.45)
just	Fréq. du mot just	Mots outils	80.3 (0.9)	1.6 (0.44)
has	Fréq. du mot has	Mots outils	73.2 (0.9)	1.78 (0.32)
itself	Fréq. du mot itself	Mots outils	75.0 (1.2)	1.69 (0.43)
!	Fréq. du signe !	Ponctuation	81.3 (0.8)	1.51 (0.25)
"	Fréq. du signe "	Ponctuation	78.8 (0.8)	1.61 (0.31)
£	Fréq. du signe £	Ponctuation	74.8 (0.9)	1.67 (0.33)
€	Fréq. du signe €	Ponctuation	75.0 (1.0)	1.64 (0.38)
#	Fréq. du signe #	Ponctuation	79.5 (1.0)	1.5 (0.29)
$	Fréq. du signe $	Ponctuation	77.2 (0.9)	1.64 (0.34)
%	Fréq. du signe %	Ponctuation	77.5 (0.9)	1.58 (0.36)
&	Fréq. du signe &	Ponctuation	78.5 (1.0)	1.67 (0.42)
’	Fréq. du signe ’	Ponctuation	79.8 (0.9)	1.55 (0.36)
’	Fréq. du signe ’	Ponctuation	78.5 (0.8)	1.6 (0.36)
(	Fréq. du signe (	Ponctuation	76.2 (0.8)	1.75 (0.43)
)	Fréq. du signe)	Ponctuation	81.0 (0.8)	1.62 (0.33)
*	Fréq. du signe *	Ponctuation	78.8 (1.1)	1.52 (0.34)
+	Fréq. du signe +	Ponctuation	79.3 (0.9)	1.63 (0.3)
,	Fréq. du signe ,	Ponctuation	79.3 (0.8)	1.54 (0.34)
-	Fréq. du signe -	Ponctuation	80.2 (0.9)	1.63 (0.32)
.	Fréq. du signe .	Ponctuation	81.0 (0.9)	1.52 (0.32)
/	Fréq. du signe /	Ponctuation	76.7 (1.1)	1.66 (0.41)

:	Fréq. du signe :	Ponctuation	75.2 (0.9)	1.63 (0.36)
;	Fréq. du signe ;	Ponctuation	77.5 (0.8)	1.66 (0.34)
<	Fréq. du signe <	Ponctuation	79.0 (0.9)	1.59 (0.3)
=	Fréq. du signe =	Ponctuation	80.2 (1.0)	1.65 (0.41)
>	Fréq. du signe >	Ponctuation	77.8 (1.2)	1.63 (0.37)
?	Fréq. du signe ?	Ponctuation	74.7 (0.8)	1.65 (0.32)
@	Fréq. du signe @	Ponctuation	80.7 (0.9)	1.53 (0.36)
[	Fréq. du signe [	Ponctuation	78.2 (0.9)	1.56 (0.37)
	Fréq. du signe	Ponctuation	77.8 (0.9)	1.64 (0.33)
]	Fréq. du signe]	Ponctuation	77.8 (1.1)	1.58 (0.38)
^	Fréq. du signe ^	Ponctuation	75.3 (0.9)	1.67 (0.3)
_	Fréq. du signe _	Ponctuation	82.5 (0.8)	1.49 (0.26)
‘	Fréq. du signe ‘	Ponctuation	77.3 (1.0)	1.69 (0.4)
{	Fréq. du signe {	Ponctuation	77.7 (0.9)	1.63 (0.32)
|	Fréq. du signe |	Ponctuation	79.7 (0.8)	1.54 (0.25)
}	Fréq. du signe }	Ponctuation	80.7 (1.0)	1.59 (0.36)
	Fréq. du signe	Ponctuation	78.0 (0.8)	1.6 (0.33)
Saut de ligne	Fréq. de saut de ligne	Ponctuation	77.0 (1.1)	1.61 (0.34)
yules_K	Diversité de Yule's K	Indices	78.2 (1.0)	1.58 (0.47)
shannon_entr	Entropie de Shannon	Indices	80.5 (0.8)	1.53 (0.33)
simpsons_ind	Indice de Simpsons	Indices	78.2 (0.9)	1.58 (0.32)
flesh_ease	Lisibilité de Flesh	Indices	76.3 (0.8)	1.61 (0.28)
flesh_cincade	Lisibilité de Flesch-Kincaid	Indices	82.8 (0.80)	1.53 (0.33)
dale_call	Compréhension de Dale-Chall	Indices	77.7 (0.8)	1.6 (0.28)
gunnin_fox	Difficulté de Gunning-Fox	Indices	77.8 (1.0)	1.59 (0.33)
-LRB-	Fréq. du TAG -LRB-	TAG	81.5 (0.7)	1.5 (0.33)
-RRB-	Fréq. du TAG -RRB-	TAG	78.8 (1.0)	1.63 (0.33)
ADD	Fréq. du TAG ADD	TAG	77.2 (1.0)	1.58 (0.39)
AFX	Fréq. du TAG AFX	TAG	79.5 (1.1)	1.59 (0.32)
CC	Fréq. du TAG CC	TAG	76.8 (1.2)	1.61 (0.39)
CD	Fréq. du TAG CD	TAG	77.8 (0.8)	1.6 (0.37)
DT	Fréq. du TAG DT	TAG	79.7 (1.2)	1.59 (0.54)
EX	Fréq. du TAG EX	TAG	78.2 (0.9)	1.63 (0.39)
FW	Fréq. du TAG FW	TAG	77.2 (1.1)	1.68 (0.42)
HYPH	Fréq. du TAG HYPH	TAG	79.0 (0.9)	1.65 (0.31)
IN	Fréq. du TAG IN	TAG	78.3 (0.9)	1.64 (0.36)
JJ	Fréq. du TAG JJ	TAG	79.0 (1.0)	1.56 (0.31)
JJR	Fréq. du TAG JJR	TAG	81.0 (0.9)	1.57 (0.33)
JJS	Fréq. du TAG JJS	TAG	77.2 (10.6)	1.53 (0.36)
LS	Fréq. du TAG LS	TAG	80.0 (1.1)	1.5 (0.29)
MD	Fréq. du TAG MD	TAG	77.3 (1.1)	1.66 (0.4)
NFP	Fréq. du TAG NFP	TAG	77.8 (0.7)	1.64 (0.24)
NN	Fréq. du TAG NN	TAG	80.7 (0.9)	1.54 (0.28)
NNP	Fréq. du TAG NNP	TAG	76.0 (0.9)	1.67 (0.31)
NNPS	Fréq. du TAG NNPS	TAG	79.5 (0.8)	1.64 (0.36)
NNS	Fréq. du TAG NNS	TAG	79.2 (0.8)	1.51 (0.33)
PDT	Fréq. du TAG PDT	TAG	78.8 (0.9)	1.55 (0.29)
POS	Fréq. du TAG POS	TAG	80.5 (0.8)	1.52 (0.32)
PRP	Fréq. du TAG PRP	TAG	76.7 (0.9)	1.62 (0.35)

PRP$	Fréq. du TAG PRP$	TAG	80.7 (0.8)	1.51 (0.31)
RB	Fréq. du TAG RB	TAG	76.0 (1.1)	1.72 (0.38)
RBR	Fréq. du TAG RBR	TAG	76.8 (0.8)	1.62 (0.3)
RBS	Fréq. du TAG RBS	TAG	79.3 (0.9)	1.62 (0.3)
RP	Fréq. du TAG RP	TAG	76.7 (0.8)	1.62 (0.33)
SYM	Fréq. du TAG SYM	TAG	78.3 (1.2)	1.58 (0.35)
TO	Fréq. du TAG TO	TAG	79.3 (0.7)	1.59 (0.33)
UH	Fréq. du TAG UH	TAG	78.5 (0.8)	1.58 (0.34)
VB	Fréq. du TAG VB	TAG	77.7 (1.2)	1.62 (0.39)
VBD	Fréq. du TAG VBD	TAG	77.3 (0.8)	1.78 (0.35)
VBG	Fréq. du TAG VBG	TAG	77.7 (1.1)	1.57 (0.34)
VBN	Fréq. du TAG VBN	TAG	77.2 (1.1)	1.61 (0.26)
VBP	Fréq. du TAG VBP	TAG	76.3 (1.1)	1.69 (0.34)
VBZ	Fréq. du TAG VBZ	TAG	79.5 (0.8)	1.6 (0.31)
WDT	Fréq. du TAG WDT	TAG	76.3 (1.2)	1.72 (0.45)
WP	Fréq. du TAG WP	TAG	76.5 (0.8)	1.64 (0.3)
WP$	Fréq. du TAG WP$	TAG	77.7 (1.2)	1.7 (0.42)
WRB	Fréq. du TAG WRB	TAG	76.8 (0.8)	1.62 (0.23)
XX	Fréq. du TAG XX	TAG	75.7 (0.7)	1.65 (0.3)
CARDINAL	Fréq. de l'entité nommée CARDINAL	NER	77.3 (1.0)	1.67 (0.44)
DATE	Fréq. de l'entité DATE	NER	77.8 (0.9)	1.62 (0.34)
EVENT	Fréq. de l'entité EVENT	NER	80.2 (1.1)	1.58 (0.42)
FAC	Fréq. de l'entité FAC	NER	75.7 (1.4)	1.7 (0.43)
GPE	Fréq. de l'entité GPE	NER	77.2 (0.8)	1.6 (0.3)
LANGUAGE	Fréq. de l'entité LANGUAGE	NER	79.8 (0.9)	1.6 (0.34)
LAW	Fréq. de l'entité LAW	NER	78.8 (0.7)	1.52 (0.34)
LOC	Fréq. de l'entité LOC	NER	78.8 (0.9)	1.47 (0.18)
MONEY	Fréq. de l'entité MONEY	NER	79.7 (0.9)	1.63 (0.41)
NORP	Fréq. de l'entité NORP	NER	76.8 (0.8)	1.58 (0.32)
ORDINAL	Fréq. de l'entité ORDINAL	NER	79.0 (0.8)	1.63 (0.27)
ORG	Fréq. de l'entité ORG	NER	80.0 (0.9)	1.56 (0.32)
PERCENT	Fréq. de l'entité PERCENT	NER	78.2 (1.0)	1.63 (0.34)
PERSON	Fréq. de l'entité PERSON	NER	77.5 (0.9)	1.63 (0.32)
PRODUCT	Fréq. de l'entité PRODUCT	NER	78.0 (1.1)	1.62 (0.37)
QUANTITY	Fréq. de l'entité QUANTITY	NER	79.5 (1.2)	1.58 (0.41)
TIME	Fréq. de l'entité TIME	NER	77.2 (0.9)	1.58 (0.24)
WORK_OF_ART	Fréq. de l'entité WORK_OF_ART	NER	77.0 (0.9)	1.6 (0.37)

TABLE A.1: Tableau détaillant l'ensemble des marqueurs du style introduit en section 3.4.2.

Bibliographie

[Men87] T. C. Mendenhall. "The Characteristic Curves of Composition". In : *Science* 9.214 (1887), p. 237-249. ISSN : 00368075, 10959203. URL : http://www.jstor.org/stable/1764604 (visité le 20/07/2023).

[Har54] Zellig S. Harris. "Distributional Structure". In : WORD 10.2-3 (1954), p. 146-162. DOI : 10.1080/00437956.1954.11659520. URL : https://doi.org/10.1080/00437956.1954.11659520.

[Fir57] J. R. Firth. "A synopsis of linguistic theory 1930-55." In : 1952-59 (1957), p. 1-32.

[Wei66] Joseph Weizenbaum. "ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine". In : *Commun. ACM* 9.1 (jan. 1966), p. 36-45. ISSN : 0001-0782. DOI : 10.1145/365153.365168. URL : https://doi.org/10.1145/365153.365168.

[Ioa91] Steven E. Shreve Ioannis Karatzas. *Brownian Motion and Stochastic Calculus*. Springer New York, NY, 1991.

[Hol94] David I. Holmes. "Authorship attribution". In : *Computers and the Humanities* 28.2 (avr. 1994), p. 87-106. ISSN : 1572-8412. DOI : 10.1007/BF01830689. URL : https://doi.org/10.1007/BF01830689.

[HS97] Sepp Hochreiter et Jürgen Schmidhuber. "Long Short-term Memory". In : *Neural computation* 9 (déc. 1997), p. 1735-80. DOI : 10.1162/neco.1997.9.8.1735.

[Lec+98] Y. Lecun et al. "Gradient-based learning applied to document recognition". In : *Proceedings of the IEEE* 86.11 (1998), p. 2278-2324. DOI : 10.1109/5.726791.

[TPB99] Naftali Tishby, Fernando C Pereira et William Bialek. "The Information Bottleneck Method". In : *The 37th annual Allerton Conference on Communication, Control, and Computing* (1999), p. 368-377.

[BDV00] Yoshua Bengio, Réjean Ducharme et Pascal Vincent. "A Neural Probabilistic Language Model". In : *Advances in Neural Information Processing Systems*. Sous la dir. de T. Leen, T. Dietterich et V. Tresp. T. 13. MIT Press, 2000. URL : https://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf.

[Glo+04] Amir Globerson et al. "Euclidean Embedding of Co-Occurrence Data". In : *Advances in Neural Information Processing Systems*. Sous la dir. de L. Saul, Y. Weiss et L. Bottou. T. 17. MIT Press, 2004. URL : https://proceedings.neurips.cc/paper_files/paper/2004/file/ec1f850d934f440cfa8e4a18d2cf5463-Paper.pdf.

[Kar04] Jussi KARLGREN. "The Wheres and Whyfores for Studying Textual Genre Computationally". In : *AAAI Technical Report (7)* (2004), p. 68-70.

[Ros+04] Michal ROSEN-ZVI et al. "The Author-Topic Model for Authors and Documents". In : *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. UAI '04. Banff, Canada : AUAI Press, 2004, p. 487-494. ISBN : 0974903906.

[ZZ05] Ying ZHAO et Justin ZOBEL. "Effective and Scalable Authorship Attribution Using Function Words". In : *Information Retrieval Technology*. Sous la dir. de Gary Geunbae LEE et al. Berlin, Heidelberg : Springer Berlin Heidelberg, 2005, p. 174-189. ISBN : 978-3-540-32001-2.

[Fra+06] Georgia FRANTZESKOU et al. "Source Code Author Identification Based on N-gram Author Profiles". In : *Artificial Intelligence Applications and Innovations*. Boston, MA : Springer US, 2006, p. 508-515. ISBN : 978-0-387-34224-5.

[Sch+06] Jonathan SCHLER et al. "Effects of Age and Gender on Blogging." In : *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*. AAAI, 2006, p. 199-205. URL : http://dblp.uni-trier.de/db/conf/aaaiss/aaaiss2006-3.html#SchlerKAP06.

[SSG07a] Purnamrita SARKAR, Sajid M. SIDDIQI et Geoffrey J. GORDON. "Approximate Kalman Filters for Embedding Author-Word Co-occurrence Data over Time". In : *Statistical Network Analysis: Models, Issues, and New Directions*. Sous la dir. d'Edoardo AIROLDI et al. Berlin, Heidelberg : Springer Berlin Heidelberg, 2007, p. 126-139. ISBN : 978-3-540-73133-7.

[SSG07b] Purnamrita SARKAR, Sajid M. SIDDIQI et Geogrey J. GORDON. "A Latent Space Approach to Dynamic Embedding of Co-occurrence Data". In : *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*. Sous la dir. de Marina MEILA et Xiaotong SHEN. T. 2. Proceedings of Machine Learning Research. San Juan, Puerto Rico : PMLR, mars 2007, p. 420-427. URL : https://proceedings.mlr.press/v2/sarkar07a.html.

[Sta09] Efstathios STAMATATOS. "A survey of modern authorship attribution methods". In : *Journal of the American Society for Information Science and Technology* 60.3 (2009), p. 538-556. DOI : https://doi.org/10.1002/asi.21001. eprint : https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.21001. URL : https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21001.

[EK13] Sara EL manarelbouanani et Ismail KASSOU. "Authorship Analysis Studies: A Survey". In : *International Journal of Computer Applications* 86 (déc. 2013). DOI : 10.5120/15038-3384.

[Mik+13a] Tomas MIKOLOV et al. "Distributed Representations of Words and Phrases and their Compositionality". In : *Advances in Neural Information Processing Systems*. Sous la dir. de C.J. BURGES et al. T. 26. Curran Associates, Inc., 2013. URL : https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

[Mik+13b] Tomas MIKOLOV et al. "Efficient Estimation of Word Representations in Vector Space". In : *CoRR* abs/1301.3781 (2013). URL : http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho et Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In : *arXiv preprint arXiv:1409.0473* (2014).

[Cho+14] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar : Association for Computational Linguistics, oct. 2014, p. 1724-1734. DOI : 10.3115/v1/D14-1179. URL : https://aclanthology.org/D14-1179.

[DL14] Sascha Diwersy et Dominique Legallois. "L'apport de la méthode des motifs aux analyses phraséologiques en discours". In : *Approches théoriques et empiriques en phraséologie*. Nancy, France, 2014. URL : https://hal.science/hal-03546271.

[KW14] Diederik P Kingma et Max Welling. "Auto-encoding variational bayes". In : *Proceedings of the International Conference on Learning Representations (ICLR)* (2014).

[KY14] Winter Koppel Moshe et Yaron. "Determining if two documents are written by the same author". In : *Journal of the Association for Information Science and Technology* 65.1 (2014), p. 178-187. DOI : https://doi.org/10.1002/asi.22954. URL : https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.22954.

[LM14] Quoc Le et Tomas Mikolov. "Distributed Representations of Sentences and Documents". In : *Proceedings of the 31st International Conference on Machine Learning*. Sous la dir. d'Eric P. Xing et Tony Jebara. T. 32. Proceedings of Machine Learning Research 2. Bejing, China : PMLR, juin 2014, p. 1188-1196. URL : https://proceedings.mlr.press/v32/le14.html.

[PSM14] Jeffrey Pennington, Richard Socher et Christopher Manning. "GloVe: Global Vectors for Word Representation". In : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar : Association for Computational Linguistics, oct. 2014, p. 1532-1543. DOI : 10.3115/v1/D14-1162. URL : https://aclanthology.org/D14-1162.

[SZB14] Yanir Seroussi, Ingrid Zukerman et Fabian Bohnert. "Authorship Attribution with Topic Models". In : *Computational Linguistics* 40.2 (juin 2014), p. 269-310. ISSN : 0891-2017. DOI : 10.1162/COLI_a_00173. eprint : https://direct.mit.edu/coli/article-pdf/40/2/269/1803926/coli_a_00173.pdf. URL : https://doi.org/10.1162/COLI%5C_a%5C_00173.

[Bow+15] Samuel R. Bowman et al. "A large annotated corpus for learning natural language inference". In : *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal : Association for Computational Linguistics, sept. 2015, p. 632-642. DOI : 10.18653/v1/D15-1075. URL : https://aclanthology.org/D15-1075.

[Iyy+15a] Mohit Iyyer et al. "Deep Unordered Composition Rivals Syntactic Methods for Text Classification". In : *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China : Association for Computational Linguistics, juill. 2015, p. 1681-1691. DOI : 10.3115/v1/P15-1162. URL : https://aclanthology.org/P15-1162.

[Iyy+15b] Mohit IYYER et al. "Deep Unordered Composition Rivals Syntactic Methods for Text Classification". In : *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China : Association for Computational Linguistics, juill. 2015, p. 1681-1691. DOI : 10.3115/v1/P15-1162. URL : https://aclanthology.org/P15-1162.

[Ken+15] Tom KENTER et al. "Ad Hoc Monitoring of Vocabulary Shifts over Time". In : *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. CIKM '15. Melbourne, Australia : Association for Computing Machinery, 2015, p. 1191-1200. ISBN : 9781450337946. DOI : 10.1145/2806416.2806474. URL : https://doi.org/10.1145/2806416.2806474.

[Kir+15] Ryan KIROS et al. "Skip-Thought Vectors". In : *Advances in Neural Information Processing Systems*. Sous la dir. de C. CORTES et al. T. 28. Curran Associates, Inc., 2015. URL : https://proceedings.neurips.cc/paper_files/paper/2015/file/f442d33fa06832082290ad8544a8da27-Paper.pdf.

[ZLP15] Han ZHAO, Zhengdong LU et Pascal POUPART. "Self-adaptive hierarchical sentence model". In : *arXiv preprint arXiv:1504.05070* (2015).

[Ami+16] Silvio AMIR et al. "Modelling Context with User Embeddings for Sarcasm Detection in Social Media". In : *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany : Association for Computational Linguistics, août 2016, p. 167-177. DOI : 10.18653/v1/K16-1017. URL : https://aclanthology.org/K16-1017.

[Din+16] Steven H. H. DING et al. "Learning Stylometric Representations for Authorship Analysis". In : *CoRR* abs/1606.01219 (2016). arXiv : 1606.01219. URL : http://arxiv.org/abs/1606.01219.

[EM16] Steffen EGER et Alexander MEHLER. "On the Linearity of Semantic Change: Investigating Meaning Variation via Dynamic Graph Models". In : *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany : Association for Computational Linguistics, août 2016, p. 52-58. DOI : 10.18653/v1/P16-2009. URL : https://aclanthology.org/P16-2009.

[HLJ16] William L. HAMILTON, Jure LESKOVEC et Dan JURAFSKY. "Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change". In : *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany : Association for Computational Linguistics, août 2016, p. 1489-1501. DOI : 10.18653/v1/P16-1141. URL : https://aclanthology.org/P16-1141.

[Jaw+16] Ganesh JAWAHAR et al. "Author2Vec: Learning Author Representations by Combining Content and Link Information". In : avr. 2016, p. 49-50. DOI : 10.1145/2872518.2889382.

[Rea+16] Andrew J. REAGAN et al. "The emotional arcs of stories are dominated by six basic shapes". In : *EPJ Data Science* 5.1 (nov. 2016), p. 31. ISSN : 2193-1127. DOI : 10.1140/epjds/s13688-016-0093-1. URL : https://doi.org/10.1140/epjds/s13688-016-0093-1.

[RGB16] Sebastian Ruder, Parsa Ghaffari et John G. Breslin. "Character-level and Multi-channel Convolutional Neural Networks for Large-scale Authorship Attribution". In : *CoRR* abs/1609.06686 (2016). arXiv : 1609.06686. URL : http://arxiv.org/abs/1609.06686.

[SPK16] Xing Shi, Inkit Padhi et Kevin Knight. "Does String-Based Neural MT Learn Source Syntax?" In : *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas : Association for Computational Linguistics, nov. 2016, p. 1526-1534. DOI : 10.18653/v1/D16-1159. URL : https://aclanthology.org/D16-1159.

[Ale+17] Alexander A Alemi et al. "Deep variational information bottleneck". In : *Proceedings of the International Conference on Learning Representations (ICLR)* (2017).

[Ami+17] Silvio Amir et al. "Quantifying Mental Health from Social Media with Neural User Embeddings". In : *Proceedings of the 2nd Machine Learning for Healthcare Conference*. Sous la dir. de Finale Doshi-Velez et al. T. 68. Proceedings of Machine Learning Research. PMLR, août 2017, p. 306-321. URL : https://proceedings.mlr.press/v68/amir17a.html.

[BM17] Robert Bamler et Stephan Mandt. "Dynamic Word Embeddings". In : *Proceedings of the 34th International Conference on Machine Learning*. Sous la dir. de Doina Precup et Yee Whye Teh. T. 70. Proceedings of Machine Learning Research. PMLR, août 2017, p. 380-389. URL : https://proceedings.mlr.press/v70/bamler17a.html.

[Con+17] Alexis Conneau et al. "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data". In : *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark : Association for Computational Linguistics, sept. 2017, p. 670-680. DOI : 10.18653/v1/D17-1070. URL : https://aclanthology.org/D17-1070.

[Kam+17] Jaap Kamps et al. "Research and Advanced Technology for Digital Libraries 21st". In : *Proceedings: 21st International Conference on Theory and Practice of Digital Libraries, 2017, Thessaloniki, Greece*. 2017. ISBN : 978-3-319-67007-2. DOI : 10.1007/978-3-319-67008-9.

[Nea+17] Tempestt Neal et al. "Surveying Stylometry Techniques and Applications". In : *ACM Comput. Surv.* 50.6 (nov. 2017). ISSN : 0360-0300. DOI : 10.1145/3132039. URL : https://doi.org/10.1145/3132039.

[NC17] Xing Niu et Marine Carpuat. "Discovering Stylistic Variations in Distributional Vector Space Models via Lexical Paraphrases". In : *Proceedings of the Workshop on Stylistic Variation*. Copenhagen, Denmark : Association for Computational Linguistics, sept. 2017, p. 20-27. DOI : 10.18653/v1/W17-4903. URL : https://aclanthology.org/W17-4903.

[SVS17] Yunita Sari, Andreas Vlachos et Mark Stevenson. "Continuous N-gram Representations for Authorship Attribution". In : *Proceedings of the 15th Conference of the European Chapter of the ACL: Volume 2, Short Papers*. Valencia, Spain : ACL, avr. 2017, p. 267-273. URL : https://aclanthology.org/E17-2043.

[Sta17] Efstathios Stamatatos. "Authorship Attribution Using Text Distortion". In : *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain : Association for Computational Linguistics, avr. 2017, p. 1138-1149. URL : https://aclanthology.org/E17-1107.

[Szw17] Piotr Szwed. "Authorship Attribution for Polish Texts Based on Part of Speech Tagging". In : mai 2017, p. 316-328. ISBN : 978-3-319-58273-3. DOI : 10.1007/978-3-319-58274-0_26.

[Vas+17] Ashish Vaswani et al. "Attention is All you Need". In : *Advances in Neural Information Processing Systems*. Sous la dir. d'I. Guyon et al. T. 30. Curran Associates, Inc., 2017. URL : https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[Aka+18] Reina Akama et al. "Unsupervised Learning of Style-sensitive Word Vectors". In : *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia : Association for Computational Linguistics, juill. 2018, p. 572-578. DOI : 10.18653/v1/P18-2091. URL : https://aclanthology.org/P18-2091.

[Amm+18] Waleed Ammar et al. "Construction of the Literature Graph in Semantic Scholar". In : *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. New Orleans - Louisiana : Association for Computational Linguistics, juin 2018, p. 84-91. DOI : 10.18653/v1/N18-3011. URL : https://aclanthology.org/N18-3011.

[BZM18] Dainis Boumber, Yifan Zhang et Arjun Mukherjee. "Experiments with Convolutional Neural Networks for Multi-Label Authorship Attribution". In : *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan : European Language Resources Association (ELRA), mai 2018. URL : https://aclanthology.org/L18-1409.

[Cer+18] Daniel Cer et al. "Universal Sentence Encoder for English". In : *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium : Association for Computational Linguistics, nov. 2018, p. 169-174. DOI : 10.18653/v1/D18-2029. URL : https://aclanthology.org/D18-2029.

[Con+18] Alexis Conneau et al. "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties". In : *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia : Association for Computational Linguistics, juill. 2018.

[EM18] Hassaan Elahi et Haris Muneer. *Identifying Different Writing Styles in a Document Intrinsically using Stylometric Analysis*. The complete code and detailed documentation is available on the attached Github Link: https://github.com/harismuneer/Writing-Styles- Classification-Using-Stylometric-Analysis. Juill. 2018. DOI : 10.5281/zenodo.2538334. URL : https://doi.org/10.5281/zenodo.2538334.

[GF18] Martin Gerlach et Francesc Font-Clos. "A standardized Project Gutenberg corpus for statistical analysis of natural language and quantitative linguistics". In : *CoRR* abs/1812.08092 (2018). arXiv : 1812.08092. URL : http://arxiv.org/abs/1812.08092.

[Jai+18] Sarthak Jain et al. "Learning Disentangled Representations of Texts with Application to Biomedical Abstracts". In : *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium : Association for Computational Linguistics, nov. 2018, p. 4683-4693. DOI : 10.18653/v1/D18-1497. URL : https://aclanthology.org/D18-1497.

[Jas+18] Johannes Jasper et al. "Authorship Verification on Short Text Samples Using Stylometric Embeddings". In : *Analysis of Images, Social Networks and Texts*. Sous la dir. de Wil M. P. van der Aalst et al. Cham : Springer International Publishing, 2018, p. 64-75.

[LDB18] Yang Li, Nan Du et Samy Bengio. "Time-Dependent Representation for Neural Event Sequence Prediction". In : 2018. URL : https://openreview.net/pdf?id=HyrT5Hkvf.

[OLV18] Aäron van den Oord, Yazhe Li et Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding". In : *CoRR* abs/1807.03748 (2018). arXiv : 1807.03748. URL : http://arxiv.org/abs/1807.03748.

[Pet+18] Matthew E. Peters et al. "Deep Contextualized Word Representations". In : *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana : Association for Computational Linguistics, juin 2018, p. 2227-2237. DOI : 10.18653/v1/N18-1202. URL : https://aclanthology.org/N18-1202.

[Rad+18] Alec Radford et al. "Improving language understanding by generative pre-training". In : (2018).

[RDT18] Swayambhu Nath Ray, Shib Sankar Dasgupta et Partha Talukdar. "AD3: Attentive Deep Document Dater". In : *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium : Association for Computational Linguistics, nov. 2018, p. 1871-1880. DOI : 10.18653/v1/D18-1213. URL : https://aclanthology.org/D18-1213.

[RB18] Maja Rudolph et David Blei. "Dynamic Embeddings for Language Evolution". In : *Proceedings of the 2018 World Wide Web Conference*. WWW '18. Lyon, France : International World Wide Web Conferences Steering Committee, 2018, p. 1003-1011. ISBN : 9781450356398. DOI : 10.1145/3178876.3185999. URL : https://doi.org/10.1145/3178876.3185999.

[SSV18] Yunita Sari, Mark Stevenson et Andreas Vlachos. "Topic or Style ? Exploring the Most Useful Features for Authorship Attribution". In : *27th International conference on computational lingustics* (2018), p. 343-353. URL : https://www.aclweb.org/anthology/C18-1029.

[Sta18] Efstathios STAMATATOS. "Masking topic-related information to enhance authorship attribution". In : *Journal of the Association for Information Science and Technology* 69.3 (2018), p. 461-473. DOI : https://doi.org/10.1002/asi.23968. eprint : https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.23968. URL : https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.23968.

[SW18] Kalaivani SUNDARARAJAN et Damon WOODARD. "What represents "style" in authorship attribution?" In : *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA : Association for Computational Linguistics, août 2018, p. 2814-2822. URL : https://aclanthology.org/C18-1238.

[Vas+18] Shikhar VASHISHTH et al. "Dating Documents using Graph Convolution Networks". In : *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia : Association for Computational Linguistics, juill. 2018, p. 1605-1615. DOI : 10.18653/v1/P18-1149. URL : https://aclanthology.org/P18-1149.

[Yao+18] Zijun YAO et al. "Dynamic Word Embeddings for Evolving Semantic Discovery". In : *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. WSDM '18. Marina Del Rey, CA, USA : Association for Computing Machinery, 2018, p. 673-681. ISBN : 9781450355810. DOI : 10.1145/3159652.3159703. URL : https://doi.org/10.1145/3159652.3159703.

[Zha+18] Richong ZHANG et al. "Syntax Encoding with Application in Authorship Attribution". In : *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium : Association for Computational Linguistics, oct. 2018, p. 2742-2753. DOI : 10.18653/v1/D18-1294. URL : https://aclanthology.org/D18-1294.

[BN19] Dorothea Kolossa BENEDIKT BOENNINGHOFF Steffen Hessler et Robert M. NICKEL. "Explainable Authorship Verification in Social Media via Attention-based Similarity Learning". In : *IEEE International Conference on Big Data (IEEE Big Data 2019), Los Angeles, CA, USA, December 9-12, 2019*. 2019.

[Boe+19] Benedikt BOENNINGHOFF et al. "Similarity Learning for Authorship Verification in Social Media". In : *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, p. 2457-2461. DOI : 10.1109/ICASSP.2019.8683405.

[Cla+19a] Kevin CLARK et al. "What Does BERT Look at? An Analysis of BERT's Attention". In : *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy : Association for Computational Linguistics, août 2019, p. 276-286. DOI : 10.18653/v1/W19-4828. URL : https://aclanthology.org/W19-4828.

[Cla+19b] Kevin CLARK et al. "What Does BERT Look at? An Analysis of BERT's Attention". In : *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy : Association for Computational Linguistics, août 2019, p. 276-286. DOI : 10.18653/v1/W19-4828. URL : https://aclanthology.org/W19-4828.

[DLD19] Edouard Delasalles, Sylvain Lamprier et Ludovic Denoyer. "Learning Dynamic Author Representations with Temporal Language Models". In : *CoRR* abs/1909.04985 (2019). arXiv : 1909.04985. URL : http://arxiv.org/abs/1909.04985.

[Dev+19] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In : *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota : Association for Computational Linguistics, juin 2019, p. 4171-4186. DOI : 10.18653/v1/N19-1423. URL : https://aclanthology.org/N19-1423.

[GSR19] Shriya TP Gupta, Jajati Keshari Sahoo et Rajendra Kumar Roul. "Authorship Identification Using Recurrent Neural Networks". In : *Proceedings of the 2019 3rd International Conference on Information System and Data Mining*. ICISDM 2019. Houston, TX, USA : Association for Computing Machinery, 2019, p. 133-137. ISBN : 9781450366359. DOI : 10.1145/3325917.3325935. URL : https://doi.org/10.1145/3325917.3325935.

[HP19] Xiaolei Huang et Michael J. Paul. "Neural Temporality Adaptation for Document Classification: Diachronic Word Embeddings and Domain Adaptation Models". In : *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy : Association for Computational Linguistics, juill. 2019, p. 4113-4123. DOI : 10.18653/v1/P19-1403. URL : https://aclanthology.org/P19-1403.

[JH19] Fereshteh Jafariakinabad et Kien A. Hua. "Style-Aware Neural Model with Application in Authorship Attribution". In : *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 2019, p. 325-328. DOI : 10.1109/ICMLA.2019.00061.

[Lam+19] Guillaume Lample et al. "Multiple-Attribute Text Rewriting". In : *International Conference on Learning Representations*. 2019. URL : https://openreview.net/forum?id=H1g2NhC5KQ.

[Mah+19] Suraj Maharjan et al. "Jointly Learning Author and Annotated Character N-gram Embeddings: A Case Study in Literary Text". In : *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. Varna, Bulgaria : INCOMA Ltd., sept. 2019, p. 684-692. DOI : 10.26615/978-954-452-056-4_080. URL : https://aclanthology.org/R19-1080.

[Oh+19] Seong Joon Oh et al. "Modeling uncertainty with hedged instance embedding". In : *Proceedings of the International Conference on Learning Representations*. 2019.

[Rad+19] Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In : 2019.

[Raf+19] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In : *CoRR* abs/1910.10683 (2019). arXiv : 1910.10683. URL : http://arxiv.org/abs/1910.10683.

[RG19] Nils Reimers et Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In : *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China : Association for Computational Linguistics, nov. 2019, p. 3982-3992. DOI : 10.18653/v1/D19-1410. URL : https://aclanthology.org/D19-1410.

[SZL19] Wei Song, Chen Zhao et Lizhen Liu. "Multi-Task Learning for Authorship Attribution via Topic Approximation and Competitive Attention". In : *IEEE Access* 7 (2019), p. 177114-177121. DOI : 10.1109/ACCESS.2019.2957152.

[BPC20] Iz Beltagy, Matthew E. Peters et Arman Cohan. "Longformer: The Long-Document Transformer". In : *CoRR* abs/2004.05150 (2020). arXiv : 2004.05150. URL : https://arxiv.org/abs/2004.05150.

[Bha+20] Sarthak Bhagat et al. "Disentangling Multiple Features in Video Sequences using Gaussian Processes in Variational Autoencoders." In : *European Conference on Computer Vision (ECCV)*. 2020.

[Bro+20] Tom Brown et al. "Language Models are Few-Shot Learners". In : *Advances in Neural Information Processing Systems*. Sous la dir. de H. Larochelle et al. T. 33. Curran Associates, Inc., 2020, p. 1877-1901. URL : https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[Fab+20] Maël Fabien et al. "BertAA : BERT fine-tuning for Authorship Attribution". In : *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*. NLP Association of India (NLPAI), déc. 2020, p. 127-137. URL : https://aclanthology.org/2020.icon-main.16.

[Hay+20] Julien Hay et al. "Representation learning of writing style". In : *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*. Online : Association for Computational Linguistics, nov. 2020, p. 232-243. DOI : 10.18653/v1/2020.wnut-1.30. URL : https://aclanthology.org/2020.wnut-1.30.

[Jaf+20] Somayeh Jafaritazehjani et al. "Style versus Content: A distinction without a (learnable) difference?" In : *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online) : International Committee on Computational Linguistics, déc. 2020, p. 2169-2180. DOI : 10.18653/v1/2020.coling-main.197. URL : https://aclanthology.org/2020.coling-main.197.

[Lan+20] Zhenzhong Lan et al. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In : *International Conference on Learning Representations*. 2020. URL : https://openreview.net/forum?id=H1eA7AEtvS.

[Le+20] Hang Le et al. "FlauBERT: Unsupervised Language Model Pre-training for French". In : *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France : European Language Resources Association, mai 2020, p. 2479-2490. ISBN : 979-10-95546-34-4. URL : https://aclanthology.org/2020.lrec-1.302.

[Lia+20] Keng-Te Liao et al. "Explaining Word Embeddings via Disentangled Representation". In : *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China : Association for Computational Linguistics, déc. 2020, p. 720-725. URL : https://aclanthology.org/2020.aacl-main.72.

[Liu+20a] Tao Liu et al. “Sentence matching with syntax-and semantics-aware bert”. In : *Proceedings of the 28th International Conference on Computational Linguistics*. 2020, p. 3302-3312.

[Liu+20b] Yinhan Liu et al. *Ro{BERT}a: A Robustly Optimized {BERT} Pretraining Approach*. 2020. URL : https://openreview.net/forum?id=SyxS0T4tvS.

[Mar+20] Louis Martin et al. “CamemBERT: a Tasty French Language Model”. In : *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online : Association for Computational Linguistics, juill. 2020, p. 7203-7219. DOI : 10.18653/v1/2020.acl-main.645. URL : https://aclanthology.org/2020.acl-main.645.

[Wu+20] Xiaodong Wu et al. “Author2Vec: A Framework for Generating User Embedding”. In : *CoRR* abs/2003.11627 (2020). arXiv : 2003.11627. URL : https://arxiv.org/abs/2003.11627.

[Zah+20] Manzil Zaheer et al. “Big Bird: Transformers for Longer Sequences”. In : *Advances in Neural Information Processing Systems*. Sous la dir. de H. Larochelle et al. T. 33. Curran Associates, Inc., 2020, p. 17283-17297. URL : https://proceedings.neurips.cc/paper_files/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf.

[Zha+20] Mengjie Zhao et al. “Masking as an Efficient Alternative to Finetuning for Pretrained Language Models”. In : *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online : Association for Computational Linguistics, nov. 2020, p. 2226-2241. DOI : 10.18653/v1/2020.emnlp-main.174. URL : https://aclanthology.org/2020.emnlp-main.174.

[Amb+21] Spurthi Amba Hombaiah et al. “Dynamic Language Models for Continuously Evolving Content”. In : *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. KDD ’21. Virtual Event, Singapore : Association for Computing Machinery, 2021, p. 2514-2524. ISBN : 9781450383325. DOI : 10.1145/3447548.3467162. URL : https://doi.org/10.1145/3447548.3467162.

[Bai+21] Jiangang Bai et al. “Syntax-BERT: Improving pre-trained transformers with syntax trees”. In : *arXiv preprint arXiv:2103.04350* (2021).

[Bom+21] Rishi Bommasani et al. “On the Opportunities and Risks of Foundation Models”. In : *CoRR* abs/2108.07258 (2021). arXiv : 2108.07258. URL : https://arxiv.org/abs/2108.07258.

[Gou21] Antoine Gourru. “Apprentissage de représentations d’auteurs et de documents : approches probabilistes à partir de représentations pré-entraînées.” Theses. Université de Lyon, nov. 2021. URL : https://theses.hal.science/tel-03783746.

[KH21] Dongyeop Kang et Eduard Hovy. “Style is NOT a single variable: Case Studies for Cross-Stylistic Language Understanding”. In : jan. 2021, p. 2376-2387. DOI : 10.18653/v1/2021.acl-long.185.

[Laz+21] Angeliki Lazaridou et al. “Mind the Gap: Assessing Temporal Generalization in Neural Language Models”. In : *Advances in Neural Information Processing Systems*. Sous la dir. d’A. Beygelzimer et al. 2021. URL : https://openreview.net/forum?id=730mmrCfSyy.

[MBH21] Rabeeh Karimi Mahabadi, Yonatan Belinkov et James Henderson. "Variational Information Bottleneck for Effective Low-Resource Fine-Tuning". In : *International Conference on Learning Representations*. 2021.

[Riv+21] Rafael A. Rivera-Soto et al. "Learning Universal Authorship Representations". In : *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online et Punta Cana, Dominican Republic : Association for Computational Linguistics, nov. 2021, p. 913-919. DOI : 10.18653/v1/2021.emnlp-main.70. URL : https://aclanthology.org/2021.emnlp-main.70.

[SJN21] Pavel Savov, Adam Jatowt et Radoslaw Nielek. "Predicting the Age of Scientific Papers". In : Krakow, Poland : Springer-Verlag, 2021, p. 728-735. ISBN : 978-3-030-77960-3. DOI : 10.1007/978-3-030-77961-0_58. URL : https://doi.org/10.1007/978-3-030-77961-0_58.

[TGV21] Enzo Terreau, Antoine Gourru et Julien Velcin. "Writing Style Author Embedding Evaluation". In : *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*. Punta Cana, Dominican Republic : Association for Computational Linguistics, nov. 2021, p. 84-93. DOI : 10.18653/v1/2021.eval4nlp-1.9. URL : https://aclanthology.org/2021.eval4nlp-1.9.

[TG21] Martina Toshevska et Sonja Gievska. "A Review of Text Style Transfer using Deep Learning". In : *CoRR* abs/2109.15144 (2021). arXiv : 2109.15144. URL : https://arxiv.org/abs/2109.15144.

[WN21] Anna Wegmann et Dong Nguyen. "Does It Capture STEL? A Modular, Similarity-based Linguistic Style Evaluation Framework". In : *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online et Punta Cana, Dominican Republic : Association for Computational Linguistics, nov. 2021, p. 7109-7130. DOI : 10.18653/v1/2021.emnlp-main.569. URL : https://aclanthology.org/2021.emnlp-main.569.

[ZJ21] Jian Zhu et David Jurgens. "Idiosyncratic but not Arbitrary: Learning Idiolects in Online Registers Reveals Distinctive yet Consistent Individual Styles". In : *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online et Punta Cana, Dominican Republic : Association for Computational Linguistics, nov. 2021, p. 279-297. DOI : 10.18653/v1/2021.emnlp-main.25. URL : https://aclanthology.org/2021.emnlp-main.25.

[Gou+22a] Antoine Gourru et al. "Dynamic Gaussian Embedding of Authors". In : WWW '22. Virtual Event, Lyon, France : Association for Computing Machinery, 2022, p. 2109-2119. ISBN : 9781450390965. DOI : 10.1145/3485447.3512084. URL : https://doi.org/10.1145/3485447.3512084.

[Gou+22b] Antoine Gourru et al. "Dynamic Gaussian Embedding of Authors". In : *Proceedings of the 2022 The Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2022.

[RGR22] Guy D. Rosin, Ido Guy et Kira Radinsky. "Time Masking for Temporal Language Models". In : WSDM '22. Virtual Event, AZ, USA : Association for Computing Machinery, 2022, p. 833-841. ISBN : 9781450391320. DOI : 10.1145/3488560.3498529. URL : https://doi.org/10.1145/3488560.3498529.

[Wan+22] Rose E WANG et al. "Language modeling via stochastic processes". In : *International Conference on Learning Representations*. 2022. URL : https://openreview.net/forum?id=pMQwKL1yctf.

[WSN22] Anna WEGMANN, Marijn SCHRAAGEN et Dong NGUYEN. "Same Author or Just Same Topic? Towards Content-Independent Style Representations". In : *Proceedings of the 7th Workshop on Representation Learning for NLP*. Dublin, Ireland : Association for Computational Linguistics, mai 2022, p. 249-268. DOI : 10.18653/v1/2022.repl4nlp-1.26. URL : https://aclanthology.org/2022.repl4nlp-1.26.

[HBL23] Lucia Bouza HEGUERTE, Aurélie BUGEAU et Loïc LANNELONGUE. In : (sept. 2023). DOI : 10.1088/2515-7620/acf81b. URL : https://doi.org/10.1088%2F2515-7620%2Facf81b.

[Ope23] OPENAI. *GPT-4 Technical Report*. 2023. arXiv : 2303.08774 [cs.CL].

[Som+23] Vignesh Ram SOMNATH et al. "Aligned Diffusion Schrödinger Bridges". In : *The 39th Conference on Uncertainty in Artificial Intelligence*. 2023. URL : https://openreview.net/forum?id=BkWFJN7_bQ.

[Tou+23] Hugo TOUVRON et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv : 2302.13971 [cs.CL].

[Zha+23] Wayne Xin ZHAO et al. *A Survey of Large Language Models*. 2023. arXiv : 2303.18223 [cs.CL].

[Bou+19] Dainis BOUMBER et al. *Robust Authorship Verification with Transfer Learning*. EasyChair Preprint no. 865. EasyChair, 2019. DOI : 10.29007/9nf3.

Table des figures

Liste des tableaux