Capturing Style in Author and Document Representation

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—A wide range of Deep Natural Language Processing (NLP) models integrates continuous and low dimensional representations of words and documents. Surprisingly, very few models study representation learning for authors. These representations can be used for many NLP tasks, such as author identification and classification, or in recommendation systems. A strong limitation of existing works is that they do not explicitly capture writing style, making them hardly applicable to literary data. We therefore propose a new architecture based on Variational Information Bottleneck (VIB) that learns embeddings for both authors and documents with a stylistic constraint. Our model fine-tunes a pre-trained document encoder. We stimulate the detection of writing style by adding predefined stylistic features making the representation axis interpretable with respect to writing style indicators. We evaluate our method on three datasets: a literary corpus extracted from the Gutenberg Project, the Blog Authorship Corpus and IMDb62, for which we show that it matches or outperforms strong/recent baselines in authorship attribution while capturing much more accurately the authors stylistic aspects.**

***Index Terms*—Gaussian Embedding, Document Embedding, Author Embedding, Variational Information Bottleneck, Writing Style**

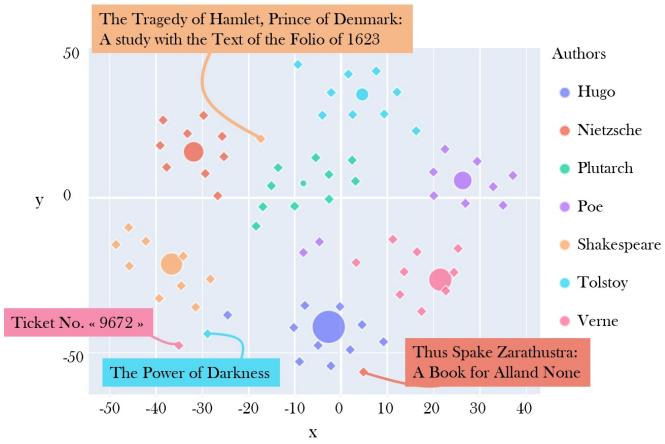

Fig. 1. **Author and book representations from R-PGD.** We here present a 2D projection with T-SNE of VADES documents and authors embeddings on R-PGD. Books are represented with diamond, authors with dot. The bigger the dot, the bigger the author variance learnt. More detailed description in section V

I. Introduction

Deep models for Natural Language Processing are usually based on Transformers, and they rely on latent intermediate representations. These representations are usually built in a self-supervised manner on a language modeling task, such as Masked Language Modeling (MLM) [1] or auto-regressive training [2]. They constitute a good feature space to solve downstream tasks, for example classification or generation, even though some of those tasks are still difficult to handle with prompt-based generative models like ChatGPT [3]. Additionally, some efforts have been made to benefit from large pretrained model to represent documents [4], [5] and even authors, with contributions like Usr2Vec [6], Aut2Vec [7], and DGEA [8]. The main drawback of these models is that they were shown by [9] to mainly focus on topics rather than on stylistic features of the text. It turns out that capturing writing style can be of much interest for some applications.

When working with literacy data or for forensic investigation [10], practitioners are generally interested in detecting similarities in writing style regardless of the topics covered by the authors. The author style can be defined as every writing choice made without semantic information, often study through various linguistic and syntactic features. As demonstrated by [9], most author embedding techniques rely on the semantic content of documents: a poem and a fiction writing on flowers will be placed closer in the latent space, regardless of their strong differences in sentence construction, structure, etc.

As an answer to these limitations, we propose a new model that builds a representation space which captures writing style by using stylistic metrics as additional input features. We follow [11] and leverage the Variational Information Bottleneck (VIB) framework [12], that was shown to outperform the classical pointwise contrastive training. More precisely, we propose to use it to fine tune a pretrained document encoder (such as [4]) and author representations on an authorship attribution task. This is, to our knowledge, the first time that this framework is applied to author representation learning. Then, we add an additional term in the objective function to enforce the representations to capture stylistic features. We name this new model VADES. Using pretrained models allows to benefit from accurate intermediate text representations,

built on ready-to-use language resources. In Figure 1, we present a subset of authors from the Project Gutenberg and the representation of the documents they wrote. The size of author's vector is proportional to its variance, learnt by using the VIB framework. As expected, some outlier productions from authors in term of style (e.g., Thus Spake Zarathustra from Nietzsche) lie closer in the representation space to books of the same genre. More precisely, our model allows 1) to capture author and document style, 2) to build an interpretable representation space to be used by researchers in linguistic, literature and public at large, 3) to predict stylistic features such as readibility index, NER frequencies, more accurately than every existing neural based methods, 4) accurately identify document's author, even when they are unknown.

After a presentation of related works, we introduce the theoretical foundations of the VIB framework, we then describe our model and how it is optimized. In the last section, we present experimental results on two tasks: author identification and stylistic features prediction. Our experiments demonstrate that our model outperforms or matches existing author embedding methods, in addition to being able to infer representations for unseen documents, measure semantic uncertainty of authors and documents, and capture author stylistic information.

II. Related Works

A. Author Embedding Models

Word embedding, popularized by [13], was then extended to document embedding by the same authors. More recent works [4] propose different aggregation functions of word embeddings, based on LSTM, Transformers, and Deep Averaging Networks, to build (short) document level representations. The aggregations is learnt through classification or document pairing. More recently, [5] proceed in a similar way by fine tuning a BERT model [1].

There are also specific works focusing on author embeddings. The Author Topic Model (ATM) [14] is a hierarchical graphical model, optimized through Gibbs sampling. It produces a distribution over jointly learnt topic factors that can be used as author features. Aut2vec [7] allows to learn representations of authors and documents that can separate true observed pairs and negative sampled (document, author) pairs. The distance between two representations modifies an activation function producing a probability that the pair is observed in the corpus. This approach concatenates two sub models: the Link Info model, which takes pairs of collaborating authors, and the Content Info model, which uses pairs of author and documents. It cannot infer representations of unseen documents and authors: the embeddings are parameters of an embedding layer. The Usr2vec model [6] learns author representation from pretrained word vectors. Authors use the same objective than [13], and add an author id to learn the representations.

B. Writing Style-oriented Embedding Models

While there is no consensual definition of writing style, it has always been a widely addressed research topic. In computational linguistic, the approach of [15] is often cited as a reference and gives the following definition: *"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."*, and the author to conclude further to the *"impossibility of drawing a clean line between meaning and style"*. That's why style is commonly defined as every writing choice without semantic information.

Based on this definition, it is hard, if not impossible, to produce a clear annotated dataset classifying different writing style. The workaround in most studies is to identify the most useful stylistic features to associate an author to its production. It starts in the 19th century with [16] and the most basic features (e.g., word and punctuation frequencies, hapax legomena, average sentence length). More recent works focus on function words frequencies [17], hybrid variables such as character n-grams [18], [19] or even Part-Of-Speech (POS) and Name Entity Recognition (NER) tag frequencies, using authorship prediction as evaluation.

Several methods try to use these stylistic features to learn document representations. For example, [20] use Doc2Vec on documents of character trigrams annotated regarding their position in the word or if they contain punctuation (NGRAM Doc2Vec). According to the authors, it allows to capture both content and writing style. In an other work, words and POS tags embeddings are learnt together before passing them through a CNN to get a sentence representation [21]. Then these sentences are fed into an LSTM with a final attention layer to compute document representation. This model is trained on the authorship attribution task.

Some works claim to capture this information in an unsupervised manner. DBert-ft [22] fine-tunes DistilBERT on the authorship attribution task, assuming that an author writing style must be consistent over its documents, and thus, that this task allows to build a "stylometric latent space" when the model is trained on a reference set. Yet, for all above models, no author representation is explicitely learnt.

III. Our model: VADES

A. Goal and VIB Framework

We deal with a set of documents, such as literature or blog posts. We assume each document is written by one author. Each document of indice d is preprocessed to extract a vector z_d^f of $r = 300$ stylistic features following [9].

Our goal is threefold: i) We want to build author and document representations *in the same space* $\mathbb{R}^r$ such that their proximity captures their stylistic similarity (Figure 1), ii) We want to learn a measure of variability in style for each document and author , and iii) We want our model to incorporate an on-the-shelf pre-trained text encoder such as Sentence-BERT or USE to benefit from their complex language understanding, fine-tuned on the dataset at hand using the objective we have just defined. To do that, we build an architecture based on the Variational Information Bottleneck (VIB) framework.

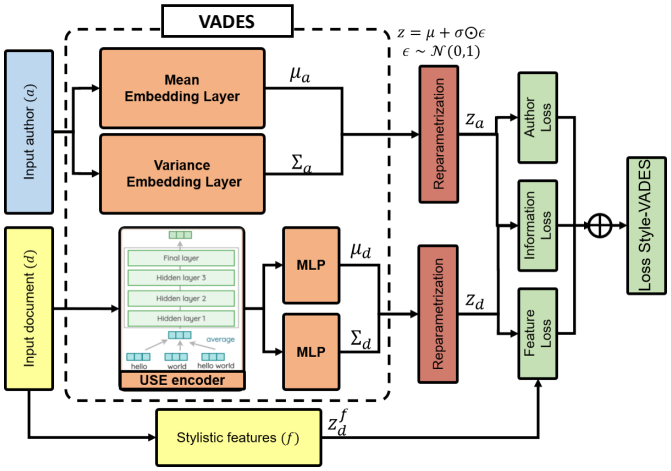

Fig. 2. **VADES in one picture.** We draw a single representation z_d using the reparametrization trick. Authors mean and variance are trainable parameters (embedding layers). L_{VADES} computes the probability of the author/document pair to be observed, plus a regularization term and a stylometric features-based loss, see Eq.5.

The VIB framework is a variational extension of the Information Bottleneck principle [23] proposed by [12]. The general objective function is, for a set of observations x, to associate labels y and latent representations z of these observations:

$$\arg\max_z I(z,y) - \beta I(z,x), \tag{1}$$

where I is the well-known Mutual Information measure, defined as:

$$I(x,y) = \int\int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} d_x d_y. \tag{2}$$

Information Bottleneck aims at maximally compressing the information in z, such that z is highly informative regarding the labels, i.e. z can be used to predict the labels y. With y being associated with a set of relevant stylistic features, we would like to maximize the stylistic information captured by the representation, while minimizing the semantic one. $\beta \geq 0$ is a hyper-parameter that controls the balance between the two sub-objectives.

In this approach, $p(z|x)$ (the "encoding law") is defined by modeling choices. Most of the time, the mutual information is intractable. We then obtain a lower bound of Eq.1 by using variational approximations thanks to [12]:

$$-L_{vib} = \mathbb{E}[\log q(y|z)] - \beta KL(p(z|x)||q(z)) \tag{3}$$

where $q(y|z)$ is a variational approximation of $p(y|z)$ and $q(z)$ approximates $p(z)$. Maximizing Eq.3 leads to increasing Eq.1.

B. VIB for Embedding with Stylistic Constraints

[24] propose to use this framework to learn probabilistic representations of images. They leverage an instance of this framework based on siamese networks with a (soft) contrastive loss objective function, to separate positive observed pairs of images ($y = 1$) and negative examples ($y = 0$). We extend this model to document and author embedding with stylistic constraint. Each author a (resp. document d) is associated to a stochastic representation z_a (resp. z_d) that is unobserved (i.e., latent). Additionally, each document is associated to a stylistic feature vector z_d^f that is beforehand extracted from the corpus with usual NLP toolkits. We assume that the dimensions of z_a, z_d and z_d^f are the same (r).

We build a set of pairs (a, d) with label $y_a = 1$ if a wrote d. We additionally draw k negative pairs (a', d) for each observed pair, associated with label $y_a = 0$, where a' is *not* an author of d. The encoding laws ($p(z|x)$) for authors and documents are normal laws. To capture stylistic information, we also build a set of pairs (d, d) with label $y_f = 1$ and we draw k negative pairs (d, d') for each observed pair, associated with label $y_f = 0$. These pairs are used to train the stylistic objective : the representation z_d of a document should be close to its feature vector z_d^f.

We learn the following parameters for each author a: mean μ_a and diagonal variance matrix with diagonal σ_a^2 (these are embedding layers). For a document d, we use a trainable text encoder to map a document's content to a vector $d_0 \in \mathbb{R}^{r_0}$. We then build the document mean $\mu_d = f(d_0) \in \mathbb{R}^r$ and diagonal variance matrix with diagonal $\sigma_d^2 = g(d_0) \in \mathbb{R}^r$. As we will show later, the dimension r should match the number of stylistic features to gain in comprehension of the learning space, but the text encoder can output vectors of any dimension (here, r_0). Following [12], [24], f and g are neural networks. We give more details on f, g (the "encoding functions"), and the text encoder later.

Following [24], the probability of a label is the soft contrastive loss:

$$\begin{aligned} q(y_a = 1|z_a, z_d) &= \sigma(-c_a||z_a - z_d||_2 + e_a) \\ q(y_f = 1|z_d, z_d^f) &= \sigma(-c_f||z_d - z_d^f||_2 + e_f), \end{aligned} \tag{4}$$

where σ is the sigmoid function, $c_a, c_f > 0$ and $e_a, e_f \in \mathbb{R}$. We introduce an additional parameter $\alpha \in [0, 1]$ to control the importance given to the features and to the authorship prediction objective. We can define the loss function (to minimize) based on the VIB framework as follows:

$$\begin{aligned} \mathcal{L} = &-(1-\alpha)\mathbb{E}_{p(z_a|x_a),p(z_d|x_d)}[\log q(y_a|z_a, z_d)] \\ &- \alpha\mathbb{E}_{p(z_d|x_d)}[\log q(y_f|z_d, z_d^f)] \\ &+ \beta\left(KL(p(z_a|x_a)||q(z_a)) + KL(p(z_d|x_d)||q(z_d))\right) \end{aligned} \tag{5}$$

Here, $\alpha = 0$ will produce representations that well predict the author-document relation but will not capture the stylistic features of the documents, as shown by [9]. With $\alpha = 1$, on the contrary, the model will simply bring document embeddings closer to their feature vectors. Hence, the value of α needs to be carefully tuned *on the dataset*, regarding if the corpus is writing style specific or not thanks to domain knowledge.

Eventually, computing the expected values in Eq.(5) is intractable for a wide range of encoders. We therefore approximate it by sampling L examples by observation (here, a triplet

document, author, feature vector), following $p(z|x)$ as done in [24]. We get (the same goes for feature vector/documents pairs) :

$$\mathbb{E}[\log q\left(y_a|z_a, z_d\right)] \approx \frac{1}{L}\sum_{l=1}^{L} \log q(y_a|z_a^{(l)}, z_d^{(l)}) \qquad (6)$$

We then use the reparametrization trick, following what is done in VAE [25]:

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

This loss can now be minimized using backpropagation. In Figure 2, we show a schematic representation of our model, called **VADES** for Variational Author and Document Representations with Style.

C. Encoding Functions and Choice of the Encoder

The entering bloc of our model for documents is a text encoder, mapping a document in natural language to a vector in $\mathbb{R}^r$. Many deep architectures could be used here and trained from scratch. Nevertheless, we propose to use a pretrained text encoder.

Models that are pretrained on large datasets are now easily available online [1]. They have been proved successful on many NLP tasks with a simple fine-tuning phase (the only constraint being to avoid catastrophic forgetting). Additionally, the VIB framework allows to naturally introduce a pretrained text encoder as shown by [11]. The encoder's output should then be mapped to document mean and variance. Both [8], [11] map the text encoder output to the document's mean (the f function) and variance (the g function) using a Multi Layer Perceptron (MLP). This approach is simple, and fast. In our experiments, we build f and g as two-layer MLP with *tanh* and linear activation with same input and intermediate dimensions (r_0). Note that the output dimension of f and g should be the same as the number of stylistic features (r).

Several constraints arise regarding the pretrained encoder itself. We would like our model to be able to capture stylistic information from a given document. As shown in [9], [26], state-of-the-art models trained on large datasets already capture complex grammatical and syntactic notions in their representations, and therefore have the explanatory power requested for our objective. Moreover, our model must be able to deal with long text as it will be used in a literary context. Processing novels, dramas, essays, where the writing style interferes the most. This is a serious problem: for example, the widely used BERT model is limited to 512 tokens. Alternative models such as [27] allow to apply transformers to long documents. To circumvent this issue, we use the Deep Averaging Network implementation of the Universal Sentence Encoder (USE) from [4]. It has several advantages over the latter works: it gives no length constraint, it is faster than transformer-based methods and it outperforms Sentence-BERT on stylistic features prediction [9]. The test of other encoder models is left to future works. Finally, note that our model is language agnostic (as it depends on a out-of-the-box text encoder) and can infer representations for unseen documents.

[1] e.g., https://huggingface.co/models

IV. Authorship Attribution Datasets

A. IMDb Corpus

The IMDb (Internet Movie Database) corpus is one of the most used ones regarding the authorship attribution task. It was introduced by [28] and is composed of $271,000$ movie reviews from $22,116$ online users. However, most of the works are evaluated on the reduction of this dataset to only 62 authors with 1000 texts for each (IMDb62). Thus, we benchmark our model on IMDb62. As shown later, the task of authorship attribution on this corpus is more or less solved, due to the low number of authors.

B. Project Gutenberg Dataset

The Project Gutenberg is a multilingual library of more than 60,000 e-books for which U.S. copyright has expired. It is freely available and started in 1971. We gathered the corpus using [29]. Most of the books are classical novels, dramas, essays, etc. from different eras, which is relevant when studying writing style and represents quite well our context of application. To keep the most authors possible, we randomly sample 10 texts for each author with such a production, leaving 664 authors in our Reduce Project Gutenberg Dataset (R-PGD) (10 times more than IMDB). To be able to deal with such works, we only keep the 200 first sentences of each book.

C. Blog Authorship Corpus

This dataset is composed of 681,288 posts from 19,320 authors gathered in the early 2000s by [30]. There are approximately 35 posts and 7,250 words by user. We only take 500 bloggers with at least 50 blogposts to build our reduced dataset of the Blog Authorship Corpus (R-BAC). This dataset is also used in several authorship attribution benchmark, only keeping the top 10 or 50 authors with most productions. We will also test our model on these extraction of the corpus.

These two last datasets (PGD and BAC) represent two common uses of author embedding (classic literature and web analysis) with a large number of authors. Usual datasets for authorship attribution (CCAT50, NYT, IMDb62) contain far less classes, further from our context of a web extracted corpus (from Blogger or Wordpress for example)... They are also stylistically and structurally different, allowing to evaluate our approach on various textual formats. For each dataset, we perform a 80/20 train-test stratified split.

V. Experiments

A. Parameter Setting and Competitors

In this section, we present implementation details for our method and competitors. For the encoder functions f and g, we use the architectures presented in the previous section with batch normalization and dropout equal to 0.2 with L2 regularization ($1e-5$). Grid search parameters are detailed in Table II. For L, we obtain a good trade-off between accuracy and speed with $L = 10$, as we quickly reach a plateau of

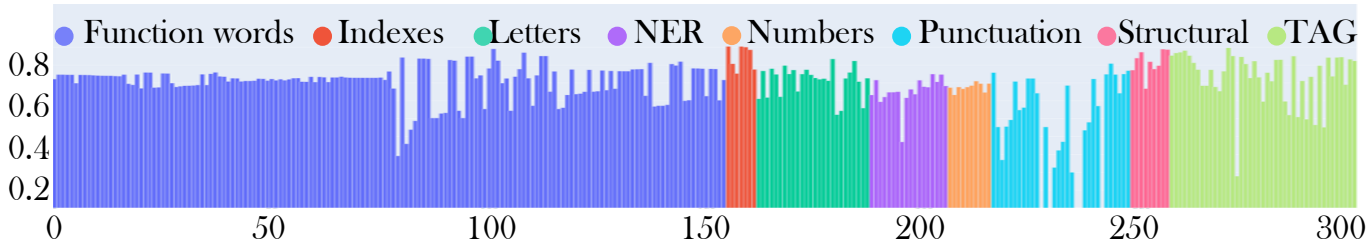

Fig. 3. **Correlation score between i^{th} embedding coordinates and i^{th} stylistic feature for VADES representation on R-PGD.** A few values in the Punctuation categories are null as they were not found anywhere in the corpus.

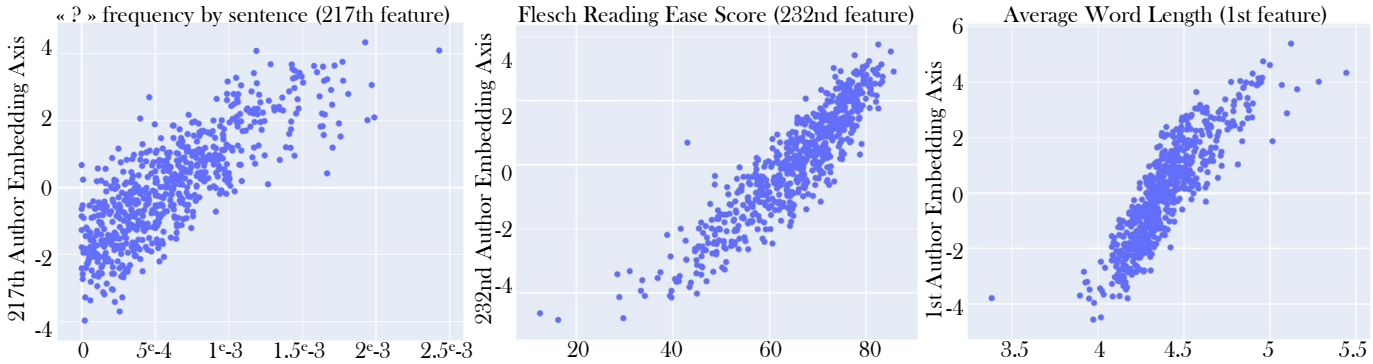

Fig. 4. **i^{th} embedding axis against i^{th} stylistic feature for each author representation, for a selection of 4 given features** We can see correlation between each feature and their respective embedding axis.

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 I
Descriptive statistics for the 3 datasets and their decomposition.
BAC : Blog Authorship Corpus, PGD : Project Gutenberg Dataset.

Hyperparameter grid search	
Hyperparameter	**Grid**
# negative pairs	{1, 5, **10**, 20}
Monte Carlo sampling	{1, 5, **10**, 20}
Learning rate	{1e-2, **1e-3**, 1e-4, 1e-5}
β	{1e-1, 1e-2, ..., **1e-12**}
Feature loss	{L2, **Cross-Entropy**}

TABLE II
Grid search used for hyperparameter selection.
Selected value in bold.

performance when increasing its value. We can summarize the tuning of α as follows:

- $\alpha = 0$ implies no feature loss and stylistic information,
- $\alpha = 0.5$ gives the same importance to feature loss and author loss,
- $\alpha = 0.9$ pushes feature loss to boost style detection.

For each task we selected α thanks to a grid-search to ensure the best results possible. We train the model for 15 epochs on R-PGD and R-BAC, and for 5 epochs on IMDB, BAC10 and BAC50 as the number of authors is around ten times smaller. We use a partition of 2 GPUs V100. On a single GPU, training the model on the R-PGD dataset takes around 10 hours. In the following section, we report the results for the best version of VADES only. As an ablation study, to justify the use of both the VIB framework and stylistic features, we compare our model with and without these components (respectively called *VADES no-VIB* and *VADES ($\alpha = 0$)*. The code is available on github and will be shared if the paper is accepted. All the datasets are available online.

We compare our model with several baselines. We use [20] (NGRAM Doc2Vec), a simple average based version of USE [4] (a document representation is built from the average of its sentence encoding, and an author representation is an average of its documents). We also compare our approach to DBert-ft [22], a document embedding method where DistilBERT is fine-tuned on the authorship attribution task. The author embeddings are built by averaging the representations of the documents it wrote. We use the parameters detailed in the authors' implementation[2].

B. Evaluation Tasks

We first evaluate the baselines and VADES regarding how well each method captures writing style. As writing style is a complex and a still discussed notion, there is no supervised

[2]https://github.com/hayj/DBert-ft

dataset to evaluate how a model can grasp it. We therefore use a proxy task that consists in predicting stylistic feature from the latent representations. We follow the experimental protocol of [9]. The stylistic features are extracted using spacy word and sentence tokenizer, POS-tagger and Name Entity Recognition, spacy English stopwords and nltk CMU Dictionary. For each author, we aim to predict the value of all stylistic features from their embeddings. Each feature is standardized before regression. We use an SVR with Radial Basis Function (rbf) kernel as it offers both quick training time and best results among other kernels in our experiments. We evaluate models using Mean Squared Error (MSE) following a 10-fold cross validation scheme.

Secondly, we perform authorship attribution, the task of predicting the author of a given document. We compare VADES with several other authorship attribution methods even though they do not necessarily perform representation learning. Each dataset is split into train and test sets with a 80/20 ratio. For our model, we repeated 5 times the evaluation scheme. For embedding method without classification head, we associate each document with its most plausible author using cosine similarity. We use accuracy to evaluate these results (the percentage of correctly predicted authors out of all data points).

C. Results on capturing writing style

As explained earlier, we use the author embeddings to perform regression and predict each stylistic features. As shown in Table IV, only using a simple logistic regression on these stylistic features allows to reach decent scores in authorship attribution, close to these of Universal Sentence Encoder, which is a state-of-the-art method in sentence embedding. As they contain strictly no topic information, it demonstrates how good they are as a proxy of writing style. Thus, a model able to capture them is able to capture writing style.

Results on the style MSE metric are shown in Table III. As expected, our model easily outperforms every baseline on all axes. DBert-ft, only trained on the authorship attribution objective performs the worst. Even though this approach is based on fine-tuned language models which already capture syntactic and grammatical notions [26], this is not the information that seems to be retained by the network when trained on the author attribution task. This is consistent with what was shown in [9]. The models may mainly focus on the semantic information to predict author-document relation. Interestingly, we observe that a simple average of USE representations performs quite well, which confirms that it can successfully capture complex linguistic concepts. VADES is guided by the feature loss to do so.

On a qualitative note, Figure 1 shows a toy example of a T-SNE 2D projection of well-known authors from the R-PGD dataset and their books (we use $\alpha = 0.5$). The objects are distributed in the space across clear author specific clusters. The most interesting observation is related to documents that are outside of their author cluster: *Thus Spake Zarathustra: A Book for All and None* by Nietzsche is a philosophical poem, closer to Hugo, while the rest of its production is mostly essays. The same conclusion goes with *The Power of Darkness* by Tolstoï, a 5 acts drama, whose embedding is closer to Shakespeare than to Tolstoï novels. The version of Hamlet presented here is fully commented, and thus is closer to analytical and philosophical works of Nietzsche and Plutarch as shown on the figure. We also represent the variance learnt by the model in the size of the author dot. Hugo, who wrote famous novels as well as poetry and dramas, has a greater variance than other authors.

D. Interpretability of the Representation Space

As we use the L_2 distance between document representations and stylistic feature vectors, each of the 300 embedding axes correspond to one given stylistic feature. The soft contrastive loss allows to ensures the L_2 constraint (bringing document embedding and stylistic features vectors closer) while being more flexible than a simple regression loss. When experimenting with the latter, the task showed up to be too hard and disadvantageous regarding both authorship attribution scores and writing style loss.

On Figure 3, we show the Pearson correlation score between the i^{th} stylistic feature and the corresponding embedding axis. These correlation values are always maximum for each feature regarding every other embedding coordinate. To further illustrate the interpretability of the embedding space, Figure 4 shows a selection of 4 stylistic features, the representation value of the matching coordinate for each author. The representation space learnt by VADES is interpretable in terms of writing style. In the context of a multidisciplinary project, involving several searchers in literature and linguistic this is a significant added value.

E. Results on the Authorship Attribution Task

Results on the authorship attribution task for IMDb62 and Blog Authorship Corpus are presented respectively in Table IV against state-of-the-art solutions (not necessarily embedding models). On both datasets, our model ranks in top 4 (or 5 for IMDB62), outperforming recent competitors while authorship attribution is not its main task. Our model is outpaced by Syntax CNN [39], DBert-ft [22] and BertAA [40], two variants of BERT fine tuned on the authorship attribution task. As shown by [40], BERT and DistilBERT are really tailored for balanced datasets with short texts such as IMDB62 and Blog Authorship Corpus. The DBert-ft model splits every document in 512 chunks during training, building an even bigger corpus with important improvement, but it is hardly reproducible with our feature loss. BertAA feeds encoded documents from a finetuned BERT together with a set of stylistic features and of most frequent bi-grams and tri-grams to a Logistic Regression. It clearly allows to better perform on Blog Authorship Corpus as this dataset is a mix of several genres and styles, compared to IMDB62 concerning only movie reviews. This confirms our use of stylistic features. Syntax CNN encodes each sentence of a document separately with its syntax. Unfortunately, this model was hardly reproducible

Embedding	Average MSE Regression Score along with standard deviation (SVR Model) on R-PGD dataset Letters	Numbers	Structural	Punctuation	Func. words	TAG	NER	Indexes
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)
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)
DBert-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)
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)**

Embedding	Average MSE Regression Score along with standard deviation (SVR Model) on R-BAC dataset Letters	Numbers	Structural	Punctuation	Func. words	TAG	NER	Indexes
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)
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)
DBert-ft	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 III

Feature prediction on R-PGD and R-BAC.

MSE score (standard deviation in parenthesis) on the prediction of stylistic features from author embedding on the R-BAC dataset using SVR. The 300 stylistic features are grouped by families. In bold the best scores for each axis. Our model (α value in parenthesis) performs best with $\alpha = 0.9$.

Approach	IMDb62 62 authors	Blog Authorship Corpus 10 authors	 50 authors
USE	60.2 (0.2)	40.7 (0.1)	24.7 (0.2)
Stylistic features + LR	88.2 (0.1)	40.9 (0.2)	28.4 (0.2)
LDA+Hellinger* [31]	82	52.5	18.3
Impostors* [32]	x	35.4	22.6
Word Level TF-IDF*	91.4	x	x
CNN-Char* [33]	91.7	61.2	49.4
C.Att + Sep.Rec.* [34]	91.8	x	x
Token-SVM* [28]	92.5	x	x
SCAP* [35]	94.8	48.6	41.6
Cont. N-gram* [36]	94.8	61.3	52.8
(C+W+POS)/LM* [37]	95.9	x	x
N-gram + Style* [38]	95.9	x	x
N-gram CNN* [39]	x	63.7	53.1
Syntax CNN* [39]	96.2	64.1	56.7
DBert-ft [22]	**96.7 (0.2)**	64.3 (0.2)	58.5 (0.2)
BertAA* [40]	93.0	**65.4**	**59.7**
VADES no-VIB (0.5)	91.3 (0.1)	60.9 (0.2)	50.2 (0.2)
VADES (0.0)	94.9 (0.2)	62.6 (0.2)	52.4 (0.2)
VADES (0.1)	95.6 (0.2)	63.8 (0.2)	53.8 (0.2)

TABLE IV

Authorship Attribution accuracy on IMDb62 and Blog Authorship Corpus

Results with * are gathered from other papers, x is for missing results on a given dataset. Best model in bold and second underlined, standard deviation in parenthesis. We here compare our model (in parenthesis α value) with several authorship attribution models. Our model compete with SOTA model while learning meaningful representations regarding writing style for documents and authors.

and cannot be tested in feature regression using intermediate representation. For VADES lower values of α allow to reach the best accuracy in authorship attribution on these datasets. Additional information bring by stylistic features benefit to the authorship attribution when texts are longer.

Finally, we compare our model to no-VIB and without feature loss (Table IV, III). Both variations underperform on both tasks. First, the VIB paradigm offers more versatility than fixed document and author representation which is key to grasp a complex notion such as writing style. Stylistic features can be sparse, using them directly as document encoder strongly degrade results. The gaussian paradigm of VIB helps to deal with it. Then, the feature loss brings additional information for authorship prediction, as shown by BertAA, which use it to improve BERT classification results. Here, our framework enable to use it directly for document and author embeddings.

VI. Conclusion

In this article, we presented VADES, a new author and document embedding method which leverages stylistic features. It has several advantages compared to existing works: it easily integrates any pretrained text encoder, it allows to compare authors and documents of any length (e.g., for authorship attribution), build an interpretable representation space by incorporating widely used stylistic features in computational linguistic. It is also able to infer representations for unseen documents at the opposite of most prior approaches. We demonstrated that VADES outperforms existing embedding baselines in stylistic feature prediction, often by a large margin, while staying competitive in authorship attribution.

In further experiments, we will incorporate modern text encoders, such as LLaMA [41]. They are much more difficult to adapt to this task, but as most recent Large Language Model are trained in an autoregressive way, they might have the expressive power needed to grasp stylistic aspects of authors productions.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the ACL: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, *et al.*, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020.

[3] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, "Is chatgpt a general-purpose natural language processing task solver?," *arXiv:2302.06476*, 2023.

[4] D. Cer, Y. Yang, S.-y. Kong, N. Hua, Limtiaco, and al., "Universal sentence encoder for english," in *Proceedings of the 2018 Conference on EMNLP: System Demonstrations*, pp. 169–174, 2018.

[5] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *Proceedings of the International Conference on EMNLP*, 2019.

[6] S. Amir, G. Coppersmith, P. Carvalho, M. J. Silva, and B. C. Wallace, "Quantifying mental health from social media with neural user embeddings," in *Proceedings of the Machine Learning for Healthcare Conference*, pp. 306–321, 2017.

[7] S. Ganguly, M. Gupta, V. Varma, V. Pudi, *et al.*, "Author2vec: Learning author representations by combining content and link information," in *Proceedings of the 25th International Conference Companion on World Wide Web*, pp. 49–50, International World Wide Web Conferences Steering Committee, 2016.

[8] A. Gourru, J. Velcin, C. Gravier, and J. Jacques, "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.

[9] E. Terreau, A. Gourru, and J. Velcin, "Writing style author embedding evaluation," in *Proceedings of the 58th Annual Meeting of the ACL, 2nd Workshop on Evaluation and Comparison of NLP Systems*, pp. 84–93, 2021.

[10] M. Yang and K.-P. Chow, "Authorship attribution for forensic investigation with thousands of authors," in *ICT Systems Security and Privacy Protection*, (Berlin, Heidelberg), pp. 339–350, Springer Berlin Heidelberg, 2014.

[11] R. K. Mahabadi, Y. Belinkov, and J. Henderson, "Variational information bottleneck for effective low-resource fine-tuning," in *International Conference on Learning Representations*, 2021.

[12] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[14] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 487–494, 2004.

[15] J. Karlgren, "The wheres and whyfores for studying textual genre computationally," *AAAI Technical Report (7)*, pp. 68–70, 2004.

[16] T. C. Mendenhall, "The characteristic curves of composition," *Science*, vol. ns-9, 1887.

[17] Y. Zhao and J. Zobel, "Effective and scalable authorship attribution using function words," in *Information Retrieval Technology*, (Berlin, Heidelberg), pp. 174–189, Springer Berlin Heidelberg, 2005.

[18] E. Stamatatos, "On the robustness of authorship attribution based on character n-gram features," *Journal of Law and Policy*, vol. 21, pp. 421–439, 01 2013.

[19] R. Schwartz, O. Tsur, A. Rappoport, and M. Koppel, "Authorship attribution of micro-messages," in *Proceedings of the 2013 Conference on EMNLP*, (Seattle, Washington, USA), ACL, Oct. 2013.

[20] S. Maharjan, D. Mave, and e. a. Shrestha, "Jointly learning author and annotated character N-gram embeddings: A case study in literary text," *International Conference RANLP*, 2019.

[21] F. Jafariakinabad and K. A. Hua, "Style-aware neural model with application in authorship attribution," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 325–328, IEEE, 2019.

[22] J. Hay, B.-L. Doan, F. Popineau, and O. Ait Elhara, "Representation learning of writing style," in *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, (Online), pp. 232–243, ACL, Nov. 2020.

[23] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *The 37th annual Allerton Conference on Communication, Control, and Computing*, p. 368–377, 1999.

[24] S. J. Oh, K. Murphy, J. Pan, J. Roth, F. Schroff, and A. Gallagher, "Modeling uncertainty with hedged instance embedding," in *Proceedings of the International Conference on Learning Representations*, 2019.

[25] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[26] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "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*, vol. abs/1906.04341, 2019.

[27] M. Zaheer, G. Guruganesh, K. A. Dubey, and e. a. Ainslie, "Big bird: Transformers for longer sequences," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17283–17297, 2020.

[28] Y. Seroussi, I. Zukerman, and F. Bohnert, "Authorship Attribution with Topic Models," *Computational Linguistics*, vol. 40, pp. 269–310, 06 2014.

[29] M. Gerlach and F. Font-Clos, "A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics," *Entropy*, vol. 22, no. 1, p. 126, 2020.

[30] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker, "Effects of age and gender on blogging.," in *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pp. 199–205, AAAI, 2006.

[31] S. El, manarelbouanani and I. Kassou, "Authorship analysis studies: A survey," *International Journal of Computer Applications*, vol. 86, 12 2013.

[32] W. Koppel, Moshe and Yaron, "Determining if two documents are written by the same author," *Journal of the Association for Information Science and Technology*, vol. 65, no. 1, pp. 178–187, 2014.

[33] S. Ruder, P. Ghaffari, and J. G. Breslin, "Character-level and multi-channel convolutional neural networks for large-scale authorship attribution," *CoRR*, vol. abs/1609.06686, 2016.

[34] W. Song, C. Zhao, and L. Liu, "Multi-task learning for authorship attribution via topic approximation and competitive attention," *IEEE Access*, vol. 7, pp. 177114–177121, 2019.

[35] G. Frantzeskou, E. Stamatatos, S. Gritzalis, and S. Katsikas, "Source code author identification based on n-gram author profiles," in *Artificial Intelligence Applications and Innovations*, (Boston, MA), pp. 508–515, Springer US, 2006.

[36] Y. Sari, A. Vlachos, and M. 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), pp. 267–273, ACL, Apr. 2017.

[37] J. Kamps, G. Tsakonas, Y. Manolopoulos, L. Iliadis, and I. Karydis, "Research and advanced technology for digital libraries 21st," in *Proceedings: 21st International Conference on Theory and Practice of Digital Libraries, 2017, Thessaloniki, Greece*, 2017.

[38] Y. Sari, M. Stevenson, and A. Vlachos, "Topic or Style ? Exploring the Most Useful Features for Authorship Attribution," *27th International conference on computational lingustics*, pp. 343–353, 2018.

[39] R. Zhang, Z. Hu, H. Guo, and Y. Mao, "Syntax encoding with application in authorship attribution," in *Proceedings of the 2018 Conference on EMNLP*, (Brussels, Belgium), pp. 2742–2753, ACL, Oct.-Nov. 2018.

[40] M. Fabien, E. Villatoro-Tello, P. Motlicek, and S. Parida, "BertAA : BERT fine-tuning for authorship attribution," in *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pp. 127–137, NLP Association of India (NLPAI), Dec. 2020.

[41] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.