# Writing Style Author Embedding Evaluation

**Enzo Terreau** **Antoine Gourru** **Julien Velcin**
Université de Lyon, Lyon 2, ERIC UR3083
{enzo.terreau, antoine.gourru, julien.velcin}@univ-lyon2.fr

## Abstract

Learning authors representations from their textual productions is now widely used to solve multiple downstream tasks, such as classification, link prediction or user recommendation. Author embedding methods are often built on top of either Doc2Vec (Le and Mikolov, 2014) or the Transformer architecture (Devlin et al., 2019). Evaluating the quality of these embeddings and what they capture is a difficult task. Most articles use either classification accuracy or authorship attribution, which does not clearly measure the quality of the representation space, if it really captures what it has been built for. In this paper, we propose a novel evaluation framework of author embedding methods based on the writing style. It allows to quantify if the embedding space effectively captures a set of stylistic features, chosen to be the best proxy of an author writing style. This approach gives less importance to the topics conveyed by the documents. It turns out that recent models are mostly driven by the inner semantic of authors' production. They are outperformed by simple baselines, based on state-of-the-art pretrained sentence embedding models, on several linguistic axes. These baselines can grasp complex linguistic phenomena and writing style more efficiently, paving the way for designing new style-driven author embedding models.

## 1 Introduction

Since the early work of Mikolov et al. 2013a, and more recent models based on the Transformer architecture (Devlin et al., 2019), continuous word representations have been key in processing and analyzing textual data. It led to a prolific research on learning meaningful representations of larger-scale textual entities, such as paragraph, document or even authors. Learning author embeddings can be used to solve several downstream tasks, such as user recommendation (Nayyeri et al., 2020) and classification (Benton and Dredze, 2018). Additionally, metadata (e.g., graph structure (Gourru et al., 2020), timestamp (Delasalles et al., 2019)) are often used to guide the representation learning process, in particular with application to social media. This is referred as user embedding, while author embedding only focuses on the textual production of users.

Despite a growing literature in both author and user embedding (Maharjan et al., 2019; Wu et al., 2020), it is usually difficult to tell what these representations really capture from its textual production. Although most recent models reach more than 90% accuracy in authorship attribution on several datasets (Maharjan et al., 2019), none of the existing works tried to determine if the embedding space captures topic preferences, topological information, sentiment or stylistic features. Getting a better understanding of what these spaces really capture can be a real asset to design new machine learning models.

For instance, one could be interested in linking several authors with similar writing style even though they deal with different topics. It is even more important with literacy data or for specific tasks such as authorship attribution for forensic investigation (Amir et al., 2017; Ganguly et al., 2016; Kumar et al., 2019). Being able to compare and determine which embedding methods are the more relevant in such a context is essential. Unfortunately, in most previous works, the evaluation of author embedding relies solely on classification accuracy, which demonstrates the need for a more robust and richer evaluation framework. As stated in (Conneau et al., 2018), most evaluation methods for embedding are based on downstream tasks, which does not fully assess the quality of the embedding space, if it really captures what is has been built for.

To tackle this issue, we propose a novel experimental scheme to evaluate author representations based on her writing style. Even though there is no

consensual definition of what the writing style is, many works have tried to identify the most relevant features to characterize it. In this article, we consider low-level structural and syntactical features uncorrelated with topics, and we show that most author embedding methods are in fact essentially driven by semantic. The code is made available at the following address: Style Embedding Evaluation

## 2 Related Works

In this section, we present an overview of existing author embedding models and evaluation frameworks. We also briefly review prior works on stylistic features selection through authorship attribution.

Author embedding consists in learning, for each author, a vector representation in a low dimensional space. In this space, the proximity between two vectors should relate to the similarity in authors' textual production.

Part of the literature focuses on user embedding. While works on author and user embedding may look similar, they should not be mistaken. User embedding usually refers to the context of social media where, in addition to the user textual production, several metadata (e.g., retweet, likes, links) are used to guide the learning process. Here, we mostly focus on methods that leverage the textual content only. Nevertheless, our experimental protocol method can be used to evaluate user embedding models in a stylistic way, in combination with other metrics. We leave this perspective to future work.

### 2.1 Author Embedding models

Representation learning in NLP took a huge step forward with (Mikolov et al., 2013b) and (Mikolov et al., 2013a) who proposed two neural models to learn word vectors, based on the distributional hypothesis. Each word embedding is learnt by solving a word co-occurrence prediction task. The Doc2Vec model (Le and Mikolov, 2014) extends these models to document embedding by adding a document id to the word context. More recently, Devlin et al. (2019) proposed another word representation method, the BERT model, that reaches state-of-the-art in various downstream tasks. Based on the Transformer architecture, each word representation is contextualized and thus different given the sentence in which it occurs. Today, Doc2Vec and BERT methods and their extensions constitute the basic bricks of every author embedding method.

Recent works apply representation learning to authors by solving several downstream tasks, such as author classification and link prediction. For example, the Aut2Vec model (Ganguly et al., 2016) is built on top of (Le and Mikolov, 2014). Document and author embeddings are initialized using Doc2Vec. Then, they train a single hidden layer model to perform authorship prediction (the Content Info model). The idea is to bring author representation closer to the content she/he produced. This simple model is paired with a link-info model using the same idea for co-authorship graph, which we do not develop here. Maharjan et al. (2019) use Doc2Vec formulation on documents of character trigrams. According to them, character trigrams should better capture both semantic content and writing style, as it was shown by previous studies (Sapkota et al., 2015; Stamatatos, 2013; Schwartz et al., 2013). The trigrams are also annotated according to their position in a given word or if they contain punctuation or not, following the idea of Sapkota et al. (2015). A few BERT-based methods recently emerged. Wu et al. (2020) use BERT to build representations of each author's posts. They are then aggregated using a bidirectionnal GRU. It allows to tackle authors with a various number of posts. This architecture is trained on authorship classification with a Multi-Layer Perceptron on top. One can also mention several methods dealing with dynamic author embedding (Kumar et al. (2019), Delasalles et al. (2019)). While our metric can also be applied to such contexts, we choose to work first in a static setting.

### 2.2 Author Embedding evaluation

Previous works use different strategies to evaluate the quality of author representations. Ganguly et al. (2016) perform link prediction (predicting if two authors have already co-authored a paper based on their embedding) and clustering. The latter requires an annotated dataset, the final metric being the Normalized Mutual Information after simple clustering through K-Means for example. Link prediction supposes we have additional information to text content if we want to build the author network. Maharjan et al. (2019) and Wu et al. (2020) also evaluate their models using an annotated dataset, through user depression prediction and gender prediction (Reddit MBTI9k dataset) for the first one, and book likeability prediction for the second one

(Goodread corpus). A simple classification model (e.g., SVM, MLP) is trained to predict the class for each author through its embedding. The accuracy score then allows to compare each method.

Maharjan et al. (2019) use authorship attribution to evaluate the quality of their model. Authorship attribution consists in predicting the author of a given document. It requires that document and author representations lie in the same space. It is performed either by clustering or simply by computing the cosine similarity between a document embedding and each author's embeddings to get either an accuracy score or a coverage error. This task could be a reference to evaluate author embedding. Being able to perfectly associate an author with its production ensures that the method efficiently captures each author's writing habits and characteristics. However, one of the biggest issues of authorship attribution is the lack of interpretability. It fails to reveal if a given author embedding method is more based on the content/topics or on the author writing style. To fully understand the distinction between writing style and content we can mention the book *Exercises in Style* of French author Raymond Queneau, who wrote the same story 99 times, but in 99 different ways (see Table 1). Although the story which is told remains the same, the choice of words and complexity of each sentence strongly differs. A way to get around this issue is to evaluate one's method on at least two datasets with various profiles. Sari et al. (2018) show that the decisive features to discriminate the author of a document can either be topic based or style based, depending on the dataset under study. Using at least two datasets when evaluating author embedding methods is a good step to better understand the model capacities.

Finally, author embedding evaluation methods are mostly centered on narrow downstream tasks and do not fully quantify the quality of the embedding space. For example, likeability prediction in (Maharjan et al., 2019) only measures one precise aspect of what the embedding space can capture. Conneau et al. (2018) therefore propose a large range of probing tasks to evaluate sentence embeddings. Following their idea, we propose an evaluation method for author embedding strictly based on writing style.

### 2.3 Stylometric Analysis

In this section we detailed which textual features are the most used to efficiently characterize an author's style. Authorship attribution is commonly used to identify an author way of writing and the most relevant features related to it. (Juola and Stamatatos, 2013; Stamatatos, 2013; Sapkota et al., 2015; Sari et al., 2018) propose a huge variety of textual extracted properties to tackle the problem of authorship attribution. The main breakthrough is that character n-grams are one of the most efficient and versatile features. It provides insight on both writing style and topic content of a given document. Sapkota et al. (2015) even show that character n-grams can be enhanced with position based affixes and punctuation n-grams.

More classic features (e.g., word and punctuation frequencies, hapax legomena, dislegomena) are also used with good efficiency since the 19th century (Mendenhall, 1887). They are often combined with function word frequencies (Zhao and Zobel (2005)) as they improve the performance of classifiers in authorship detection. Sari et al. (2018) performed ablation study on the authorship attribution problem, with style features (punctuation, function words, word length, etc.), topical features (frequencies of most common word n-grams) and hybrid variables (frequencies of most common character n-grams). Doing so, they reach state-of-the-art performance on two out of four datasets. They allow to identify the most useful and easily retrievable stylometric features to identify an author writing style without topic information. However, they do not rely on author embedding and cannot evaluate whether the embedding captures the writing style or not.

## 3 New proposed framework for author embedding evaluation

A classification model usually evaluates whether an embedding method successfully captures an author's topic preferences. Here, we want to evaluate how well the embedding captures its way of writing. As stated earlier, simple stylistic features are a good proxy of the authors' writing style. These features can easily be extracted from a corpus and aggregated by author. Training a simple regression model (typically linear regression or more complex methods, such as support vector regression) using author embedding to predict these features would allow to compare these representations in their abil-

| Writing style | Extract |
|---|---|
| Notation | *Two hours later, I meet him in the Cour de Rome, in front of the gare Saint-Lazare.* |
| Litotes | *Two hours later I met him again;* |
| Metaphorically | *In a bleak, urban desert, I saw it again that self-same day, ...* |
| Retrograde | *I met him in the middle of the Cour de Rome, after having left him rushin avidly towards a seat.* |
| Surprises | *Two hours after, guess whom I met in front of the gare Saint-Lazare!* |

Table 1: We illustrate here the distinction between writing style and semantic thanks to Raymond Queneau. For each writing style the same passage of the story is shown.

ity to separate writing styles. Even better, it would show if the representation space effectively captures each stylistic feature. Figures 1 and 2 show the intuition behind this simple idea. To the best of our knowledge, this method is the first to evaluate this aspect of author embedding. At the opposite of classification tasks, there is no metric such as accuracy for regression that gives an absolute measure of performance. Here, we propose to use the Mean Square Error as regression score, in a 10-fold cross validation scheme. First results show how poorly current author embedding method capture writing style and thus set a new goal for new methods to come.

### 3.1 Selection of key stylistic features

The selection of the stylistic features to retrieve from the corpus is key. It needs to fully embrace each author's writing style and specificity, whether it is based on phonetic, syntax, or structural. It should not be topic related. Based on (Zhao and Zobel, 2005), (Elahi and Muneer, 2018), (Sari et al., 2018), we choose a total of 301 stylistic features that are summed up in Table 3. Each of these features are aggregated into categories extracted from the aforementioned references according to their nature. Although none of the aforementioned works study POS and NER tags as stylistic features, (Szwed, 2017) shows that it can be used for authorship attribution with good results. (Feng et al., 2012; Ganjigunte Ashok et al., 2013) demonstrate that they are effective markers of the syntactic structure of a text, performing sentence type identification with POS tags. We therefore incorporate these features in our metric. As a test, we perform authorship attribution using these variables on the Project Gutenberg dataset with a simple logistic regression and a various number of authors. Results are shown in Table 2. We use two metrics, accuracy, and coverage error. Coverage error computes how far we need to go through the ranked scores to cover the true labels. Using style-based features only, we are able to reach an accuracy of 96% with 10 authors and 88% with 50 authors. The averaged coverage error is always near 2 (correct author has the second highest score in average in prediction). Best authorship attribution methods reach accuracy score between 90% and 95% (Sari et al., 2018; Ruder et al., 2016), depending on the dataset and the number of authors. These features are thus a good proxy of an author writing style as no topic information directly flows through the selected variables. Of course, there are correlation between style and topics. Strong line break frequency attest of poetry, strong PERSON NER-tag frequency attest of novels and so on. Are writing style and topics strictly separable remains an open question (Subramanian et al., 2018). Here, we tried to keep variables with least topic information possible.

| Authorship Attribution scores | | |
|---|---|---|
| **Number of authors** | **Accuracy** | **Coverage error** |
| 10 | 0.96 | 1.04 |
| 50 | 0.88 | 1.80 |
| 100 | 0.79 | 2.28 |

Table 2: Authorship attribution with logistic regression using only stylistic features. With no direct topic information, we are able to reach 96% accuracy with only 10 authors. These was performed on the full Project Gutenberg dataset with a random authors sample.

## 4 Evaluation

### 4.1 Competitors

To test our metric, we select several author embedding methods among the ones presented in Section 2.1. The Content Info model of Ganguly et al. (2016) (embedding size 512) and the annotated ngram based Doc2Vec model of Maharjan et al. (2019), with embedding size fixed to 300 (referred

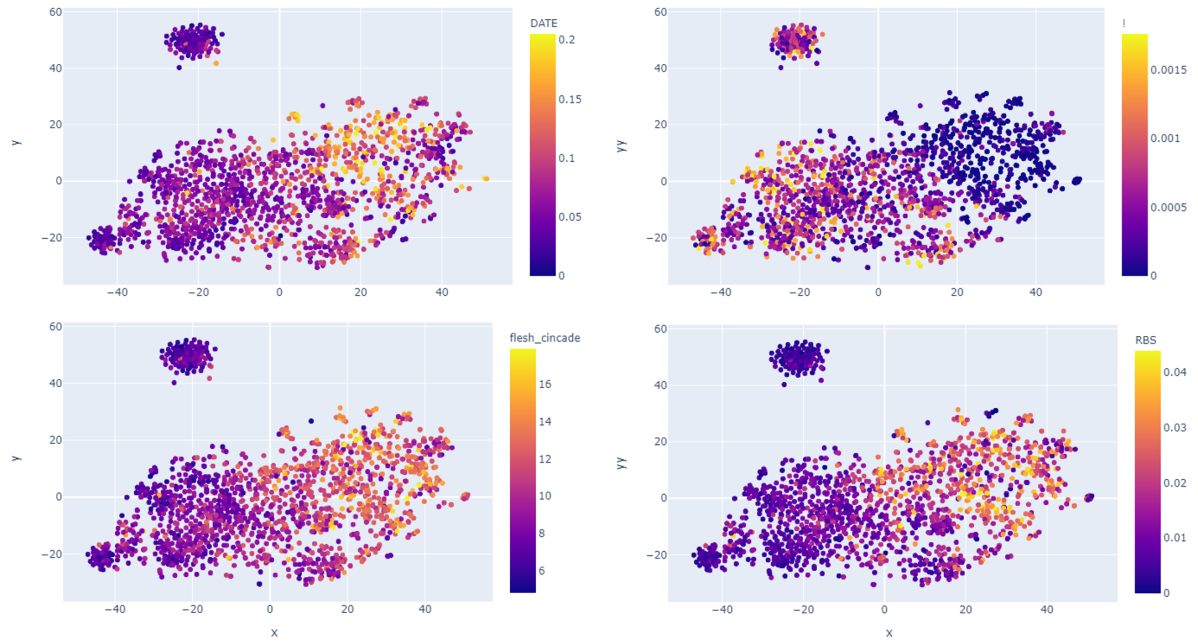


Figure 1: We project NGRAM DOC2VEC embeddings on the full Gutenberg dataset into a 2D space with t-SNE and we represent the gradient of 4 selected stylistic features (before standardization): DATE entity frequency, Exclamation mark frequency, Flesh-Cincade readability index and superlative adverb frequency - RBS. Clear tendencies appear, which motivates our method.

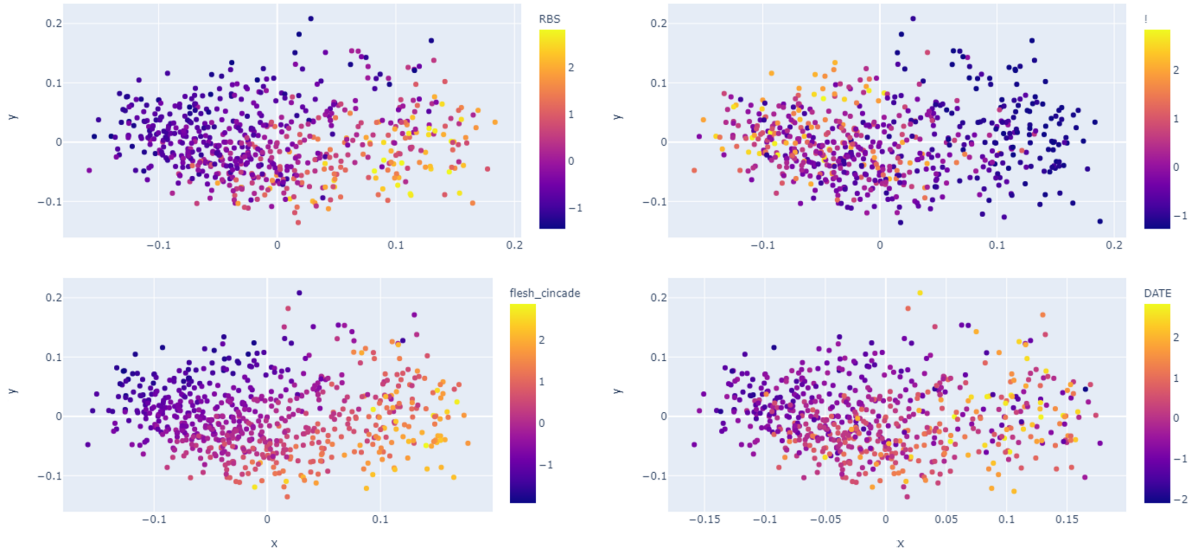


Figure 2: We project USE embeddings on the reduced Gutenberg dataset into a 2D space with t-SNE and we represent the gradient of 4 selected stylistic features (before standardization): DATE entity frequency, Exclamation mark frequency, Flesh-Cincade readability index and superlative adverb frequency - RBS. Clear tendencies appear, which motivates our method.

| Type of features | Examples | Number of features |
|---|---|---|
| Letters | Letters frequencies | 26 |
| Numbers | Numbers frequencies | 10 |
| Structural | Avg word length, Hapax Legomena, Syllable count, ... | 9 |
| Punctuation | Punctuation sign frequencies | 16 |
| Function words | Function words (does, once, doing, ...) frequencies | 174 |
| Tag | Pos tag frequencies | 43 |
| Ner | Name Entity Recognition tag frequencies | 18 |
| Indexes | Complexity and readability indexes | 7 |

Table 3: List of stylistic features selected and their categories. Frequencies are computed by sentence.

as NGRAM Doc2Vec in this paper). We also add two state-of-the-art sentence embedding methods: a) Universal Sentence Encoder (USE) (Cer et al., 2018), based on a Deep Averaging Network (DAN) built on top of a Bag Of Word (BOW) vector, and b) Sentence BERT (Reimers and Gurevych, 2019), built on top of BERT. Author embedding is then calculated by simply averaging every sentence representation of an author (embedding size 512 for USE and 768 for SBERT). Whereas it performs well on several downstream tasks for sentence embedding, we expect lower results in author embedding, both because of the averaging of full documents, and as it is not trained specifically to retrieve writing style. We choose not to test the most recent (Wu et al., 2020), which is closer to user embedding, and to focus on well-established method.

### 4.2 Datasets & framework

For the stylistic features extraction, we use spacy word and sentence tokenizer, POS-tagger and NER. We use nltk set of English stopwords and nltk CMU Dictionary for syllable count. Each feature is standardized before regression. The regression algorithm used is 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 apply our evaluation protocol to datasets of different natures, using most recent author embedding approaches. First, we experiment with a Lyrics dataset [1], consisting in a set of 47 singers from various music genre (Bob Dylan to Eminem, including Prince and Radiohead). There are around 2,300 verses by author. This dataset is rather small and unusual but is a good illustration of our approach. Nevertheless, poetry and by extension song lyrics are the type of document where one could expect literary style to express the most.

We also experiment on a Project Gutenberg dataset extracted following (Gerlach and Font-Clos, 2018) paper. 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. This dataset is often used in NLP, whether for its literacy aspect or for automatic translation. Here, we focus on the texts written in English, randomly sampling 10 books for each author. As most of the books are novels, we only take the 200 first sentences of each book, to eventually obtain 664 authors with 2,000 sentences by author. We refer to this subset as the "reduced Project Gutenberg dataset" in the upcoming paragraphs.

Finally, we use part of the Blog Authorship Corpus. This dataset is composed of 681,288 posts from 19,320 authors gathered in the early 2000s. 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 BlogAuthorshipCorpus.

These three datasets allow to cover different aspect of writing style, from pure literature to social media. To compare how well each embedding also captures content, we extract the 10 most relevant topics using LDA from the reduced version of project Gutenberg. We then perform a topic prediction task based on embeddings with an SVM. Results are presented in Table 4.

### 4.3 Results

All results are presented in Table 5, 6 and 7. For the Gutenberg and Blog Authorship datasets, they are summed up in Figure 3. Although smaller, the Lyrics dataset gives clear tendencies which are confirmed by the results on Blog Authorship and Gutenberg. The Content Info and NGRAM

[1] https://www.kaggle.com/paultimothymooney/poetry

| Topic prediction on Gutenberg dataset | | |
|---|---|---|
| **Embedding method** | **Accuracy** | **Coverage error** |
| Content-Info | 0.81 | 1.32 |
| SBERT | 0.76 | 1.38 |
| USE | 0.74 | 1.43 |
| NGRAM Doc2Vec | 0.73 | 1.44 |

Table 4: Topic prediction from author embeddings with an SVM. Topic are retrieved with an LDA with 10 topics. We see that each model efficiently captures each author's topic preferences, Content-Info model being the best at this task.

Doc2Vec models designed to build consistent author embeddings both obtain the worst MSE scores. USE and SBERT outperform them on almost every axis defined among our stylistic features. USE and SBERT are powerful models, pretrained on huge corpora with multitask training. It seems that they are able to capture linguistic notions, which is not achievable by current author embedding models.

As expected, the Content Info model performs poorly. It is based on BOW representations thus unable to grasp any structural, syntactic or punctuation-based information such as TAG or NER, even more after word tokenization. This model strictly focuses on topic preferences, as shown in Table 4, reaching top accuracy on topic prediction.

Character n-grams are known as the best features to capture both style and content, even more when they are annotated regarding their position in a word or if they contain punctuation as in (Sapkota et al., 2015). Here we show that Maharjan et al. (2019)'s method does not properly capture complex syntactic notions, suffering from the reduced window size in Doc2Vec formulation. Similarly to the Content Info model, it cannot detect TAG or NER, but grasps punctuation with ease. As function words are not filtered during preprocessing, all the tested models perform equally well along this axis.

The real surprise here is how well SBERT and USE perform in capturing complex grammatical and linguistic notions such as TAG, readability and complexity indexes. They also perform well along the structural axis (e.g., average word/sentence length, hapax legomena, short words frequency). Clark et al. (2019) show that each BERT attention head naturally focuses on different linguistic phenomena in a sentence. For example, in some heads, direct objects attend to their verbs. In others, auxiliary verbs mostly put attention to the verb they modify, and so on. Our experiment seems to show that this information is propagated to the author embedding. This is why transformer-based models, not even fine-tuned on a specific author-based downstream task, capture writing style notions so well. For USE, we use the DAN version, which performs non-linear transformations on word and bigram embedding averages over sentences. Despite the BOW assumption made in the model, it is the best embedding model regarding our metric. The DAN model successfully retrieves complex linguistic information, showing that a syntactic treatment of sentences is not a prerequisite to effectively represent them, as stated in the original paper (Iyyer et al., 2015). This gives a huge improvement in terms of computation time against costly transformer-based models. Training models on multiple tasks with a huge corpus allows to skip the need to process semantically sentences and documents. USE is trained on question answering, next and previous sentence prediction, and the SNLI task. We could expect the Transformer version of USE to have even better results, at the cost of a higher computation time. Relying on these pretrained models seems to be the path to develop new author embedding methods to better capture the writing style, as improvements can still be done on several axes.

## 5 Conclusion

We presented a new evaluation framework for author embedding focusing on the writing style. This evaluation scheme is based on the extraction of stylistic features which represent a good proxy of an author way of writing. We show that simple baselines outperform recent author embedding models in predicting most of those stylistic features. These baselines rely on state-of-the-art sentence embedding models which capture complex linguistic notion thanks to multitask training on several big corpora. This demonstrates the need to develop new author embedding models that can grasp the author writing style. If models relying on Doc2Vec show clear limit in this task, USE with a DAN architecture seems to be a way to go.

## References

Silvio Amir, Glen Coppersmith, Paula Carvalho, Mario J Silva, and Byron C Wallace. 2017. Quan-

| Embedding | Average MSE Regression Score along with standard deviation (SVR Model) on Lyrics Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Letters | Numbers | Structural | Punctuation | Func. words | TAG | NER | Indexes |
| **USE** | **0.78 (0.21)** | **0.84 (0.08)** | **0.56 (0.28)** | **0.73 (0.24)** | **0.93 (0.17)** | **0.43 (0.31)** | **0.58 (0.26)** | **0.32 (0.24)** |
| **Content Info** | 0.81 (0.21) | 1.08 (0.10) | 0.78 (0.21) | 0.96 (0.14) | 0.94 (0.15) | 0.98 (0.06) | 0.99 (0.07) | 0.90 (0.12) |
| **Ngram Doc2Vec** | 1.00 (0.08) | 1.07 (0.06) | 1.01 (0.04) | 1.01 (0.03) | 1.04 (0.06) | 1.00 (0.04) | 0.99 (0.04) | 0.98 (0.018) |
| **SBERT** | 0.79 (0.20) | 1.12 (0.10) | 0.77 (0.22) | 0.97 (0.11) | 0.94 (0.12) | 0.97 (0.06) | 0.96 (0.09) | 0.82 (0.21) |

Table 5: MSE score (standard deviation in parenthesis) on the prediction of stylistic features from author embedding on the Lyrics dataset using SVR. In bold the best score for every axis. Surprisingly, both author embedding methods are outperformed by simply averaging USE Embeddings when it comes to writing style representation.

| Embedding | Average MSE Regression Score along with standard deviation (SVR Model) on Gutenberg dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Letters | Numbers | Structural | Punctuation | Func. words | TAG | NER | Indexes |
| **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)** |
| **Content Info** | 0.67 (0.22) | 0.87 (0.12) | 0.54 (0.18) | 0.67 (0.16) | 0.71 (0.19) | 0.65 (0.17) | 0.74 (0.13) | 0.50 (0.15) |
| **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) |
| **SBERT** | 0.67 (0.27) | 0.90 (0.07) | 0.41 (0.19) | 0.62 (0.26) | 0.71 (0.21) | 0.51 (0.27) | 0.69 (0.18) | 0.32 (0.18) |

Table 6: MSE score (standard deviation in parenthesis) on the prediction of stylistic features from author embedding on the Gutenberg dataset using SVR. In bold, the best scores for each axis.

| Embedding | Average MSE Regression Score along with standard deviation (SVR Model) on Blog Authorship Corpus dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Letters | Numbers | Structural | Punctuation | Func. words | TAG | NER | Indexes |
| **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)** |
| **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) |
| **SBERT** | 0.68 (0.28) | 0.85 (0.03) | 0.48 (0.19) | 0.80 (0.12) | 0.85 (0.16) | 0.66 (0.20) | 0.82 (0.18) | 0.41 (0.16) |

Table 7: MSE score (standard deviation in parenthesis) on the prediction of stylistic features from author embedding on the Blog Authorship Corpus dataset using SVR. In bold the best scores for each axis.

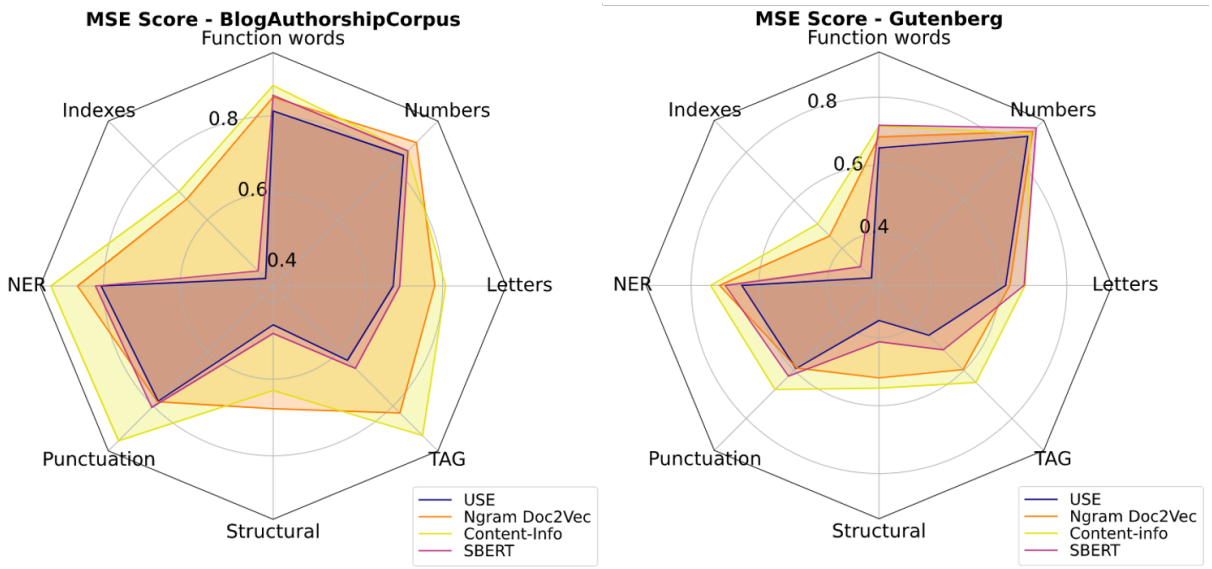


Figure 3: We represent previous regression results on spider charts to better visualize on which axis each embedding method performs the best.

tifying mental health from social media with neural user embeddings. In *Proceedings of the Machine Learning for Healthcare Conference*, pages 306–321.

Adrian Benton and Mark Dredze. 2018. Using author embeddings to improve tweet stance classification. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 184–194, Brussels, Belgium. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. 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*, volume abs/1906.04341.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. 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)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Edouard Delasalles, Sylvain Lamprier, and Ludovic Denoyer. 2019. Learning Dynamic Author Representations with Temporal Language Models. *CoRR*, abs/1909.04985.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186.

Hassaan Elahi and Haris Muneer. 2018. 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.

Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Characterizing stylistic elements in syntactic structure. *EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Proceedings of the Conference*, (July):1522–1533.

Soumyajit Ganguly, Manish Gupta, Vasudeva Varma, Vikram Pudi, et al. 2016. Author2vec: Learning author representations by combining content and link information. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 49–50. International World Wide Web Conferences Steering Committee.

Vikas Ganjigunte Ashok, Song Feng, and Yejin Choi. 2013. Success with style: Using writing style to predict the success of novels. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1764, Seattle, Washington, USA. Association for Computational Linguistics.

Martin Gerlach and Francesc Font-Clos. 2018. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *CoRR*, abs/1812.08092.

Antoine Gourru, Julien Velcin, and Julien Jacques. 2020. Gaussian embedding of linked documents from a pretrained semantic space. In *IJCAI*, pages 3912–3918.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. 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)*, pages 1681–1691, Beijing, China. Association for Computational Linguistics.

Patrick Juola and Efstathios Stamatatos. 2013. Overview of the author identification task at pan 2013. *CEUR Workshop Proceedings*, 1179.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting dynamic embedding trajectory in temporal interaction networks. *CoRR*, abs/1908.01207.

Quoc V. Le and Tomás Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32(2), pages 1188–1196.

Suraj Maharjan, Deepthi Mave, Prasha Shrestha, Manuel Montes-Y-Gómez, Fabio A. González, and Thamar Solorio. 2019. Jointly learning author and annotated character N-gram embeddings: A case study in literary text. *International Conference Recent Advances in Natural Language Processing, RANLP*, 2019-Septe.

T. C. Mendenhall. 1887. The characteristic curves of composition. *Science*, ns-9(214S):237–246.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mojtaba Nayyeri, Sahar Vahdati, Xiaotian Zhou, Hamed Shariat Yazdi, and Jens Lehmann. 2020. Embedding-based recommendations on scholarly knowledge graphs. In *The Semantic Web*, pages 255–270, Cham. Springer International Publishing.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *CoRR*, abs/1609.06686.

Upendra Sapkota, Steven Bethard, Manuel Montes, and Thamar Solorio. 2015. Not all character n-grams are created equal: A study in authorship attribution. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–102, Denver, Colorado. Association for Computational Linguistics.

Yunita Sari, Mark Stevenson, and Andreas Vlachos. 2018. Topic or Style ? Exploring the Most Useful Features for Authorship Attribution. *27th International conference on computational lingustics*, pages 343–353.

Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. 2013. Authorship attribution of micro-messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891, Seattle, Washington, USA. Association for Computational Linguistics.

Efstathios Stamatatos. 2013. On the robustness of authorship attribution based on character n-gram features. *Journal of Law and Policy*, 21:421–439.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text style transfer. *CoRR*, abs/1811.00552.

Piotr Szwed. 2017. Authorship attribution for polish texts based on part of speech tagging. pages 316–328.

Xiaodong Wu, Weizhe Lin, Zhilin Wang, and Elena Rastorgueva. 2020. Author2vec: A framework for generating user embedding. *CoRR*, abs/2003.11627.

Ying Zhao and Justin Zobel. 2005. Effective and scalable authorship attribution using function words. In *Information Retrieval Technology*, pages 174–189, Berlin, Heidelberg. Springer Berlin Heidelberg.