Dynamic Gaussian Embedding of Authors

Antoine Gourru
Université de Lyon, Lyon 2, ERIC UR3083
Laboratoire Hubert Curien, UMR CNRS 5516
Lyon, Saint-Etienne, France
antoine.gourru@gmail.com

Julien Velcin
Université de Lyon, Lyon 2, ERIC UR3083
Lyon, France
julien.velcin@univ-lyon2.fr

Christophe Gravier
Laboratoire Hubert Curien, UMR CNRS 5516
Saint-Etienne, France
christophe.gravier@univ-st-etienne.fr

Julien Jacques
Université de Lyon, Lyon 2, ERIC UR3083
Lyon, France
julien.jacques@univ-lyon2.fr

ABSTRACT

Authors publish documents in a dynamic manner. Their topic of interest and writing style might shift over time. Tasks such as author classification, author identification or link prediction are difficult to solve in such complex data settings. We propose a new representation learning model, **DGEA** (for Dynamic Gaussian Embedding of Authors), that is more suited to solve these tasks by capturing this temporal evolution. We formulate a general embedding framework: author representation at time t is a Gaussian distribution that leverages pre-trained document vectors, and that depends on the publications observed until t. The representations should retain some form of multi-topic information and temporal smoothness. We propose two models that fit into this framework. The first one, **K-DGEA**, uses a first order Markov model optimized with an Expectation Maximization Algorithm with Kalman Equations. The second, **R-DGEA**, makes use of a Recurrent Neural Network to model the time dependence. We evaluate our method on several quantitative tasks: author identification, classification, and co-authorship prediction, on two datasets written in English. In addition, our model is language agnostic since it only requires pre-trained document embeddings. It outperforms existing baselines by up to 18% on an author classification task on a news articles dataset.

CCS CONCEPTS

• **Information systems** → *Document representation*; • **Computing methodologies** → *Natural language processing*; *Neural networks*; *Latent variable models.*

KEYWORDS

Representation Learning, Dynamic Gaussian Embedding, Author Embedding, Document Embedding

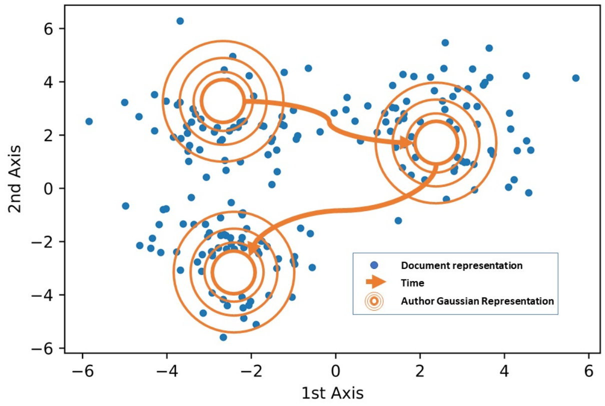

Figure 1: *Output of DGEA* - We here present an illustration of the dynamic Gaussian embedding of an author (in orange) in the same semantic space than pre-trained document representation. We illustrate documents' representations as dots (in blue). They are built using document encoders such as the Universal Sentence Encoder [16]. The author's representations are Gaussian distributions that evolve over time (indicated as an orange arrow).

ACM Reference Format:
Antoine Gourru, Julien Velcin, Christophe Gravier, and Julien Jacques. 2022. Dynamic Gaussian Embedding of Authors. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3512084

1 INTRODUCTION

1.1 Context

Text is the dominant form of information exchange on the Web. Authors' interests and personality can often be determined by their publication history. On social media, such as Twitter, users' publications are a mix of personal diary entries and speak-up stances. Authors' publications are often observed on a large temporal window. This information is crucial. For example, in the scientific literature, researchers tend to publish in narrow domains when considering short-time windows, though their research interests significantly vary at the scale of a decade due to the evolution of their scientific fields, job relocation, new positions, etc.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France.

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512084

As the amount of available textual data gets bigger, automatic tools are developed to scrap it, store it, and process it. Information Retrieval study the organization of corpora to facilitate the access to relevant information. Usual Information Retrieval tasks, such as recommendation or expert finding [11], require a coarse-grain representation of authors. This task is referred to as Representation Learning. It uses machine learning approaches to build representations of textual objects (words, documents, author) into a (low dimensional) Euclidean space.

To capture the dynamic information and to consider the drift of authors interest, we propose to learn multi-purpose *dynamic representations* of authors in the same space than documents embeddings. The representations should demonstrate smooth temporal trajectories, i.e., they should incorporate some form of time dependence. As we consider a discrete representation of time, we learn, for each author, a sequence of representations in a semantic space $\mathbb{R}^r$. Additionally, as authors can write on different subjects, we propose to learn *Gaussian representations* to capture the author semantic uncertainty. We illustrate our main objective in Figure 1.

1.2 Motivation

Learning one author's representation by time bin is a strong modeling assumption. We provide here empirical evidences that support this assumption and therefore the relevance to tackle this problem. We start with an illustrative example. In Table 1, we present 5 decades of Geoffrey Hinton's publications. The selected articles are the most cited. His research interests clearly evolved. He initially worked on cognitive science then progressively changed to neural network and general machine learning. We extend this observation to a significant pool of authors. To this end, we consider the S2G corpus, composed of machine learning articles' titles from Ammar et al. [2], each of them being associated to one or several authors. The articles were published between 1985 and 2017. We first construct vectorial representations in $\mathbb{R}^r$ of each document in the dataset with the Universal Sentence Encoder [13]. We then compute, for each author, the cosine similarity between the representations of the documents she/he wrote in her/his last year of publication T and the representations of the documents she/he wrote in the previous year (T-1), in the one preceding this last year (T-2) and in her/his very first year of publication (T1). We then compute the average similarity, for each author, and then the overall average over the corpus. We also trained a baseline classifier, which is a two-layers MLP (700 units) using ReLU activation and dropout (rate set to 0.1), that predicts the document's authors from the USE representations. We use documents in T as training data, and evaluate performance at T-1, T-2 and T1 (in terms of micro-F1). For comparison, the micro-F1 on a validation set of 10% of the documents in T is close to 13%. We obtain the results presented in Table 2.

On average, the representations of an author's documents are closer between T and T-1 and between T and T-2 than between T and T1. We perform a Student's t test on the means per author and obtain that the similarities T vs T-1 are significantly greater than T vs T1 (p-value : 1.6e-24), greater at T vs T-2 than T vs T1 (p-value : 8.6e-20) but not significantly greater between T vs T-1 and T vs T-2 (p-value :0.059). Finally, we observe a significant drop in the author's prediction performance between T-1/T-2 and T1. This shows that the representations are much closer between T, T-1 and T-2 than between T and T1. On this dataset, the representations of the documents produced by the authors have evolved, demonstrating the necessity to learn dynamic representations of authors.

1.3 Related works

Representation Learning has been successfully applied to Natural Language Processing (NLP), e.g. for information retrieval [32] or text generation [12]. User embedding consists in learning representations for numerical services users (usually web services). Users are usually associated with heterogeneous data [21, 37], such as link information, purchases history, text, etc. In this article, we study a sub-domain of representation learning for users that focuses only on the *textual information*. We call it *author representation learning*.

Several approaches go beyond the naive Term frequency representation of documents. In [31], authors propose to learn sentence representations with an encoder-decoder architecture, while in [3], authors average pre-trained word embeddings and modify the obtained vectors using a PCA to produce the document representation. Recent approaches [13, 16] train different aggregation functions (LSTM, Transformers, and Deep averaging networks) to map matrices of word embeddings to a single vector. The function is trained using pairs of related documents. Reimers and Gurevych [39] proceed in a similar way by fine tuning a BERT model [18] using a siamese networks approach.

Representation learning for authors has been less studied. The Author Topic Model (ATM) [41], based on LDA [9] learns low dimensional vectors for each author. This vector is a distribution over latent topics. Aut2Vec [20] uses textual and link information to build users' representations. They propose two models. The first one, the Content Info model, is based on the textual content only. It uses a Deep Neural Network. For a given pair of author/document, the network outputs the probability that the author wrote the document, computed using the distance between the author and the document representation. Usr2vec [1] makes use of pre-trained word embeddings, using a method close to Le and Mikolov [33].

Very few methods take the temporal structure of the publications into account. Blei and Lafferty [8] propose a graphical model to learn topics with smooth temporal evolution. Recent works focus on Dynamic Topic Embedding such as [19]. Nevertheless, this literature does not model author level representation. Sarkar et al. [43] use Kalman Equations to learn author and word representations that evolve over time. The conditional probability of co-occurrence depends on author and word latent factors and uses the CODE formulation [22]. Nevertheless, they need to use several approximations, such as a Taylor approximation of the variational bound. Furthermore, the transition matrix (fixing the temporal evolution) is shared among the authors and treated as an hyperparameter. Delasalles et al. [17] propose to use a recurrent neural network to model the evolution of author language models over time. The network uses functions of static and dynamic representations to predict the next time step language model using a residual transition model. These two approaches model author-word co-occurrences only and cannot infer representations for unseen authors.

Title	Year of publication
"Some demonstrations of the effects of structural descriptions in mental imagery"	1979
"Phoneme recognition using time-delay neural networks"	1989
"Unsupervised learning: foundations of neural computation"	1999
"Learning multiple layers of features from tiny images"	2009
"When does label smoothing help?"	2019

Table 1: *Five decades of Geoffrey Hinton's publication record* - We here present the titles of the articles and their year of publication. We provide the most cited article (according to Google Scholar) of the considered year. His research interest clearly evolved over the year: he started with cognitive science, and progressively shifted to neural network/machine learning.

	T-1	T-2	T1
Average cosine similarity (T vs ...)	0,233	0,228	0,201
Micro-F1 for author prediction (in %)	4,74	3,13	0,87

Table 2: *Quantifying the authors' publication evolution* - Comparison of average cosine similarities between S2G authors' documents representations at T vs. T-1, T vs. T-2, and T vs. T1. We also provide the results of a simple classifier that identify the author of a document trained on the authors' documents representations at time T (their last year of publication).

Furthermore, an important limitation of these previous works is that they learn vectorial representations. However, author publications often address several topics. Recent approaches therefore learn an uncertainty measure in addition to vectors for each document, such as [24, 27, 34]. These models learn document (not author) representations as Gaussian distributions. The GELD method we proposed in a previous article Gourru et al. [24] allows to learn document representations from pre-trained word embedding. We use a generative hypothesis: word vectors (and citations) are drawn from Gaussian distributions representing the documents. These approaches do not provide any way to capture the temporal dependencies for such representations. In a nutshell, none of previous works applied this approach to learn author-level representations as Gaussian distributions in the same space than documents.

1.4 Proposition

We propose an original general framework to learn Dynamic Gaussian Embeddings of Authors (DGEA) in a continuous latent space $\mathbb{R}^r$. It is general in the sense that it can deal with any document representation, and in the sense that we provide a high-level theoretical description of the temporal dependence. The key novelties are that 1) authors are represented as Gaussian distribution, 2) these representations lie in a same space than documents, represented as data point, and 3) authors representations are also meant to capture the temporal dependencies, so that their representation is time dependent. We formulate a simple hypothesis that leverages pre-trained document representations: documents vectors in $\mathbb{R}^r$ are generated by a Gaussian distribution, whose parameters depend on the author. We then introduce two types of temporal dependencies. The parameters (means and variances) are either 1) smoothed, so that the parameters of consecutive time slices are close, modeling the dependence as a prior at the parameter level, or 2) a function of the entire publication history, i.e., the means and variances at time T are conditionally independent of the values of the means and variances in the previous time slices. We then propose two instances of this general framework: K-DGEA, based on Kalman smoothing, and R-DGEA, which uses a recurrent neural network (the representation of the author at time T is a function of all her/his publication history).

We show that using the dynamic information allows to improve authorship identification, co-authorship prediction and author classification on two English datasets. To mitigate the risk that our experiments are dependent to the text encoder at use, we employ three different settings the encode the document titles: InferSent [16], USE [13] and SBERT [39], that are all reported to be efficient for short-texts representations. Nevertheless, our framework can be applied to any document representation, including embedding of long documents obtained using methods such as [5, 15].

Our model has many advantages over existing work: 1) the R-DGEA model (the second instance of DGEA) can infer representations for unseen authors, which avoids re-training the network (our approach is the only existing method with this inductive property), 2) the authors are represented as Gaussian distributions, so they are associated with a variance measure, allowing to give additional information in a recommendation process for example, 3) it easily incorporates pre-trained document representations, so that our method can evolve as new approaches for learning document representations are discovered, and 4) our model obtains the best overall performances on the evaluated tasks.

2 THE DGEA FRAMEWORK

2.1 Generative model

We consider a corpus of timestamped documents. Each document has a fixed representation in a semantic space $\mathbb{R}^r$, computed using any pre-trained document encoder, e.g. [13, 16], [23, 45] for linked documents, or [5] for long documents. As done in previous works [4, 42], we split the corpus into T time bins indexed by $t \in \{1, \ldots, T\}$. Each author i ($1 \leq i \leq N$) is associated with the bag of document embeddings $\mathcal{D}_i^{(t)} = \{d_{i,1}^{(t)}, d_{i,2}^{(t)}, \ldots\}$ she/he wrote at time t. In our notations, $d_{i,1}^{(t)}$ is the embedding of the first document wrote by author i at time t. We introduce the notation $|\mathcal{D}_i^{(t)}| = n_{i,t}$, the number of documents published by the author i at time t. We also use the following notation for vector sequences: $x_{1:T} = (x_1, x_2, ..., x_{T-1}, x_T)$.

We posit that the vectors in $\mathcal{D}_i^{(t)}$ are independently drawn from a diagonal Gaussian distribution, meaning that the j-th element of this set follows $d_{i,j}^{(t)} \sim \mathcal{N}(\phi_{i,t}, \sigma_{i,t}^2 I)$, with $\phi_{i,t}, \sigma_{i,t}^2 \in \mathbb{R}^r$. Note that this means that the Gaussian distributions are author dependent. We choose to use a Gaussian for several reasons: it allows to learn a measure of semantic uncertainty, and it simplifies the calculation. A strong hypothesis is that documents in the same time bin are exchangeable. We believe this hypothesis to be reasonable when choosing a relevant temporal resolution. Obviously, the mean and variance are not independent of the author publication history. We need to introduce some form of *time dependence*. In Figure (1), we show what the output should look like: authors are represented as smooth trajectories of Gaussians in a pre-trained document embedding space. We propose two different approaches to model temporal dependencies: stochastic dependence and functional dependence.

2.2 Stochastic dependence: the K-DGEA model

The simplest way to model the temporal evolution is to state that the authors' representations evolve according to a Markov model, i.e., that the sequences of means and variances follow:

$$\begin{aligned} p(\phi_{i,1:T}) &= p(\phi_{i,1}) \prod_{t=2}^{T} p(\phi_{i,t}|\phi_{i,1:t-1}) \\ p(\sigma_{i,1:T}^2) &= p(\sigma_{i,1}^2) \prod_{t=2}^{T} p(\sigma_{i,t}^2|\sigma_{i,1:t-1}^2). \end{aligned} \tag{1}$$

With $\mathcal{D}$ the set of observations, i.e. $\{\{\mathcal{D}_i^{(t)}\}_{i=1}^N\}_{t=1}^T$, and $\mathcal{D}_i = \{\mathcal{D}_i^{(t)}\}_{t=1}^T$, we obtain the following likelihood:

$$\mathcal{L}(\mathcal{D}) = \prod_i \int p(\sigma_{i,1:T}^2) p(\phi_{i,1:T}) p(\mathcal{D}_i|\phi_{i,1:T}, \sigma_i^2) d_{\phi_{i,1:T}, \sigma_{i,1:T}^2}. \tag{2}$$

This likelihood is generally difficult to maximize due to the multiple integration and to the time dependence. Here, there is no closed form solution, contrary to classical Linear Dynamic Gaussian Models [6]. We therefore consider a simpler model. We use a first order Markov model on the authors' means. We learn a single variance per author, i.e., the variance does not depend on time. This second assumption is obviously simplistic, but it is computationally convenient, and it allows us to maximize the likelihood efficiently: we can rewrite it in such a way as to make use of Kalman's equations [6]. At each step, the author's mean evolves according to a Gaussian distribution depending only on the previous time step, and with a diagonal variance:

$$\phi_{i,t} \sim \mathcal{N}(\phi_{i,t-1}, \delta_i^2 I) \quad \text{and} \quad \phi_{i,1} \sim \mathcal{N}(\phi_{i,0}, \delta_{i,0}^2 I), \tag{3}$$

where $\phi_{i,0}$ and $\delta_{i,0}^2$ are the initial parameters to estimate. We obtain:

$$\begin{aligned} \mathcal{L}(\mathcal{D}; \psi) &= \prod_i \int p(\phi_{i,1:T}) p(\mathcal{D}_i|\phi_{i,1:T}, \sigma_i^2) d_{\phi_{i,1:T}} \\ &= \prod_i \int p(\phi_{i,1:T}) \prod_t \prod_j p(d_{i,j}^{(t)}|\phi_{i,t}, \sigma_i^2) d_{\phi_{i,1:T}} \end{aligned} \tag{4}$$

where $\psi = \{\phi_{i,0}, \delta_{i,0}^2, \delta_i^2, \sigma_i^2\}_{1 \le i \le N}$. For each author, we need to identify the values of the parameters ψ that maximize $\mathcal{L}(\mathcal{D}; \psi)$. As we show in Appendix C, $p(d_{i,j}^{(t)}|\phi_{i,t}; \sigma_i^2)$ can be rewritten as:

$$\prod_j p(d_{i,j}^{(t)}|\phi_{i,t}; \sigma_i^2) = C_t \omega(\sigma_i^2, \mathcal{D}_i^{(t)}) \mathcal{N}(\bar{d}_{i,t}; \phi_{i,t}, r_{i,t}^2 I) \tag{5}$$

with C_t a positive constant. $\bar{d}_{i,t}$, $r_{i,t}^2$ and $\omega(\sigma_i^2, \mathcal{D}_i^{(t)})$ are defined as :

$$\bar{d}_{i,t} = \frac{1}{n_{i,t}} \sum_l d_{i,j}^{(t)},\ r_{i,t}^2 = \frac{\sigma_i^2}{n_{i,t}},\ S_{i,t,r}^2 = \frac{1}{n_{i,t}} \sum_j (d_{i,j,r}^{(t)} - \bar{d}_{i,t,r})^2 \tag{6}$$

$$\text{and} \quad \omega(\sigma_i^2, \mathcal{D}_i^{(t)}) = \exp\left(-\frac{n_{i,t}}{2} \sum_r \frac{S_{i,t,r}^2}{\sigma_{i,r}^2}\right) \left(\prod_r \sigma_{i,r}^2\right)^{-\frac{n_{i,t}-1}{2}}. \tag{7}$$

In this notation, r refers to the dimension of the semantic space. Moreover, $\omega(\sigma_i^2, \mathcal{D}_i^{(t)})$ does not depend on $\phi_{i,t}^2$. The likelihood becomes:

$$\begin{aligned} \mathcal{L}(\mathcal{D}; \psi) = &\underbrace{\prod_i^N \prod_{t=1}^T C_t \omega(\sigma_i^2, \mathcal{D}_i^{(t)})}_{\text{Normalizing factor}} \\ &\times \underbrace{\int p(\phi_{i,1:T}) \prod_{t=1}^T \mathcal{N}(\bar{d}_{i,t}; \phi_{i,t}, r_{i,t}^2 I) d_{\phi_{i,1:T}}}_{\text{Simple dynamic linear system}}. \end{aligned} \tag{8}$$

We obtain a simpler dynamical system, where instead of multiple observations per time step, we can aggregate the observations and consider the mean only. Apart from a normalization factor, our objective is now similar to the likelihood maximization of the simple model:

$$\phi_{i,t} \sim \mathcal{N}(\phi_{i,t-1}, \delta_i^2 I) \quad \text{and} \quad \bar{d}_{i,t} \sim \mathcal{N}(\phi_{i,t}, r_{i,t}^2 I). \tag{9}$$

$\phi_{i,t}$ is the latent state which evolves with noise δ_i^2 and $\bar{d}_{i,t}$ the measurement, with error $r_{i,t}^2$. We can therefore apply the standard Kalman equations [28, 38]. We use an EM model with forward and backward recursions. We present the mathematical details in Appendix B.

2.3 Functional dependence : the R-DGEA model

Instead of treating the temporal information as a prior structuring the sequences of latent variables, the author representation is a function of the whole publication history:

$$\phi_{i,t} = f(\mathcal{D}_i^{(1)}, ..., \mathcal{D}_i^{(t-1)}) \text{ and } \sigma_{i,t} = h(\mathcal{D}_i^{(1)}, ..., \mathcal{D}_i^{(t-1)}). \tag{10}$$

The likelihood becomes:

$$\mathcal{L}(\mathcal{D}) = \prod_{i,t,j} p(d_{i,j}^{(t)}|f(\mathcal{D}_i^{(1:t-1)}), h(\mathcal{D}_i^{(1:t-1)})). \tag{11}$$

We propose to use neural networks capable of modeling sequential data, such as Recurrent Neural Networks (RNN). Let us define

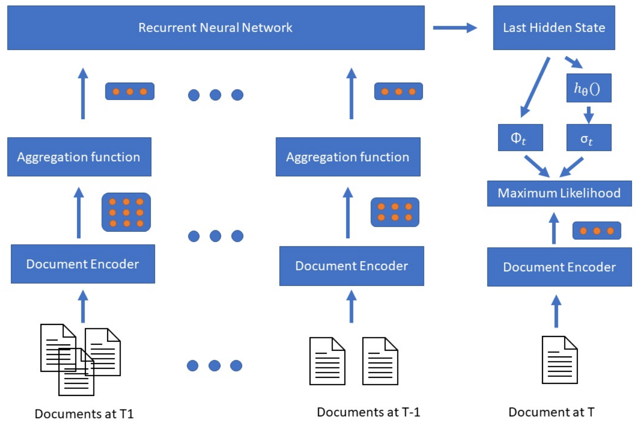

Figure 2: *Illustration of the R-DGEA architecture* - The R-DGEA model takes a sequence of document representations as input, which are aggregated by time bin. The RNN outputs a vector, which is the mean of the author at time T. This vector is then fed to a MLP that outputs the variance (similar to VAE [30]). The likelihood of the documents written at time T is then maximized.

f_θ a Recurrent Neural Network [14, 25] or a Transformer [44] with shared parameters, and g an aggregation function mapping from $\mathbb{R}^{n_{i,j,t} \times r}$ to $\mathbb{R}^r$, with n_j the number of words in the j-th document of $\mathcal{D}_i^{(t)}$. We get :

$$\phi_{i,t} = f_\theta(g(\mathcal{D}_i^{(1)}), ..., g(\mathcal{D}_i^{(t-1)})) \tag{12}$$

For the variance, we propose to use a multi-layer perceptron mapping the means to the variance vector. The function is therefore $\log \sigma_{i,t}^2 = h_{\theta'}(\phi_{i,t})$. This is similar to the Deep Averaging Network of Cer et al. [13] and what is done in Variational Auto Encoder [30]. This model can also be seen as a degenerated and dynamic version of the Mixture Density Networks [7]. The intuition is that some areas of the latent semantic space will be associated to a higher variance. In Appendix, we present and evaluate different other architectures, showing that mapping the mean to the variance using an MLP performs best. We present in Figure 2 a general overview of R-DGEA architecture.

2.4 Differences between K-DGEA and R-DGEA

The K-DGEA model is a smoothing approach. It cannot predict the authors' representations at time $t + 1$: the most probable value of ϕ_{t+1} will be ϕ_t, with an uncertainty given by the transition variance, δ_i. On the other hand, R-DGEA predicts the representation at time $t+1$ given the publication history up to time t. It is thus able to infer the representations of authors who have not been seen during the training phase. Another important difference is that, in R-DGEA, the transition parameters θ are shared by all authors (they are the RNN parameters), contrary to K-DGEA (the δ vectors are specific to each author).

	NYT			S2G		
Method	SBERT	InferSent	USE	SBERT	InferSent	USE
Average	19.07	13.70	12.51	26.98	27.83	23.54
N-DGEA	22.72	21.22	17.39	28.93	28.72	26.99
K-DGEA	**17.54**	**12.65**	12.16	**26.35**	**24.71**	**22.74**
R-DGEA	17.83	13.41	**11.09**	26.89	26.57	23.42

Table 3: *Author identification* - Coverage-error on the author identification task (the lower the better). The coverage error computes the worst rank (as a percent) of the nearest neighbor that is a true author of the document. ATM, Usr2Vec, Aut2Vec and DAR are missing as they do not model document level representations. R-DGEA with USE document embedding performs best, with 11.09% on NYT, K_DGEA with USE on S2G with 22.74%. Adding the temporal information improves the performance in author identification.

ATM		40.78	
Aut2Vec		30.36	
Usr2Vec		23.22	
DAR (static)		**21.49**	
DAR (dynamic)		42.25	
DAR (concat)		33.17	
Method	SBERT	InferSent	USE
N-DGEA	31.39	33.82	30.69
K_DGEA	24.91	26.97	23.96
R-DGEA	22.67	22.89	21.55

Table 4: *Author link prediction* - Coverage-error (the lower the better) for author link prediction on the S2G dataset, i.e. the percent of nearest neighbors that we need to keep to cover all the true author's co-authors.

.

3 EVALUATION

3.1 Datasets

We use two datasets (S2G and NYT) written in English, prepared by Delasalles et al. [17]. S2G is a corpus of machine learning articles from Ammar et al. [2]. The articles were published between 1985 and 2017. Each article is associated with its authors, venue, and year of publication. The textual content is the title only. It has 45, 496 documents and 1117 authors. NYT is a set of news articles from the New York Times written between 1990 and 2015, gathered by [46]. It is composed of the article's title, its authors, and topics (e.g., sport, art, business). It has 41, 249 documents and 542 authors. We can only access the year of each document. We therefore use that information as the unit of time bins.

3.2 Evaluated methods

We compare our approach, DGEA (Dynamic Gaussian Embedding for Authors)[1] against the most recent baselines. We compare our method to three static author embedding methods, ATM [41], the

[1] https://github.com/AntoineGourru/DGEA

Train/Test ratio	10%	30%	50%	70%
ATM	23.7 (4.5)	31.7 (1.7)	34.4 (1.3)	35.5 (2.5)
Aut2Vec	27.2 (2.6)	32.3 (2.1)	34.5 (1.6)	36.2 (2.21)
Usr2Vec	34.7 (4.1)	41.7 (2.8)	43.8 (2.4)	45.4 (2.4)
DAR (static)	34.9 (3.1)	42.9 (2.2)	43.3 (1.7)	45.21 (2.5)
DAR (dynamic)	18.6 (2.3)	24.9 (2.7)	27.1 (3.0)	29.6 (2.7)
DAR (concat)	34.5 (2.1)	43.6 (1.4)	44.1 (1.5)	47.4 (2.4)
N-DGEA_I	34.0 (2.4)	41.6 (1.1)	44.9 (2.2)	46.7 (2.2)
K-DGEA_I	41.7 (3.0)	51.3 (1.9)	53.1 (2.3)	**56.0** (3.3)
R-DGEA_I	**43.7** (2.8)	50.8 (1.9)	53.2 (1.8)	53.2 (2.5)
N-DGEA_S	29.3 (2.4)	34.8 (2.2)	36.5 (2.6)	37.2 (2.5)
K-DGEA_S	37.2 (3.0)	42.1 (2.6)	44.8 (2.0)	45.3 (3.3)
R-DGEA_S	39.9 (2.2)	46.6 (2.1)	49.5 (2.5)	51.4 (2.7)
N-DGEA_U	34.1 (2.2)	41.3 (2.5)	45.1 (2.4)	46.8 (2.3)
K-DGEA_U	42.3 (2.5)	50.8 (2.7)	**54.2** (2.4)	54.9 (3.4)
R-DGEA_U	42.8 (2.1)	**51.0** (1.2)	52.4 (3.0)	53.9 (4.0)

Table 5: *Author classification (NYT)* - We can associate each author to a class. We then perform classification (using the author representation as features). We display the Micro-F1 for different train/test ratios and provide the standard deviation in parentheses. Our approaches outperform all the existing methods.

Train/Test ratio	10%	30%	50%	70%
ATM	31.4 (0.9)	32.4 (1.7)	33.5 (1.6)	33.5 (2.8)
Aut2Vec	23.2 (1.4)	24.0 (1.3)	24.6 (1.9)	24.9 (2.0)
Usr2Vec	34.6 (1.2)	36.9 (1.7)	36.9 (1.4)	36.1 (2.3)
DAR (static)	36.4 (1.3)	38.8 (1.1)	40.4 (1.1)	40.3 (2.3)
DAR (dynamic)	31.0 (1.1)	32.4 (0.7)	33.2 (1.5)	32.1 (2.3)
DAR (concat)	35.6 (1.5)	38.0 (0.8)	39.8 (1.5)	40.2 (1.7)
N-DGEA_I	30.9 (1.4)	34.5 (1.3)	35.2 (1.3)	35.6 (1.6)
K-DGEA_I	34.5 (2.1)	39.0 (0.8)	40.5 (1.2)	40.7 (1.5)
R-DGEA_I	**37.4** (1.1)	**40.3** (1.6)	**42.7** (1.9)	**42.6** (2.4)
N-DGEA_S	26.1 (1.6)	26.6 (0.8)	27.5 (1.4)	27.6 (1.8)
K-DGEA_S	28.1 (1.7)	30.4 (0.8)	30.9 (1.9)	31.3 (1.9)
R-DGEA_S	29.4 (0.9)	33.0 (0.9)	33.7 (1.5)	35.8 (1.6)
N-DGEA_U	27.2 (2.4)	30.5 (1.1)	31.4 (1.4)	32.5 (1.6)
K-DGEA_U	31.9 (2.1)	36.3 (1.3)	37.2 (1.3)	38.6 (2.6)
R-DGEA_U	36.4 (1.3)	38.8 (1.1)	39.7 (1.6)	40.6 (2.8)

Table 6: *Author classification (S2G)* - We can associate each author to a class. We then perform classification (using the author representation as features). We display the Micro-F1 for different train/test ratios and provide the standard deviation in parentheses. Our approaches outperform all the existing methods.

"Content-Info" model of Aut2Vec [20] and Usr2Vec [1]. Eventually, we compare our approach to DAR [17], which produces low-dimensional representations of author at each time step. We use the static representation, DAR (static), the dynamic representation, DAR (dynamic), and a concatenation of both, DAR (concat).

One could challenge that these empirical observations may be biased depending on the used text encoder. For that reason, we evaluate three versions of our model, using SBERT [39], USE [13] and InferSent [16] as encoder. N-DGEA is the simpler setup in which all time bins are considered independent. The solution is therefore the empirical means and variances of the document written by the author at each time step. This naive approach does not use the temporal information. It allows to demonstrate the benefit of smoothing author representations through time. K-DGEA is the model optimized using Kalman Equations described in Subsection 2.2. R-DGEA is the model that uses an RNN, described in Subsection 2.3. The subscript _U refers to USE, _I to InferSent and _S to SBERT. We did not use the variance in the evaluation to be fair with approaches that use point estimation.

3.3 Parameters setting

For SBERT [39], we use the 'bert-base-nli-mean-tokens' model. For InferSent we use the Glove based pre-trained model[2] and the 300 dimensions pre-trained Glove embeddings[3]. For USE, we use version 4, available on the Tensorflow Hub[4]. For ATM, we use Gensim[5]. We use the number of topics that maximizes the c_v coherence value [40], i.e., 201 for NYT and 229 for S2G. Ganguly et al. [20] initialize the document embedding layer using PV-DBOW [33]. It leads to poor performances in our experiments. We obtain the best results with Universal Sentence Embedding and when initializing the author layer with the mean of the representations of the documents they wrote. Embedding dimension is 512, we use 256 units in the intermediate layer, and optimize with Adam and a learning rate of 0.001. We draw 10 negative examples by observed pair. For Usr2Vec, we use a learning rate of 5e-5, a margin of 1, 25 epochs and patience of 5 (number of epochs without loss decreasing needed to stop training). We train the word embeddings using Skip gram with negative sampling [35] in dimension 300. We also remove rare words (appearing less than 5 times).

For the dynamic methods, we split the documents by year. For the DAR model [17], we use authors recommended parameters, as they evaluate their model on the same datasets. For K-DGEA, we use 100 epochs and our own implementation of the Kalman based EM algorithm. For R-DGEA, we use an LSTM with a mean pooling operation, as it produces the best results on these datasets. The hidden dimension is the same as the document embeddings, and we use the last hidden state as ϕ. $h_{\theta'}()$ is a 2-layer MLP with $relu()$ and linear activation. We use batch of 32 authors with Adam [29] and a learning rate of 1e-4. *

3.4 Evaluation tasks

We evaluate how close a document representation is to its author representation (the author identification task). Then, we use the standard evaluation tasks to evaluate author representation learning as used when evaluating embeddings: link prediction was used in [20] and author classification in [1]. We then provide visualization of K-DGEA authors representations and an analysis of K-DGEA and R-DGEA learned variances.

[2] https://github.com/facebookresearch/ [3] http://nlp.stanford.edu/

[4] https://tfhub.dev/google/ [5] https://radimrehurek.com/gensim/

4 RESULTS

4.1 Author identification

We use the documents from the last year of production for each author as the test set (T+1), and the previous documents as the training set. Since S2G features multiple authors, we use the coverage error. This metric is well suited for multi-label evaluation as it measures the average number of nearest neighbors that must be considered to cover all the true authors of a given document. We normalize the error by n_a, the total number of authors in the corpus (the worst-case scenario) and provide this result in percentage. The coverage error also provides a quantification of the average rank of true authors (unlike Hamming loss). We compute the nearest neighbors of a document using cosine similarity computed between the author representations at time T and those of the documents in the test set, i.e. at $T + 1$. Since existing author representation methods (ATM, Usr2Vec, Aut2Vec, DAR) do not model document-level representations, they can hardly compete in this task. We therefore compare our approach with several simple methods. We average the representations of documents written by an author to build its representation. This approach is equivalent to N-DGEA in the naive configuration for which all documents are in the same time slice.

We present the results in Table 3. K-DGEA and R-DGEA outperform the average of document representations without temporal smoothing. USE performs best on this task: the approach seems to be able to build document representations that better discriminate authors. The USE model was pre-trained on two additional tasks compared to SBERT and InferSent: a question-answer task and a co-occurring sentence prediction task. The latter could allow the model to capture information relevant to identify the author of a document, such as writing style. We observe that the average results are worst on S2G: we recall that the models are not fine-tuned on the dataset, and that S2G contains a more specialized vocabulary than NYT. Finally, even in the best scenario, one must consider 10% of the nearest neighbors to cover all the real authors of a document. On these datasets, the task remains difficult to solve (even in the supervised setting, c.f. Table 2): we recall that we consider the title of the document only, for datasets in which the language is formal and stereotyped: there is a journalistic and a scientific "style", which are consensual and shared by all authors.

4.2 Author classification

In this task, each document is associated with a class. We posit that the author's class is the most frequent class observed in their productions. For example, Geoffrey E. Hinton published most often at the NeurIPS conference, while Christopher D. Manning published mostly at ACL, and Jure Leskovec at TheWebConf (WWW). Here, we can construct an author's class at each time slice. We use the last year of publication for evaluation. We train the classifier on the representations of the last time slice for DGEA. We use the same classifier than in existing works [45]: an SVM with L2 norm regularization. The strength of the regularization is fixed, for each method, using grid-search. We report the Micro-F1 mean and standard deviation.

We present the results in Table 5 (Nyt) and Table 6 (S2G). The best DGEA method performs up to 10 points better than all baselines on the NYT dataset. On the S2G dataset, its performance is close to that of DAR. We also observe that the performances are globally better for the dynamic representation approaches: the datasets contain a rather long production period (about 30 years), and the authors' themes of interest change over time. This observation confirms the importance to learn evolving representations.

	EV	#topics	avg-#topic	#topics$_{t+1}$	avg-#docs	#docs
			NYT			
K-variance	**0.86**	0.15	0.18	0.06	0.02	0.09
K-δ	0.09	0.17	-0.12	-0.03	**-0.72**	-0.5
R-variance	0.20	-0.01	0.01	-0.01	-0.04	-0.10
EV	1	0.39	0.24	0.05	-0.16	0.11
			S2G			
K-variance	**0.85**	0.14	0.21	0.18	0.29	0.27
K-δ	0.12	-0.37	**-0.75**	-0.45	**-0.81**	-0.56
R-variance	0.41	-0.13	-0.22	-0.15	-0.13	-0.14
EV	1	0.02	-0.08	-0.04	0.02	0.20

Table 7: *Variances correlation* - Average correlation between the variances learned by our models and different measures computed on NYT and S2G.

4.3 Link prediction

We construct the co-author graph on the S2G dataset during each author's last year of publication. We compute how well the similarity between author representations can predicts the existence of an edge. Similar to author identification, we use the coverage error which is defined as the average proportion of nearest neighbors we need to compute to cover all co-authors of the author.

We present the results in Table 4. R-DGEA with USE equals the static version of DAR (with a difference of 0.06 points). Furthermore, each R-DGEA model outperforms all other baseline models, including the dynamic version of DAR. Finally, Usr2Vec and the static version of DAR produce overall relative better results than for the other tasks. Taking into account the authors' dynamics does not seem to be a prevalent feature for link prediction. This observation could mean that scientific collaborations are perennial in time and that they: either remain little sensitive to changes in research themes, or they are not independent of these changes, in the sense that the collaboration can itself modify the thematic trajectory of the authors.

4.4 Author variance analysis

Modeling the authors as Gaussian distributions has many advantages, such as those highlighted by Bojchevski and Günnemann [10] and in our previous works Gourru et al. [24]: it captures the uncertainty of the representation and could help to interpret the results in recommendation. It also allows to use the theoretical framework of Gaussian dynamic linear models, and thus the EM algorithm as well as forward-backward equations. We present in Table 7 the average correlation between the variances learned by our models and different measures of empirical dispersion computed on NYT and S2G and USE representations. These measures are the following. Empirical variance (denoted EV) is the empirical variance of the representations of the documents written by the author over her/his entire publication history. The number of topics,

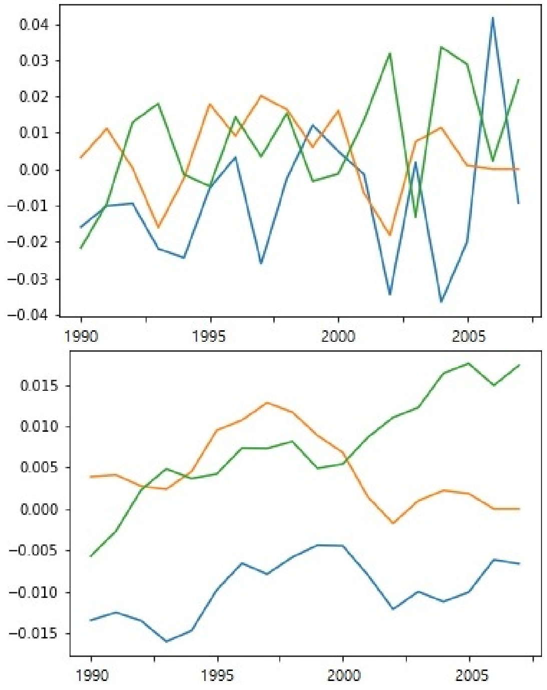

Figure 3: *Representation of 3 authors learned with N-DGEA and K-DGEA* - We plot the first axis for three authors embedding (color coded) with regard to time. We provide their representation from N-DGEA$_U$ (on top; without time smoothing), and K-DGEA$_U$ (on the bottom; with time smoothing). Using the temporal information allows to easily separate the author *trajectories.* This temporal representation smoothness can be observed for each author in the datasets.

noted #topics is the total number of different labels observed over the author's entire history. The average number of labels observed (noted avg-#topic) is obtained by averaging the number of different labels per time slice. The number of labels on the last year of production is noted #topics$_{t+1}$. The average number of documents written by an author per time slot is noted avg-#docs and the total number of documents written by an author is #docs.

The variance of an author learned with K-DGEA shows a strong correlation with the empirical variance of her/his document representations (0.86 for K-DGEA$_U$ on NYT, and 0.85 on S2G). The δ parameter (the temporal variance) seems to be negatively correlated with the average number of documents per time slot (-0.72 for K-DGEA$_U$ on NYT and -0.81 on S2G) and with the average number of topics (-0.75 on S2G): more publications and topics reduce the temporal uncertainty of an author, i.e. authors publishing a lot in many fields evolve less over time.

4.5 Comparison between DGEA settings

K-DGEA and R-DGEA consistently outperform N-DGEA: independent construction of each time slice representations is generally 10 points worse on our evaluation tasks. Thus, temporal smoothing clearly improves the performance of author representations to solve the evaluated tasks. R-DGEA performs better than K-DGEA in classification on S2G, and in link prediction. Nevertheless, K-DGEA produces better results on the S2G dataset on the author identification task and on NYT in author classification. On average, USE vector representations seem to produce better results. The results of both models remain comparable with similar document representation methods while our models have the ability to represent authors as Gaussian distribution in the same space than documents which is not available for the literature – not only do they perform better, but they also allow to address new author-based tasks such as author identification (cf subsection 4.1).

4.6 Visualization

We provide the visualization of the *trajectory* of three authors in Figure (3). We randomly selected one author, and then randomly drew two other authors from among the authors who have published as many years as the first author. On the y axis, we plot the value on the first axis of the author's representation (more precisely its mean), on the x axis the time. On the top, we show the K-DGEA_U result for 3 authors on the NYT dataset and on the bottom, the N-DGEA_U version (without temporal smoothing). The trajectories formed by the authors' dynamic representations are easy to separate, while they are strongly intertwined without temporal smoothing, reinforcing the interest of our approach.

5 CONCLUSION

We proposed the DGEA framework, that allows to learn dynamic representations of authors from any pre-trained document embeddings. We proposed two instances of this framework: K-DGEA, based on a first order Markov model, and R-DGEA a deep model with a LSTM layer. We tested these models on three tasks involving author representations (author identification, author classification, and link prediction). The two models perform similarly in our experiments. The main difference is that K-DGEA is faster and parallelizable, whereas R-DGEA can infer representations for unseen authors. The user can choose both depending on its computing resources and targeted tasks, while knowing that the performance will be equivalent. The temporal information allows to improve author identification, co-authorship prediction and author classification.

Several tracks and limitations could be studied in the future. First, we model the authors as Gaussians distributions, contrary to existing approaches. Nevertheless, even this assumption may seem limited: authors sometimes publish in distant topics. Modeling them as a mixture of Gaussians could benefit the model and reduce the number of parameters. Additionally, the document encoders integrate complex distance measures in their objective function (scalar product, Euclidean distance and concatenation of these measures). We could test other probability laws more suited to these metrics: this change would be simpler for R-DGEA (when the likelihood is well determined) than for K-DGEA: particle filters [6] allow to optimize linear models with non-Gaussian emission laws, but they are more expensive in terms of computation. Finally, the variance information could be used to improve tasks such as expert identification, by penalizing authors with large variance, but the latter is yet to be tested.

REFERENCES

[1] Silvio Amir, Glen Coppersmith, Paula Carvalho, Mario J Silva, and Byron C Wallace. 2017. Quantifying mental health from social media with neural user embeddings. *arXiv preprint arXiv:1705.00335* (2017).

[2] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. 2018. 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)*. 84–91.

[3] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017*.

[4] Robert Bamler and Stephan Mandt. 2017. Dynamic word embeddings. In *International conference on Machine learning*. PMLR, 380–389.

[5] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv:2004.05150* (2020).

[6] CM Bishop. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics), chapter 13.

[7] Christopher M Bishop. 1994. Mixture density networks. (1994).

[8] David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*. 113–120.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[10] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *Proceeding of the International Conference on Learning Representations (ICLR)*.

[11] Robin Brochier, Antoine Gourru, Adrien Guille, and Julien Velcin. 2020. New Datasets and a Benchmark of Document Network Embedding Methods for Scientific Expert Finding. In *10th International Workshop on Bibliometric-enhanced Information Retrieval co-located with 42nd European Conference on Information Retrieval*.

[12] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* 33 (2020).

[13] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 169–174.

[14] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*.

[15] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research* 12, ARTICLE (2011), 2493–2537.

[16] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 670–680.

[17] Edouard Delasalles, Sylvain Lamprier, and Ludovic Denoyer. 2019. Learning Dynamic Author Representations with Temporal Language Models. In *2019 IEEE International Conference on Data Mining (ICDM)*. 120–129.

[18] 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)*. 4171–4186.

[19] Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2019. The Dynamic Embedded Topic Model. *arXiv preprint arXiv:1907.05545* (2019).

[20] 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*. International World Wide Web Conferences Steering Committee, 49–50.

[21] Soumyajit Ganguly and Vikram Pudi. 2017. Paper2vec: Combining graph and text information for scientific paper representation. In *European conference on information retrieval*. Springer, 383–395.

[22] Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. 2007. Euclidean Embedding of Co-occurrence Data. *Journal of Machine Learning Research* 8 (2007), 2265–2295.

[23] Antoine Gourru, Adrien Guille, Julien Velcin, and Julien Jacques. 2020. Document Network Projection in Pretrained Word Embedding Space. In *European Conference on Information Retrieval*. Springer, 150–157.

[24] Antoine Gourru, Julien Velcin, and Julien Jacques. 2020. Gaussian Embedding of Linked Documents from a Pretrained Semantic Space. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*.

[25] Sepp Hochreiter, Jürgen Schmidhuber, and Corso Elvezia. 1997. LONG SHORT-TERM MEMORY. *Neural Computation* 9, 8 (1997), 1735–1780.

[26] 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)*. 1681–1691.

[27] Geng Ji, Robert Bamler, Erik B Sudderth, and Stephan Mandt. 2017. Bayesian paragraph vectors. *Symposium on Advances in Approximate Bayesian Inference* (2017).

[28] Rudolph Emil Kalman. 1960. A new approach to linear filtering and prediction problems. (1960).

[29] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

[30] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. *Proceedings of the International Conference on Learning Representations (ICLR)* (2014).

[31] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*. 3294–3302.

[32] Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query expansion using word embeddings. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. ACM, 1929–1932.

[33] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. 1188–1196.

[34] Zaiqiao Meng, Shangsong Liang, Hongyan Bao, and Xiangliang Zhang. 2019. Co-embedding attributed networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 393–401.

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

[36] Kevin P Murphy. 2007. Conjugate Bayesian analysis of the Gaussian distribution. *def* 1, $2\sigma 2$ (2007), 16.

[37] Shimei Pan and Tao Ding. 2019. Social media-based user embedding: A literature review. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*.

[38] Herbert E Rauch, F Tung, and Charlotte T Striebel. 1965. Maximum likelihood estimates of linear dynamic systems. *AIAA journal* 3, 8 (1965), 1445–1450.

[39] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the International Conference on Empirical Methods in Natural Language Processing* (2019).

[40] Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*. 399–408.

[41] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. 487–494.

[42] Maja Rudolph and David Blei. 2018. Dynamic Embeddings for Language Evolution. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1003–1011.

[43] Purnamrita Sarkar, Sajid M Siddiqi, and Geoffrey J Gordon. 2006. Approximate Kalman filters for embedding author-word co-occurrence data over time. In *ICML Workshop on Statistical Network Analysis*. Springer, 126–139.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[45] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Network representation learning with rich text information. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

[46] Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the eleventh acm international conference on web search and data mining*. 673–681.

	NYT			S2G		
Method	SBERT	InferSent	USE	SBERT	InferSent	USE
Mean	17.6	13.2	11.3	26.6	25.7	**23.0**
LSTM	32.3	15.2	11.7	47.6	31.0	27.3
Single	17.6	13.1	11.1	26.9	25.7	23.2
R-DGEA	17.8	13.4	**11.1**	26.9	26.6	23.4

Table 8: *Author Identification (with R-DGEA alternatives)* - Coverage error (the lower the better) on the author identification task for different alternatives of the R-DGEA model. The coverage error calculates the worst rank (in percent) of the nearest neighbor who is a real author of the document.

Method	SBERT	InferSent	USE
Mean	22.55	22.77	22.24
LSTM	29.24	28.52	23.13
Single	22.75	22.52	22.33
R-DGEA	22.67	22.89	**21.55**

Table 9: *Link prediction (with R-DGEA alternatives)* - Coverage error (to be minimized) for the prediction of links between authors on the S2G dataset for different alternatives of the R-DGEA models, i.e., the percentage of nearest neighbors we need to keep in order to cover all true co-authors of an author.

Train/Test ratio	10%	30%	50%	70%
Mean (I)	44.0 (3.5)	51.0 (1.5)	52.9 (1.5)	53.3 (3.2)
LSTM (I)	40.1 (3.1)	47.8 (1.6)	48.2 (2.48)	48.7 (1.5)
Single (I)	43.5 (3.3)	50.2 (1.5)	52.7 (1.7)	53.5 (3.7)
R-DGEA (I)	43.7 (2.8)	50.8 (1.9)	**53.2** (1.8)	53.2 (2.5)
Mean (S)	41.0 (2.1)	46.5 (1.8)	48.1 (2.8)	48.5 (2.9)
LSTM (S)	33.7 (2.4)	40.4 (1.5)	42.0 (2.9)	44.4 (2.7)
Single (S)	39.2 (3.0)	46.7 (1.84)	48.0 (2.5)	50.3 (2.7)
R-DGEA (S)	39.9 (2.2)	46.6 (2.1)	49.5 (2.5)	51.3 (2.7)
Mean (U)	**44.6** (2.7)	49.3 (1.7)	52.0 (2.2)	**54.4** (2.3)
LSTM (U)	43.5 (2.9)	49.8 (1.5)	52.2 (1.9)	52.8 (3.6)
Single (U)	44.3 (1.6)	49.9 (1.7)	51.3 (1.6)	53.2 (1.7)
R-DGEA (U)	42.8 (2.1)	**51.0** (1.2)	52.4 (2.9)	53.9 (4.0)

Table 10: *Author classification (with R-DGEA alternatives) on NYT* - Results in author classification on the NYT dataset. We display the Micro-F1 score for different train/test ratios and provide the standard deviation in parentheses.

A ADDITIONAL EXPERIMENTS

The R-DGEA model learns a variance per time slice using a DAN-type network [26]. We assess the appropriateness of this modeling choice by evaluating three alternatives to the original R-DGEA model. We start by removing the variance and train the network using a simple mean square error objective function (equivalent to setting the diagonal of each author's variance to one). We call this model "Mean". Next, we use an LSTM to which we feed the sequences of empirical variances instead of the DAN-like approach.

Train/Test ratio	10%	30%	50%	70%
Mean (I)	34.6 (1.3)	38.2 (2.1)	39.7 (1.5)	39.0 (1.3)
LSTM (I)	31.1 (1.7)	34.5 (1.4)	36.8 (1.7)	37.3 (1.5)
Single (I)	36.1 (1.2)	36.8 (1.5)	38.5 (1.5)	38.8 (0.9)
R-DGEA (I)	**37.4** (1.1)	**40.3** (1.6)	**42.7** (1.9)	**42.6** (2.4)
Mean (S)	32.6 (1.5)	36.2 (1.2)	36.4 (1.2)	37.8 (2.5)
LSTM (S)	26.4 (1.6)	28.9 (1.1)	30.3 (1.6)	30.7 (1.8)
Single (S)	28.9 (1.5)	30.3 (1.2)	31.8 (1.6)	33.2 (2.6)
R-DGEA (S)	29.4 (0.9)	33.0 (0.9)	33.7 (1.5)	35.8 (1.6)
Mean (U)	36.3 (1.4)	37.9 (1.0)	38.8 (1.5)	39.4 (1.3)
LSTM (U)	31.2 (2.0)	33.8 (1.4)	35.0 (2.1)	35.6 (2.1)
Single (U)	34.7 (1.0)	37.8 (0.6)	38.8 (1.3)	39.3 (1.5)
R-DGEA (U)	36.4 (1.3)	38.8 (1.1)	39.7 (1.6)	40.6 (2.8)

Table 11: *Author classification (with R-DGEA alternatives) on S2G* - Results in author classification on the S2G dataset. We display the Micro-F1 score for different train/test ratios and provide the standard deviation in parentheses.

We add to the output of the LSTM a two-layer MLP, with tanh and relu activation. This is equivalent to using:

$$\sigma^2_{i,t} = h_{\theta'}(g'(\mathcal{D}_i^{(0)}), ..., g'(\mathcal{D}_i^{(t-1)})) \tag{13}$$

,where $h_{\theta'}$ is an LSTM and $g'(\mathcal{D}_i^{(t)})$ computes the empirical variance of $\mathcal{D}_i^{(t)}$. We call this method "LSTM". In the third variant, we learn the authors' variances as parameters (i.e., a layer of vector representations). We call this method "Single".

We train the model with a batch size of 32 authors with Adam [29] and a learning rate of 1e-4, except for the "LSTM" model which we optimize with a learning rate of 1e-3. We present the results on NYT and S2G in author identification in Table 8, link prediction in Table 9, and author classification in Table 10 and Table 11.

R-DGEA with USE performs better in author identification, while the "Mean" model with USE performs better than the other models on S2G. R-DGEA with USE produces the best results in link prediction.

It is interesting to note that the LSTM variance model fails to produce good results in author identification with SBERT representations. The "Mean" model generally produces the best results in author classification on NYT, while R-DGEA outperforms the competitors on S2G.

Nevertheless, there is no significant difference between all models. As the different models have equivalent performances, we have chosen to keep the R-DGEA model presented previously for several interesting properties when applying the model to real life situations.

Indeed, the original model has several advantages over the other instances: it provides a measure of semantic uncertainty unlike mean-only learning, it has fewer parameters than LSTM-based variance learning, and it allows to infer representations of authors that has not been seen during the learning phase, unlike learning a single variance as a parameter.

B DETAILS FOR K-DGEA

We recall that the aim is to maximize :

$$\begin{aligned}\mathcal{L}(\mathcal{D};\psi) = \prod_{i}^{N} & \underbrace{\prod_{t=1}^{T} C_t \omega(\sigma_i^2, \mathcal{D}_i^{(t)})}_{\text{Normalizing factor}} \\ & \times \underbrace{\int p(\phi_{i,1:T}) \prod_{t=1}^{T} \mathcal{N}(\bar{d}_{i,t}; \phi_{i,t}, r_{i,t}^2 I) d_{\phi_{i,1:T}}}_{\text{Simple dynamic linear system}} .\end{aligned} \tag{14}$$

We use an EM model with forward and backward recursions.

In the E step, we evaluate the probability of the hidden states $\phi_{i,t}$ for $1 \leq i \leq N$ and $1 \leq t \leq T$ using the forward-backward equations, and in the M step, we maximize the expectation of the completed log-likelihood. We obtain, for the E forward step:

$$\begin{aligned} m_{i,t} &= m_{i,t-1} + \frac{v_{i,t-1} + \delta_i^2}{v_{i,t-1} + \delta_i^2 + r_{i,t}^2} \left(\bar{d}_{i,t} - m_{i,t-1}\right) \\ v_{i,t} &= \frac{(v_{i,t-1} + \delta_i^2) r_{i,t}^2}{v_{i,t-1} + \delta_i^2 + r_{i,t}^2} \end{aligned} \tag{15}$$

with the initial conditions:

$$m_{i,1} = \phi_{i,0} + \frac{\delta_{i,0}^2 \left(\bar{d}_{i,1} - \phi_{i,0}\right)}{\delta_{i,0}^2 + r_{i,1}^2} \quad \text{and} \quad v_{i,1} = (\frac{\delta_{i,0}^2 r_{i,1}^2}{\delta_{i,0}^2 + r_{i,1}^2}). \tag{16}$$

The backward E step is:

$$\begin{aligned} \widehat{m}_{i,t} &= m_{i,t} + \frac{v_{i,t}}{v_{i,t} + \delta_i^2} \left(\widehat{m}_{i,t+1} - m_{i,t}\right) \\ \widehat{v}_{i,t} &= v_{i,t} + \left(\frac{v_{i,t}}{v_{i,t} + \delta_i^2}\right)^2 \left(\widehat{v}_{i,t+1} - v_{i,t} - \delta_i^2\right) \end{aligned} \tag{17}$$

with $\widehat{m}_{i,T} = m_{i,T}$ and $\widehat{v}_{i,T} = v_{i,T}$. In the M step, we maximize :

$$\begin{aligned} \mathbb{E}_p[\log \mathcal{L}_i(\mathcal{D}_i, \phi_{i,1:T})] = & \sum_{t=1}^{T} \log \omega(\sigma_i^2, \mathcal{D}_i^{(t)}) \\ & + \mathbb{E}_p[\log p(\phi_{i,1} | \phi_{i,0}, \delta_{i,0}^2)] \\ & + \mathbb{E}_p[\sum_{t=2}^{t} \log p(\phi_{i,t} | \phi_{i,t-1}, \delta_i^2 I) \\ & + \sum_{t=1}^{t} \log p(\bar{d}_{i,t} | \phi_{i,t}, r_{i,t}^2)] \end{aligned} \tag{18}$$

with regard to the parameters (cf [6]), with:

$$\begin{aligned} \mathbb{E}\left[\phi_{i,t}\right] &= \widehat{m}_{i,t} \\ \mathbb{E}\left[\phi_{i,t}\phi_{i,t-1}^\top\right] &= \frac{v_{i,t-1}\widehat{v}_{i,t}}{v_{i,t-1} + \delta_i^2} + \widehat{m}_{i,t}\widehat{m}_{i,t-1}^\top \\ \mathbb{E}\left[\phi_{i,t}\phi_{i,t}^\top\right] &= \widehat{v}_{i,t} + \widehat{m}_{i,t}\widehat{m}_{i,t}^\top \end{aligned} \tag{19}$$

C PRODUCT OF DIAGONAL GAUSSIANS

Theorem C.1. *The product of Multivariate Gaussians Densities over independent vectors, with similar mean ϕ and diagonal variance $\sigma^2 I$, is a scaled Gaussian Density over the empirical mean of the vectors, with mean ϕ and diagonal variance $\frac{\sigma^2}{n} I$, where n is the number of random variables.*

To prove that results, we follow [36]. Let us define $x = \{x_i\}_{i=1}^n$, $x_i, \phi, \sigma^2 \in \mathbb{R}^r$.

$$\begin{aligned} \prod_i \mathcal{N}(x_i; \phi, \sigma^2 I) = & \frac{1}{(\sqrt{2\pi^r \prod_r \sigma_r^2})^n} \\ & \cdot exp\left(\sum_i -\frac{1}{2} \sum_r \frac{(x_{i,r} - \phi_r)^2}{\sigma_r^2}\right) \\ = & \frac{1}{(\sqrt{2\pi^r \prod_r \sigma_r^2})^n} \\ & \cdot \exp\left(-\frac{1}{2} \sum_r \frac{1}{\sigma_r^2} \sum_i (x_{i,r}^2 - 2x_{i,r}\phi_r + \phi_r^2)\right) \\ = & \frac{1}{(\sqrt{2\pi^r \prod_r \sigma_r^2})^n} \\ & \cdot \exp\left(-\frac{1}{2} \sum_r \frac{n}{\sigma_r^2} (s_r^2 + \bar{x}_r^2 - 2\bar{x}_r\phi_r + \phi_r^2)\right) \end{aligned} \tag{20}$$

With

$$\bar{x} = \frac{1}{n} \sum_i x_i \qquad s_r^2 = \frac{1}{n} \sum_i (x_{i,r} - \bar{x}_r)^2 \tag{21}$$

Then

$$\begin{aligned} \prod_i \mathcal{N}(x_i; \phi, \sigma^2 I) = & \frac{1}{(\sqrt{2\pi^r \prod_r \sigma_r^2})^n} \exp\left(-\frac{1}{2} \sum_r \frac{n s_r^2}{\sigma_r^2}\right) \\ & \cdot \exp\left(-\frac{1}{2} \sum_r \frac{n}{\sigma_r^2} (\bar{x}_r - \phi_r)^2\right) \\ = & \frac{\exp\left(-\frac{1}{2} \sum_r \frac{n s_r^2}{\sigma_r^2}\right)}{(\sqrt{2\pi^r \prod_r \sigma_r^2})^{n-1}} \cdot \frac{1}{\sqrt{2\pi^r \prod_r \sigma_r^2}} \\ & \cdot \exp\left(-\frac{1}{2} \sum_r \frac{n}{\sigma_r^2} (\bar{x}_r - \phi_r)^2\right) \\ = & \frac{\exp\left(-\frac{1}{2} \sum_r \frac{n s_r^2}{\sigma_r^2}\right)}{(\sqrt{2\pi^r \prod_r \sigma_r^2})^{n-1} n^{\frac{r}{2}}} \cdot \frac{1}{\sqrt{2\pi^r \prod_r \frac{\sigma_r^2}{n}}} \\ & \cdot \exp\left(-\frac{1}{2} \sum_r \frac{n}{\sigma_r^2} (\bar{x}_r - \phi_r)^2\right) \\ = & \frac{\exp\left(-\frac{1}{2} \sum_r \frac{n s_r^2}{\sigma_r^2}\right)}{(\sqrt{2\pi^r \prod_r \sigma_r^2})^{n-1} n^{\frac{r}{2}}} \cdot \mathcal{N}(\bar{x}; \phi, \frac{\sigma^2}{n} I) \end{aligned} \tag{22}$$