Apprentissage Conjoint de Représentations d'Auteurs et de Documents

Antoine Gourru*, Rohit Yadav**,*, Julien Velcin*

*ERIC UR3083, Université de Lyon, Lyon 2, France
{antoine.gourru,julien.velcin}@univ-lyon2.fr,
**Université Jean Monnet, Saint-Etienne, France
rohit.yadav@etu.univ-st-etienne.fr

Résumé. Les modèles de langue les plus récents utilisent des représentations de mots contextualisés à l'aide de Transformers. Ils ont rapidement dépassé les méthodes état de l'art dans de nombreuses tâches de traitement automatique de la langue. Des versions pré-entraînées de ces modèles sont largement utilisées, mais leur spécialisation pour résoudre une tâche spécifique reste une question centrale. Par exemple, ces méthodes ne produisent pas de représentation à l'échelle du document et de l'auteur, mais seulement du mot. Or comme le montrent Reimers et Gurevych (2019), une simple moyenne des plongements de mots ne suffit pas. En utilisant une approche dite du Variational Information Bottleneck, nous développons une architecture simple pour construire des représentations d'auteurs et de documents à partir de modèles pré-entraînés (Devlin et al., 2019). Nous évaluerons de manière quantitative et qualitative notre modèle sur deux jeux de données : un corpus d'articles scientifiques et un d'articles de presse. Notre modèle produit des représentations plus robustes que l'existant, et donne des résultats compétitifs en classification et en identification d'auteurs.

1 Introduction

Depuis les travaux de Mikolov et al. (2013), de nombreux modèles incorporent des représentations continues des mots, appelées plongements, pour résoudre des tâches de traitement automatique de la langue, par exemple de recherche d'information (Kuzi et al., 2016), génération (Brown et al., 2020), et classification (Yang et al., 2015). Grâce aux mécanismes d'attention (Bahdanau et al., 2015; Luong et al., 2015), ces vecteurs sont contextualisés, i.e. un mot possède une représentation dépendante de sa position dans la phrase ainsi que de son contexte. Une nouvelle famille de modèles de langue (Devlin et al., 2019; Brown et al., 2020) basés sur les Transformers (Vaswani et al., 2017), qui incorporent ces mécanismes, a révolutionné le domaine en dépassant l'état de l'art sur un très grand nombre de tâches de traitement automatique des langues (traduction, génération, systèmes de questions-réponses).

De nombreuses problématiques peuvent bénéficier d'une représentation conjointe des documents et des auteurs, comme la recommandation automatique ou l'identification d'auteur. Ces représentations doivent ainsi vivre dans le même espace de façon à pouvoir calculer des

proximités et construire des voisinages. Malheureusement, il n'existe pas de méthode universelle pour apprendre une représentation continue des documents et des auteurs dans un même espace à partir de vecteurs de mots : une simple moyenne des vecteurs, contextualisés ou non, semble fonctionner assez mal (Le et Mikolov, 2014; Reimers et Gurevych, 2019). On peut ajouter qu'une représentation d'un document ou d'un auteur comme un point dans un espace vectoriel n'est pas suffisante. En effet, de nombreux auteurs et documents sont associés à un contenu sémantique complexe. Certaines méthodes de représentation de documents proposent donc d'apprendre une mesure d'incertitude en plus de ces vecteurs en utilisant des auto-encodeurs variationnels (Meng et al., 2019) ou des modèles probabilistes (Ji et al., 2017; Gourru et al., 2020). Malheureusement, aucune méthode n'intègre cette mesure dans l'apprentissage de représentations des auteurs.

Dans cet article, nous proposons une nouvelle approche pour apprendre des représentations continues dans un espace de faible dimension des auteurs et des documents d'un corpus. Cette approche, basée sur la méthode du Variational Information Bottleneck (VIB) (Alemi et al., 2017), utilise des matrices de vecteurs de mots comme observation, et permet donc de réaliser un transfert de connaissance à partir de modèles de type BERT (Devlin et al., 2019; Liu et al., 2019) qui bénéficient d'un apprentissage sur des corpus volumineux. En plus de représenter les auteurs comme un vecteur, notre méthode leur associe une mesure de variance, et donc d'incertitude sémantique. Nous allons d'abord présenter les travaux existants en apprentissage de représentation pour les documents, puis pour les auteurs, dans la section 2. Après une présentation du cadre théorique VIB utilisé, nous proposons un modèle original appelé PAD en section 4. Nous évaluons notre modèle sur des tâches de classification et d'identification d'auteur en section 5 avant de conclure.

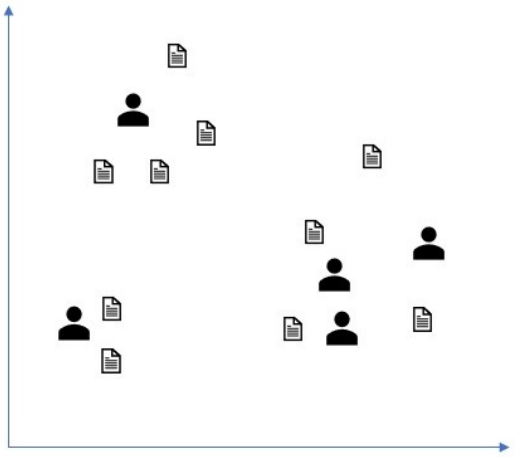

FIG. 1 – *Apprendre à représenter les auteurs et les documents dans le même espace permet de réaliser un grand nombre de tâches : clustering, recommandation, identification de l'auteur d'un document.*

2 Travaux existants

Nous allons présenter ici les différentes approches de représentation des documents et des auteurs. L'objectif général est d'apprendre, pour chaque auteur et chaque document, un vecteur dans un espace $\mathbb{R}^r$ tel que les mesures de similarité entre objets dans cet espace révèlent une forme de proximité sémantique. Les représentations apprises servent ensuite d'intermédiaires pour résoudre des tâches en aval, telles que la prédiction de lien entre auteurs ou entre documents, la classification ou le clustering.

2.1 Représentation des documents

L'approche la plus simple pour représenter un document est le sac de mots. Un document est un vecteur dans l'espace du vocabulaire, où l'entrée correspondant au mot est non nulle si le mot est présent dans le document. De nombreuses méthodes de normalisation et de pondération de ces vecteurs ont été proposées, comme TF-IDF ou Okapi Bm25. Ces représentations ont le désavantage d'être extrêmement creuses et en très grande dimension. De plus, la similarité entre deux documents utilisant des synonymes est faible, malgré un thème possible en commun.

L'objectif de l'apprentissage de représentation est donc double : i) compresser l'information en réduisant la dimension et en densifiant les vecteurs, ii) identifier un espace sémantique dans lequel deux documents évoquant les mêmes thématiques seront proches.

Différents travaux permettent de représenter les documents dans un espace sémantique latent, telle l'allocation de Dirichlet Latente de Blei et al. (2003) qui construit, au niveau de chaque document, un vecteur probabiliste sur un ensemble de thématiques. Les thématiques sont des distributions de probabilité sur le vocabulaire de mots.

Plus récemment, les travaux de Mikolov et al. (2013) ont donné un nouveau souffle à l'apprentissage de représentation de mots. Dans ce cadre, les plongements (*embedding*) de mots sont appris en résolvant une tâche de classification de paires positives de vraies cooccurrences et des paires négatives tirées aléatoirement. Les modèles Doc2Vec (Le et Mikolov, 2014) étendent cette approche au niveau du document.

Les méthodes récentes s'attellent à construire des plongements contextualisés des mots (Devlin et al., 2019). Le plongement est modifié en fonction du contexte dans lequel le mot est observé, et en fonction de sa position dans la phrase. Ces approches reposent sur le principe du mécanisme d'attention, proposé à l'origine pour résoudre des tâches de traduction (Luong et al., 2015; Bahdanau et al., 2015). Les travaux de Reimers et Gurevych (2019) proposent d'affiner (*fine-tuner*) un modèle BERT sur une tache de classification de paires de documents (*Triplet Loss*).

Une autre famille d'approches se concentre sur l'apprentissage de documents en réseau : la prise en compte du voisinage permet d'améliorer les plongements appris sur le texte seul (Yang et al., 2015; Gourru et al., 2020).

2.2 Représentation des Auteurs

Dans le modèle Author Topic Model (ATM) (Rosen-Zvi et al., 2004), les auteurs sont représentés dans un espace de thématiques, similairement à Blei et al. (2003). ATM est un modèle hiérarchique probabiliste, optimisé par échantillonnage de Gibbs.

L'approche proposée par Ganguly et al. (2016), Aut2vec (A2V), reprend le principe de Mikolov et al. (2013). Les plongements d'auteurs et de documents sont appris de façon à bien séparer des exemples auteur/document positifs et négatifs. Une paire positive correspond à un auteur et un des documents qu'il a écrits. Les exemples négatifs sont tirés aléatoirement. La distance entre les vecteurs modifie la valeur d'une fonction d'activation qui produit une probabilité que le lien soit observé. Le modèle prend la forme d'un réseau de neurones.

Peu de méthodes permettent d'apprendre des représentations d'auteurs et de documents dans un même espace. Aucune ne semble utiliser des représentations pré-apprises comme BERT (Devlin et al., 2019). De plus, il n'existe pas de méthode qui représente un auteur autrement que par un point. Nous proposons donc dans cet article un modèle pour répondre à ces trois problématiques. Il est basé sur l'apprentissage par Variational Information Bottleneck que nous allons présenter dans la section suivante.

3 PAD : Plongement d'Auteurs et de Documents

Nous présentons dans cette section notre modèle, après une introduction au principe du Variational Information Bottleneck (VIB).

3.1 Variational Information Bottleneck

L'apprentissage VIB Tishby et al. (1999) consiste à apprendre une représentation z qui maximise la compression des observations initiales x, tout en étant informative par rapport aux étiquettes y associées aux données.

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

où I est l'information mutuelle, $I(x,y) = \int\int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} d_x d_y$. Le paramètre $\beta \geq 0$ contrôle l'équilibre entre ces deux sous objectifs. Une valeur β élevée entraîne une représentation hautement compressée. Intuitivement, le premier terme de l'équation 1 encourage z à bien prédire y ; le deuxième terme encourage z à « oublier » x de façon à le compresser le plus possible. Dans ce cadre, la loi d'encodage, ou encodeur, $p(z|x)$ est un choix de modélisation et est définie explicitement. L'information mutuelle n'est généralement pas calculable en forme close. Alemi et al. (2017) proposent donc d'utiliser une approche variationnelle (Jordan et al., 1999) : une série d'approximations permet de construire une borne inférieure de l'équation 1 :

$$-L_{VIB} = \mathbb{E}_{z\sim p(z|x)}[\log q(y|z)] - \beta \mathbf{KL}(p(z|x)||q(z)) \tag{2}$$

où $q(y|z)$ est une approximation variationnelle de $p(y|z)$, qui joue le rôle de décodeur, et $q(z)$ est une approximation de l'à priori en z (*prior*). Cette borne est issue de la décomposition de l'information mutuelle. Comme dans l'inférence variationnelle, on est assuré de faire croître la valeur en équation (1) en maximisant l'équation (2). Malheureusement, l'espérance n'est souvent pas calculable en forme close et doit donc être approximée au moyen d'une méthode d'échantillonnage. Oh et al. (2019) proposent d'utiliser ce principe pour appendre des représentations probabilistes d'image, en apprenant à séparer des paires d'images observée ($y = 1$) et des exemples négatifs tirés aléatoirement ($y = 0$).

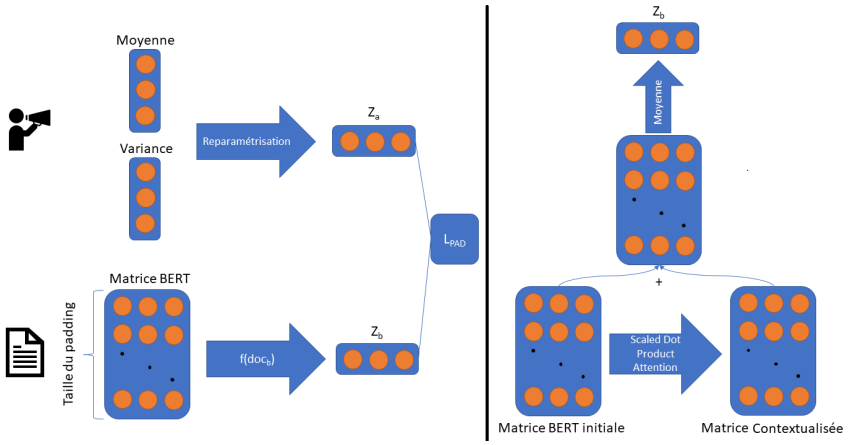

FIG. 2 – *L'architecture du modèle PAD (Plongement d'Auteurs et de Documents) à gauche. Un document est représenté par une matrice de plongements de mots contextualisés obtenue avec un modèle BERT. Les représentations latentes sont obtenues : par reparamétrisation pour l'auteur, par un encodage par attention et moyenne pour le document. La perte est ensuite calculée selon l'équation (4). À droite, l'encodeur de document :* $f(doc_b)$.

3.2 PAD

Nos données sont un ensemble d'auteurs et de documents. Chaque document peut être écrit par un ou plusieurs auteurs, comme c'est le cas dans la littérature scientifique par exemple. Le but est d'apprendre pour chaque auteur et chaque document, une représentation $z \in \mathbb{R}^r$. En plus d'un vecteur, nous voudrions apprendre une mesure d'incertitude pour chaque auteur. Par exemple, un auteur publiant dans des domaines variés devra avoir une incertitude associée élevée. Enfin, nous voudrions pouvoir intégrer un modèle de langue pré-entraîné au modèle.

Nous utiliserons l'architecture BERT (Devlin et al., 2019). La dernière couche de ce modèle contient les plongements contextualisés des mots du document encodé. La matrice est de taille fixe, c'est à dire $p \times r$, où r est la taille de l'espace latent et p est la taille du padding. Le padding correspond à la longueur de document maximale observée dans le corpus.

Nous reprenons le cadre théorique présenté en section précédente et devons donc d'abord définir la loi d'encodage, le décodeur et l'à priori. Un auteur a est encodé selon une loi normale dépendant des paramètres suivants : sa moyenne μ_a et sa variance σ_a^2.

Pour construire z_d, le document est d'abords encodé avec un modèle BERT pré-entraîné qui retourne une matrice de plongements contextualisés. Il s'agira ensuite de réaliser une agrégation de cette matrice d'embedding de mots, de façon à obtenir un unique vecteur z_d, guidée par la tâche d'identification d'auteur. Dans Reimers et Gurevych (2019), les mots sont moyennés pour construire les plongements de documents, mais c'est le modèle BERT lui-même qui est "fine tuné" pour construire des représentations adéquates par rapport à la tâche. Dans PAD, nous avons fait le choix d'apprendre une couche supplémentaire pour modifier les plongements plutôt que de "fine tuner" BERT. En effet, cela rendrait notre méthode peu exploitable sans machine puissante. De plus, le fait d'apprendre une couche d'agrégation permet d'étendre notre

modèle à n'importe quels plongements de mots, comme Word2Vec ou Glove (Mikolov et al., 2013; Pennington et al., 2014).

La matrice de plongements issue de BERT est injectée dans une couche d'attention de type "scaled dot product" (Vaswani et al., 2017). Le résultat est une nouvelle matrice de plongements contextualisés, sommée ensuite avec la matrice BERT initiale (on réalise donc une connexion résiduelle, comme dans le Transformer). Enfin, les plongements de mots obtenus sont moyennés pour construire la représentation $z_d \in \mathbb{R}^r$. Nous n'intégrons pas d'aspect probabiliste dans l'encodage du document. Nous démontrerons l'intérêt de cette fonction d'encodage en comparant nos résultats à une simple moyenne des embeddings pour construire z_d.

On obtient pour un auteur a et un document d :

$$\begin{aligned} &z_a \sim \mathcal{N}(\mu_a, \sigma_a^2) \\ &z_d = f(doc_d) = Moyenne(Attention(Bert(doc_d)) + Bert(doc_d)) \end{aligned} \tag{3}$$

Nous construisons ensuite l'ensemble des paires (a, d) positives associées au label $y = 1$ correspond au fait que l'auteur a a écrit d. Pour chaque paire, nous tirons k exemples négatif (a', d), associé au label $y = 0$. On obtient comme fonction objectif générale :

$$L_{PAD} = -\mathbb{E}_{z_a \sim p(z_a|x_a), z_d = f(doc_d)}[\log q\,(y|z_a, z_d)] + \beta \mathbf{KL}(p(z_a|x_a)||q(z_a)) \tag{4}$$

Similairement à Oh et al. (2019), la probabilité du label y est

$$q\,(y = 1|z_a, z_d) = \sigma\,(-c||z_a - z_d||_2 + e) \tag{5}$$

avec σ la fonction sigmoïde, z_a (resp. z_d) l'embedding de l'auteur a (resp. du document d), $c > 0$ et $e \in \mathbb{R}$. $q(z_a)$ une loi normale centrée réduite et l'espérance est approximée par une méthode de tirage. Plus précisément, nous utiliserons l'approche de Kingma et Welling (2014) et Oh et al. (2019). Grâce à l'astuce de la reparamétrisation, permettant la rétro propagation du gradient, la minimisation de L_{PAD} peut être réalisée par un réseau de neurones. Avec L un entier naturel, il suffit alors de calculer :

$$\begin{aligned} \mathbb{E}_{z_a \sim p(z_a|x_a), z_d = f(doc_d)}[\log q\,(y|z_a, z_d)] \approx &\frac{1}{L} \sum_{l=1}^{L} \log q\,(y|z_a, f(doc_d)) \\ &z_a = \mu_a + \sigma_a \odot \epsilon,\ \epsilon \sim \mathcal{N}(0, 1) \end{aligned} \tag{6}$$

4 Évaluation

4.1 Jeux de Données

Nous utilisons deux jeux de données préparés par Delasalles et al. (2019). Le premier jeu de données, S2G, est composé de titres d'articles scientifiques issus du domaine du machine learning. Les articles ont été publiés entre 1985 et 2017. Il a été initialement construit par Ammar et al. (2018). Les méta-données associées sont les noms des auteurs et la conférence

dans laquelle a été publié le papier (par ex. IJCAI, AAAI, ICCV). Il contient 45 496 documents et 1117 auteurs. Le deuxième jeu de données, NYT, est un ensemble de titres d'articles de presse du New York Times, initialement proposé dans Yao et al. (2018). Les méta-données sont l'auteur et la catégorie de l'article (par ex. sport, art, business). Il contient 41 249 documents et 542 auteurs.

4.2 Compétiteurs et Paramètres

Nous comparons notre méthode à trois approches. Deux méthodes d'apprentissage de représentation d'auteurs et de document : ATM (Rosen-Zvi et al., 2004) et A2V (Ganguly et al., 2016). La troisième est une méthode naïve qu'on nommera BERT$_a$ (Devlin et al., 2019). Cette dernière consiste à moyenner les représentations des mots de la dernière couche d'un modèle BERT pré-entraîné pour construire la représentation des documents. Pour les auteurs, on moyenne les représentations des documents écrits par l'auteur.

Pour A2V, nous utilisons la version "Content-Info" seulement, c'est-à-dire sans considérer les liens entre documents. Nous utilisons les paramètres suggérés par Ganguly et al. (2016) : les plongements sont de dimension 100, avec une couche cachée de 50 neurones. Les plongements de documents sont initialisés par les représentations obtenues par la méthode PV-DBOW (Le et Mikolov, 2014). L'optimiseur est Adam avec 0.001 de taux d'apprentissage, et des lots de taille 256. Pour ATM, nous utilisons l'implémention Gensim[1] avec un nombre de thématiques qui donne la valeur de cohérence c$_v$ maximale (Röder et al., 2015). Cela donne 201 thématiques pour NYT et 229 pour S2G.

Pour BERT$_a$, ainsi que pour PAD, nous utilisons la version "bert-base-uncased" d'HuggingFace (Wolf et al., 2019), ainsi que leur implémentation du modèle BERT. Nous obtenons donc des représentations d'auteurs et de documents en dimension 768. Pour PAD, plus précisément, nous tirons cinq exemples négatifs par exemple positif. Nous tirons $L = 20$ échantillons pour calculer l'équation 6. Dans nos expériences, une valeur $\beta = 1e - 13$ semble fournir les meilleurs résultats. Enfin, nous utilisons des lots de taille 256, l'optimiseur est Adam, et 0.001 de taux d'apprentissage. Nous comparons PAD à une version plus simple, PAD$_a$, dans laquelle la fonction d'encodage par attention est remplacée par une simple moyenne de la matrice d'embedding. Nous fournissons l'implémentation de notre modèle, ainsi que les jeux de données à la communauté[2].

4.3 Évaluation quantitative

La tâche d'identification d'auteur est équivalente à un problème de classification multi-labels dans le cas où le document est écrit par plusieurs auteurs, comme c'est le cas sur le jeu de donnée S2G. Nous adoptons donc les scores d'erreur de couverture CE (*coverage error*) et le score de précision moyenne de rangs LRAP (*label ranking average precision score*). Nous calculons la proximité cosinus entre documents et auteurs, puis normalisons cette mesure. L'erreur de couverture calcule combien de plus proches voisins en moyenne il faut considérer pour couvrir tous les vrais auteurs d'un document. Dans le pire des cas, cette valeur est égale au nombre d'auteurs dans le jeu de données. Plus cette valeur est faible, meilleure est la méthode.

[1] `https://radimrehurek.com/gensim/`
[2] `https://github.com/AntoineGourru/PAD`

	S2G		NYT	
	CE	LRAP	CE	LRAP
ATM	437.51	0.0397	189.07	0.0473
BERT$_a$	319.46	0.0519	136.89	0.0895
A2V	101.04	**0.4182**	**53.26**	**0.2549**
PAD$_a$	515.90	0.01	231.75	0.0268
PAD	**65.67**	0.20	58.10	0.1697

TAB. 1 – *Coverage-error (à minimiser) et Précision (à maximiser) sur les deux jeux de données étudiés*

Train/Test ratio	10%	30%	50%
ATM	30.47 (1.67)	39.11 (1.60)	42.21 (2.37)
BERT$_a$	53.89 (2.26)	63.29 (1.89)	64.46 (1.30)
A2V	11.97 (0.70)	12.26 (0.99)	11.44 (1.87)
PAD$_a$	52.09 (2.15)	58.66 (1.26)	60.11 (1.76)
PAD	**57.64** (2.49)	**65.26** (1.70)	**68.34** (1.23)

TAB. 2 – *Résultats en classification sur le jeu de données NYT en faisant varier la proportion des données d'entraînement de 10% à 50%*

LRAP est une mesure moyenne de précision pour le cas multi-classes, entre 0 et 1, où 1 est la valeur maximale de précision. Plus précisément, pour chaque vrai label de chaque échantillon, la LRAP calcule la proportion de vrais labels mieux classés. Les résultats sont présentés dans le tableau 1. ATM et BERT$_a$ ne semblent pas bien adaptés à cette tâche, sur les deux jeux de donnée. PAD donne de meilleurs résultats en couverture sur le jeu de donnée S2G mais A2V obtient un score LRAP plus élevé. Cet écart peut s'interpréter comme un ratio précision/rappel : A2V est plus précis mais donne plus souvent de mauvais scores à certains vrais auteurs que PAD. Sur NYT, A2V surpasse les autres méthodes, mais d'une marge faible par rapport à PAD. Enfin, la fonction d'encodage que nous proposons semble bien fonctionner : une simple moyenne, c'est à dire le modèle PAD$_a$, fournit les moins bons résultats.

Dans nos jeux de données, chaque document est associé à une thématique. Nous associons donc à chaque auteur la thématique qui apparaît le plus de fois dans les documents qu'il écrit. Nous utilisons ensuite une méthode classique d'évaluation (Yang et al., 2015) : nous entraînons un modèle SVM avec régularisation L2 à partir des plongements d'auteurs et calculons le score micro F1 et sa variance pour plusieurs ratios entraînement/test. La régularisation optimale est déterminée par recherche par quadrillage (*grid search*). Nous présentons les résultats dans les tableaux 2 et 3. A2V n'est pas du tout adapté à cette tâche et n'apprend pas de représentations permettant de classifier les auteurs. Seule la proximité entre auteur et documents semble interprétable dans ce modèle, et non leurs réelles positions dans l'espace latent. Sur cette évaluation, une simple moyenne de vecteur BERT donne de bons résultats. PAD obtient néanmoins les meilleurs scores sur le jeu de données S2G pour des ratios faibles : notre méthode généralise donc mieux. Sur le jeu de données NYT, PAD surpasse toutes les autres méthodes.

Train/Test ratio	10%	30%	50%
ATM	44.25 (1.40)	50.74 (1.14)	53.42 (1.30)
BERT$_a$	55.80 (1.55)	<u>62.57</u> (0.54)	**64.65** (1.28)
A2V	14.30 (1.17)	11.24 (0.94)	11.63 (1.14)
PAD$_a$	53.64 (1.97)	56.33 (1.36)	55.74 (1.51)
PAD	**60.75** (0.90)	**63.67** (0.77)	<u>63.49</u> (1.11)

TAB. 3 – *Résultats en classification sur le jeu de données S2G en faisant varier la proportion des données d'entraînement de 10% à 50%*

4.4 Évaluation Qualitative

Dans la figure 3, nous présentons une visualisation des plongements d'auteurs obtenus en appliquant PAD sur le jeu de données S2G. Nous utilisons la méthode T-SNE (Maaten et Hinton, 2008) pour réduire la dimension à 2 (avec une perplexité de 20). Les couleurs correspondent aux classes, c'est-à-dire la conférence dans laquelle chaque auteur publie le plus souvent. Les plongements semblent bien séparer les classes. On observe aussi que les auteurs proches étudient des thématiques similaires : Ian Goodfellow, Yann LeCun et Yoshua Bengio sont trois voisins, comme présentés sur la figure.

Un avantage de PAD est que les documents et les auteurs sont représentés dans le même espace. On peut donc utiliser les représentions pour de nombreuses tâches de recommandation. Le tableau 4 montre les 3 plus proches voisins de l'article : "Latent Dirichlet Allocation" et leur variance moyenne. Les plus proches voisins sont calculés selon la similarité cosinus ou la vraisemblance (calculée en fonction des moyennes et variances apprises par PAD). On observe tout d'abord que David Blei à une variance faible par rapport aux autres voisins (en termes de cosinus). En effet, il a fait sa spécialité des modèles d'Allocation de Dirichlet latente. Ensuite, la prise en compte de la variance pour calculer les plus proches voisins d'un document rapproche Michael Jordan, qui est co-auteur de l'article. Il a une variance encore plus faible. Il semblerait donc que les thématiques qu'il aborde dans notre jeu de données soient plus précises encore que David Blei.

Cosinus (Variance)	**Vraisemblance (Variance)**
David M. Blei (0.044)	David M. Blei (0.044)
Neil D. Lawrence (0.083)	Michael I. Jordan (0.033)
Ariel D. Procaccia (0.051)	Zoubin Ghahramani (0.040)

TAB. 4 – *Les 3 plus proches voisins de l'article : "Latent Dirichlet Allocation" et leur variance moyenne. Les plus proches voisins sont calculés selon la similarité cosinus ou selon la vraisemblance (qui prend donc en compte la variance)*

5 Conclusion

Dans cet article, nous avons présenté une méthode d'apprentissage de représentations d'auteurs et de documents nommée PAD. Elle repose sur un apprentissage de type *Variational In-*

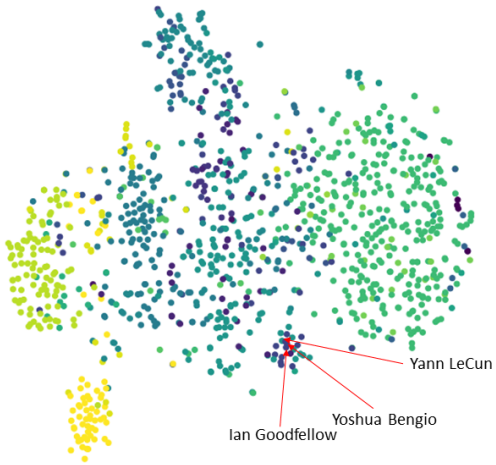

FIG. 3 – *Visualisation par T-SNE Maaten et Hinton (2008) des plongements d'auteurs sur le jeu de donnée S2G. Les couleurs correspondent aux classes (conférences les plus fréquentes).*

formation Bottleneck. PAD dépasse ou égale les scores de l'état de l'art en classification et en identification d'auteur. Elle est aussi plus robuste que le principal compétiteur qui, malgré de bonnes performances en identification, ne permet pas de bien classifier les auteurs. Chaque auteur est en plus associé à une mesure d'incertitude dont l'analyse peut permettre d'améliorer la recommandation. Nous explorerons dans de futurs travaux d'autres méthodes d'agrégation pour construire les représentations de documents.

Références

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

Ammar, W., D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. 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)*, pp. 84–91.

Bahdanau, D., K. Cho, et Y. Bengio (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Blei, D. M., A. Y. Ng, et M. I. Jordan (2003). Latent dirichlet allocation. *Journal of machine Learning research 3*(Jan), 993–1022.

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv :2005.14165*.

Delasalles, E., S. Lamprier, et L. Denoyer (2019). Learning dynamic author representations with temporal language models. In *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 120–129.

Devlin, J., M.-W. Chang, K. Lee, et K. 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)*, pp. 4171–4186.

Ganguly, S., M. Gupta, V. Varma, V. 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*, pp. 49–50. International World Wide Web Conferences Steering Committee.

Gourru, A., J. Velcin, et J. Jacques (2020). Gaussian embedding of linked documents from a pretrained semantic space. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*.

Ji, G., R. Bamler, E. B. Sudderth, et S. Mandt (2017). Bayesian paragraph vectors. *Symposium on Advances in Approximate Bayesian Inference*.

Jordan, M. I., Z. Ghahramani, T. S. Jaakkola, et L. K. Saul (1999). An introduction to variational methods for graphical models. *Machine learning 37*(2), 183–233.

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

Kuzi, S., A. Shtok, et O. Kurland (2016). Query expansion using word embeddings. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pp. 1929–1932. ACM.

Le, Q. et T. Mikolov (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pp. 1188–1196.

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, et V. Stoyanov (2019). Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692*.

Luong, M.-T., H. Pham, et C. D. Manning (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421.

Maaten, L. v. d. et G. Hinton (2008). Visualizing data using t-sne. *Journal of machine learning research 9*(Nov), 2579–2605.

Meng, Z., S. Liang, H. Bao, et X. Zhang (2019). Co-embedding attributed networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 393–401.

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

Oh, S. J., K. Murphy, J. Pan, J. Roth, F. Schroff, et A. Gallagher (2019). Modeling uncertainty with hedged instance embedding. In *Proceedings of the International Conference on Learning Representations*.

Pennington, J., R. Socher, et C. Manning (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Reimers, N. et I. Gurevych (2019). Sentence-bert : Sentence embeddings using siamese bert-networks. *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*.

Röder, M., A. Both, et A. Hinneburg (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pp. 399–408.

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

Tishby, N., F. C. Pereira, et W. Bialek (1999). The information bottleneck method. *The 37th annual Allerton Conference on Communication, Control, and Computing*, 368–377.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, et I. Polosukhin (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, et A. M. Rush (2019). Huggingface's transformers : State-of-the-art natural language processing. *ArXiv abs/1910.03771*.

Yang, C., Z. Liu, D. Zhao, M. Sun, et E. Y. Chang (2015). Network representation learning with rich text information. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Yao, Z., Y. Sun, W. Ding, N. Rao, et H. Xiong (2018). Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the eleventh acm international conference on web search and data mining*, pp. 673–681.

Summary

Most recent language models use contextualized word embedding, learnt using the Transformer architecture. They achieve state-of-the art performance on a lot of natural language processing tasks. Pretrained versions of these models are now widely used, however, their fine-tuning on specific tasks remains a central question. For example, these methods do not provide document and author level representations: a simple average of the contextualized word embedding is not good enough (Reimers et Gurevych, 2019). We develop a simple architecture based on Variational Information Bottleneck (VIB) to learn author and document representations using pre-trained contextualized word vectors (Devlin et al., 2019). We evaluate our method quantitatively and qualitatively on two datasets: a news article corpus, and a scientific article corpus. Our method produces more robust representations than existing methods and performs well in author identification and classification.