Apprentissage de l'évolution langagière dans des communautés d'auteurs

Edouard Delasalles — Sylvain Lamprier — Ludovic Denoyer

Sorbonne Université, LIP6, F-75005, Paris, France

RÉSUMÉ. *Les modèles de langue sont au cœur de nombreux de travaux, notamment dans les domaines de la recherche d'information et de la fouille de texte. Plutôt qu'une analyse fine de la sémantique des textes, ces modèles statistiques visent à extraire des distributions d'occurrence de mots dans différents contextes. Divers types d'approches ont été proposés dans la littérature, du simple modèle multinomial unigramme à des modèles à variables latentes pour la prise en compte de dépendances complexes dans les textes. Néanmoins, peu de travaux se sont portés sur la prise en compte conjointe des dépendances structurelles et temporelles dans des distributions de textes observés au cours du temps. Nous établissons ici un modèle dynamique de l'évolution langagière d'une communauté d'auteurs. En apprenant un modèle neuronal d'évolution sur des ensembles de textes produits par différents auteurs d'une communauté à différents instants, nous sommes capables d'en exploiter les dépendances latentes à des fins de prédiction des publications à venir.*

ABSTRACT. *Language models are at the heart of numerous works, specifically in the information retrieval domain. Instead of fine grained semantical analysis, theses statistical models aim at extracting words occurrence distributions in different contexts. Several approaches have been considered among the community, from simple unigram models to complex latent variables aiming at capturing subtle dependencies in texts. Nevertheless, only few works focused both on structural and temporal dependencies in text distributions observed through time. We propose a dynamic model of the stylistic evolutions of authors. By capturing influence relationships between authors, we are able to learn a dynamic language model for the prediction of distributions of texts that will publish a considered community of authors in the future time-steps.*

MOTS-CLÉS : *Apprentissage Profond, Modèle de Langue dynamique, Diffusion langagière.*

KEYWORDS: *Deep Learning, Dynamic Language Modeling, Stylistic Diffusion, Neural Networks*

1. Introduction

La modélisation de la langue est au cœur de multiples travaux de recherche depuis de nombreuses années. Alors que le domaine du traitement de la langue naturelle vise à une analyse fine du texte, les modèles de langue pour la recherche d'information ou la fouille de texte sont essentiellement basés sur des comptages de mots (ou n-grammes), avec prise en compte de dépendances plus ou moins complexes. Les travaux de ce domaine ont donné lieu à divers modèles, du simple modèle multinomial unigramme (Song et Croft, 1999) à des modèles neuronaux avec apprentissage de représentation sémantique des mots (Bengio *et al.*, 2003). Dans tous les cas, il s'agit de déterminer des probabilités d'occurrence des mots dans les textes, éventuellement en fonction de leur contexte d'émission.

Néanmoins, peu de travaux se sont portés sur la prise en compte conjointe des dépendances structurelles et temporelles dans des distributions de textes observés au cours du temps. Alors que la diffusion d'information dans les réseaux a suscité de nombreuses recherches pour l'extraction et la prédiction de dynamiques de transmission de contenu (voir par exemple (Saito *et al.*, 2009)), la quasi totalité des approches se sont bornées à l'étude de diffusion dans un cadre discret (infection ou non infection par un contenu émis par une source de propagation du réseau). Or, il paraît naturel que les dynamiques dans les communautés d'auteurs (influences inter-auteurs ou adoption de nouveaux comportements en réaction à des stimuli externes) ne se limitent pas à des réactions binaires, mais se reflètent également sur des comportements plus diffus, et notamment sur la façon de s'exprimer. Divers travaux autour de la modélisation de thématiques et leur évolution temporelle existent (voir par exemple (Wang et Mccallum, 2006) ou (Kabán et Girolami, 2002)), mais pour la plupart ils ne concernent pas le cadre multi-auteurs, ou bien ne peuvent pas être appliqués à des tâches de prédiction de distributions futures.

Nous proposons dans cet article de nous intéresser à ce phénomène d'évolution langagière dans des communautés d'auteurs dans un cadre prédictif. L'idée est de se baser sur des ensembles de textes observés sur une période passée pour prédire les distributions de mots des publications à venir pour les auteurs considérés. L'hypothèse sous-jacente est qu'il existe des régularités dans les évolutions temporelles du langage au sein d'une communauté d'auteurs, qu'il est alors possible d'exploiter en vue d'une amélioration de la prédiction par rapport à des modèles de langue statiques.

Nous établissons un modèle dynamique de l'évolution langagière d'auteurs basé sur l'apprentissage de représentation. Bien que nous ne modélisons pas explicitement les relations entre auteurs, notre modèle est capable de capturer des dynamiques d'évolution latentes via un état courant du système, permettant de conditionner les modèles de langue des différents auteurs, et dont les translations dans l'espace de représentation suivent une fonction apprise par un réseau de neurones. À la manière de (Le et Mikolov, 2014), où un modèle de langue type `Word2Vec` (Mikolov *et al.*, 2013) est conditionné par un vecteur représentant le paragraphe dont le texte à modéliser est issu, ce vecteur d'état adapte les distributions d'occurrence des mots en fonction de

la situation de l'instant considéré. Différents modèles de langue peuvent être envisagés sur ce principe, nous proposons de considérer un modèle simple multinomial unigramme ainsi qu'un modèle LSTM permettant la prise en compte du contexte des mots dans leurs probabilités d'occurrence. Le modèle proposé est finalement évalué sur un jeu de données regroupant des titres de publications scientifiques, afin d'en démontrer les avantages.

La suite de cet article s'articule de la manière suivante. La section 2 discute d'un certain nombre de travaux connexes afin de positionner l'approche par rapport à l'état de l'art. La section 3 présente les modèles proposés. Enfin, la section 4 décrit les expérimentations réalisées et en discute des résultats.

2. Travaux Connexes

L'idée de s'intéresser à l'évolution langagière dans les textes n'est pas nouvelle. En remontant d'une quinzaine d'années, on trouve des travaux sur l'évolution des thématiques dans des documents textuels, notamment celui de (Kabán et Girolami, 2002) dont le modèle basé sur des chaînes de Markov cachées vise à permettre de visualiser les évolutions temporelles des thématiques principales dans un flux textuel. Cette approche entre dans le cadre général du `Topic Detection and Tracking`, où l'idée est d'identifier les thématiques émergentes et d'en repérer les mentions dans des flux de documents. Ce modèle consistant en une extension du modèle GTM temporel de (Bishop *et al.*, 1997) à la modélisation textuelle, permet de visualiser les changements thématiques via des trajectoires sur une grille 2-D. Ce genre d'approche permet de faire du suivi de thématiques et de la segmentation de texte mais ne peut cependant pas être utilisé pour des tâches de prédiction ou de modélisation de la langue. L'approche non-markovienne proposée par (Wang et Mccallum, 2006) ne permet pas non plus de prédire des distributions pour des périodes futures, malgré une bonne capacité à repérer les évolutions de thématiques sur une période d'observations. Divers travaux ont en outre proposé des études des évolutions du vocabulaire - selon les transformations d'un graphe sémantique des termes dans (Kenter *et al.*, 2015) -, ou des dérives thématiques d'une communauté - selon les thématiques majoritaires par intervalle de temps dans (Hall *et al.*, 2008).

Plus proches des applications visées dans cet article, les modèles thématiques dynamiques tels que ceux proposés dans (Blei et Lafferty, 2006) proposent une modélisation de type `LDA` (Latent Dirichlet Allocation (Blei *et al.*, 2002)), mais où les distributions de thématiques et les distributions de mots selon les thématiques évoluent au fil du temps. Les évolutions entre distributions multinomiales successives sont modélisées par des mouvements gaussiens de leurs paramètres naturels, à la manière des filtres de Kalman. L'optimisation des modèles se fait par inférence variationnelle. Les limites de ces modèles sont cependant la nécessité de fixer le nombre de thématiques à considérer et la restriction des modèles de langue à des distributions d'occurrence de mots simples (difficile d'envisager des modèles de langue à dépendances longues tels que les LSTM par exemples dans ce cadre). Par ailleurs, ces approches sont générale-

ment contraintes à utiliser certains types de distributions conjuguées pour l'inférence des variables latentes de leurs modèles d'évolution. Notons les extensions de (Blei et Lafferty, 2006) à une version avec multiples échelles de temps (Iwata *et al.*, 2012) ou encore avec dépendances en temps continu (Wang *et al.*, 2012). En outre, (Gerrish et Blei, 2010) y introduit une notion d'influence entre documents qui pourraient s'approcher de notre objectif mais qui n'est utilisable qu'en analyse (pas de prédiction possible pour cette approche). Enfin, (Wang *et al.*, 2011) propose une approche temporelle avec prise en compte de relations entre documents via un graphe de dépendances connu a priori, ce qui sort quelque peu du cadre de cette étude où l'on fait l'hypothèse que l'on ne dispose pas de ce genre de données relationnelles.

Dans la lignée des approches par apprentissage de représentations (Bengio *et al.*, 2003) et du très populaire modèle `Word2Vec`, un récent engouement pour la modélisation temporelle a suscité divers modèles basés sur des projections de mots dans des espaces vectoriels latents tels que (Eger et Mehler, 2017) - dépendances temporelles linéaires entre représentations des mots et de leurs voisins -, (Bamler et Mandt, 2017) - modèle de `Skip-Gram` dynamique -, (Rudolph et Blei, 2017) - modèle à évolutions probabilistes exponentielles - ou encore (Yao *et al.*, 2017) - factorisation matricielle avec alignement temporel. Contrairement au modèle de (Kabán et Girolami, 2002), les éléments textuels sont projetés dans un espace continu plutôt que sur une grille discrète, ce qui permet d'employer des méthodes d'optimisation continues classiques. En outre, contrairement aux approches précédentes basées sur des distributions de thématiques avec dépendances temporelles, l'objectif de ces travaux est d'apprendre des représentations sémantiques de mots pouvant être directement utilisées dans des modèles de langue neuronaux classiques (type LSTM ou autres). Les dépendances temporelles sont considérées directement sur ces représentations de mots : chaque pas de temps considéré possède sa propre représentation du vocabulaire, où les représentations entre instants successifs respectent diverses contraintes. Cependant, il paraît difficile d'envisager ce type d'approches dans un cadre multi-auteurs, pour lequel il faudrait envisager des représentations pour chaque mot à la fois pour chaque pas de temps et pour chaque auteur. À noter enfin l'approche de (Rudolph *et al.*, 2017) pour données groupées permettant une réduction du nombre de paramètres en partageant des vecteurs de contextes entre groupes, mais pour lequel la transposition dans un cadre multi-auteurs temporel n'est pas évidente (nombre de groupes élevé, dépendances doubles, évolution temporelle vs groupes connectés).

Une alternative à la croisée des chemins entre ces différents modèles est alors de conditionner des modèles de langue appris en fonction des auteurs et des moments de publication des documents considérés. Ce genre de conditionnement a déjà été envisagé dans le cadre de la modélisation du contexte des mots dans les documents (Le et Mikolov, 2014), mais à notre connaissance pas pour l'extraction de dynamiques temporelles et structurelles dans des communautés d'auteurs. Plutôt que d'avoir une représentation vectorielle différente pour chaque mot aux différents pas de temps et pour les différents auteurs, ce qui paraît trop complexe pour être appris correctement, l'idée est de se baser sur une représentation commune mais dont le modèle de langue est modifié selon un conditionnement évolutif.

3. Modélisation dynamique de la langue

L'objectif du modèle proposé est donc d'être capable de prédire, selon des textes observés pour des auteurs sur des périodes de t_0 à t, les distributions de mots utilisés dans les textes que publieront ces mêmes auteurs à la période suivante $t+1$. Dans la suite, nous notons $X_t^a = \{x_t^{a,1}, \ldots, x_t^{a,n_t^a}\}$ l'ensemble des n_t^a textes publiés au temps t par l'auteur $a \in \mathcal{A}$, où $\mathcal{A}$ correspond à l'ensemble des auteurs considérés. Notre tâche consiste alors à maximiser la vraisemblance des textes au temps $t+1$ connaissant les textes publiés sur les périodes précédentes :

$$P(X_{t+1}|X_1 \ldots X_t) = \prod_{a \in \mathcal{A}} \prod_{x \in X_{t+1}^a} P(x|X_1 \ldots X_t)$$

Où X_t correspond à l'ensemble des textes publiés au temps t et $P(x|X_1 \ldots X_t)$ est la probabilité d'un texte x selon un modèle de langue conditionné par les textes de l'historique. Comme nous le verrons ci-dessous, ce modèle de langue peut prendre différentes formes. Dans tous les cas, nous considérons que la totalité de l'information issue de l'historique au temps t pour un utilisateur $a \in \mathcal{A}$ est contenue dans un vecteur de conditionnement $h_{t,a}$: $P(x|X_1 \ldots X_t) = P(x|h_{t,a})$ pour tout texte $x \in X_{t+1}^a$.

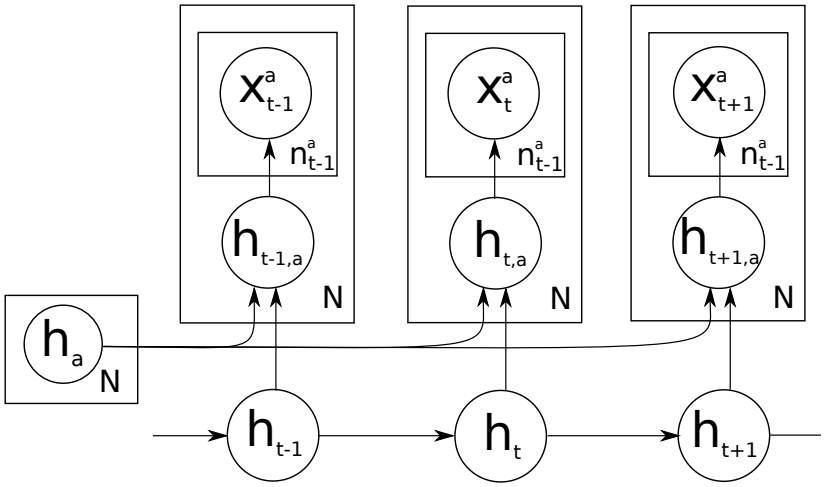

Figure 1 – Représentation graphique de notre modèle d'évolution langagière (sur trois pas de temps).

Notre architecture générale se décompose alors en deux parties distinctes, comme illustré sur la figure 1 :

– Une partie haute qui correspond à un modèle de langue conditionnel pour la génération des textes de chaque auteur a à l'instant courant t en fonction d'une entrée $h_{t,a}$ de dimension $2k$;

– Une partie basse qui décrit la manière de conditionner les modèles de langue en fonction des auteurs concernés et de l'état du langage à l'instant courant. Cette partie détermine les évolutions du système et les interactions entre entités considérées pour émettre un vecteur $h_{t,a}$ pour chaque auteur $a \in \mathcal{A}$ à chaque instant considéré t.

Nous détaillons dans la suite chacune de ces deux parties. Après avoir décrit les modèles de langue conditionnels utilisés, nous présentons donc le modèle général d'évolution permettant un conditionnement temporel de ces modèles. La section se termine par une discussion sur l'apprentissage du modèle.

3.1. *Modèles de langue conditionnels*

Les modèles de langue sont classiquement des modèles probabilistes de génération de textes, considérés comme des séquences de termes $\omega_{1\ldots n}$, où chaque ω_i correspond à un mot dans un vocabulaire Ω prédéfini. La probabilité $P(\omega_{1\ldots n})$ d'observer cette séquence étant donné le modèle de langue est donnée par :

$$P(\omega_{1\ldots n}) = \prod_{i=1}^{n} P(\omega_i|\omega_{1\ldots i-1}) \quad [1]$$

Où $P(\omega_i|\omega_{1\ldots i-1})$ est une distribution multinomiale (ou plus exactement catégorielle) d'émission du terme ω_i du vocabulaire Ω en fonction des termes précédents dans le texte. Dans notre cas, nous conditionnons nos modèles de langue selon le contexte du document, soit pour un instant t et un auteur a :

$$P(\omega_{1\ldots n}|h_{t,a}) = \prod_{i=1}^{n} P(\omega_i|\omega_{1\ldots i-1}, h_{t,a}) \quad [2]$$

Cette section décrit les deux types de modèles de langue envisagés dans cet article. Notons cependant que bien d'autres types de modèles auraient pu se brancher sur notre architecture générale puisqu'il s'agit de faire dépendre des modèles classiques d'un vecteur d'entrée issu du contexte du document (auteur et instant de publication).

3.1.1. *Modèle unigramme*

Dans ce modèle on considère une hypothèse de Markov d'ordre 1, qui suppose que la probabilité d'occurrence d'un mot dans un texte ne dépend pas des termes qui le précèdent. On a alors pour un terme en position i d'un document publié par l'auteur a à l'instant t : $P(\omega_i|\omega_{1\ldots i-1}, h_{t,a}) = P(\omega_i|h_{t,a})$. Ce type de modèle, bien que simpliste (il considère les documents comme des sacs de mots), est très utilisé dans la littérature, notamment dans la communauté de la recherche d'information,

puisqu'il permet d'éviter des représentations trop coûteuses (en terme de stockage) et trop parcimonieuses (difficilement généralisables).

Dans notre cas, il s'agit de définir une application $g_{t,a} : \Omega \rightarrow [0, 1]$ pour chacun des mots du vocabulaire en fonction du contexte d'entrée $h_{t,a}$, avec $\sum_{\omega \in \Omega} g_{t,a}(\omega) = 1$. Nous proposons la fonction SoftMax suivante :

$$P\left(\omega_i | \omega_{1\ldots i-1}, h_{t,a}\right) = \frac{e^{g_{\omega_i}(h_{t,a})}}{\sum_{\omega \in \Omega} e^{g_{\omega}(h_{t,a})}} \quad [3]$$

Où $g : \mathbb{R}^{2k} \rightarrow \mathbb{R}^{|\Omega|}$ est un réseau de neurones multi-couches prenant en entrée le vecteur de contexte $h_{t,a}$ et y associant un vecteur de réels de la taille du vocabulaire. g_ω dénote à la sortie du réseau correspondant au mot ω. Dans nos expérimentations, nous utilisons 3 couches cachées de taille k, avec activation RELU.

3.1.2. *Modèle neuronal récurrent*

Les modèles neuronaux récurrents sont des réseaux de neurones dont les modules se répètent, avec transmission d'états entre modules successifs, sur des séquences de tailles variables. Ils sont tout particulièrement adaptés pour le traitement des textes, puisqu'ils permettent naturellement de considérer l'hist55((orique des termes. Ceci est d'autant plus vrai depuis l'apparition des modèles `LSTM` (Long-Short Time Memory) et `GRU` (Gated Recurrent Unit) qui permettent, via des mécanismes d'attention, de se prémunir contre les problèmes d'estimation (i.e., *gradient vanishing*) sur les séquences de longue taille. Nous proposons d'utiliser le modèle `LSTM` pour considérer des hypothèses de Markov d'ordre supérieur à 1, tout en évitant les difficultés d'estimation et de stockage mentionnées ci-dessus pour la prise en compte de longs historiques de contextes textuels dans le cadre des modèles n-grammes.

Lors de son application au texte pour le calcul des distributions des mots, chaque module du réseau LSTM prend en entrée un mot du contexte. Ainsi, le calcul des probabilités d'émission selon un historique de taille l fait intervenir l modules identiques, éventuellement à plusieurs couches chacun. Le dernier module émet un vecteur de réels de taille $|\Omega|$ que l'on fait passer dans une fonction SoftMax pour obtenir une distribution de probabilité multinomiale.

Dans notre cas, le modèle doit non seulement considérer l'historique des mots du texte mais également le vecteur de conditionnement $h_{t,a}$:

$$P\left(\omega_i | \omega_{1\ldots i-1}, h_{t,a}\right) = \frac{e^{g_{\omega_i}(z(\omega_1)\ldots z(\omega_{i-1}), h_{t,a})}}{\sum_{\omega \in \Omega} e^{g_{\omega}(z(\omega_1)\ldots z(\omega_{i-1}), h_{t,a})}} \quad [4]$$

Où pour la position i du texte, $g : \mathbb{R}^{(i-1)w+2k} \rightarrow \mathbb{R}^{|\Omega|}$ est cette fois ci un réseau LSTM. Les $(i-1)$ mots d'historique donnés en entrée du LSTM pour le calcul de la distribution du mot i sont représentés par des vecteurs réels $z(\omega_1), \ldots, z(\omega_{i-1})$ de

dimension w, appris en même temps que les paramètres du réseau. Dans nos expérimentations, nous utilisons un module LSTM à trois couches, prenant en entrée des vecteurs de taille $w = 500$. Un des points forts du LSTM est qu'il peut considérer des séquences d'historique de tailles variables, lui permettant d'être appliqué de la même manière pour toutes les positions des textes considérés.

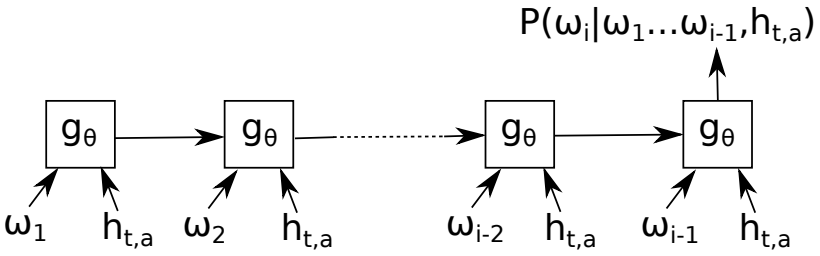

Figure 2 – Conditionnement du réseau de neurones récurrent

Notons enfin que, comme décrit en figure 2 représentant notre modèle de langue neuronal récurrent pour le calcul de la distribution du i-ème mot d'un document, nous avons fait le choix de répéter le conditionnement $h_{t,a}$ en entrée de chaque module du réseau, plutôt qu'uniquement en tant qu'état de départ du LSTM. L'idée est de faire peser ce conditionnement de manière importante sur les distributions émises.

3.2. *Conditionnement évolutif*

Cette section porte sur la partie basse de l'architecture générale présentée plus haut, à savoir le conditionnement évolutif des modèles de langue. Selon la manière dont la condition $h_{t,a}$ est calculée pour un temps t et un auteur a, le modèle peut s'en trouver largement différent et capturer différentes dynamiques / dépendances.

L'idée du modèle est que l'ensemble des auteurs partagent un même conditionnement temporel représentant l'état du système h_t à l'instant t, censé capturer les tendances courantes de la communauté d'auteurs. Dans ce modèle, chaque état h_t dépend du précédent, selon une distribution gaussienne centrée sur une évolution de cet état précédent h_{t-1}. Ainsi, pour tout $t > 1$:

$$h_t \sim \mathcal{N}(v(h_{t-1}), \sigma^2)$$

Avec $v : \mathbb{R}^k \to \mathbb{R}^k$ une fonction de transition entre états successifs du système. Dans nos expérimentations, cette fonction correspond à un réseau de neurones à 2 couches avec activations RELU. Ainsi, le système d'évolution des états suit un modèle stochastique, qui offre plus de flexibilité qu'une évolution déterministe (où h_t serait égal à la sortie de $v(h_{t-1})$). Cela permet notamment au système de gérer les cas de rupture forte pour lesquels les régularités observées ailleurs entre états successifs ne seraient pas respectées. Plutôt que de bouleverser considérablement la fonction d'évolution et

impacter trop fortement l'ensemble des états suivants ce genre de rupture ponctuelle, l'algorithme d'apprentissage peut alors choisir de ne pas se focaliser sur cette transition difficile, moyennant une pénalité dépendant de la variance σ^2 de la distribution gaussienne considérée. Cette variance σ^2 correspond à une matrice diagonale $k \times k$ apprise par le modèle, afin de lui permettre d'adapter l'aspect stochastique de l'évolution en fonction du niveau de régularité observé dans les données (voir la section suivante pour les détails de l'apprentissage du modèle).

Au vecteur h_t est concaténée une représentation h_a de l'auteur concerné (vecteur réel de taille k) pour former le vecteur de conditionnement $h_{t,a}$ de dimension $2k$. Cela permet de spécialiser le conditionnement selon l'auteur concerné. Alors que le modèle de langue apprend des distributions génériques partagées par l'ensemble des auteurs (la langue de la communauté concernée), cette spécialisation permet de s'adapter au mode d'expression de l'auteur considéré (tout en profitant de connaissances générales sur la langue employée extraites sur l'ensemble des auteurs).

Ainsi, le vecteur de conditionnement $h_{t,a}$ vise à apporter au modèle la capacité de s'adapter aux évolutions de langage générales de la communauté et de prévoir les distributions futures via le modèle d'évolution v. Mais aussi, de manière indirecte, il offre au modèle la capacité d'encoder des dépendances entre auteurs de la communauté : un utilisateur influent opérant une rupture dans son mode d'expression est supposé impacter le vecteur h_t, et ainsi modifier les distributions des auteurs qui lui sont proches. Les auteurs possédant un vecteur de conditionnement h_a proche de cet auteur influent tendent en effet à être plus impactés que les autres par cette évolution puisque leurs distributions dépendent d'activations neuronales similaires du modèle de langue considéré. La fonction v quant à elle est supposée capturer des récurrences dans les évolutions de la langue, et apporte au modèle un caractère prédictif, utile pour anticiper les distributions de documents à venir dans la communauté d'auteurs considérée.

3.3. *Apprentissage*

Soit $\Theta = (\theta, \vec{z}, \psi, \sigma, h_0, \vec{h}_a)$ l'ensemble des paramètres du modèle, où θ correspond aux paramètres du modèle de langue, $\vec{z}$ aux représentations des mots dans un espace de dimension w, ψ aux paramètres de la fonction d'évolution v, σ aux k paramètres d'écart de la gaussienne de génération des états h_t (on utilise une matrice de variance-covariance diagonale, dont les éléments de la diagonale correspondent aux éléments de σ au carré), h_0 à l'état initial du modèle et $\vec{h}_a$ à l'ensemble des vecteurs h_a pour tous les auteurs $a \in \mathcal{A}$.

La vraisemblance des documents observés selon notre modèle général est alors donnée par :

$$P(X_{1...T}|\Theta) = \quad P(X_1,\Theta) \prod_{t\in\{2,T\}} P(X_t|X_1 \dots X_{t-1},\Theta) \qquad [5]$$

$$= \qquad \int_{\vec{h}_t} p(\vec{h}_t|\Theta) \prod_{t\in\{1,T\}} P(X_t|h_t,\Theta) d\vec{h}_t$$

Avec $X_{1...T}$ l'ensemble des textes publiés jusqu'au temps T et $\vec{h}_t$ l'ensemble des vecteurs h_t de $t = 1$ à T.

Malheureusement cette vraisemblance est très difficile à maximiser directement du fait de la marginalisation sur tous les ensembles d'états $\vec{h}_t$ possibles. Nous avons alors recours à une approche variationnelle (Kingma et Welling, 2013), où l'on considère la distribution suivante :

$$q(\vec{h_t}) = \prod_{t\in\{1,T\}} q_t(h_t) \qquad [6]$$

Où les termes q_t sont des distributions variationnelles de h_t indépendantes pour chaque temps $t \in \{1,T\}$. Chacune de ces distributions sont des gaussiennes de moyenne ϕ_t et de variance η_t^2 (matrice diagonale de dimension $k \times k$). On note $\Phi = (\phi_t, \eta_t)_{t\in\{1,T\}}$ l'ensemble des paramètres variationnels de notre modèle.

En considérant la log-vraisemblance de notre modèle, et en adaptant le raisonnement de (Krishnan *et al.*, 2016) pour l'inférence dans les séquences, on a alors :

$$\log P(X_{1...T}|\Theta,\Phi)$$

$$= \log \int_{\vec{h}_t} p(\vec{h}_t|\Theta) \prod_{t\in\{1,T\}} P(X_t|h_t,\Theta) d\vec{h}_t$$

$$= \log \int_{\vec{h}_t} q(\vec{h_t}) p(\vec{h}_t|\Theta) \frac{\prod_{t\in\{1,T\}} P(X_t|h_t,\Theta)}{q(\vec{h_t})} d\vec{h}_t$$

$$\geq \int_{\vec{h}_t} q(\vec{h_t}) \log \left(p(\vec{h}_t|\Theta) \frac{\prod_{t\in\{1,T\}} P(X_t|h_t,\Theta)}{q(\vec{h_t})} \right) d\vec{h}_t$$

$$= \sum_{t\in\{1,T\}} \int_{h_t} q_t(h_t) \log P(X_t|h_t,\Theta) dh_t$$

$$+ \sum_{t\in\{1,T\}} \int_{h_{t-1}} q_{t-1}(h_{t-1}) \int_{h_t} q_t(h_t) \log \frac{p(h_t|h_{t-1},\Theta)}{q_t(h_t)} dh_{t-1} dh_t$$

$$= \sum_{t\in\{1,T\}} \mathbb{E}_{q_t(h_t)} P(X_t|h_t,\Theta) - \sum_{t\in\{1,T\}} \mathbb{E}_{q_{t-1}(h_{t-1})} KL(q_t(h_t)||p(h_t|h_{t-1}))$$

Avec q_0 une distribution Dirac centrée sur les paramètres h_0 et KL la divergence de Kullback-Leibler mesurant l'erreur d'approximation réalisée en calculant l'espérance sur la distribution variationnelle q plutôt que selon la distribution d'évolution

p. L'inégalité appliquée dans cette dérivation est obtenue grâce au théorème de Jensen sur les fonctions concaves. Elle permet d'obtenir une borne inférieure de la log-vraisemblance, qu'il est bien plus facile d'optimiser (grâce au passage du log dans les produits) par des méthodes de Monte-Carlo (échantillonnage des états).

La divergence de Kullback-Leibler entre deux gaussiennes possède une forme analytique. Cela permet de réécrire notre borne inférieure de vraisemblance, notée $\tilde{\mathcal{L}}(\Theta, \Phi)$, de la manière suivante :

$$\begin{aligned}\tilde{\mathcal{L}}(\Theta, \Phi) = & \sum_{t \in \{1,T\}} \mathbb{E}_{q_t(h_t)} P(X_t|h_t, \Theta) + \frac{Tk}{2} \\ & -\frac{1}{2}\left(T \sum_{i=0}^{k-1} \log \sigma_i^2 - \sum_{t \in \{1,T\}} \sum_{i=0}^{k-1} \log \eta_{t,i}^2 + \sum_{t \in \{1,T\}} \sum_{i=0}^{k-1} \frac{\eta_{t,i}^2}{\sigma_i^2} \right. \\ & \left. + \sum_{t \in \{1,T\}} \mathbb{E}_{q_{t-1}(h_{t-1})} (v(h_{t-1}) - \phi_t)' (\sigma^2)^{-1} (v(h_{t-1}) - \phi_t) \right)\end{aligned}$$

Où l'on note A' la transposée matricielle d'une matrice A et où σ_i^2 et $\eta_{t,i}^2$ correspondent respectivement à la i-ème composante des diagonales de σ^2 et η_t^2. Cette réécriture permet de gager en stabilité par rapport à une version où l'on échantillonnerait aussi les éléments de la KL.

Il s'agit maintenant de maximiser cette borne $\tilde{\mathcal{L}}(\Theta, \Phi)$ selon les paramètres Θ et Φ. Nous employons pour cela un algorithme de rétro-propagation du gradient stochastique (Kingma et Welling, 2013), dans lequel les états ne sont pas échantillonnés directement à partir de q mais sont calculés de manière déterministe selon une composante stochastique tirée d'une gaussienne centrée réduite (astuce de re-paramétrisation). Ainsi, pour tout t :

$$h_t = \epsilon_t \times \eta_t + \phi_t, \text{ avec } \epsilon_t \sim \mathcal{N}(0_k, I_k)$$

Où $\times$ correspond à une multiplication terme à terme des deux vecteurs (η_t est le vecteur de paramètres à partir duquel la matrice diagonale η_t^2 est formée). Tout l'aspect stochastique de l'algorithme d'inférence est alors cantonné au tirage de ϵ_t. Cette re-paramétrisation permet d'obtenir une formulation du gradient non-biaisée malgré la nécessité d'échantillonner les états pour le calcul de $\tilde{\mathcal{L}}(\Theta, \Phi)$ (afin de remplacer les espérances par des moyennes de K échantillons de chaque état). L'optimisation se fait par mini-batchs de M textes tirés aléatoirement parmi les N exemples d'apprentissage (M=128 et $K = 1$ dans nos expérimentations).

Notons enfin que nous considérons en fait un maximum à posteriori dans lequel l'ensemble des paramètres sont conditionnés par un prior $p(\Theta, \Phi)$ gaussien centré sur le vecteur nul et de variance-covariance diagonale λI (avec I la matrice identité et λ un hyper-paramètre scalaire). Cela permet d'éviter le sur-apprentissage du modèle et de conserver des variances raisonnables concernant les paramètres d'évolution stochastique. Dans nos expérimentations, nous utilisons $\lambda = 1$ pour tous les paramètres.

4. Expérimentations

Corpus Semantic Scholar

L'ensemble des expérimentations effectuées ont été réalisées sur le corpus `Semantic Scholar` [1]. Ce corpus contient des articles scientifiques parus entre 1991 et 2016. Tous les articles sont annotés avec l'année de parution et les auteurs. Nous avons extrait les articles publiés par le sous-ensemble des 1000 auteurs les plus prolifiques. Cela nous a permis de former un jeu de données composé des titres de 1087131 articles en informatique, neuroscience, biologie et médecine. Sur ces titres d'articles, nous avons extrait un vocabulaire de 24449 mots.

Modèles comparés

– U : Un modèle multinomial unigramme classique obtenu par simple comptage de proportions de mots dans les documents

– DTM : Le modèle dynamique décrit dans (Blei et Lafferty, 2006) (extension de LDA avec déplacements gaussiens des paramètres naturels des multinomiales entre pas de temps successifs). Deux versions sont considérées : DTM-1 considérant une unique distribution de mots et DTM-20 qui sélectionne pour chaque document une distribution (thématique) parmi 20 selon une approche d'inférence variationnelle ;

– DU-a : Le modèle unigramme décrit en section 3.1.1 avec conditionnement uniquement selon l'auteur ($h_{t,a} = (h_a, 0_k)$)[2] ;

– DU-t : Le modèle unigramme décrit en section 3.1.1 avec conditionnement uniquement selon l'instant ($h_{t,a} = (0_k, h_t)$) ;

– DU-t,a : Le modèle unigramme décrit en section 3.1.1 avec conditionnement selon l'auteur et l'instant ($h_{t,a} = (h_a, h_t)$) ;

– DR : Le modèle récurrent décrit en section 3.1.2 avec conditionnement constant ($h_{t,a} = 0_{2k}$)

– DR-a : Le modèle récurrent décrit en section 3.1.2 avec conditionnement uniquement selon l'auteur ($h_{t,a} = (h_a, 0_k)$) ;

– DR-t : Le modèle récurrent décrit en section 3.1.2 avec conditionnement uniquement selon l'instant ($h_{t,a} = (0_k, h_t)$) ;

– DR-t,a : Le modèle récurrent décrit en section 3.1.2 avec conditionnement selon l'auteur et l'instant ($h_{t,a} = (h_a, h_t)$) ;

Prédiction de modèles de langage

Nous avons entraîné tous nos modèles sur les articles parus entre 1991 et 2010 (inclus), et nous avons utilisé les articles de l'année 2011 pour sélectionner les modèles et les hyper-paramètres. L'apprentissage sur ces vingt années de publications a pris

1. http://labs.semanticscholar.org/corpus/
2. Où 0_l correspond à u vecteur de taille l.

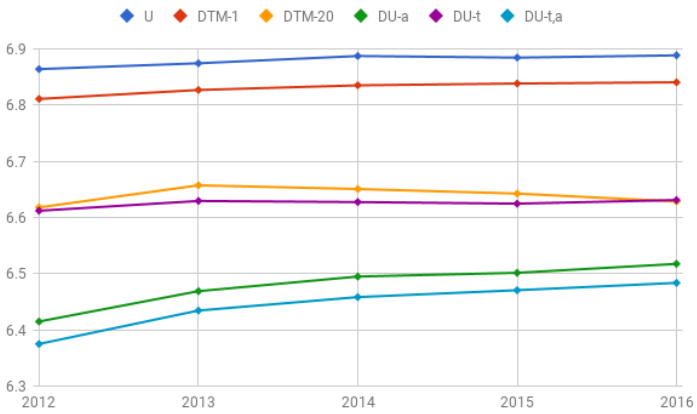

Figure 3 – Log-Vraisemblance négative à $t+1$ entre 2012 et 2016 sur les modèles unigrammes

environ 24h sur une carte graphique Nvidia TITAN Xp Pascal. Nous avons ensuite affiné nos modèles en ajoutant successivement les articles des années 2011 à 2015, en relevant à chaque fois les résultats de tests à l'année $t+1$. Pour chaque nouvelle année, nous affinons les modèles en les entraînant pendant 10 époques sur les données d'entraînement jointes aux nouvelles données (environ 1h30 d'apprentissage). Pour nos modèles évolutifs, on génère l'état de l'année de test $t+1$ en considérant son espérance : $h_{t+1} = v(\phi_t)$.

Nous rapportons sur les figures 3 (pour les modèles `U`, `DTM-1`, `DTM-20`, `DU-a`, `DU-t` et `DU-t,a`) et 4 (pour les modèles `DR`, `DR-a`, `DR-t` et `DR-t,a`) les résultats en log-vraisemblance négative moyenne (valeurs faibles préférables) à $t+1$ pour les années 2012 à 2016, selon le protocole décrit ci-dessus. Comme attendu, le modèle unigramme simple (comptage) donne les moins bon résultats. Le modèle DTM-1 correspond à un modèle similaire mais où les paramètres de la multinomiale sont autorisés à se déplacer entre pas de temps successifs. On observe que cette liberté permet au modèle d'améliorer les résultats par rapport au modèle statique. Le modèle `DTM-20` possède une liberté supplémentaire : elle peut sélectionner pour chaque document parmi 20 distributions de termes celle qui semble le mieux lui correspondre. Bien que ce modèle donne de meilleurs résultats que `DTM-1`, les performances sont biaisées car `DTM-20` se base sur les données à $t+1$ pour inférer pour chaque document de test la distribution à utiliser, ce qui ne correspond pas à notre cadre prédictif dans lequel les données à $t+1$ ne sont pas disponibles.

On observe sur la figure 3 que nos trois modèles obtiennent de meilleurs résultats que `DTM` et le modèle unigramme simple (`U`). Le modèle `DTM` avec 20 thématiques et le modèle `DU-t` sont très proches, mais encore une fois `DTM` à besoin d'inférer les thématiques à partir des données de test, ce qui n'est pas possible en pratique. On note aussi

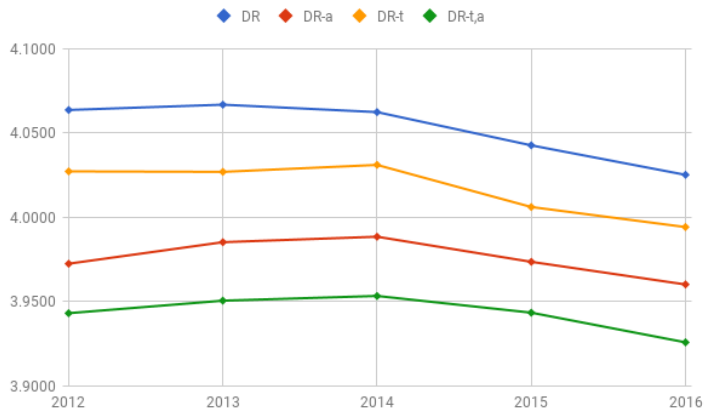

Figure 4 – Log-Vraisemblance négative à $t+1$ entre 2012 et 2016 sur les modèles récurrents

que le plus gros gain de performance est obtenu par l'ajout de représentation des auteurs. Cela est attendu puisque les articles proviennent de communautés scientifiques différentes qui n'ont pas le même vocabulaire. Enfin, on voit que le meilleur modèle unigramme est `DU-t,a`. L'association d'une représentation temporelle dynamique et d'une représentation des auteurs donne les meilleurs log-vraisemblances à $t+1$. Cela prouve que l'on arrive bien a capturer une évolution temporelle dans les modèles de langues des auteurs.

Enfin, on observe sur la figure 4 que tous les modèles LSTM gagnent en moyenne approximativement 2.5 points de log-vraisemblance par rapport aux modèles unigrammes, montrant ainsi l'intérêt des réseaux de neurones récurrents pour la modélisation du langage. Alors que l'utilisation de ce genre de réseaux n'est pas envisageable avec des modèles types `DTM` qui sont cantonnés à des distributions simples des termes pour assurer la phase d'inférence variationnelle avec priors conjugués, à notre connaissance notre modèle est le seul à offrir cette possibilité dans un cadre à dépendances temporelles (sauf approches de dépendances sur les représentations des termes écartées en raison de leur complexité). On observe également que l'ajout des représentations des auteurs donne de meilleurs résultats que le modèle conditionné uniquement par rapport à l'état temporel partagé. Mais c'est encore une fois le modèle qui associe les deux représentations qui offrent les meilleurs performances, confirmant ainsi l'intérêt des deux aspects pour la prédiction de modèle de langages.

5. Conclusion

Au fil des années, les modes d'expression d'une communauté d'auteurs évoluent, en fonction des modes du moment, des influences internes ou externes à la commu-

nauté, des transformations du monde. Il apparaît alors important d'être à même de capturer ces dynamiques, autant pour des visées d'analyse de la communauté et de ses tendances qu'à des fins de prédiction des publications à venir. Dans cet article nous avons proposé un modèle cherchant à répondre à ces besoins identifiés, en exploitant les avancées récentes de l'apprentissage neuronal et des techniques de représentation d'entités. Le modèle stochastique proposé cherche à capturer les dynamiques d'évolution des modes d'expression dans une communauté, en exploitant les dépendances entre temps successifs. Modélisant un état ponctuel du monde partagé par les différents auteurs de la communauté, l'idée était de permettre une inclusion des influences entre auteurs dans les estimations des tendances futures.

Les résultats expérimentaux obtenus, bien que partiels, sont prometteurs puisque le modèle proposé permet une amélioration de la prédiction langagière sur différents pas de temps d'une communauté d'auteurs scientifiques. Des travaux sont en cours pour l'application du modèle à d'autres jeux de données plus complets. Son application pour des tâches de reconnaissance d'auteurs ou de recherche d'experts est également considérée. Par ailleurs, bien qu'étant capable de capturer des influences entre auteurs, ces relations intra-communauté ne sont pas représentées de manière explicite dans le modèle proposé. Afin de s'orienter pleinement vers des modèles de diffusion continue dans les communautés, nous travaillons actuellement sur une extension basée sur des modèles d'attention neuronaux (Luong *et al.*, 2015), pour lesquels l'état courant du système peut être utilisé pour déterminer les auteurs les plus en adéquation avec le mode d'expression courant d'un individu, et être en mesure d'exploiter les tendances actuelles du voisinage de cet individu. Cela apportera non seulement une meilleure capacité de copie des modes d'expression entre auteurs, mais fournira également en sortie une description des relations d'influence régissant les dynamiques d'échange de la communauté.

Remerciements

Ce travail a été effectué avec le support du projet ANR LOCUST (2015-2019).

6. Bibliographie

Bamler R., Mandt S., « Dynamic Word Embeddings », *ArXiv e-prints*, February, 2017.

Bengio Y., Ducharme R., Vincent P., Jauvin C., « A Neural Probabilistic Language Model », *Journal of Machine Learning Research*, 2003.

Bishop C. M., Hinton G. E., Strachan I. G. D., « GTM through time », *Fifth International Conference on Artificial Neural Networks (Conf. Publ. No. 440)*, p. 111-116, Jul, 1997.

Blei D. M., Lafferty J. D., « Dynamic Topic Models », *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, ACM, New York, NY, USA, p. 113-120, 2006.

Blei D. M., Ng A. Y., Jordan M. I., « Latent Dirichlet Allocation », *Journal of Machine Learning Research*, vol. 3, p. 2003, 2002.

Eger S., Mehler A., « On the Linearity of Semantic Change : Investigating Meaning Variation via Dynamic Graph Models », *CoRR*, 2017.

Gerrish S. M., Blei D. M., « A Language-based Approach to Measuring Scholarly Impact », ICML'10, Omnipress, USA, p. 375-382, 2010.

Hall D., Jurafsky D., Manning C. D., « Studying the History of Ideas Using Topic Models », EMNLP '08, ACL, Stroudsburg, PA, USA, p. 363-371, 2008.

Iwata T., Yamada T., Sakurai Y., Ueda N., « Sequential Modeling of Topic Dynamics with Multiple Timescales », *ACM Trans. Knowl. Discov. Data*, vol. 5, n° 4, p. 19 :1-19 :27, February, 2012.

Kabán A., Girolami M. A., « A Dynamic Probabilistic Model to Visualise Topic Evolution in Text Streams », *Journal of Intelligent Information Systems*, vol. 18, n° 2, p. 107-125, Mar, 2002.

Kenter T., Wevers M., Huijnen P., de Rijke M., « Ad Hoc Monitoring of Vocabulary Shifts over Time », *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, ACM, New York, NY, USA, p. 1191-1200, 2015.

Kingma D. P., Welling M., « Auto-encoding variational bayes », *arXiv preprint arXiv :1312.6114*, 2013.

Krishnan R. G., Shalit U., Sontag D., « Structured Inference Networks for Nonlinear State Space Models », *ArXiv e-prints*, September, 2016.

Le Q. V., Mikolov T., « Distributed Representations of Sentences and Documents », *ICML'14*, 2014.

Luong M., Pham H., Manning C. D., « Effective Approaches to Attention-based Neural Machine Translation », *CoRR*, 2015.

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., « Distributed Representations of Words and Phrases and their Compositionality. », *NIPS'14*, 2013.

Rudolph M., Blei D., « Dynamic Bernoulli Embeddings for Language Evolution », *ArXiv e-prints*, March, 2017.

Rudolph M. R., Ruiz F. J. R., Athey S., Blei D. M., « Structured Embedding Models for Grouped Data », *CoRR*, 2017.

Saito K., Kimura M., Ohara K., Motoda H., « Learning Continuous-Time Information Diffusion Model for Social Behavioral Data Analysis », *Proceedings of the 1st Asian Conference on Machine Learning : Advances in Machine Learning*, ACML '09, Springer-Verlag, Berlin, Heidelberg, p. 322-337, 2009.

Song F., Croft W. B., « A General Language Model for Information Retrieval », *Proceedings of the Eighth International Conference on Information and Knowledge Management*, CIKM '99, ACM, New York, NY, USA, p. 316-321, 1999.

Wang C., Blei D. M., Heckerman D., « Continuous Time Dynamic Topic Models », *CoRR*, 2012.

Wang E., Silva J., Willett R., Carin L., « Dynamic relational topic model for social network analysis with noisy links », *2011 IEEE Statistical Signal Processing Workshop (SSP)*, p. 497-500, June, 2011.

Wang X., Mccallum A., « Topics over time : A non-Markov continuous-time model of topical trends », *in SIGKDD*, 2006.

Yao Z., Sun Y., Ding W., Rao N., Xiong H., « Discovery of Evolving Semantics through Dynamic Word Embedding Learning », *CoRR*, 2017.