Apprentissage de Modèles de Langue Neuronaux pour la Recherche d'Information

Nicolas Despres* — **Sylvain Lamprier** * — **Benjamin Piwowarski** *

* *Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris. Email: prenom.nom@lip6.fr*

RÉSUMÉ. *La recherche d'information (RI) ad-hoc se heurte à différentes difficultés, notamment liées à des discordances de vocabulaire entre requête et documents, ainsi qu'à la prise en compte de dépendances séquentielles entre les termes de la requête. Les récents modèles de langue neuronaux sont capables de capturer différents types de dépendances, grâce à une représentation distribuée des mots, mais nécessitent de gros volumes de données pour être entrainés efficacement. Jusqu'alors, ces modèles n'ont pas été utilisés directement pour des tâches de RI classiques, pour lesquelles l'estimation d'un modèle de langue pour chaque document est requise. Nous proposons une approche basée sur des transformations spécifiques (à chaque document) d'une représentation générique (apprise sur l'ensemble du corpus), pour définir des modèles de langue neuronaux pour la RI ad-hoc.*

ABSTRACT. *Information Retrieval (IR) faces different difficulties, notably those related to vocabulary mismatch issues and term dependencies. In the last few years, language models based on neural networks have been proposed to deal with both term dependencies and vocabulary mismatch issues in complex natural language processing tasks. However, to be efficient, these models require huge amounts of training data. They have thus never been employed for IR ad-hoc tasks directly, where the estimation of one language model per document is required. We propose an approach based on the specialization of a generic language model, learned on the whole document collection, by a set of document-specific parameters, to define neural language models fitted for ad-hoc IR tasks.*

MOTS-CLÉS : *Apprentissage de Représentation, Recherche d'Information, Réseau de Neurones.*

KEYWORDS: *Representation Learning, Information Retrieval, Neural Network.*

1. Introduction

Pour améliorer l'efficacité des recherches, la Recherche d'Informations (RI) vise depuis longtemps à prendre en compte les longues dépendances entre les termes et à aborder les questions de discordance de vocabulaire. Ces deux problèmes ont été abordés par différentes approches, des plus empiriques aux plus théoriques, mais aucune de celles proposées jusqu'ici ne résolvent les deux problèmes *à la fois*. Cet article propose une approche basée sur les récentes avancées dans les modèles de langue neuronaux.

La prise en compte des dépendances entre les termes de la requête (comme les mots composés, par exemple) permet d'augmenter la précision d'un système de RI. Plusieurs modèles probabilistes ont été proposés (Fagan, 1987 ; Song et Croft, 1999 ; Metzler et Croft, 2005), mais, dans tous les cas, le problème se résume à calculer des estimations précises des modèles de langue n-grammes (ou l'une de ses variantes), à savoir des modèles de langue où la distribution de probabilité d'un terme dépend d'une séquence finie de termes le précédant.

La prise en compte des relations sémantiques (comme la synonymie) augmente le rappel en permettant de retrouver des documents pertinents qui ne contiennent pas exactement les termes de la requête, mais utilisent des termes sémantiquement liés. Ceci est particulièrement important en raison de l'asymétrie entre les documents et les requêtes. Deux approches différentes sont principalement utilisées pour faire face à ce problème. La première consiste à utiliser le (pseudo) retour de pertinence pour ajouter à la requête des termes qui ne figuraient pas initialement. La seconde utilise des modèles distribués, comme l'indexation sémantique latente (Deerwester, Scott *et al.*, 1990), où les termes et les documents sont représentés dans un espace latent, qui peut être probabiliste ou vectoriel. Cependant, aucune de ces approches ne prend en compte les dépendances entre les termes.

Dans cet article, nous montrons que les modèles de langue neuronaux, qui exploitent une représentation distribuée des mots dans un espace continu, gèrent naturellement les dépendances entre termes, et ont donc un potentiel intéressant en RI. Les modèles de langue neuronaux (Bengio *et al.*, 2003) ont été utilisés avec succès dans de nombreuses tâches de traitement du langage naturel. Leur atout principal réside dans leur capacité à considérer conjointement les longues dépendances entre termes et les relations sémantique des mots. Cependant, ces modèles de langue requièrent un très grand nombre de données pour être appris efficacement, et la plupart des ces travaux se sont concentrés sur la mise au point de modèles de langue génériques (i.e. d'un ensemble de documents). Cela rend alors impossible leur application directe aux tâches de RI ad-hoc. L'apprentissage par maximisation de vraisemblance d'un modèle individuel pour chaque document de la collection, comme c'est le cas dans le cadre des approches de RI basées sur des modèles de langues n-grammes classiques, ne permet pas l'estimation correcte de paramètres, puisque issus d'un volume trop restreint d'observations de dépendances.

Une alternative intéressante a été proposée par (Le et Mikolov, 2014) qui ont récemment publié un modèle de langue neuronal dans lequel ils suggèrent de représenter le contexte (un document ou un paragraphe) comme un vecteur qui modifie le modèle de langue, ce qui évite la coûteuse construction d'un modèle individuel pour chaque document considéré et permet de tirer parti de dépendances générales observées sur le corpus. Notre travail est basé sur leurs conclusions. Nous suivons une approche dans laquelle les dépendances entre mots sont modifiées par les représentations des documents considérés. Nos contributions sont les suivantes : (i) Nous généralisons le modèle proposé dans (Le et Mikolov, 2014), en définissant de nouvelles manières de prendre en compte les spécificités individuelles de chaque document ; (ii) Nous appliquons ce modèle aux tâches ad-hoc de recherche d'information.

2. Travaux connexes

Les premières approches pour le traitement des dépendances entre les termes dans le domaine de la RI consistaient à étendre le modèle de représentation de texte dit en « sac de mot », en incluant les bigrammes dans le vocabulaire. Une telle approche a été suivie pour les modèles vectoriels (Fagan, 1987) puis pour les modèles de langue (Song et Croft, 1999 ; Srikanth et Srihari, 2002 ; Gao *et al.*, 2004), où les auteurs ont proposé d'utiliser un mélange des modèles de langue unigramme et bigramme, la différence demeurant dans la manière d'estimer le modèle de langue bigramme ou la manière de sélectionner les bigrammes. Cette approche n'a pas eu le succès escompté, probablement parce que cela entraîne l'utilisation d'observations peu nombreuses (Zhai, 2008) menant à des estimations inexactes des probabilités de bigramme. Dans cet article, nous étudions une autre manière de modéliser les distributions de n-grammes, avec n non limité à 1 ou 2.

Des modèles probabilistes plus sophistiqués ont été proposés pour prendre en compte des dépendances plus longues : Les modèles de proximité orientés région combinent le score de plusieurs modèles (Metzler et Croft, 2005 ; Bendersky et Croft, 2012 ; Blanco et Boldi, 2012), chacun traitant d'une dépendance spécifique entre les termes de la question. Ces travaux sont en quelque sorte orthogonaux au nôtre, puisque nous ne cherchons pas à combiner différentes sources d'informations, mais plutôt à examiner si un modèle de langue paramétrique peut capturer une séquence de termes caractéristiques.

L'une des techniques les plus utilisées pour répondre à la question de la discordance de vocabulaire est l'ajout (artificiel) de termes à la question basé sur un ensemble de (pseudo) documents pertinents fondé sur le pseudo-retour de pertinence (Manning *et al.*, 2008). Ce procédé a montré, dans certains cas, une amélioration des résultats de la recherche, au risque de dériver de la question initiale. L'emploi du pseudo-retour de pertinence est orthogonal à notre approche, mais pourrait être envisagé comme une extension pour estimer un modèle de langue des questions pertinentes (Lavrenko, 2010).

Une analyse globale peut être utilisée pour enrichir la représentation du document en s'appuyant sur la cooccurrence d'informations à l'échelle de la collection : Les techniques de réduction de dimension telles que LSI, ont été proposées pour répondre aux questions de discordance de vocabulaire (Deerwester, Scott *et al.*, 1990). L'idée est de représenter à la fois le document et la question dans un espace latent et de calculer un score de pertinence basé sur cette représentation. Toutefois, dans la pratique, ces modèles ne fonctionnent pas bien car de nombreux termes spécifiques au document sont considérés comme du bruit (Wang *et al.*, 2013). Il est donc nécessaire de combiner les scores de ces modèles latents avec les scores des modèles standards de RI tels que BM25 pour observer une amélioration de l'efficacité (Wang *et al.*, 2013 ; Deveaud *et al.*, 2013). Dans cet article, nous utilisons une combinaison d'un modèle de langue neuronal spécifique au document avec un modèle multinomial unigramme standard.

L'idée d'utiliser des réseaux de neurones pour construire des modèles de langue a émergé ces dix dernières années. Ce thème de recherche s'inscrit dans le récent domaine de « l'apprentissage de représentations ». Bengio et al. (Bengio *et al.*, 2003) présente l'un des premiers travaux sur les modèles de génération de texte (à l'échelle des mots) construit avec des réseaux de neurones. Ce type de modèle est caractérisé par une représentation distribuée des mots dans un espace vectoriel et par un état (un vecteur de $\mathbb{R}^n$) qui représente le contexte[1]. Cet état définit une distribution de probabilité sur les mots, et peut être mis à jour après chaque nouvelle observation (mot), permettant ainsi de définir un modèle de langue sur un texte complet. Cette manière de représenter du texte dans un espace latent a été largement explorée depuis avec des applications en détection de sentiment (Socher *et al.*, 2012) ou à la traduction automatique (Schwenk, 2013).

Plus proche de la RI, l'idée de représenter le contexte par un état dans un espace vectoriel a été exploitée par Palangi et al. (Palangi *et al.*, 2014) qui a proposé d'utiliser l'état obtenu à la fin du document (resp. de la question) comme une représentation vectorielle de l'ensemble du document (resp. de la question). Le score de pertinence est alors égal au cosinus entre le vecteur de la question et celui du document. Palangi et al. ont entraîné le modèle sur les clics des utilisateurs du moteur de recherche, et ont observé que leur modèle était en mesure de mieux ordonner les documents.

Contrairement à ces travaux, la représentation d'un document n'est pas le résultat d'un encodage successif du contexte mais conditionne le processus génératif, i.e. un jeu de paramètres est utilisé pour modifier un modèle probabiliste génératif préalablement appris.

Ces modèles probabilistes génératifs que nous appellerons *paramétriques* ont d'abord été appliqués dans le cadre de la reconnaissance du locuteur (Wilson et Bobick, 1999). Leur utilisation est nécessaire car ces systèmes doivent s'adapter assez rapidement à un nouveau locuteur. L'idée générale de ces modèles est de représenter le contexte comme un vecteur dans un espace vectoriel de petite dimension. Ce

1. ce contexte peut être calculé de façon récursive, en considérant l'état précédent ainsi que le mot courant, ou bien en utilisant une fenêtre avant le mot à prédire

vecteur modifie le modèle génératif appris sur l'ensemble des locuteurs. Comme la distribution de sortie dépend non seulement de l'état, mais aussi du contexte, un modèle peut exprimer un grand nombre de distributions de probabilité avec un nombre limité de paramètres supplémentaires.

Cette idée a été exploitée par Le et Mikolov (Le et Mikolov, 2014). Ils ont étudié les performances d'un modèle de langue paramétrique sur une tâche d'analyse des sentiments et une tâche de récupération de documents connexes, où les relations entre les questions sont modélisées par la distance entre leurs représentations respectives dans l'espace projeté considéré. Nous proposons dans cet article d'étendre cette approche en concevant des modèles plus génériques dédiés aux tâches ad-hoc de RI.

3. Réseaux de neurones pour la RI

Dans cette section, nous rappelons d'abord les principes des modèles de langue classiques de RI avant de présenter notre contribution. Comme la formulation de la plupart des modèles fait appel aux séquences, pour clarifier et raccourcir les notations, nous définissons $X_{i...j}$ comme la séquence $X_i, X_{i+1}, \ldots, X_{j-1}, X_j$ et supposons que la séquence est vide quand $i > j$.

Les modèles de langue sont des modèles probabilistes générant du texte – considéré comme une séquence de termes. Si un texte est composé d'une séquence de termes $t_{1...n}$, où chaque t_i correspond à un mot dans un vocabulaire prédéfini, on peut calculer la probabilité $P\left(t_{1...n}|M\right)$ d'observer cette séquence étant donné le modèle de langue M. Appliqué à la RI, l'approche la plus courante (Zhai, 2008) consiste à faire l'hypothèse que la pertinence d'un document d pour une question q est égale à la probabilité que le modèle de langue du document M_d génère la question.

$$P\left(d \text{ pertinent pour } q\right) = P\left(q|\, M_d\right)$$

où M_d est le modèle de langue du *document* à proprement parler, qui est le modèle (appartenant à la famille de modèle $\mathcal{M}$) maximisant la probabilité d'observer le document d composé des termes $d_{1...N}$. On a ainsi :

$$M_d = \underset{M \in \mathcal{M}}{\operatorname{argmax}}\, P\left(d|M\right) = \underset{M \in \mathcal{M}}{\operatorname{argmax}} \sum_{i=1}^{N} \log P\left(d_i|d_{1...i-1}, M\right) \qquad [1]$$

Parmi les différentes familles de modèles génératifs, la famille des modèles de n-grammes multinomiaux est la plus utilisée en RI, avec n habituellement égal à 1. Les modèles multinomiaux suivent l'hypothèse de Markov en supposant que l'occurrence d'un terme ne dépend que des $n-1$ termes précédents.

Avec $n = 1$, nous obtenons un modèle unigramme simple, qui ne tient pas compte du contexte de l'expression (à noter que dans ce cas le dénominateur est égal à la longueur du document). Par exemple, dans un document à propos de Boston, le terme « sentier » est plus susceptible de se produire avant le mot « liberté » que dans d'autres

documents. Il est important de prendre en compte cette information pour construire des modèles de RI plus précis, puisqu'un *bon* modèle pour un document traitant de Boston donnerait une probabilité plus élevée au terme « sentier » lorsqu'il apparaît avant le mot « liberté », et les documents correspondants obtiendraient donc un score plus élevé avec les questions contenant la séquence « le sentier de la liberté ». Des travaux comme ceux de (Song et Croft, 1999) ont exploré l'utilisation de modèles de langue avec $n > 1$. Dans ce cas, les modèles sont capables de capturer les dépendances entre termes, mais généralement au prix d'une complexité plus élevée et d'une perte des capacités de généralisation, en raison de la rareté des données – plus les séquences sont longues, plus elles sont improbables dans un document d, quand bien même ces séquences sont fortement liées au contenu de d du point de vue de l'utilisateur. Avec $n \geq 2$, les probabilités estimées sont dans la plupart des cas égales à 0.

Même pour les unigrammes ($n = 1$), l'estimation donnée par la méthode du maximum de vraisemblance peut être mauvaise et ignorer des documents simplement parce qu'ils ne contiennent pas un terme de la question, alors même qu'ils contiennent plusieurs occurrences de tous les autres. Pour éviter ce type de problèmes, des techniques de lissage sont utilisées pour empêcher la probabilité d'être nulle (et donc qu'un document ait un score égal à 0) en mélangeant le modèle de langue du document avec le modèle de langue de la collection [2] noté M_C. Un modèle de langue de la collection M_C correspond au modèle de langue qui maximise la probabilité d'observer les séquences de termes contenus dans les documents de la collection C.

On utilise classiquement le lissage de Jelinek-Mercer qui consiste en un mélange du modèle de langue du document et du modèle de langue de la collection. Étant donné un coefficient de lissage $\lambda \in [0, 1]$, le modèle de langue du document devient :

$$P(t_i|t_{i-n+1\ldots i-1}, \lambda, d, C) = \\ (1-\lambda)P(t_i|t_{i-n+1\ldots i-1}, M_d) + \lambda P(t_i|t_{i-n+1\ldots i-1}, M_C) \quad [2]$$

Mais même pour de faibles valeurs de n, le lissage à partir de la collection pourrait ne pas être efficace. Dans cet article, nous proposons de développer de nouveaux modèles de langue incluant des méthodes de lissage plus sophistiquées qui sont en mesure de traiter à la fois de longues dépendances entre les termes et le problème de discordance de vocabulaire.

3.1. *Modèles de langue neuronaux pour la RI*

Les représentations distribuées de mots et de documents sont connues depuis longtemps en RI (Deerwester, Scott *et al.*, 1990). Elles permettent de surmonter le problème de la parcimonie des données que nous venons d'évoquer en s'appuyant sur les relations spatiales entre les objets représentés. Cela a été exploité en RI pour faire

2. Il peut s'agir de la collection contenant le document, ou d'une toute autre collection de documents.

face au problème de discordance de vocabulaire, mais l'idée de tirer parti de ce type de représentation pour les modèles de langue est plus récente (Bengio *et al.*, 2003). Ces modèles sont construits en utilisant des réseaux de neurones (d'où leur appellation) et offrent plusieurs avantages du fait de l'utilisation d'un espace continu : (i) la possibilité de considérer des dépendances plus longues ($n > 2$) ; (ii) la disparition des problèmes liés à l'estimation des probabilités pour les mots peu fréquents dans un document.

Dans cet article, nous proposons d'inclure un tel modèle de langue dans la formule classique de calcul de la probabilité de chaque terme d'une question de RI. Nous proposons ainsi de remplacer le modèle de langue général de l'équation 2 par un modèle de langue neuronal *spécifique au document* d :

$$P(t_i|t_{i-n+1\ldots i-1}, d) \quad = \quad (1 \ - \ \lambda)P_U(t_i|d) \ + \ \lambda P_{NN}(t_i|t_{i-n+1\ldots i-1}, d) \quad [3]$$

où $P_{NN}(t_i|t_{i-n+1\ldots i-1}, d)$ est la probabilité d'observer le terme t_i après la séquence $t_{i-n+1\ldots i-1}$ dans le document d selon notre modèle de réseau de neurones et P_U est le modèle de langue uni-gramme. Ceci revient à introduire les dépendances entre les termes et la proximité sémantique dans un modèle de langue uni-gramme classique qui ne seraient pas en mesure de capturer de telles relations. Idéalement, nous aimerions que les meilleures performances soient atteintes pour $\lambda = 1$, mais dans la pratique la mixture permet de conserver une probabilité plus forte pour les termes spécifiques au document. Nous discutons ce point dans la conclusion de cet article.

Deux types de modèles sont détaillés ci-après : (i) Un modèle de langue neuronal *général* défini sur l'ensemble de la collection (section 3.1.1) ; (ii) Un modèle de langue neuronal *spécifique* au document estimé sur l'ensemble de la collection et le document étant classé (section 3.1.2). Notez que dans ce cas, nous sommes intéressés par la performance du modèle lorsque λ est proche de 1 (idéalement 1) car cela signifie que le modèle spécifique au document est suffisamment précis pour représenter pleinement le document. Ces deux modèles sont représentés dans la figure 1, où la partie noire correspond à la partie commune aux deux modèles et les parties vertes et bleues représentent respectivement le modèle général et celui spécifique au document.

3.1.1. *Modèle de langue neuronal général*

Il existe deux types de modèles de langue neuronal. Ceux qui prennent en compte un contexte virtuellement infini (réseaux récurrents) et ceux qui ne prennent en compte qu'un nombre limité de termes précédents (réseaux de convolution). Les travaux que nous présentons sont basés sur ces derniers, mais il serait intéressant d'étudier dans le futur des approches basées sur des réseaux récurrents.

Dans notre cas, l'entrée du réseau de neurones (Figure 1) correspond aux $(n-1)^{\text{ème}}$ termes précédents. Pour les $n-1$ premiers mots d'un document, nous utilisons un terme spécial dit de « remplissage » (*padding*). À chaque terme t_i correspond un vecteur $z_i^t \in \mathbb{R}^{m_0}$. Les $n-1$ vecteurs sont ensuite transformés par une fonction ϕ (décrite ci-dessous) en un vecteur d'état s dans $\mathbb{R}^{m_f}$ où f est la dernière couche

du réseau de neurones correspondant au calcul du vecteur de contexte. Ce vecteur sert d'entrée au classifieur (HSM sur la figure 1) qui calcule la probabilité d'observer chacun des termes du vocabulaire en fonction de ce vecteur de contexte.

Lors de nos expériences, nous avons utilisé trois architectures de réseaux de neurones, i.e. trois différentes fonctions pour ϕ. Ces trois modèles se distinguent par une complexité croissante, et devraient donc être à même de modéliser des dépendances plus fines.

3.1.1.1. Model 1 (M1) : linear$(m_1) - \tanh$

La première couche transforme linéairement les $n-1$ vecteurs $z_j \in \mathbb{R}^{m_0}$ en un vecteur d'état $s \in \mathbb{R}^{m_1}$. Autrement dit, on a :

$$l_1 = \sum_{j=1}^{n-1} A_j z_j + b$$

où les A_j sont des matrices de dimensions $m_1 \times k$ et b un vecteur de biais dans $\mathbb{R}^{m_1}$. La deuxième couche introduit une non-linéarité en calculant la tangente hyperbolique ($\tanh$) de chaque composante de son entrée :

$$\forall j \; l_{2j} = \tanh(l_1 j)$$

Dans ce modèle, la fonction ϕ a $(n-1) \times m_0 \times m_1$ paramètres.

3.1.1.2. Model 2 (M2) : linear$(m_1) - \tanh -$linear$(m_2) - \tanh$

La deuxième fonction ϕ que nous considérons est une extension de la première à laquelle nous ajoutons une couche linéaire (la matrice B de dimensions $m_2 \times m_1$) et une couche non-linéaire ($\tanh$). Dans ce modèle, la fonction ϕ a $(n-1) \times m_0 \times m_1 + m_1 \times m_2$ paramètres.

3.1.1.3. Model 3 (M2Max) : linear$(\kappa m_1) - \max(\kappa) -$ linear$(m_2) - \tanh$

Pour construire le troisième modèle, nous remplaçons la deuxième couche (tangente hyperbolique) du modèle précédent par une autre fonction non-linéaire : une couche de regroupement maximale (*max-pooling*). Les couches de regroupement maximales sont utiles aux réseaux de neurones profonds, car elles introduisent un invariant (Kalchbrenner *et al.*, 2014), et permettent d'apprendre plus facilement le contexte dépend par exemple de deux mots qui peuvent être séparés par un espace ("le vase est bleu" et "le grand vase est bleu"). Cette couche est définie par un paramètre κ (valant 4 dans nos expériences)

$$\text{max-pooling}_\kappa(x)_j = \max\{x_{\kappa \times (j-1)}, \ldots, x_{\kappa \times j - 1}\}$$

Dans ce modèle, la fonction ϕ a $(n-1) \times \kappa \times m_0 \times m_1 + m_1 \times m_2$ paramètres.

Pour les trois modèles, d'une séquence de $n-1$ termes $(t_{1 \ldots n-1})$, nous obtenons ensuite un vecteur $\phi(z_{1 \ldots n-1})$ résumant l'information. Chaque vecteur z_i est ensuite

utilisé pour calculer la distribution de probabilité sur les termes $p(t|z_i)$ – la probabilité que chaque mot du vocabulaire apparaisse après une séquence donnée de $n-1$ termes.

Dans notre modèle, nous utilisons une fonction exponentielle normalisée hiérarchique (HSM) qui permet de calculer la probabilité d'une classe (sachant le contexte) de manière efficace lorsque le nombre de classes est grand (Morin et Bengio, 2005). Le HSM est composé d'un arbre (binaire) dont les feuilles correspondent aux mots du vocabulaire. Cette fonction est définie par :

$$HSM_t(v) = \prod_{s \in \text{path}(t)} \frac{1}{1 + \exp(b_s(t) \times x_s \cdot v)} \quad [4]$$

où HSM_t représente la composante correspondant au mot t dans la distribution encodée par la couche HSM du réseau de neurones, v correspond au vecteur d'entrée de la fonction, path(t) correspond à l'ensemble des nœuds qui forment le chemin menant à la feuille qui représente le mot t, x_s est le vecteur attaché au nœud interne s de l'arbre et $b_s(t)$ vaut -1 (resp. 1) si le chemin vers le mot t emprunte la branche gauche (resp. droite) du nœud s. Dans nos expériences, l'arbre de la fonction HSM est un arbre d'Huffman, mais d'autres choix sont possibles (en particulier, il serait intéressant de grouper ensemble les mots proches sémantiquement).

Ceci nous permet de calculer facilement la distribution de probabilité conditionnelle $P_{NN}(t|t_{1\ldots n-1})$ pour le prochain nœud $w \in \Omega$, connaissant la précédente séquence des $n-1$ mots observés :

$$P_{NN}(t|t_{1\ldots n-1}) = HSM_t(\phi(t_{1\ldots n-1})) \quad [5]$$

Enfin, pour obtenir un modèle opérationnel capable de calculer la probabilité de générer une question donnée, par rapport à notre modèle générique, il est nécessaire d'apprendre la représentation des mots, les paramètres du HSM et les paramètres de chaque couche. Cet ensemble de paramètres est noté θ. Le problème d'apprentissage peut donc être formulé comme la maximisation de la probabilité d'observer le document d dans la collection $\mathcal{D}$ de documents :

$$\Theta^* = \underset{\Theta}{\text{argmax}} \sum_{d \in \mathcal{D}} \sum_{i=0}^{|d|} w(d_i) \log\ HSM_{d_i}(\phi(z^d_{i-n+1\ldots i-1})) \quad [6]$$

où $w(d_i)$ est le poids utilisé pour minimiser l'importance des termes fréquemment rencontrés dans la collection (Mikolov *et al.*, 2013).

3.1.2. *Modèle de langue neuronal dépendant du document*

Le modèle de langue neuronal générique présenté précédemment prend en compte les dépendances et les relations sémantiques entre les mots pour l'ensemble des documents de la collection. Ce modèle de langue peut être une bonne alternative au modèle de langue uni-gramme multinomial de la collection utilisé pour le lissage (Equation [2]). Cependant, nous pensons que la prise en compte des spécificités du

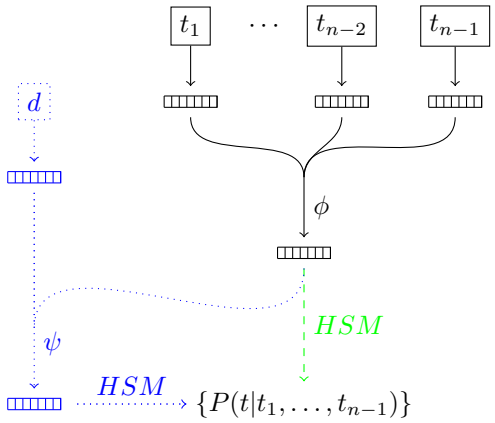

Figure 1. *Architecture générale des modèles de langue neuronaux générique (en noir et vert hachuré) et spécifique au document (en noir et bleu pointillé).*

document en cours de classement doit conduire à un meilleur modèle de langue, et donc à de meilleurs résultats de recherche, car les termes plus spécifiques au document devraient obtenir des probabilités plus fortes.

Apprendre un modèle de langue neuronal spécifique pour tous les documents de la collection est impossible pour les mêmes raisons que pour les modèles de langue à n-gramme avec $n > 1$: l'apprentissage des dépendances entre termes sur les séquences contenues dans un unique document serait sujet au sur-apprentissage, puisque l'échantillon de n-grammes est faible. Pour surmonter ce problème, nous suivons l'approche de (Le et Mikolov, 2014), où la distribution de probabilité d'un modèle de langue neuronal générique est modifiée par un petit ensemble de paramètres qui peuvent être appris de manière plus fiable à partir de l'observation d'un seul document. La dimension d'un tel vecteur (de 100 à 200) est typiquement beaucoup plus petite que le nombre de paramètres d'une distribution multinomiale classique (taille du vocabulaire). Cette approche est intéressante, car elle permet d'abord apprendre un modèle de langue sur l'ensemble de la collection, et ensuite d'apprendre seulement *à le modifier* pour un document spécifique. L'emploi de paramètres pour modifier le comportement d'un modèle probabiliste génératif a été utilisé dans de nombreux travaux en traitement du signal, comme la reconnaissance des gestes (Wilson et Bobick, 1999), où le modèle doit être rapidement adapté à un utilisateur spécifique. Ces modèles bénéficient d'une grande source d'informations (la collection de tous les gestes ou de tous les documents dans notre cas) pour être assez généraux et en même temps peuvent être rendus suffisamment spécifiques pour décrire un seul utilisateur ou un document.

La modification du modèle de langue neuronal est présentée dans la figure 1 par la partie bleue en pointillé. Un vecteur $z_d \in \mathbb{R}^{m_f}$ spécifique au document est utilisé pour modifier le vecteur représentant le contexte du prochain mot au sein du modèle

de langue générique appris sur toute la collection. Le vecteur ainsi modifié est ensuite utilisé pour générer une distribution de probabilité en utilisant un HSM, comme pour le modèle de langue générique.

Dans cet article, nous considérons deux opérations de « fusion » $\psi : \mathbb{R}^{m_f} \times \mathbb{R}^{m_f} \rightarrow \mathbb{R}^{m_f}$ qui associent le vecteur d'état s donné par ϕ et le vecteur z_d spécifique au document à :

– leur somme, à savoir $\psi(s, z_d) = s + z_d$

– leur produit composante par composante, c'est-à-dire $\psi(s, z_d) = s \odot z_d$

La première est celle utilisée dans (Le et Mikolov, 2014), et correspond à un déplacement dans l'espace thématique afin de se rapprocher des mots spécifiques au document. La seconde suppose que certains composants du vecteur d'état sont liés à des champs sémantiques différents – si certaines composantes de z_d sont nulles, alors les champs sémantiques correspondants ne sont pas considérés.

Ces deux fonctions sont encore simples, mais elles peuvent sensiblement modifier la répartition du modèle de langue. En reprenant l'exemple sur Boston, les composantes de z_d biaiseraient le modèle pour les mots susceptibles de se produire dans ce type de documents, et ainsi augmenteraient la probabilité de trouver « chemin » avant « liberté ». Ce biais est obtenu en transformant le vecteur d'état afin qu'il soit plus orthogonal aux vecteurs du HSM qui mènent au mot « chemin » qu'à ceux qui mènent aux autres mots associés à « liberté ».

Notez que, comme la solution à notre problème d'optimisation n'est pas unique, des solutions équivalentes (vis à vis de la fonction de coût) pourraient être obtenues par rotation – ce qui, par conséquent, aurait un impact sur le bénéfice de l'utilisation de ce type de modifications. Nos expériences montrent cependant qu'il y a un gain associé à l'utilisation de fonctions, même simples, comme la multiplication ou l'addition terme à terme. Nos futurs travaux exploreront des transformations du vecteur d'état plus sophistiquées et plus appropriées.

En théorie, ce modèle de langue spécifique au document pourrait être utilisé seul (c.-à-d. sans lissage, car il est basé sur un modèle de langue général) pour estimer la probabilité de générer une question. En pratique, comme montré dans les expériences ci-dessous, le modèle n'est pas encore suffisamment puissant pour y parvenir. Toutefois, combiné avec un modèle unigramme multinomial classique telle que proposé par l'équation 3, il permet d'observer des améliorations intéressantes pour les tâches ad-hoc de RI. C'est une première étape vers une solution formelle pour la prise en compte des discordances de vocabulaire et des dépendances entre les termes.

4. Expériences

Nous avons utilisé les collections TREC-1 à TREC-8 pour nos expériences. Le modèle de base est BM25 (Robertson et Zaragoza, 2009) avec les paramètres usuels ($k_1 = 1.2$ and $b = 0.5$).

Nom	Modèle	ϕ	word/HSM
M1	linéaire(100) - tanh	40,000	75,043,700
M2	linéaire(100) - tanh - linéaire(100) - tanh	50,000	75,043,700
M2Max	linéaire(400) - max(4) - linéaire(100) - tanh	170,000	75,043,700

Tableau 1. *Modèles et nombre de paramètres pour la fonction ϕ (3ème colonne) et la représentation des mots/le classificateur ()*

Pour apprendre les modèles de langue, nous avons pré-traité l'ensemble des collections de documents de chaque TREC [3] en utilisant le stemmer de Porter et nous n'avons pas utilisé d'anti-dictionnaire. En revanche, nous avons retiré les mots apparaissant moins de 5 fois dans le jeu de données, car il est difficile d'apprendre une représentation fiable pour des termes apparaissant peu de fois. Ces derniers sont capturés par le modèle de langue uni-gramme (terme P_U de l'équation 3). La taille du vocabulaire ainsi obtenu est de 375 219 mots. Nous avons utilisé word2vec (Mikolov *et al.*, 2013)[4] pour pré-calculer les représentations des mots et les paramètres du HSM. Les paramètres de word2vec sont ceux par défaut (fenêtre de taille 5, pas d'apprentissage de 0.5). Nous avons utilisé 5 itérations sur le corpus afin d'apprendre une représentation fiable des mots apparaissant un petit nombre de fois.

Nous avons ensuite appris les modèles de langue génériques (Section 3.1.1) donnés dans le tableau 1. Nous pouvons voir également dans ce tableau que l'essentiel des paramètres est lié à la représentation des mots.

Lors de l'apprentissage, nous avons utilisé une descente de gradient stochastique par lots, avec un pas de gradient décroissant en fonction du nombre d'itérations :

$$\epsilon_k = \frac{\epsilon_0}{1 + k \times \delta}$$

où $\epsilon_0 = 0.1$ et $\delta = 2e - 4$. Le nombre d'itération a été fixé de manière empirique à 100000, ce qui correspond à (1) avoir vu cinq fois chaque mot du corpus (2) une vraisemblance qui n'évolue presque plus.

Pour chaque question, nous avons utilisé BM25 pour sélectionner 100 documents. Pour ces 100 documents, les paramètres spécifique au document z_d ont été appris. Nous avons également utilisé une descente de gradient – dans ce cas, RProp (Riedmiller et Braun, 1993) a été utilisé, car il permet une convergence rapide sur un petit ensemble de paramètres (pas initial fixé à 0.1). Nous avons itéré jusqu'à obtenir une différence entre les paramètres décrivant le document de deux itérations successives inférieur à $1e - 4$.

3. Les besoins du HSM décide de la taille minimale du corpus (Chen *et al.*, 2015)

4. Le code source est disponible à l'adresse suivante `https://code.google.com/p/word2vec/`

Les résultats sont donnés Figure 2 où nous comparons les différents modèles proposés avec le modèle de langue basé sur un lissage Jelinek-Mercer ainsi que BM25, pour les métriques MAP et G-MAP (moyenne géométrique des précisions moyenne). La métrique G-MAP est intéressante car elle est plus sensible à l'amélioration des questions les plus difficiles que la métrique MAP.

Les neufs modèles sont indiqués en utilisant le nom du modèle (M1, M2 ou M2max) suivi de la méthode utilisée pour fusionner le vecteur de contexte avec le vecteur représentant le document (# correspond au modèle neutre – sans adaptation au document, $+$ au modèle additif et $*$ au modèle multiplicatif). Les abscisses correspondent à la valeur de λ (équations 2 et 3).

Les résultats montrent que les performances des différents modèles de langue (neuronaux ou classiques) sont proches, et qu'en fonction du jeu de test les conclusions sont inversées. Il n'y a donc pas encore d'amélioration liée à l'utilisation de ces modèles. Des études qualitatives ont montré que (1) les mots très spécifiques au document mais apparaissant peu dans le corpus ont des probabilités qui sont sous-évaluées par les modèles neuronaux, des modèles permettant de modifier de manière plus fine le contexte sont donc nécessaire ; et (2) le classifieur HSM a des problèmes pour les mots dont la fréquence d'occurrence est faible. Des expériences complémentaires sont nécessaires pour résoudre ces problèmes.

Les autres observations que l'on peut faire au niveau des résultats est que l'influence des modèles neuronaux est plus grande sur les questions difficiles (la différence pour le G-MAP est plus importante que la différence pour le MAP). Nous espérons donc qu'en ayant un modèle plus fin pour les documents, cette différence va s'accentuer.

Finalement, les modèles plus complexes semblent avoir de meilleurs performances : M2 et M2Max sont souvent supérieurs ou comparables au modèle M1. Utiliser des modèles plus complexes, et des modèles récursifs, peut potentiellement être également une source d'amélioration des résultats.

5. Conclusion

Dans cet article, nous avons proposé d'utiliser des modèles de langue neuronaux paramétriques pour la recherche d'information. Nous avons proposé pour chaque modèle deux variantes, une correspondant au modèle de langue général (collection), ainsi qu'un modèle spécifique à un document décrit par un petit nombre de paramètres modifiant le processus génératif. Ces paramètres peuvent donc être appris même pour des documents de petite taille.

Les expériences conduites avec les collections TREC-1 à TREC-8, où nous avons comparé un modèle de langue classique avec les modèles neuronaux, montrent que les modèles actuels n'apportent pas d'amélioration en performance. Des analyses qualitatives des résultats montrent que l'utilisation de tels modèles demande à résoudre des

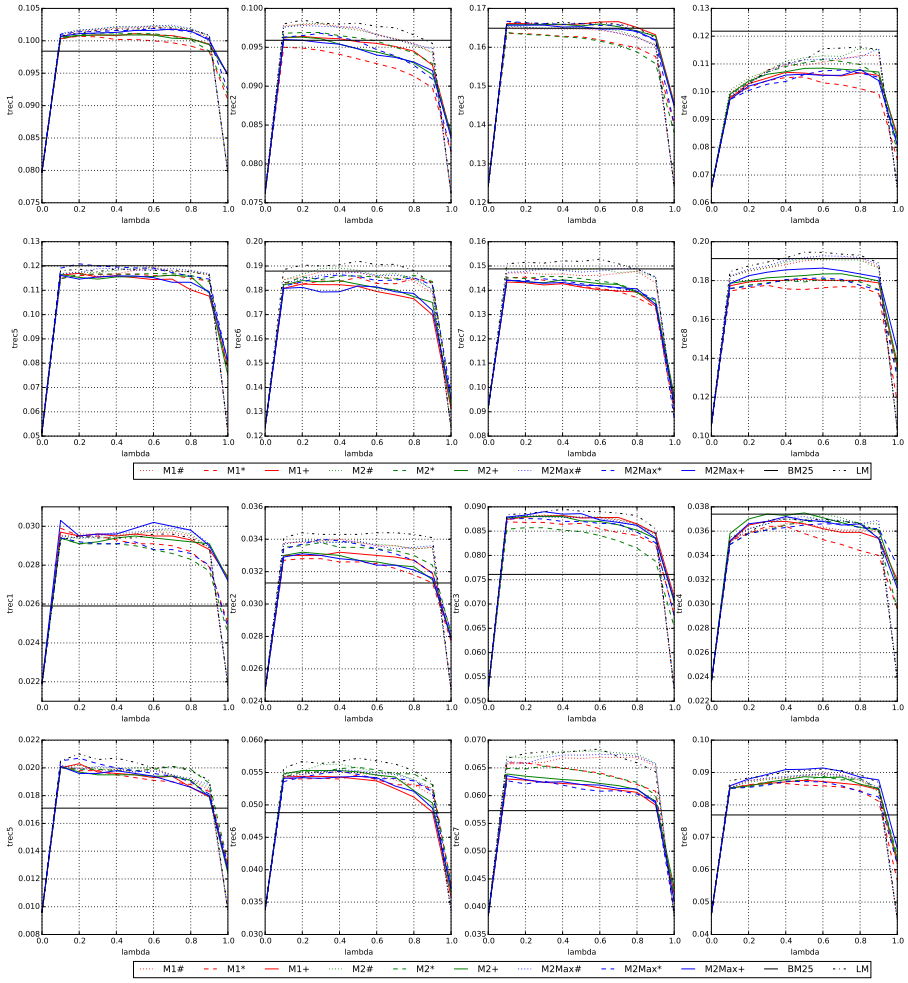

Figure 2. *MAP (haut) et G-MAP (bas) pour TREC-1 à TREC-8. La performance de BM25 est donnée pour référence (courbe constante), et les résultats pour les différents modèles de langue sont donnés en fonction de* λ*. Pour les modèles basés sur les réseaux de neurones, les résultats des trois modèles M1, M2 et M2Max sont donnés pour les trois variantes : # correspond au modèle général, + (resp. *) pour le modèle spécifique basé sur l'addition (resp. la multiplication).*

problèmes qui ne se posent pas pour d'autres utilisations des modèles neuronaux. En particulier, la probabilité des mots spécifiques au document est sous-évaluée, ce qui explique pourquoi le modèle ne se comporte pas bien lorsqu'il est utilisé sans lissage : il parait donc nécessaire d'utiliser des transformations plus complexes.

Bien que ces résultats ne montrent pas de différence substantielle, nous pensons que de tels modèles sont intéressant car l'idée de modifier un modèle génératif par un petit ensemble de paramètres décrivant un contexte particulier (document, paragraphe, phrase, etc.) permet d'envisager d'avoir à terme des modèles prenant en compte à la fois les spécificités d'un document et un modèle de langue général. Nous pensons que, outre l'étude de nouveaux opérateurs permettant de modifier le processus génératif, il serait important d'utiliser des techniques de pseudo-retour de pertinence, qui permettrait de ne plus avoir de problème d'asymétrie entre la question et les documents.

6. Bibliographie

Bendersky M., Croft W. B., « Modeling higher-order term dependencies in information retrieval using query hypergraphs », *SIGIR '12*, 2012.

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

Blanco R., Boldi P., « Extending BM25 with multiple query operators », *SIGIR '12*, 2012.

Chen W., Grangier D., Auli M., « Strategies for Training Large Vocabulary Neural Language Models », *CoRR*, 2015.

Deerwester, Scott, Dumais, Susan T, Furnas, George W, Landauer, Thomas K, Harshman, Richard, « Indexing by latent semantic analysis », 1990.

Deveaud R., SanJuan E., Bellot P., « Unsupervised Latent Concept Modeling to Identify Query Facets », *OAIR'13*, 2013.

Fagan J. L., « Automatic Phrase Indexing for Document Retrieval : An Examination of Syntactic and Non-Syntactic Methods. », *SIGIR'87*, 1987.

Gao J., Nie J.-Y., Wu G., Cao G., « Dependence language model for information retrieval », *SIGIR'04*, 2004.

Kalchbrenner N., Grefenstette E., Blunsom P., « A Convolutional Neural Network for Modelling Sentences », *arXiv.org*, 2014.

Lavrenko V., *A Generative Theory of Relevance*, Springer, 2010.

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

Manning C. D., Raghavan P., Schütze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

Metzler D., Croft W. B., « A Markov random field model for term dependencies », *SIGIR*, 2005.

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

Morin F., Bengio Y., « Hierarchical Probabilistic Neural Network Language Model. », *AISTATS 2005*, 2005.

Palangi H., Deng L., Shen Y., Gao J., He X., Chen J., Song X., Ward R., « Semantic Modelling with Long-Short-Term Memory for Information Retrieval. », *arXiv.org*, 2014.

Riedmiller M., Braun H., « A direct adaptive method for faster backpropagation learning : the RPROP algorithm », *IEEE International Conference on Neural Networks*, 1993.

Robertson S. E., Zaragoza H., *The Probabilistic Relevance Framework : BM25 and Beyond*, Foundations and Trends in Information Retrieval, 2009.

Schwenk H., « Continuous Space Translation Models for Phrase-Based Statistical Machine Translation », *Proceedings of COLING 2012 : Posters*, 2013.

Socher R., Manning C. D., Huval B., Ng A. Y., « Semantic compositionality through recursive matrix-vector spaces », *EMNLP-CoNLL '12 : Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012.

Song F., Croft W. B., « A general language model for information retrieval », *CIKM'99*, 1999.

Srikanth M., Srihari R. K., « Biterm language models for document retrieval. », *SIGIR'02*, 2002.

Wang Q., Xu J., Li H., Craswell N., « Regularized Latent Semantic Indexing : A New Approach to Large-Scale Topic Modeling », *ACM TOIS*, 2013.

Wilson A. D., Bobick A. F., « Parametric hidden Markov models for gesture recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

Zhai C., « Statistical Language Models for Information Retrieval : A Critical Review. », *Foundations and Trends in Information Retrieval*, 2008.