TABLE DES MATIÈRES

1 CONTEXTE ET CONTRIBUTION DE LA THÈSE 1
1 Contexte et problématique . . . 1
1.1 Contexte de la thèse . . . 1
1.2 Problématique de la thèse . . . 3
2 Contributions . . . 5
3 Organisation du mémoire . . . 6

I SYNTHÈSE DES TRAVAUX DE L'ÉTAT-DE-L'ART 9

2 PRÉDICTION DE L'EMPLACEMENT À PARTIR DE FLUX D'INFORMATIONS DANS LES RÉSEAUX SOCIAUX 11
1 Introduction à la recherche d'information géographique . . . 14
1.1 Concepts fondamentaux . . . 14
1.2 Index et index géographique . . . 18
1.2.1 La nécessité d'indexer les documents . . . 18
1.2.2 Indexation avec des listes inversées . . . 19
1.2.3 Indexation spatiale . . . 20
1.2.4 Indexation spatio-textuelle . . . 22
1.3 Requête géographique . . . 25
1.4 Ordonnancement de pertinence . . . 26
1.4.1 Notion de pertinence pour la RIG . . . 26
1.4.2 Calculer et combiner la similarité spatiale . . . 28
1.5 Principale problématique et périmètre de la thèse . . . 29
2 Résolution de la portée géographique des géotextes . . . 32
2.1 Prédiction de l'emplacement du contenu généré par l'utilisateur . . . 33
2.1.1 Inférence de l'emplacement à partir du contenu . . . 34
2.1.2 Inférence de l'emplacement à partir du contexte . . . 35
2.2 Prédiction de l'emplacement mentionné . . . 36
2.2.1 Reconnaissance de l'emplacement mentionné . . . 37
2.2.2 Désambiguïsation de l'emplacement mentionné . . . 38
2.3 Prédiction sémantique de l'emplacement . . . 39
2.3.1 Appariement d'objets non-géotaggés . . . 40
2.3.2 Appariement d'objets géotaggés . . . 42
3 Discussion . . . 43

4 Conclusion 44

3 RÉSEAUX DE NEURONES POUR LA REPRÉSENTATION DISTRIBUÉE ET L'APPARIEMENT DE TEXTES ET DE GÉOTEXTES 47
1 Réseaux de neurones et apprentissage profond : concepts préliminaires 49
1.1 Concepts préliminaires 49
1.1.1 Neurone formel 49
1.1.2 Paramètres libres 50
1.1.3 Fonction de combinaison 51
1.1.4 Fonction d'activation 51
1.2 Réseau de neurones artificiels 52
1.3 Architectures populaires en recherche d'information 54
1.3.1 Réseau de neurones à convolution (CNN) 54
1.3.2 Réseau de neurones récurrents (RNN) 56
1.3.3 Transformer 57
1.4 Algorithmes d'apprentissage des modèles neuronaux 58
1.4.1 Fonction de coût 59
1.4.2 Rétropropagation du gradient 60
1.5 Surapprentissage et régularisation 62
2 Représentations distribuées de textes et de géotextes 63
2.1 Représentations distribuées de textes 64
2.1.1 Représentations distribuées des mots 64
2.1.2 Représentations distribuées des phrases 68
2.1.3 Apprentissage augmenté par des ressources externes 71
2.2 Représentations distribuées de géotextes augmentées par les contextes spatiaux 78
2.2.1 Représentations distribuées des mots 79
2.2.2 Représentations distribuées des géotextes 81
3 Réseaux de neurones profonds pour l'appariement de textes 84
3.1 Formulation unifiée des modèles d'ordonnancement 85
3.2 Modèles axés sur la représentation 87
3.3 Modèles axés sur l'interaction 90
4 Discussion 94
5 Conclusion 97

II CONTRIBUTIONS 99

4 RÉGULARISATION SPATIALE DE PLONGEMENTS LEXICAUX 101
1 Contexte et motivations 102
2 Problématiques et définitions 104
2.1 Concepts et définitions 104
2.1.1 Objets géotextuels 104

2.1.2 Distance géographique et répartition spatiale 106
2.2 Définition du problème 107
3 Apprentissage a posteriori des plongements lexicaux spatiaux 108
3.1 Aperçu général de la solution 108
3.2 Détection des sens locaux des mots 109
3.2.1 Méthode de clustering : algorithme des *k*-moyennes 110
3.2.2 Méthode probabiliste : méthode de Parzen-Rosenblatt111
3.3 Régularisation a posteriori des plongements lexicaux 116
3.3.1 Correction des plongements lexicaux 116
3.3.2 Intégration des contraintes spatiales 117
4 Évaluation expérimentale 121
4.1 Retour sur l'Hypothèse 1 (QR1) 122
4.2 Évaluation intrinsèque : similarité de POIs (QR2) 124
4.2.1 Cadre expérimental 124
4.2.2 Résultats 129
4.3 Évaluation extrinsèque : prédiction sémantique de l'emplacement (QR3) 131
4.3.1 Cadre expérimental 133
4.3.2 Résultats 135
5 Bilan 138

5 MODÈLE NEURONAL POUR LA PRÉDICTION SÉMANTIQUE DE L'EMPLACEMENT 141
1 Contexte et motivations 142
2 Définition du problème 146
3 Architecture du réseau de neurones 147
3.1 Aperçu général de la solution 147
3.2 Modélisation des interactions locales 149
3.2.1 Représentation des géotextes 149
3.2.2 Matrice des interactions locales 150
3.2.3 Représentations latentes des interactions locales . . 151
3.3 Modélisation des interactions globales 154
3.3.1 Interaction globale spatiale 154
3.3.2 Interaction globale textuelle 155
3.4 Calcul du score d'appariement 156
3.5 Apprentissage du modèle 156
4 Cadre expérimental 157
4.1 Jeux de données 158
4.2 Scénarios d'évaluation 159
4.3 Modèles de référence 161
4.4 Détails de mise en œuvre 163
4.4.1 Représentations distribuées des mots 163

4.4.2 Configuration des modèles proposés 164
4.4.3 Configuration des modèles de référence 164
4.4.4 Détails du protocole d'évaluation 165
5 Résultats de l'évaluation 166
5.1 Évaluation intrinsèque du facteur d'amortissement spatial **(QR1)** 166
5.1.1 Comparaison des distributions du facteur d'amortissement spatial. 167
5.1.2 Impact du facteur d'amortissement sur les similarités mot à mot 168
5.2 Analyse des scénarios (**QR2**) 170
5.2.1 Analyse de l'histogramme d'appariement 170
5.2.2 Effet des interactions locales 174
5.2.3 Effet des interactions globales 175
5.3 Comparaison des performances (**QR3**) 176
5.3.1 Analyse des performances globales 177
5.3.2 Analyse qualitative d'un échantillon de tweets . . . 180
5.4 Analyse de sensibilité des paramètres de **SGM** (**QR4**) 184
5.4.1 Impact du nombre de classes 184
5.4.2 Impact du rayon r 185
6 Bilan 185

III CONCLUSION 187

BIBLIOGRAPHIE 195

LISTE DES FIGURES

Figure 2.1 Latitude et longitude sur la Terre. 14
Figure 2.2 Distance du grand cercle entre Toulouse et Toronto. 16
Figure 2.3 Exemples de projections cartographiques. 17
Figure 2.4 Coordonnées des pays Européens dans le système UTM. . . 17
Figure 2.5 Exemple d'une liste inversée. 19
Figure 2.6 Exemple d'un index spatial quadratique. 21
Figure 2.7 Exemple d'un index spatial en R-arbre. 22
Figure 2.8 Exemple d'un index géographique. 23
Figure 2.9 Exemple d'un index primaire spatial. 24
Figure 2.10 Exemple d'un index primaire textuel. 24
Figure 2.11 Exemple d'une page Wikipedia avec différentes formes de géoréférences. 31
Figure 2.12 Illustration du contenu d'un tweet, du contexte et du réseau Twitter, ainsi que de trois types de lieux. 34
Figure 2.13 Architecture du modèle supervisé bayésien (sBM). 42
Figure 3.1 Exemple d'un neurone formel : le perceptron. 50
Figure 3.2 Courbes des fonctions d'activation courantes. 52
Figure 3.3 Architecture d'un perceptron multicouche. 53
Figure 3.4 Réseau de neurones à convolution LeNet-5. 55
Figure 3.5 Schéma d'un réseau de neurones récurrents. 56
Figure 3.6 Architecture générale d'un *transformer*. 58
Figure 3.7 Architecture des configurations du *Word2Vec*. 66
Figure 3.8 Architecture du modèle *ParagraphVector*. 70
Figure 3.9 Illustration de l'espace vectoriel transformé. 75
Figure 3.10 Illustration de l'approche *post-specialisation*. 78
Figure 3.11 Enrichissement d'un tweet avec des contextes géographiques situés à des rayons D croissants. 80
Figure 3.12 Entraînement des plongements lexicaux sur des collections différentes. 81
Figure 3.13 Illustration des étapes de génération de l'arbre binaire avec le modèle POI2Vec. 82
Figure 3.14 Illustration des couches *check-ins* et *texte* du modèle CAPE. . 84
Figure 3.15 Architecture générale décrivant un modèle neuronal unifié pour l'appariement de textes. 86

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Figure 3.16 Architectures générales des modèles neuronaux axés sur la représentation et l'interaction. 88
Figure 3.17 Architecture du modèle d'appariement Arc-I. 89
Figure 3.18 Architecture du modèle d'appariement DRMM. 92
Figure 3.19 Architecture du modèle d'appariement Arc-II. 93
Figure 4.1 Illustration des différents sens locaux du mot « *football* ». . . 103
Figure 4.2 Exemple d'un POI issu de Foursquare. 105
Figure 4.3 Exemple d'un tweet géotaggé. 105
Figure 4.4 Illustration de la répartition spatiale d'un mot et de son centre géographique. 106
Figure 4.5 Exemple du partitionnement en k-moyennes. 111
Figure 4.6 Carte de chaleur des densités calculées en utilisant la méthode d'estimation par noyau. 113
Figure 4.7 Exemple d'identification des sens locaux d'un mot. 115
Figure 4.8 Exemple de sélection des mots proches et des mots distants (k-moyennes). 119
Figure 4.9 Exemple de sélection des mots proches et des mots distants (KDE). 121
Figure 4.10 Similarité du cosinus des plongements lexicaux traditionnels et amortis. 123
Figure 4.11 Aperçu de la tâche d'évaluation sur Amazon M. Turk. . . . 125
Figure 4.12 Similarité du cosinus des représentations distribuées régularisées. 138
Figure 5.1 Exemple d'un tweet associé à un POI et ses candidats. . . . 142
Figure 5.2 Analyse des scores de pertinences thématique et spatiale en utilisant la vérité terrain de paires tweet-POI. 144
Figure 5.3 Architecture du modèle **SGM**. 148
Figure 5.4 Illustration de la matrice de représentation d'un tweet. . . . 150
Figure 5.5 Carte de chaleur des concentrations de POIs dans les villes de New York et de Singapour. 159
Figure 5.6 Comparaison des distributions de $F(d^*)$. 167
Figure 5.7 Distribution des coefficients de corrélation de Spearman calculés entre les variables de rang rgI_l et rgI'_l. 168
Figure 5.8 Visualisation des poids $\mathbf{W}^{(2)}$ associés aux histogrammes d'interaction pour le modèle **SGM**. 173
Figure 5.9 Évolution de l'*Acc@k* pour le modèle **SGM** et une sélection de modèles de référence. 178
Figure 5.10 Évolution de l'*Acc*@1 du modèle $\mathbf{SGM}_{I_l}$ selon le nombre de classes. 184

LISTE DES TABLEAUX

Tableau 2.1 Critères d'évaluation de la pertinence géographique. 28
Tableau 3.1 Équations et dérivées des fonctions d'activation courantes. . 51
Tableau 3.2 Liste des fonctions objectifs courantes. 59
Tableau 4.1 Configuration des scénarios utilisés pour l'évaluation intrinsèque. 127
Tableau 4.2 Comparaison de l'efficacité des plongements lexicaux spatiaux et des modèles de référence sur la tâche de similarité des POIs (corrélation de Spearman ρ). 129
Tableau 4.3 Configuration des scénarios utilisés pour l'évaluation extrinsèque. 134
Tableau 4.4 Comparaison des performances des modèles **CM-W**$^{s}_{kde}$ et **CE-W**$^{s}_{kde}$ par rapport aux différents scénarios et modèles de référence. 136
Tableau 5.1 Statistiques des jeux de données de NY et de SG. 158
Tableau 5.2 Configuration des scénarios utilisés pour l'évaluation. 160
Tableau 5.3 Synthèse des modèles de référence utilisés pour évaluer la qualité de notre contribution pour la tâche de prédiction sémantique de l'emplacement. 161
Tableau 5.4 Exemple d'ordonnancement de paires de mots $(w_i^{(t)}, w_j^{(p)})$ issues des matrices d'interaction I'_l et I_l 169
Tableau 5.5 Comparaison des performances du modèle **SGM** selon différentes configurations. 171
Tableau 5.6 Comparaison des performances du modèle **SGM** par rapport aux modèles de référence. 177
Tableau 5.7 Analyse qualitative des performances du modèle **SGM**. . . 181
Tableau 5.8 Exemples d'échec et de succès d'appariements effectués par le modèle **SGM**. 182
Tableau 5.9 Comparaison des performances du modèle **SGM** pour différentes valeurs de rayon r. 185

Rapport-Gratuit.com

1

CONTEXTE ET CONTRIBUTION DE LA THÈSE

1 Contexte et problématique

1.1 *Contexte de la thèse*

Le recherche d'information (RI) est un domaine de l'informatique qui permet à un système de recherche d'information (SRI) de sélectionner, à partir d'une collection de documents, ceux susceptibles de répondre aux besoins de l'utilisateur exprimés sous la forme d'une requête (Salton, 1968). Le contenu des documents peut être du texte, des sons, des images, des vidéos ou des données. Un SRI possède trois fonctions fondamentales, qui définissent le modèle de recherche : représenter le contenu des documents, représenter le besoin de l'utilisateur, et comparer ces deux représentations pour calculer un score de pertinence entre le document et la requête. Traditionnellement, l'adéquation entre une requête et un document s'appuie sur la concordance entre les sujets de la requête et du document sur la base de leurs contenus en termes de mots, concepts ou sujets.

Si le traitement de l'information textuelle était la prérogative de la RI traditionnelle, le traitement de l'information géographique, conservée en tant que données structurées, était celle des systèmes d'information géographique (SIG), qui permettaient d'accéder aux informations géographiques via des combinaisons de cartes numériques et de bases de données. Cependant, les applications de SIG ont montré leurs limites, notamment à cause de l'augmentation exponentielle des ressources peu structurées combinant l'information géographique et textuelle (McCurley, 2001). Pourtant, l'importance de la localisation dans la RI semble aujourd'hui évidente. Comme tous ce que nous faisons se déroule dans un contexte géographique, il n'est pas surprenant que de plus en plus de requêtes formulées par les utilisateurs aient une orientation géographique (Sanderson et Kohler, 2004; Aloteibi et Sanderson, 2014; Reichenbacher *et al.*, 2016). C'est donc pour répondre à ce besoin croissant d'accès public à l'information géographique que la RI traditionnelle a naturellement évoluée vers la recherche d'information géographique

(RIG) et le développement de systèmes de recherche d'information géographique (SRIG). Cependant, les documents et les requêtes étant peu ou pas structurées, l'information géographique, formulée en langage naturel, est ambiguë et incertaine (Amitay *et al.*, 2004). En effet, ces dernières décennies, le développement des technologies de l'information ont facilité la création et l'échange de données, notamment via les réseaux sociaux numériques (RSNs) tels que Facebook, Twitter ou Instagram. Les RSNs sont devenus des espaces d'échanges populaires permettant d'établir des liens sociaux et de partager de l'information textuelle. De plus, avec la connectivité croissante des utilisateurs et la prévalence des appareils mobiles, un nombre croissant de ces données textuelles incluent des géotags. Appelés géotextes, ces derniers ouvrent ainsi de nouvelles opportunités pour faire le lien entre le monde social en ligne et le monde physique, et développer de nouvelles applications pour répondre aux besoins du monde réel, tant pour la RI ad-hoc (Chen *et al.*, 2006), que pour la recommandation (Shaw *et al.*, 2013) ou le résumé spatio-temporel (Li *et al.*, 2015) par exemple.

Il est donc nécessaire d'adapter les SRI traditionnels afin d'intégrer l'aspect géographique dans les tâches d'appariement, que ce soit pour l'indexation des documents (Vaid *et al.*, 2005), le calcul de la pertinence (Martins *et al.*, 2005) ou l'évaluation des résultats (Mandl, 2011). Par ailleurs, les SRIG doivent être capables de détecter et résoudre des références à des lieux dans les documents afin d'en déterminer la portée géographique. Cette étape cruciale passe par une étape de géoréférencement (ou prédiction de l'emplacement), et peut se distinguer en trois problématiques de recherche qui sont : la prédiction de l'emplacement du contenu généré par l'utilisateur, la prédiction de l'emplacement mentionné dans le texte et la prédiction sémantique de l'emplacement. La prédiction sémantique de l'emplacement consiste à apparier des publications, géotaggées ou non, à des objets spatiaux sémantiquement liés, généralement représentés par des points d'intérêts (POIs). Cette tâche a largement été étudiée pour des documents traditionnels, tels que Wikipedia (Roller *et al.*, 2012) ou des pages web (Amitay *et al.*, 2004). Cependant, la nature des publications issues des RSNs, appelées géotextes, rend la plupart des approches traditionnelles ou dédiées inefficaces. En effet, les géotextes sont généralement des textes de courte longueur (p. ex. 280 caractères pour un tweet) avec une utilisation fréquente de mots non conventionnels, comme des abréviations ou des erreurs syntaxiques. Ainsi l'ambiguïté des termes et l'inadéquation du vocabulaire entre le contenu des publications en ligne et celui des données plus structurées ne permet pas de résoudre efficacement des tâches de RIG. Les travaux présentés dans ce manuscrit s'intéressent donc à l'intégration de la dimension spatiale pour la représentation et l'appariement de géotextes dans le cadre de tâches de RIG.

1.2 *Problématique de la thèse*

Les travaux présentés dans ce manuscrit s'inscrivent dans le contexte général de la RIG avec un intérêt particulier pour la représentation et l'appariement de géotextes issus des RSNs. Plus précisément, nous abordons les deux principales problématiques suivantes : *(1)* l'exploitation de la sémantique distributionnelle au travers des plongements lexicaux pour améliorer les représentations des géotextes ; *(2)* l'appariement d'objets géotextuels pour déterminer la portée géographique des géotextes via une tâche de prédiction sémantique de l'emplacement.

Concernant la problématique de représentation, nous tentons de lever les limitations de la représentation en sacs de mots et de l'appariement exact qui s'appuie sur un simple comptage des mots en communs. Les modèles d'appariement de textes récents utilisent largement les représentations de mots qui s'appuient sur la sémantique distributionnelle. Appelées plongements lexicaux (Mikolov *et al.*, 2013a,b; Pennington *et al.*, 2014), ces représentations tirent parti des caractéristiques sémantiques des mots présents dans un texte, permettant ainsi d'extraire les liens sémantiques entre les séquences de textes à l'aide d'opérations vectorielles. En effet, en capturant la mesure dans laquelle les mots se produisent dans des contextes similaires, les plongements lexicaux sont capables d'encoder la similarité sémantique étant donné que les représentations des mots similaires seront proches dans l'espace vectoriel. Cependant, la sémantique distributionnelle présente quelques limites pour intégrer la dimension spatiale. Elle ne permet pas de lever le problème de polysémie, puisque tous les sens d'un même mot sont représentés dans un seul vecteur. Néanmoins, certains travaux ont montré l'existence de langages sensibles à la localisation qui sont suivis par une variation des mots et des sujets en fonction des contextes géospatiaux (Backstrom *et al.*, 2008; Han *et al.*, 2012; Laere *et al.*, 2014). Par exemple, dans le contexte géographique de la France, le terme *football* désigne un sport qui se joue avec le pied et un ballon rond, tandis que dans le contexte géographique des États-Unis, il désigne un sport qui se joue principalement à la main, avec un ballon ovale. De plus, le similarités explicites entre les mots, telles qu'elles peuvent être établies dans une ressource externe, peuvent ne pas l'être si leur apparition dans un même contexte est insuffisante dans le corpus d'apprentissage. Dès lors, sachant que les géotextes sont par définition dépendant d'une localisation, il convient d'intégrer les spécificités géographiques des mots dans les représentations distribuées des mots pour améliorer leur représentation.

Concernant la problématique de prédiction sémantique de l'emplacement, nous exploitons d'une part les résultats en lien avec la sensibilité des langages à la localisation (exploitée également pour la représentation) et d'autre part la force des approches neuronale pour pallier le problème du défaut sémantique, pour

proposer un modèles neuronal d'appariement de géotextes. Seuls quelques travaux ce sont intéressés à cette tâche (Dalvi *et al.*, 2009a; Zhao *et al.*, 2016). Ces derniers reposent principalement sur des modèles de langues spécifiques ou des modèles bayésiens pour pallier les spécificités des géotextes, évoqués précédemment (faible longueur, texte non conventionnel, etc.). Le principal inconvénient de ces approches réside dans la difficulté et le coût de l'estimation des probabilités conditionnelles de la pertinence thématique à travers les régions géographiques. Une alternative envisageable serait d'utiliser des architectures neuronales, qui ont largement fait leurs preuves en RI (Onal *et al.*, 2017; Guo *et al.*, 2019). Plusieurs modèles, s'appuyant sur différentes architectures ont ainsi été proposés pour traiter des tâches liées à l'appariement de textes. Ces derniers sont très efficaces, puisqu'ils s'attaquent aux problèmes de rareté des données et de discordance de vocabulaire (Onal *et al.*, 2017), qui sont les principaux obstacles rencontrés dans la tâche de prédiction sémantique de l'emplacement à partir de géotextes. Par ailleurs, la tâche étant très sensible à la localisation, il convient de trouver une architecture qui permet d'exploiter les contextes spatiaux par le biais d'interactions spatialement sensibles entre les mots, afin de comprendre la proximité de leurs sens, dans un contexte spatial spécifique. De plus, confortés par les travaux précédents (Dalvi *et al.*, 2009a,b; Zhao *et al.*, 2016), il semble claire que les signaux d'appariement de pertinence, au niveau des objets, sont tout aussi critiques que les signaux de correspondance mot à mot pour estimer la relation et la pertinence entre deux objets géotextuels.

Ainsi, dans le cadre de cette thèse nous proposons d'exploiter la sémantique relationnelle augmentée par l'information spatiale pour améliorer les représentations et l'appariement des objets géotextuels. Pour cela, nous nous attachons dans ce manuscrit à répondre aux questions suivantes :

— Comment exploiter la sémantique distributionnelle et l'information spatiale pour améliorer la représentation des mots et des objets géotextuels pour des tâches de RIG portés sur l'appariement de géotextes ?

  1. Comment déterminer les spécificités locales des mots ?
  2. Comment intégrer les contraintes spatiales pour mieux apprendre les représentations distribuées géosensibles ?

— Comment intégrer la sémantique distributionnelle dans un modèle d'apprentissage d'ordonnancement pour améliorer l'efficacité de la prédiction sémantique de l'emplacement ?

  1. Quelle représentation des géotextes, proposée à l'entrée du réseau de neurones, permet d'intégrer au mieux la connaissance sémantique et spatiale ?

2. Comment combiner les signaux d'appariement local mot à mot et les signaux d'appariement global textuel et spatial pour capturer conjointement la proximité sémantique et la proximité spatiale ?

2 Contributions

Les travaux réalisés dans cette thèse s'inscrivent dans la définition de modèles de représentation de géotextes pour l'appariement et la représentation d'objets géotextuels pour la prédiction sémantique de l'emplacement qui peut être associé au sujet du tweet. Plus particulièrement, nous nous intéressons à l'appariement de tweets avec des POIs. Nous proposons de tirer parti des contextes géographiques et de la sémantique distributionnelle pour résoudre la tâche de prédiction sémantique de l'emplacement. Nos contributions sont les suivantes :

1. *Augmenter les plongements lexicaux par l'information spatiale.* Notre première contribution repose sur approche de régularisation a posteriori de plongements lexicaux qui exploite la répartition spatiale des mots pour identifier des relations sémantiques locales entre mots ainsi que des sens locaux de mots. Plus spécifiquement, nous proposons deux solutions, l'une s'appuyant sur un partitionnement spatial, l'autre sur un partitionnement probabiliste, pour déceler les spécificités locales des différents sens d'un mot. Nous proposons ensuite de corriger les plongements lexicaux préalablement entraînés, à l'aide d'une fonction de régularisation qui intègre les répartitions spatiales. Cette dernière a pour objectif de rapprocher dans l'espace vectoriel les représentations des mots proches spatialement, et d'éloigner les représentations des mots distants spatialement. Nous menons une évaluation empirique et des analyses qualitatives et quantitatives pour valider nos hypothèses de recherche et montrer l'efficacité des plongements lexicaux spatiaux que nous proposons d'intégrer dans des tâches de similarité de géotextes et de prédiction sémantique de l'emplacement. Les résultats de l'évaluation montrent que l'utilisation des représentations régularisées permet d'améliorer significativement les performances par rapport aux modèles de référence, tant sur la tâche de similarité de POIs, avec une précision de $0,426$, que sur la tâche de prédiction sémantique de l'emplacement, avec une précision de $0,535$.
2. *Exploitation des réseaux de neurones profonds pour réaliser un appariement sémantique de tweets avec des POIs.* Nous proposons dans la deuxième contribution une architecture neuronale acyclique permettant d'apprendre une fonction d'appariement de tweets pour la prédiction sémantique de l'emplacement. À notre connaissance, il s'agit d'une des premières approches neuronales pour résoudre cette tâche. Contrairement aux architectures de RI existantes,

notre approche s'appuie sur un apprentissage conjoint, c.-à-d. dans le même espace de représentation, des interactions locales mot à mot et globales textuelles et spatiales entre des paires de tweet-POI. Plus spécifiquement, nous proposons de calculer les signaux d'appariement exact des interactions local mot à mot et de les lisser à l'aide d'un facteur d'amortissement spatial afin de discriminer les similarités sémantiques de paires de mots. Les interactions locales ainsi lissées sont traitées à l'aide d'histogrammes d'appariement et envoyés dans un perceptron multicouche en vue d'obtenir une représentation latente des interactions locales. Ces dernières permettent ainsi de révéler des motifs de similarités de paires de mots guidés par l'information spatiale. Les interactions globales considèrent quant à elles, la force de l'interaction entre le tweet et le POI à la fois du point de vue spatial, à travers une distance géographique entre les objets géotextuels, et du point de vue sémantique via une proximité sémantique de leurs représentations latentes. Les interactions locales et globales sont ensuite combinées dans un réseau de neurones. Ce dernier est entraîné pour maximiser la distance entre la similarité de paires tweet-POI pertinentes et la similarité de paires non-pertinentes. Une évaluation expérimentale et des analyses qualitatives approfondies sont menées pour valider notre approche. Nous obtenons une efficacité supérieure aux modèles de référence, avec une score de précision supérieur à $0,7$ sur deux jeux de données issus du monde réel. De plus, nous présentons une analyse approfondie de l'impact des principaux paramètres de notre modèle sur la performance globale.

3 Organisation du mémoire

Cette thèse est constituée d'un chapitre introductif ainsi que de trois principales parties, dont la première présente la synthèse des travaux de l'état-de-l'art, la suivante détaille nos principales contributions et la dernière conclut le manuscrit et discute des perspectives de recherche.

Le Chapitre 1 introduit la thèse. Il présente le contexte, les problématiques de recherche abordées et les contributions issues de nos travaux.

La première partie de cette thèse, intitulée *Synthèse des travaux de l'état-de-l'art*, présente le contexte de nos travaux. Elle englobe les deux chapitres suivants :

— Le Chapitre 2, *Prédiction de l'emplacement à partir de flux d'information dans les réseaux sociaux*, introduit dans la Section 1, les notions inhérentes à la RIG. Nous commençons par détailler les concepts fondamentaux (Section 1.1) avant de présenter les spécificités des SRIG en ce qui concerne l'indexation

des documents (Section 1.2), les requêtes géographiques (Section 1.3) et l'ordonnancement de pertinence (Section 1.4), ainsi que la problématique de résolution de la portée géographique des documents (Section 1.5). Nous présentons ensuite dans la Section 2 de ce chapitre, les travaux de l'état-de-l'art sur la résolution de la portée géographique des documents des RSNs. Plus précisément, nous détaillons les différentes approches, regroupées en trois niveaux, selon le type de géoréférencement effectué : la prédiction de l'emplacement du contenu généré par l'utilisateur (Section 2.1), la prédiction de l'emplacement mentionné (Section 2.2) et la prédiction sémantique de l'emplacement (Section 2.3). Dans la Section 3, nous discutons les principales limites de ces approches.

— Le Chapitre 3, *Réseaux de neurones pour la représentation distribuée et l'appariement de textes et géotextes*, introduit, dans la Section 1 les notions relatives aux réseaux de neurones, en détaillant les concepts (Section 1.1 et Section 1.2), quelques architectures populaires en RI (Section 1.3) ainsi que l'algorithme d'apprentissage (Section 1.4). Nous revenons aussi sur les problématiques inhérentes aux modèles neuronaux (Section 1.5). Nous poursuivons ce chapitre en présentant, dans la Section 2, les différents modèles de l'état-de-l'art pour la représentation distribuée de textes (Section 2.1) et de géotextes (Section 2.2). Nous continuons en détaillant les architectures neuronales utilisées pour l'appariement de textes dans la Section 3. Plus précisément, nous introduisons une formulation unifiée des modèles d'ordonnancement (Section 3.1) et détaillons les différentes approches, regroupées en deux catégories, celles orientées représentation (Section 3.2) et celles orientées interaction (Section 3.3). Nous terminons en discutant les principales limites des modèles de représentation et d'appariement dans la Section 4.

La deuxième partie de cette thèse, intitulée *Contributions*, présente les contributions de cette thèse. Elle englobe les deux chapitres suivants :

— Le Chapitre 4, *Régularisation spatiale de plongements lexicaux*, présente la première partie de nos contributions pour l'apprentissage de plongements lexicaux augmentés par des connaissances issues des répartitions spatiales des mots. Nous commençons par rappeler le contexte de nos travaux dans la Section 1. Nous détaillons ensuite dans la Section 2, les concepts abordés dans nos contributions (Section 2.1) et notre problématique de recherche (Section 2.2). Nous présentons dans la Section 3 notre méthode de régularisation a posteriori des plongements lexicaux, avec la détection des sens locaux des mots (Section 3.2) et l'algorithme de régularisation (Section 3.3). Pour mesurer la qualité de nos plongements lexicaux, nous réalisons plusieurs évaluations expérimentales, décrites dans la Section 4. Plus spécifiquement, dans la Section 4.1, nous commençons par valider notre hypothèse de recherche. Nous continuons dans la Section 4.2 en évaluant la qualité intrin-

sèque des plongements lexicaux spatiaux sur une tâche de similarité de POIs. Enfin, dans la Section 4.3, nous réalisons une évaluation extrinsèque des représentations à l'aide de la tâche de prédiction sémantique de l'emplacement. Finalement, nous concluons de chapitre dans la Section 5.

— Le Chapitre 5, *Modèle neuronal pour la prédiction sémantique de l'emplacement*, présente la deuxième partie de nos contributions relative à l'apprentissage d'un modèle d'interaction pour la prédiction sémantique de l'emplacement. Après avoir rappelé le contexte dans la Section 1, nous formalisons dans la Section 2 la problématique de recherche et l'hypothèse qui vont guider nos travaux. Nous décrivons ensuite dans la Section 3 l'architecture de notre réseau de neurones, en détaillant les calculs des interactions locales (Section 3.2) et des interactions globales (Section 3.3) utilisées pour calculer le score d'appariement (Section 3.4), ainsi que l'algorithme d'apprentissage (Section 3.5). Nous détaillons le protocole d'évaluation dans la Section 4 et présentons les résultats obtenus dans la Section 5. Enfin, nous concluons ce chapitre dans la Section 6.

En conclusion, nous faisons le bilan des travaux réalisés et synthétisons les éléments originaux de nos contributions. Nous présentons ensuite les différentes perspectives d'évolution de nos travaux.

Partie I

SYNTHÈSE DES TRAVAUX DE L'ÉTAT-DE-L'ART

2

PRÉDICTION DE L'EMPLACEMENT À PARTIR DE FLUX D'INFORMATIONS DANS LES RÉSEAUX SOCIAUX

Introduction

L'information géographique est un type d'information qui permet d'associer des entités et des contenus à des lieux physiques (p. ex. des pays, des régions, des villes ou des points d'intérêts). Elle est enregistrée sur une grande variété de supports et de types de documents. Il existe d'innombrables livres, rapports, images et cartes sur papier, mais aussi des bases de données informatiques et des cartes numériques, ainsi qu'un grand nombre de pages web contenant des textes, des images géoréférencées et des versions numériques d'articles ou de livres. Historiquement, cette information, conservée en tant que données structurées, était la prérogative des systèmes d'information géographique (SIG), qui permettaient d'accéder aux informations géographiques via des combinaisons de cartes numériques et de bases de données.

L'importance de la localisation dans la recherche d'information (RI) semble aujourd'hui évidente. Une grande partie des informations disponibles sur le web sont spécifiques à une zone géographique (Vaid *et al.*, 2005; Delboni *et al.*, 2007; Vasardani *et al.*, 2013). De plus, comme tout ce que nous faisons se déroule dans un contexte géographique, il n'est pas surprenant que de nombreuses requêtes sur le web aient une orientation géographique, que ce soit pour trouver le restaurant le plus proche, obtenir des informations sur une ville ou trouver des photos d'un monument par exemple. Nous estimons que 13% à 15% des requêtes soumises à des SRI contiennent des noms de lieux ou des termes géographiques (Sanderson et Kohler, 2004; Gan *et al.*, 2008; Aloteibi et Sanderson, 2014), et plus d'un tiers des recherches mobiles sont liées à la localisation [1]. De fait, les services s'appuyant sur la localisation, dans lesquels l'emplacement actuel ou prévu de l'utilisateur est utilisé comme information contextuelle en temps réel, se propagent à un rythme

1. https://www.thinkwithgoogle.com/consumer-insights/

effréné, avec pour cible principale, les utilisateurs des smartphones (Reichenbacher *et al.*, 2016). C'est donc ce besoin croissant d'accès public à l'information géographique qui a été une motivation majeure pour explorer le domaine de la recherche d'information géographique (RIG), une extension du domaine de la RI (Baeza-Yates et Ribeiro-Neto, 1999), qui cherche à développer un système de recherche spatialisé et à soutenir les besoins d'information géographique des utilisateurs en utilisant notamment les métadonnées géographiques des documents (Jones et Purves, 2009).

Cependant, l'information géographique présente dans les documents, formulée en langage naturel, est généralement ambiguë et incertaine (Amitay *et al.*, 2004; Clough *et al.*, 2004). Par exemple, les lieux mentionnés peuvent correspondre à différents lieux (p. ex. Paris, en plus d'être la capitale de la France faire référence à plus de soixante villes dans le monde). Ils peuvent aussi être nommés en utilisant des mots courants (*Park*, *Hope* et *Independence* sont des villes américaines) ou des noms propres (*Washington* et *Houston* sont des noms de villes ou de personalités). Il peut aussi y avoir une ambiguïté de référence, qui se produit lorsqu'un lieu est associé à plusieurs noms, comme « *la ville rose* » ou « *la cité des violettes* » pour désigner la ville de Toulouse. L'ambiguïté rend donc la résolution des références aux lieux intrinsèquement contextuelle. Une étape importante pour aborder la désambiguïsation est la détermination de la portée géographique du document, c.-à-d. l'ensemble des lieux référencés par le document et pertinents pour son contenu (Andogah *et al.*, 2012; Alexopoulos *et al.*, 2012; Silva *et al.*, 2006). De ce fait, l'accès efficace aux documents, dans lesquels nous pouvons déduire une pertinence géographique, nécessite des méthodes capables de reconnaître la présence de références géographiques et de les résoudre sans équivoque. Cela inclut l'interprétation automatisée des noms de lieux et des relations spatiales dans les requêtes et les documents. La RIG s'attache ainsi à résoudre ces défis en améliorant la qualité de la recherche d'information géographiquement spécifique (Purves *et al.*, 2008).

Un système de recherche d'information (SRIG) doit donc être capable de détecter et résoudre des références à des lieux, typiquement, mais pas exclusivement, sous la forme de noms de lieux ou de toponymes plus formels, à partir de documents non structurés (Purves *et al.*, 2018). En s'appuyant sur cette résolution de toponymes ou pas, un SRIG doit également être en mesure de déterminer la portée géographique des documents et des requêtes. En d'autres termes, il doit répondre à la question « sur quelle localisation porte le sujet du document ou de la requête ? ». Pour cela, les SRIG, comme les SRI traditionnels, se composent généralement de trois composants majeurs à savoir, l'analyseur de requêtes qui permet de considérer les représentations cognitives de l'espace et la manière dont elles peuvent influencer le langage utilisé pour formuler la requête, un ou plusieurs index qui permettent de récupérer les documents pertinents qui seront ensuite

classés par le système d'information, et une interface pour présenter les résultats à l'utilisateur. Cependant, les SRIG possèdent d'autres composants clefs, comprenant toujours un répertoire toponymique (aussi appelé *gazetteer*) ou index géographique, qui enregistre les noms de lieux et les informations associées telles que les coordonnées ou d'autres informations géographiques structurées permettant de faire le pont entre la RI traditionnelle (Manning *et al.*, 2009) et la science de l'information géographique (Goodchild, 2010).

Depuis quelques années, de nombreux SRIG se sont développés. L'un des exemples de système les plus anciens est le *Geo-referenced Information Processing System* (GIPSY) (Larson, 1996) qui permettait la recherche au sein de bibliothèques numériques. D'autres systèmes, tels que les projets *Web-a-Where* (Amitay *et al.*, 2004), *Spatially Aware Search Engine for Information Retrieval on the Internet* (Purves *et al.*, 2007) ou *Spatio-Textual Extraction on the Web Aiding Retrieval of Document* (STEWARD) (Lieberman *et al.*, 2007) ont suivi pour la RIG à partir de pages web. Les articles de presse se sont aussi révélés être de très riches sources d'information. L'exemple le plus marquant est celui de *NewsStand* (Teitler *et al.*, 2008), qui s'est concentré sur la collecte et le résumé de nouvelles en temps réel avec l'utilisation d'index géographiques adaptés. Enfin, d'autres chercheurs ont utilisé les méthodes de RIG pour résumer de grands corpus de textes, à des échelles plus grossières, et pour des régions géographiques beaucoup plus vastes (allant des États-Unis au monde entier). Les systèmes comme *TextGrounder* (Brown *et al.*, 2012) et *Franken-Place* (Adams *et al.*, 2015) s'appuient sur des approches d'apprentissage automatique pour géoréférencer le contenu sans recourir aux index géographiques. Plus récemment, avec l'avènement des réseaux sociaux numériques (RSNs), de nouveaux axes de recherche ont émergé, valorisant ainsi les données sociales pour le résumé spatio-temporel (Li *et al.*, 2015; Liu *et al.*, 2016; Zhang *et al.*, 2016), la recommandation (Shaw *et al.*, 2013; Min *et al.*, 2015; Xie *et al.*, 2016) ou la détection de l'emplacement (Li *et al.*, 2011a; O'Hare et Murdock, 2013; Fang et Chang, 2014).

Dans le contexte de cette thèse, nous nous intéressons principalement à la prédiction sémantique de l'emplacement des documents non structurés, plus spécifiquement des géotextes issus des RSNs. Nous l'aborderons à la fois sous l'angle de la représentation d'objets géotextuels et de leur appariement. Dans les sections suivantes, nous commençons par introduire, dans la Section 1, les notions inhérentes à la RIG en détaillant les concepts abordés dans ce manuscrit. Nous détaillons ensuite dans la Section 2 les principales approches mises en œuvre pour résoudre la porté géographique des documents issus des RSNs. Enfin, dans la Section 3, nous discutons des limites de ces approches dans le cadre de tâches de RIG.

1 Introduction à la recherche d'information géographique

La RIG est interdisciplinaire et peut être considérée comme une extension du domaine de la RI. Le but de cette section est donc de présenter un ensemble de concepts généralement formulés en RIG, en ce qui concerne la façon dont l'espace et les données sont conceptualisés, représentés et analysés.

1.1 *Concepts fondamentaux*

Nous introduisons dans cette section les concepts hérités de la géographie, tels que le système géodésique, les coordonnées géographiques, la distance géographique et la projection cartographique.

Définition 2.1 (**Système géodésique**)**.** Les systèmes de coordonnées géographiques fournissent une méthode quantitative pour enregistrer l'emplacement par rapport à un point de référence connu, c'est-à-dire l'origine du système de coordonnées. Les systèmes de référence couramment utilisés sont *les systèmes géodésiques*, qui permettent d'exprimer les positions au voisinage de la Terre. Ces systèmes reposent sur une ellipsoïde dont les paramètres de définition sont un centre O, un demi-grand axe a et un aplatissement f. Actuellement, il existe plus de 4 300 systèmes de référence, chacun ayant des paramètres différents. Néanmoins, le système géodésique le plus répandu est le *WGS 84* (*World Geodetic System 1984)* [2], notamment utilisé par le système de positionnement par satellite GPS [3].

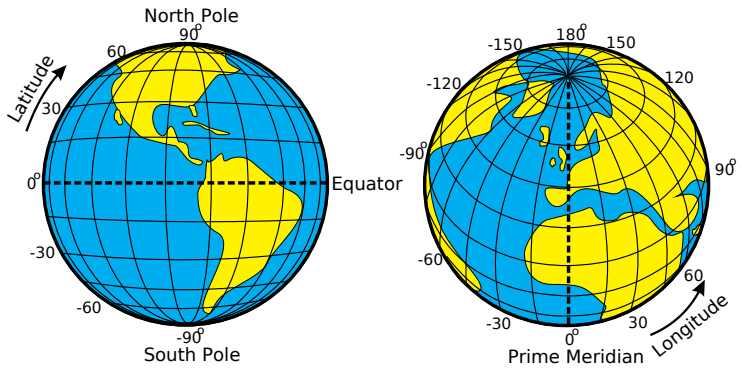


Figure 2.1 – Latitude et longitude sur la Terre. (©*Wikimedia*)

2. La définition des paramètres est la suivante : $a = 6\,378$ km ; $f = 1/298$.
3. *Global Positioning System*.

Définition 2.2 (Coordonnées géographiques). Le système de coordonnées géographiques est un système de coordonnées qui permet à chaque emplacement de la Terre d'être matérialisé par un ensemble de chiffres, lettres ou symboles. Les coordonnées géographiques découlent d'un système géodésique et sont généralement représentées sous la forme d'une *latitude* (notée *lat* ou φ) et d'une *longitude* (notée *lon* ou λ). Elles enregistrent les angles par rapport à des plans de référence. Comme montré sur la Figure 2.1, la latitude est une valeur angulaire exprimant le positionnement Nord ou Sud d'un point de la Terre par rapport à l'équateur. La longitude est quant à elle une valeur angulaire exprimant le positionnement Est ou Ouest d'un point de la Terre par rapport au méridien de Greenwich. Il convient de noter que, comme la Terre n'est pas parfaitement ronde, la longitude n'est pas une mesure proportionnelle et l'écart (ou distance) entre deux longitudes varie selon la latitude. À titre d'exemple, au niveau de l'équateur (latitude de 0°), un écart de 1° de longitude représente 111,3 km, tandis qu'à Saint-Pétersbourg (latitude de 59°), un écart de 1° de longitude ne vaut plus que 55,80 km.

Définition 2.3 (Distance géographique). Le calcul de la distance entre des coordonnées géographiques est établi sur un certain niveau d'abstraction, qui ne fournit donc pas une distance exacte. Les abstractions courantes pour la distance entre deux points géographiques sont : une surface plane, une surface sphérique et une surface ellipsoïdale.

Une approximation plane de la surface de la Terre peut être utilisée sur de petites distances. Dans ce cas, la précision du calcul devient de plus en plus imprécise au fur et à mesure que la distance entre les points devient grande, et que les points se rapprochent des pôles géographiques. Avec une approximation plane, la distance géographique entre deux points x et y de coordonnées (φ_x, λ_x) et (φ_y, λ_y) est généralement calculée en utilisant la formule de la distance euclidienne :

$$dist_E(x,y) = ||y - x||_2 = \sqrt{(\varphi_y - \varphi_x)^2 + (\lambda_y - \lambda_x)^2} \tag{2.1}$$

Dans le cas d'une approximation sphérique de la surface de la Terre, le taux d'erreur est de 0,5% (Navy, 1987). Cette approximation permet d'appliquer la trigonométrie sphérique pour effectuer le calcul de distance. Nous utilisons la formule de Haversine (De Smith et Goodchild, 2007) pour déterminer la distance du grand cercle[4] entre deux points d'une sphère, à partir de leurs longitudes et latitudes. La distance Haversine entre deux points x et y de coordonnées (φ_x, λ_x) et (φ_y, λ_y) est donnée par :

$$dist_H(x,y) = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_y - \varphi_x}{2}\right) + \cos(\varphi_x)\cos(\varphi_y)\sin^2\left(\frac{\lambda_y - \lambda_x}{2}\right)}\right)$$

4. La distance du grand cercle, plus généralement appelée distance orthodromique ou simplement orthodromie, est la plus petite distance entre deux points sur une sphère.

(2.2)

avec r le rayon de la Terre approximé par une sphère (soit 6 378 km). La Figure 2.2 schématise la distance du grand cercle entre Toulouse et Toronto.

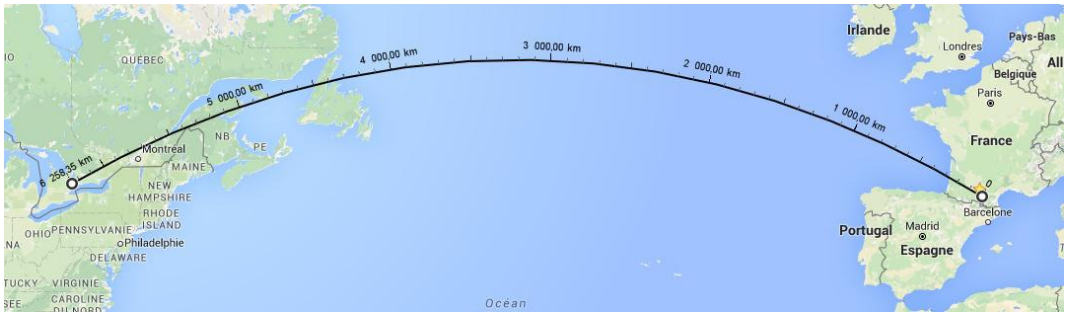


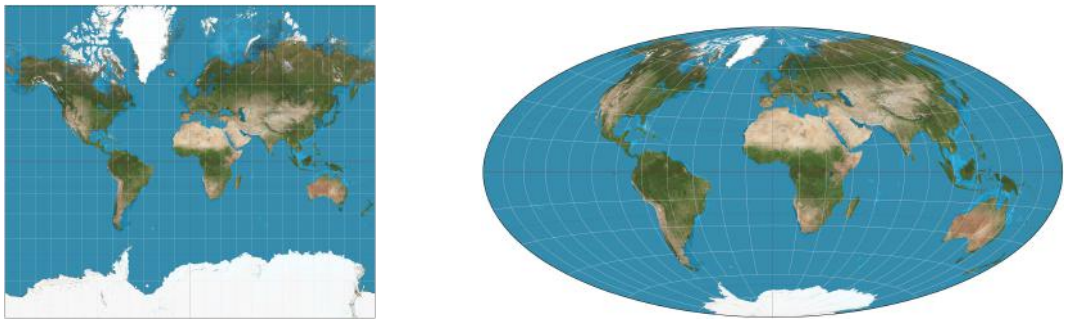
Figure 2.2 – Distance du grand cercle entre Toulouse et Toronto. (©*Fabrice Arnaud*)

Dans le cas d'une approximation ellipsoïdale de la surface de la Terre (une sphère oblate), la précision est d'environ 0,5 mm (Vincenty, 1975). Pour calculer cette distance, Vincenty (1975) a proposé deux méthodes itératives. La première méthode, directe, permet de calculer l'emplacement d'un point qui est à une distance et un azimut (c.-à-d. direction) donnés d'un autre point. La deuxième méthode, indirecte, permet de calculer la distance géographique et l'azimut entre deux points donnés. Cette mesure est largement utilisée en géodésie lorsque des précisions élevées sont requises. Bien que la méthode inverse de Vincenty soit plus précise que les distances détaillées ci-dessus, elle repose sur une méthode itérative qui peut se révéler coûteuse en temps d'exécution.

Dans la suite de ce manuscrit, sauf indication contraire, nous utiliserons par défaut la distance Haversine pour calculer la distance géographique entre deux coordonnées. Par commodité, $dist_H(x, y)$ sera simplifiée par $dist(x, y)$.

Définition 2.4 (Projection cartographique). La projection cartographique est un ensemble de techniques géodésiques permettant de représenter une surface non plane dans son ensemble sur la surface plane d'une carte. Il existe différents types de projection, chacune ayant des propriétés diverses, déformant plus ou moins les formes et les aires des continents, comme le présente la Figure 2.3. La plus commune étant celle de Mercator (Figure 2.3a). La projection ne doit pas être confondue avec le système de coordonnées géographiques qui permet de localiser un point à la surface de la Terre.

Définition 2.5 (Projection UTM). La projection Transverse Universelle de Mercator ou *Universal Transverse Mercator* (UTM) en anglais, dont un extrait est présenté en Figure 2.4, est un type de projection cartographique conforme de la surface de la Terre. Une projection conforme permet de conserver les angles et donc les



(a) Projection de Mercator (b) Projection de Hammer

Figure 2.3 – Exemples de projections cartographiques. (©*Wikimedia*)

formes des continents. Pour couvrir la surface de la Terre, celle-ci est découpée en 60 fuseaux de 6 degrés en séparant l'hémisphère Nord et l'hémisphère Sud, soit au total 120 zones. Le système étant rectangulaire et mesuré en kilomètres, nous pouvons directement recouper chaque zone en grilles plus fines de 200 mètres par 200 mètres par exemple.

Figure 2.4 – Coordonnées des pays Européens dans le système UTM. (©*Wikimedia*)

Définition 2.6 (**Objet géotextuel**)**.** Un objet géotextuel, ou plus simplement un géotexte, est un objet textuel géotaggé, c.-à-d. associé à des coordonnées géographiques. Les géotextes sont par exemple des tweets géotaggés sur Twitter, des points d'intérêts (restaurants, bars, musée,etc.) sur Foursquare ou Google Place, des photos géotaggées sur Instagram ou Flickr, etc..

1.2 *Index et index géographique*

Comme pour les SRI traditionnels, l'index est la composante du SRIG qui contient des références à tous les documents connus du système, soit parce qu'ils ont été indexés en parcourant le web, soit parce qu'ils ont été ajoutés manuellement dans l'index à partir de diverses sources de données. Dans cette section, nous détaillons les approches communes en matière d'indexation (Section 1.2.1), avec l'utilisation de listes inversées (Section 1.2.2), l'indexation spatiale (Section 1.2.3) et des combinaisons d'index textuels et spatiaux (Section 1.2.4).

1.2.1 *La nécessité d'indexer les documents*

L'objectif d'un index est de fournir un accès rapide aux documents qui répondent aux besoins spatiaux et thématiques des utilisateurs formulés sous la forme d'une requête. Les structures permettant d'accéder aux documents contenant des termes spécifiques sont appelées listes inversées ou index inversés (Manning *et al.*, 2009). Les SRIG utilisent généralement des index inversés combinés avec des méthodes d'indexation spatiale pour déterminer efficacement les documents se rapportant à des régions spatiales spécifiques.

La vitesse d'accès étant un critère critique, il est important que l'index puisse accéder à toutes les données requises pour ordonner les documents pertinents. Ainsi, il pourrait y avoir des millions de documents qui répondent aux besoins spatiaux et textuels exprimés dans la requête, mais l'utilisateur souhaite généralement ne se voir présenter que les quelques documents les plus susceptibles de l'intéresser, tels que déterminés par les critères d'ordonnancement de pertinence (Section 1.4). Les éléments nécessaires au classement peuvent être stockés dans l'index et pourraient inclure, sans s'y limiter, le nombre de documents qui contiennent un terme particulier, la fréquence d'apparition d'un terme dans un document spécifique et l'endroit où ce terme apparaît dans le document.

En pratique, les index pour la recherche sur le web contiennent des informations sur des milliards de documents et diverses stratégies sont donc employées pour accélérer le processus de recherche. Par exemple, un ou plusieurs index sont utilisés pour effectuer une première recherche sur l'ensemble des documents qui contiennent au moins un des termes de la requête. Ce processus est extrêmement rapide car il s'agit d'une simple correspondance booléenne, et il réduit considérablement le nombre de documents à ordonner. L'ensemble de documents qui résulte de cette première sélection peut ensuite être affiné par d'autres index qui donnent accès aux éléments utilisés pour effectuer l'ordonnancement, en tenant compte par exemple de la distance par rapport à l'endroit où se trouve l'utilisa-

teur et du profil personnel de l'utilisateur. Il peut donc y avoir plusieurs couches d'index et de caches, permettant d'améliorer la rapidité de la recherche et la qualité des résultats. Les informations spatiales, plus complexes à traiter sur le plan informatique en termes de calcul, sont généralement intégrées lors de l'ordonnancement d'un sous-ensemble de documents très restreint. La requête étant spécifique à l'emplacement de l'utilisateur, les résultats ne peuvent généralement pas être mis en cache et réutilisés pour répondre aux besoin d'autres utilisateurs. Habituellement, l'index comprend des métadonnées structurées identifiant la portée géographique d'un document, et d'autres informations extraites au moment de l'indexation ou récupérées à partir d'autres collections de données.

1.2.2 *Indexation avec des listes inversées*

Terme	Liste des identifiants de documents
Terme 1	D1, D2, D4, D7
Terme 2	D1,D6, D8, D9
Terme 3	D2, D3
Terme n	...

Figure 2.5 – Exemple d'une liste inversée dans laquelle chaque enregistrement contient un terme et une liste avec les références des documents dans lequel le terme apparaît.

Lors de l'indexation standard des documents, avec une liste inversée, aussi appelée index inversé (Witten *et al.*, 1994), l'index se compose de paires « *clé :valeur* », où les clés sont l'ensemble de tous les termes qui figurent dans les documents, et pour chaque terme, correspond la liste des documents (la valeur) qui contiennent le terme, comme le présente la Figure 2.5. Pour permettre à l'index de prendre en charge l'accès aux phrases citées, les listes doivent inclurent des informations sur l'endroit du document où le terme apparaît réellement. Afin de faciliter les procédures d'ordonnancement, l'index contient également d'autres informations (nombre de documents contenant le terme, fréquence d'apparition du terme, etc.). Dans le cas d'une requête sous la forme d'un sac de mots, la liste inversée est utilisée pour trouver, pour chaque terme de la requête, les documents qui contiennent ce terme. Ce processus est assez simple dans la mesure où chaque terme de recherche est associé à la clé correspondante dans l'index et la liste d'affichage de cette clé est renvoyée. Pour trouver les documents qui contiennent tous les termes de recherche, il suffit de calculer l'intersection des listes correspondant à chaque terme de recherche.

Les index inversés ont un rôle primordial dans la RIG, dans la mesure où ils permettent d'accéder à des documents qui contiennent les termes de la requête

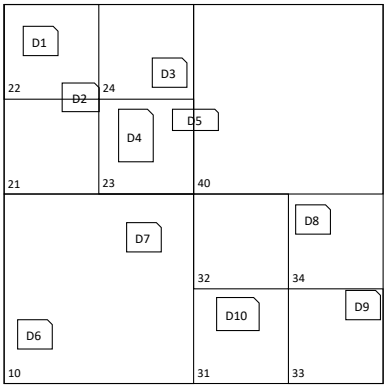
d'un utilisateur. Ils peuvent également être utilisés pour rechercher les documents qui contiennent les noms des lieux cités dans les requêtes géotextuelles. En effet, ces derniers sont traités exactement de la même manière que les autres termes dans le processus d'indexation. Pour des requêtes géographiques assez simples, cette approche fonctionne assez bien comme l'ont montré les performances des premiers SRIG (Ding *et al.*, 2000). Cependant, il existe plusieurs situations dans lesquelles ce type d'indexation ne suffit pas. Par exemple, lorsque le document pertinent contient un toponyme différent de celui utilisé dans la requête. Il se peut également qu'un document fasse référence à des emplacements situés dans le lieu désigné dans la requête, sans le mentionner explicitement. Une autre situation qui ne peut être résolue par cette approche est lorsqu'une requête inclut un qualificatif spatial (p. ex. dans un rayon de x km, à proximité de), auquel cas l'utilisateur s'intéresse à une région liée au lieu mentionné mais pas directement à celui-ci. Les limites de l'interrogation purement toponymique ont conduit au développement d'approches alternatives d'indexation combinant l'indexation par liste inversée avec des techniques d'indexation spatiale.

1.2.3 *Indexation spatiale*

L'objectif d'un index spatial est d'aider le SRIG à retrouver les objets géoréférencés stockés qui se situent dans l'empreinte géographique de la requête. Dans ce système, les objets sont des documents qui sont référencés à des emplacements spécifiques, via des objets géométriques (polygones, cercles, points) représentant l'empreinte géographique des documents [5]. Dans la littérature, il existe deux approches principales pour construire un index géographique, l'approche orientée vers l'espace et l'approche orientée vers l'objet. Dans les deux cas, l'index peut être considéré comme un ensemble de paires « *clé :valeur* », où la clé représente une cellule spatiale qui contient (ou recoupe) la géométrie des documents auxquels les valeurs se réfèrent. L'accès aux paires « *clé :valeur* » s'effectue soit par une base de données standard indexant la structure des données, comme l'arbre B+ (Bayer, 1997), soit par une structure d'indexation spécifiquement adaptée aux données multidimensionnelles. < Dans les méthodes d'indexation orientées vers l'espace, les cellules clés sont généralement de forme rectangulaire et servent à quadriller toute les régions couvertes par le corpus de documents. Dans sa forme la plus simple, les cellules de l'index spatial sont disposées sur une grille régulière de taille fixe. Chaque cellule est identifiée par un code de localisation représentant un point ou une région de l'espace. Ces codes peuvent servir de clés dans un arbre B+ par exemple.
Bien que la méthode d'indexation par une grille régulière soit souvent utilisée,

5. L'empreinte géographique d'un document peut être vue comme la ou les régions sur lesquelles se focalise le document.

elle présente l'inconvénient de ne pas limiter le nombre de documents associés à chaque cellule, ce qui peut entraîner des surcoûts de performance. Pour contrer cette difficulté, il est possible de découper la grille avec des cellules de tailles variables, comme le montre la Figure 2.6. Les cellules sont subdivisées de manière récursive jusqu'à ce qu'elle ne contiennent ou ne croisent pas plus d'un nombre seuil spécifié de documents. Les cellules sont généralement identifiées par des codes de localisation, déterminés à partir des coordonnées de la cellule. Dans la Figure 2.6, les objets étiquetés D1, D2, ... sont des documents dont les empreintes géographiques coupent leur cellule respective.



Id cellule	Ids document
10	D6, D7
21	D2
22	D1, D2
23	D4, D5
24	D3
31	D10
32	
33	D9
34	D8
40	D5

Figure 2.6 – Exemple d'un index spatial quadratique. Chaque cellule est associée à une liste des documents dont les empreintes géographiques recoupent une ou plusieurs cellules.

Concernant les méthodes d'indexation orientées objets, les cellules sont aussi rectangulaires, mais leurs dimensions sont adaptées à la géométrie des différents documents. De fait, les cellules peuvent se chevaucher les unes avec les autres. La forme la plus courante d'un index orienté objet est le R-arbre (Guttman, 1984). Dans cette configuration, les rectangles sont organisés en hiérarchie, de sorte que les rectangles du niveau le plus bas sont regroupés en rectangles plus grands, qui peuvent eux-mêmes être contenus dans des rectangles de niveau supérieur. Un exemple de R-arbre est présenté dans la Figure 2.7. Les clés d'un R-arbre sont les dimensions des rectangles et la structure d'indexation est proche de celle d'un arbre B+, les feuilles contiennent les références des documents dont les empreintes géographiques sont contenues dans leur rectangle correspondant.

Lorsque le SRIG traite une requête avec un index spatial, la région représentée par l'empreinte géographique de la requête doit être mise en correspondance avec les cellules de l'index. Dans une première étape de filtrage, les cellules qui couvrent l'empreinte de la requête sont identifiées, de sorte que les documents liés à ces cellules puissent être récupérés. Comme les cellules consultées doivent couvrir complètement l'empreinte géographique de la requête, et donc éventuel-

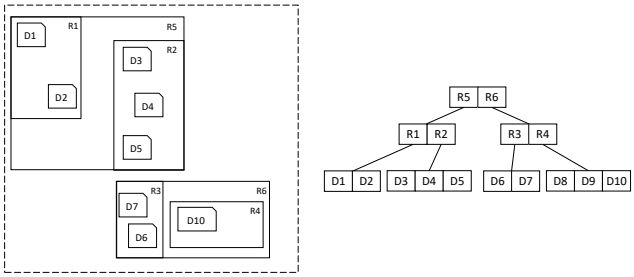


Figure 2.7 – Exemple d'un index spatial en R-arbre. Chaque feuille contient des entrées pour les documents dont les empreintes géographiques sont contenues dans le rectangle correspondant.

lement s'étendre au-delà de celle-ci, elles peuvent référencer certains documents situés en dehors de la limite de l'empreinte de la requête. Dans une deuxième étape, le SRIG affine la liste des documents récupérés en déterminant lesquels se trouvent réellement à l'intérieur de l'empreinte géographique de la requête ou le chevauchent.

1.2.4 *Indexation spatio-textuelle*

L'indexation spatiale et textuelle peuvent être combinées de différentes manières, dans des architectures distinctes ou jointes.

Une approche simple pour combiner les méthodes d'indexation spatiale et textuelle consiste à utiliser des index spatiaux et textuels séparément, et à fusionner les résultats obtenus par les requêtes (Vaid *et al.*, 2005; Chen *et al.*, 2006; Brisaboa *et al.*, 2010). Dans cette configuration, l'index spatial est utilisé pour récupérer les documents qui se rapportent aux contraintes spatiales formulées dans la requête, tandis que l'index inversé récupère les documents qui contiennent les termes de la requête. Les résultats peuvent ensuite être croisés pour ne conserver que les documents qui répondent aux exigences spatiales et textuelles. L'ordonnancement de pertinence peut être effectué en combinant les scores de pertinence textuelle et spatiale. Dans les travaux de Vaid *et al.* (2005), l'index spatial enregistre, pour chaque cellule spatiale, les identifiants des documents dont les empreintes géographiques recoupent la cellule. Chen *et al.* (2006) ont quant à eux introduit une cinquantaine d'approches utilisant des grilles régulières avec un index inversé séparé. Dans leurs approches, le principal index spatial est une grille régulière, de taille 1024×2014 ou 256×256. Chaque empreinte géographique de document est représentée par une ou plusieurs « zones », où chaque zone correspond à un rectangle de délimitation minimum (RDM) d'une région, telle qu'une ville, à laquelle le document se réfère. Ces RDM sont eux-mêmes décrits par une série de codes

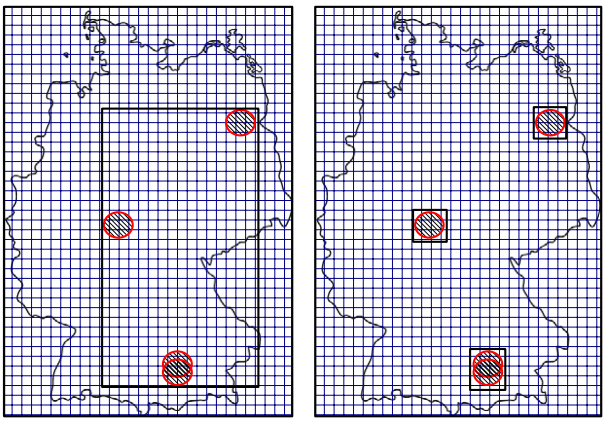


Figure 2.8 – Exemple d'un index géographique proposé par Chen *et al.* (2006) présentant une empreinte géographique et son RDM (à gauche) et les zones qui en résultent avec leurs RDMs (à droite).

de localisations établis sur des courbes de remplissage d'espace. Un exemple d'index spatial est présenté dans la Figure 2.8. Dans un autre exemple d'indexation séparée, Brisaboa *et al.* (2010) ont créé une forme d'index spatial hiérarchique dans lequel les nœuds font références à des lieux plutôt qu'à des cellules spatiales arbitraires. Chaque nœud est associé à une liste de documents qui se rapportent au lieu et au RDM du lieu. Le contenu textuel des documents est quant à lui accessible par un fichier inversé séparé.

Pour une meilleure intégration de l'indexation spatiale et textuelle, une autre approche consiste à utiliser l'index spatial comme filtre principal de la requête et à inclure un index secondaire qui filtre les autres caractéristiques textuelles de la requête (Vaid *et al.*, 2005; Zhou *et al.*, 2005; Cong *et al.*, 2009; Li *et al.*, 2011b). Cela conduit dans un premier temps à un index primaire spatial dans lequel chaque nœud d'un index spatial (correspondant à une cellule ou à une région) est associé à un index inversé des documents qui se rapportent à ce nœud. Ainsi, étant donné une requête composée de termes T et d'une empreinte géographique S, l'index inversé associé à chaque cellule de l'index spatial qui croise S est recherché pour trouver les documents qui comprennent les termes de la requête $t \in T$. Les méthodes d'indexation spatiales primaires ont principalement été implémentées avec des grilles régulières (Vaid *et al.*, 2005) et des R-arbres (Zhou *et al.*, 2005). La Figure 2.9 illustre l'idée d'un index spatial primaire avec des grilles (à gauche) et un R-arbre (à droite). Cong *et al.* (2009) et Li *et al.* (2011b) se sont appuyés sur des structures en arbre plus complexes, appelées *IR-tree*, permettant de récupérer les k documents les plus pertinents en utilisant une combinaison pondérée de termes spatiaux et textuels.

Id cellule	Index Inversé
10	Index Inversé 1
21	Index Inversé 2
22	Index Inversé 3
23	Index Inversé 4
24	Index Inversé 5
31	Index Inversé 6
33	Index Inversé 7
34	Index Inversé 8
40	Index Inversé 9

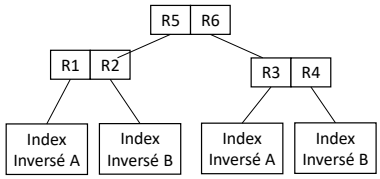


Figure 2.9 – Exemple d'un index primaire spatial avec une décomposition de l'espace en grilles (à gauche) et un R-arbre (à droite).

Dans la même idée que les index primaires spatiaux, une approche alternative repose sur des index primaires textuels (Vaid *et al.*, 2005; Chen *et al.*, 2006; Christoforaki *et al.*, 2011). Les documents de l'index inversé de chaque terme sont organisés par un index spatial dédié pour chaque terme, comme l'illustre la Figure 2.10. Dans ce cas, pour chaque terme de l'index inversé, son index spatial respectif est recherché pour trouver les documents qui recoupent l'empreinte géographique de la requête. Vaid *et al.* (2005) ont constaté que ce schéma avait de meilleures performances que l'indexation primaire spatiale. Chen *et al.* (2006) ont proposé une variante de cette méthode, appelée *Space Filling Inverted Index*, dans laquelle l'indexation spatiale est réalisée en stockant, pour chaque terme, des références aux zones. Christoforaki *et al.* (2011) ont quant à eux proposé une variante de cette dernière approche, dans laquelle les index sont organisés selon l'ordre spatial de la courbe de remplissage de l'espace, afin d'améliorer les temps de calcul. Une forme hybride de l'index primaire textuel, appelée S2In a été proposée par Rocha-Junior *et al.* (2011), dans laquelle une distinction est faite entre les termes les plus fréquents et les moins fréquents, afin de permettre des requêtes retournant les top-k documents. Pour chaque terme les plus fréquents, un R-arbre agrégé est construit avec les données stockées dans les nœud non foliaires. Il est conçu pour prendre en charge l'élagage de l'espace de recherche en fonction de la pertinence décroissante des documents. Les documents contenant des termes moins fréquents sont

Terme	Liste des identifiants de documents
Terme 1	IndexSpatialT1[D1, D3, D6, D8]
Terme 2	IndexSpatialT2[D2, D3, D7]
Terme 3	IndexSpatialT3[D1, D2, D6, D9]
Terme n	IndexSpatialTn[...]

Figure 2.10 – Exemple d'un index primaire textuel.

stockés dans des blocs analogues à des index inversés en ce sens qu'ils stockent les identités des documents qui contiennent un terme spécifique.

1.3 *Requête géographique*

Les SRIG offrent de nombreuses fonctionnalités permettant d'exploiter des informations géographiques. Nous pouvons les classer en trois catégories : le prétraitement des données, la recherche ou l'extraction d'information dans une région particulière, et l'analyse de l'information géographique (Longley *et al.*, 2015). Les fonctions de recherche et d'extraction de l'information à partir d'une base de données spatiale permet à l'utilisateur de formuler des requêtes en spécifiant des emplacements précis (avec des coordonnées) ou avec des termes de relations spatiales (près de, à 5 km de). Il existe de nombreuses méthodes pour analyser spatialement l'information géographique, telles que les mesures de distance, de surface et de volume, les caractéristiques spatiales de distribution des phénomènes, les interactions entre l'espace et le temps, etc.

Dans ce qui suit, nous détaillons les principales méthodes d'interrogation spatiale généralement utilisées lors de la recherche avec un SRIG. Ces méthodes se caractérisent par l'utilisation de trois types de relations spatiales permettant de spécifier la relation entre ce qui doit être extrait et un emplacement de référence. Ces types de relations sont la proximité, la topologie et l'orientation.

Requête de proximité. Dans une requête de proximité, les données sont extraites à une certaine distance de l'emplacement de référence. Par exemple, il est possible de trouver des documents relatifs à des lieux situés à une distance donnée d'une ville ou à une certaine distance d'une rivière ou d'une route. Plutôt que de spécifier une distance exacte, un utilisateur pourrait spécifier une relation de proximité sans contrainte précise sur la distance. L'emplacement des documents est représenté par une empreinte géographique, généralement sous la forme d'un point ou d'un rectangle. Cette géométrie est comparée à l'emplacement de la géométrie de l'objet de référence afin de mesurer la distance. Un exemple de requête serait « *quelles sont les villes accessibles à* 100 *km de Toulouse ?* »

Requête topologique. Les relations topologiques définissent la nature de la connectivité entre une paire d'objets géométriques. Les plus courantes de ces relations sont *à l'intérieur*, dans laquelle la géométrie de l'objet est entièrement présente dans la géométrie de référence, *rencontre* ou *touche*, dans laquelle la limite d'un objet coïncide avec la limite d'un autre, *recoupe*, dans laquelle seule une partie d'un objet coïncide avec l'intérieur de l'autre, *égale*, dans laquelle les deux géométries sont identiques, et *disjoint* qui fait référence à une séparation complète des

objets. Un exemple de requête topologique serait de trouver les documents dont la portée géographique se situe à l'intérieur des frontières d'un pays donné. Un autre exemple est la recherche de documents qui se rapportent aux pays voisins d'un pays.

Requête d'orientation ou de direction. Bien que la direction soit largement considérée comme l'un des principaux types de relations spatiales, elle est rarement mise en œuvre dans les SRIG ou les opérateurs d'interrogation de bases de données spatiales. Un exemple de requête serait de trouver les documents relatifs à des zones « *au Nord de Toulouse* », mais comme la notion de « *Nord* » est vague et dépend du contexte, il n'y a pas d'interprétation standard. De même, il n'est pas courant, lors de la recherche de données géospatiales, de vouloir spécifier des relations directionnelles de manière quantitative, comme un angle particulier.

1.4 *Ordonnancement de pertinence*

1.4.1 *Notion de pertinence pour la RIG*

Étant donné une requête, les SRIG doivent être capables de récupérer et ordonner les documents indexés qui sont pertinents, c.-à-d. qui répondent aux besoins d'information de l'utilisateur. Les documents sont généralement classés selon une fonction/modèle d'ordonnancement de la pertinence, et les documents les mieux classés sont retournés à l'utilisateur. En RI traditionnelle, la fonction d'ordonnancement de la pertinence s'appuie, à son niveau le plus simple, sur le calcul de similarités des correspondances des mots entre la requête et le document, bien que dans la pratique, de nombreuses caractéristiques sont utilisées pour coder des informations contextuelles importantes par rapport aux besoins d'information des utilisateurs (Baeza-Yates et Ribeiro-Neto, 1999). La principale différence entre la RI traditionnelle et la RIG est l'importance primordiale de la pertinence géographique des documents pour saisir la proximité géographique, le périmètre et d'autres relations spatiales. Celle-ci est souvent mesurée en utilisant des méthodes de similarité spatiale et d'ordonnancement. La plupart des SRIG traitent séparément les composantes textuelles et spatiales de la recherche, en combinant les scores de similarité textuelle et spatiale pour produire une seule liste ordonnée.

La plupart des architectures comportent une composante d'ordonnancement. Soit une requête q et un document d, le système applique une fonction/modèle d'ordonnancement de de la pertinence, $f(q, d)$, pour attribuer un score aux documents retournés. Les documents sont triés en fonction de leurs scores. En RIG, le système combine généralement les attributs spatiaux (le *où*) et textuels (le *quoi*)

pour calculer le score de pertinence. Un processus d'ordonnancement comprend les étapes suivantes (Martins *et al.*, 2005) :

1. transformer l'emplacement et les opérateurs spatiaux de la requête en une ou plusieurs empreintes géographiques. Les opérateurs spatiaux peuvent être utilisés en modifiant l'empreinte géométrique associée aux toponymes de la requête ou en générant des toponymes supplémentaires (c.-à-d. en élargissant la requête) ;
2. quantifier le degré de correspondance entre la requête et les empreintes géographiques du document en utilisant une mesure de similarité spatiale ;
3. produire un classement des documents qui correspondent à la ou les empreintes géographiques de la requête. Le classement s'appuie sur la similarité entre l'empreinte de la requête et celle du document qui est combinée avec le classement des documents effectué à partir de similarités non spatiales.

Nous l'avons vu, en RI traditionnelle comme en RIG, la notion de pertinence doit être prise en compte, car la fonction clé d'un SRIG est d'attribuer un score de pertinence aux documents, pour estimer la mesure dans laquelle ils sont susceptibles de répondre aux besoins d'information spatiale d'un utilisateur. Sabbata *et al.* (2012) ont décrit la pertinence géographique comme « la pertinence d'une entité géographique dans un contexte d'utilisation spécifique ». Ainsi, une entité géographique peut se référer à des documents géoréférencés, mais aussi à des entités physiques individuelles dans le monde réel.

Les SRIG doivent donc garantir que les informations récupérées sont pertinentes par rapport au sujet thématique et aussi par rapport à la partie spatiale de la requête de l'utilisateur ou à sa localisation (p. ex. sa position actuelle). La pertinence est jugée à partir des relations spatiales entre l'emplacement exprimé dans le besoin d'information et les empreintes spatiales identifiées dans un document (Cai, 2002). Les aspects spatiaux et thématiques constituent donc les conditions de base de la pertinence dans un SRIG et coïncident avec la notion de pertinence thématique utilisée dans les SRI traditionnels. Dans de nombreuses situations, en particulier celles impliquant des utilisateurs mobiles, le temps est également un facteur important qui influence la pertinence. Par exemple, Palacio *et al.* (2010) ont décrit un SRIG qui indexe et récupère des documents en fonction des dimensions thématiques, spatiales et temporelles. Sabbata *et al.* (2012) ont étudié vingt-neuf critères possibles liés à la pertinence géographique. Ces critères, résumés dans le Tableau 2.1, sont regroupés en quatre classes : ceux liés aux propriétés de l'entité géographique, ceux liés à la géographie, ceux qui saisissent la manière dont l'entité est représentée dans le système d'information, et ceux qui servent à juger la manière dont l'information est présentée à l'utilisateur.

Propriétés	Géographie	Information	Présentation
Actualité Adéquation Couverture géographique Nouveauté	Proximité spatiale Proximité temporelle Proximité spatio-temporelle Direction Visibilité Hiérarchie Groupe Colocalisation Règles d'association	Spécificité Disponibilité Exactitude Devise Fiabilité Vérification Affectivité Curiosité Familiarité Variété	Accessibilité Clarté Tangibilité Dynamisme Qualité de présentation

Tableau 2.1 – Critères d'évaluation de la pertinence géographique (Sabbata *et al.*, 2012).

1.4.2 *Calculer et combiner la similarité spatiale*

Les mesures de similarité spatiale sont utilisées pour estimer ou inférer la pertinence géographique des documents pour une requête donnée (Hill, 2009; Larson, 2011; Cai, 2011). Le processus d'ordonnancement dans le SRIG s'appuie donc sur la quantification de la similarité entre l'empreinte de la requête et celle du document. Le score de similarité est une estimation de la pertinence par rapport au besoin de l'utilisateur, et les documents sont classés par ordre décroissant des scores de pertinence. Une hypothèse courante veut que les documents spatialement plus proches de l'emplacement de la requête soient plus pertinents pour l'utilisateur que ceux qui en sont plus éloignés (Tobler, 1970).

Généralement, les SRIG commencent par identifier les correspondances topologiques et géométriques. Une mesure de similarité spatiale est ensuite utilisée pour calculer la force de ces correspondances en fonction du degré du chevauchement spatial. Différentes mesures, prenant en compte les tailles relatives de la requête et les empreintes des documents, peuvent être utilisées. Cette méthode rejoint la méthode de normalisation de la longueur des documents utilisée en RI traditionnelle (Baeza-Yates et Ribeiro-Neto, 1999). Par exemple, en supposant que q est la la région couverte par la requête, d celle couverte par le document et o la zone de chevauchement entre q et d, la similarité $sim(q, d)$ pourrait être exprimée comme (Hill, 2009; Larson, 2011) :

$$sim(q, d) = 2 \times \frac{o}{q + d} \tag{2.3}$$

Larson et Frontiera (2004) et Frontiera *et al.* (2008) ont proposé d'autres mesures de similarité spatiale, notamment la distance de Hausdorff, une mesure de comparaison des formes qui calcule la distance entre deux sous-ensembles d'un espace

métrique noté (E, δ). Pour chaque point $x \in q$, nous trouvons simplement la distance la plus courte par rapport à l'autre ensemble d. La distance maximale trouvée parmi tous les points de q est conservée comme la distance de Hausdorff.

$$sim(q,d) = \max \left\{ \sup_{y \in d} \delta(X,q), \quad \sup_{x \in q} \delta(x,d) \right\}. \tag{2.4}$$

D'autres facteurs, tels que le comptage de la population, les relations ontologiques (Andrade et Silva, 2006) ou une combinaison de plusieurs facteurs, p. ex. la distance et les relations ontologiques (Larson et Frontiera, 2004; Zaila et Montesi, 2015), peuvent être utilisés pour influencer le score de similarités.

Dans les SRIG qui traitent séparément les composantes thématiques et spatiales d'une requête, il est nécessaire de combiner les scores de pertinence pour ne conserver qu'une seule liste ordonnée. Cela peut être réalisé en calculant un score de similarité textuelle et spatiale combiné pour chaque document, ou en fusionnant les listes. Dans le cas d'une combinaison de scores, une méthode courante consiste à utiliser une combinaison linéaire (Andrade et Silva, 2006; Larson, 2011; Chen *et al.*, 2013). Pour une requête q et un document d, le score combiné $comb(q,d)$ est donné par :

$$comb(q,d) = \alpha_1 \cdot textsim(q_t, d_t) + \alpha_2 \cdot geosim(q_s, d_s) \tag{2.5}$$

où $textsim(q_t, d_t)$ représente la similarité thématique, calculée en utilisant par exemple le modèle BM25 (Robertson et Jones, 1976), et $geosim(q_s, d_s)$ la similarité spatiale. Les poids α_1 et α_2 sont les poids qui permettent de refléter l'importance relative des composants géographiques et textuels, et $\alpha_1 + \alpha_2 = 1$. Martins *et al.* (2005) ont proposé d'autres méthodes pour combiner les scores. D'autres approches permettant de combiner des listes de classement de documents utilisent les positions de classement des documents (p. ex. méthode de Borda) ou utilisent une combinaison de méthodes s'appuyant sur le score et le classement (Palacio *et al.*, 2010). D'autres approches ont fait appel à des modèles probabilistes pour combiner les scores et prédire la pertinence des documents en réponse à une requête (Frontiera *et al.*, 2008), ou des méthodes d'apprentissage automatique dérivées du *Learning to Rank* (Liu, 2009).

1.5 *Principale problématique et périmètre de la thèse*

La qualité d'un SRIG repose fondamentalement sur sa capacité à déterminer la portée géographique des documents et des requêtes. Bien qu'une étape essentielle, elle reste difficile en raison de la richesse et de l'ambiguïté du langage naturel (Amitay *et al.*, 2004; Clough *et al.*, 2004).

Dans un texte, les références explicites ou implicites aux lieux sont communément appelées géoréférences. Le géoréférencement, aussi appelé prédiction de l'emplacement, implique d'associer des informations contenues dans les documents à un lieu physique. Les géoréférences peuvent prendre plusieurs formes, par exemple, tous les éléments suivants sont des géoréférences qui peuvent nous permettre de localiser la Tour Eiffel à Paris :

— Champ de Mars, 5 Avenue Anatole - Adresse postale
— Tour Eiffel - Nom formel du lieu (aussi appelé toponyme)
— 75007 Paris - Code Postal
— 48.858423,2.2942882 - Coordonnées géographiques
— Dame de Fer - Surnom de la Tour Eiffel

Les géoréférences ont un certain nombre de propriétés importantes (Hill, 2009). Elles doivent être sans ambiguïté, c.-à-d. qu'elles ne doivent se référer qu'à un seul lieu, ce qui est généralement le cas dans un cadre de référence donné. Les géoréférences doivent également être, dans la mesure du possible, persistantes dans le temps. En pratique, c'est rarement les cas, les noms des lieux peuvent changer, les systèmes de coordonnées peuvent évoluer, etc.. Enfin, les géoréférences sont généralement associées à une granularité implicite.

Le point de départ est généralement un document contenant un texte écrit en langage naturel. La Figure 2.11 présente la page Wikipedia associée à la Tour Eiffel et comprend plusieurs formes de géoréférences, telles que des coordonnées géographiques, des toponymes (p. ex. Tour Eiffel) et d'autres références à des régions géographiques (p. ex. Paris). En lisant ce type de document, l'identification des références à l'emplacement aide à comprendre son contexte géographique. Ainsi, nous pouvons facilement identifier les lieux importants mentionnés dans le document et éliminer les références à des lieux qui ne sont pas pertinents. Réaliser cette tâche automatiquement est un défi central mais complexe. Bien que les références à la localisation exprimées en langage naturel puissent prendre de nombreuses formes, la plupart des travaux en RIG se sont concentrés sur le traitement des toponymes et des adresses (Leidner, 2006).

À partir d'un document, la tâche prédiction de l'emplacement, consiste donc à identifier sans ambiguïté les références géographiques et, en général, à leur attribuer des coordonnées spatiales. Cette tâche a été largement étudiée pour des articles Wikipedia (Roller *et al.*, 2012; Wing et Baldridge, 2014), des pages web (Amitay *et al.*, 2004; Zong *et al.*, 2005) et des documents traditionnels (Woodruff et Plaunt, 1994; Purves *et al.*, 2007). Diverses techniques de traitement et d'analyse de texte ont été utilisées pour effectuer ce processus, la plus répandue étant la reconnaissance et la désambiguïsation d'entités nommées (Larson, 1996; McCurley, 2001; Leidner et Lieberman, 2011), qui consiste à identifier dans le texte

Figure 2.11 – Exemple d'une page Wikipedia avec différentes formes de géoréférences, telles que le nom de ville, la région et les coordonnées géographiques. (©*Wikipedia*)

les passages qui font référence un lieu, et à déterminer l'emplacement physique unique auquel ils se rapportent. Ces géoréférences peuvent ensuite être utilisées pour déterminer la portée géographique du document, c.-à-d. l'emplacement ou les emplacements auxquels le contenu du document, ou une partie de celui-ci, est supposé être associé (Ding *et al.*, 2000; Andogah *et al.*, 2012).

Dans le cadre de ce manuscrit, nous nous intéressons à la problématique de résolution de la portée géographique des géotextes sous l'angle de la prédiction de l'emplacement. Plus spécifiquement, nous intéressons à des géotextes issus des RSNs. En conséquence, nous développons dans la suite de ce chapitre, les différentes approches de l'état-de-l'art permettant de répondre à cette problématique.

2 Résolution de la portée géographique des géotextes

Ces dernières années, les RSNs tels que Facebook, Twitter ou Instagram sont devenus des espaces populaires d'échanges permettant d'établir des liens sociaux et de partager de l'information textuelle et audiovisuelle. Avec l'augmentation de la connectivité des utilisateurs et la prédominance des smartphones, de plus en plus de ressources associées à des coordonnées géographiques, c.-à-d. des géotextes, sont créés chaque jour. Cela ouvre de nombreuses opportunités pour faire le lien entre le monde social en ligne et le monde physique, et développer des nouvelles applications de RIG permettant de répondre aux besoins du monde réel. Twitter est par exemple connu pour être une plateforme efficace pour détecter les sujets émergents, notamment les foyers d'épidémies ou les régions touchées par des catastrophes naturelles (Cheong et Cheong, 2011; Kumar *et al.*, 2011). Connaître la porté géographique des publications sur les RSNs, permet de comprendre ce qui se passe dans la vie réelle et peut ainsi aider aux rapports d'urgence (Imran *et al.*, 2015; Kumar et Singh, 2019) et la gestion de crise (Vieweg *et al.*, 2010; Lingad *et al.*, 2013), mais pas seulement. De nombreuses applications de RIG ont vu le jour, en particulier pour l'assistance touristique, avec la recommandation d'événements (Yuan *et al.*, 2013; Yin *et al.*, 2015) ou de POIs (Deveaud *et al.*, 2015; Bothorel *et al.*, 2018), ou encore le résumé spatio-temporel (Rakesh *et al.*, 2013; Mallela *et al.*, 2017). La maturité croissante des approches d'apprentissage automatique et la nécessité de méthodes généralisables applicables à de très grands volumes de données en temps réel ont conduit à une nouvelle famille de méthodes qui, plutôt que d'explorer les toponymes explicitement contenus dans le texte, cherchent à apprendre comment l'emplacement est décrit de manière plus générale dans le texte (Ahern *et al.*, 2007; Kinsella *et al.*, 2011; O'Hare et Murdock, 2013). L'idée est qu'à partir des nombreux documents associés à des coordonnées, il est possible d'identifier des ensembles de mots qui sont associés à des régions particulières de l'espace. L'ensemble des fréquences de mots pour une région donnée est appelé modèle de langue. Un premier exemple a été proposé par Ahern *et al.* (2007), qui ont utilisé des méthodes de *clustering* (k-moyennes) combinées avec le TfIdf pour sélectionner les mots-clés significatifs contenus dans les tags Flickr, qui ont ensuite été attribués aux cellule d'une grille à différents niveaux de granularité.

Ainsi, si le problème de résolution de la portée géographique des documents, sous l'angle de la prédiction de l'emplacement a largement été étudié pour des documents traditionnels (Wikipedia, pages web), il reste aujourd'hui un défi pour ceux issus des RSNs. Par exemple, la taille des tweets, limitée à 280 caractères, exige de la brièveté dans l'écriture, ce qui donne lieu à un vocabulaire informel uniquement utilisé dans les RSNs. De plus, les publications en ligne ont tendance

à comporter de nombreuses abréviations non standards, des erreurs typographiques, l'utilisation d'émoticônes, d'ironie, de sarcasmes et de sujets populaires, appelés *hashtags* (Cheng *et al.*, 2010; Liu *et al.*, 2012). Ces textes non conventionnels et non structurés rendent les approches classiques de TALN et de RI peu efficaces, conduisant à un défi intéressant pour l'analyse de contenu social. L'étude de l'état-de-l'art sur l'appariement géographique des contenus générés par les utilisateurs des RSNs révèle l'existence de trois grands axes de recherche : la prédiction de l'emplacement du contenu généré par l'utilisateur, la prédiction de l'emplacement mentionné dans le texte et la prédiction sémantique de l'emplacement (Ajao *et al.*, 2015; Zheng *et al.*, 2018; Haldar *et al.*, 2019).

Dans cette section, nous décrivons les trois principales approches pour la prédiction de l'emplacement. Nous commençons par aborder dans la Section 2.1, la prédiction de l'emplacement du contenu généré par l'utilisateur. Nous continuons ensuite dans la Section 2.2 en présentant les approches pour la prédiction de l'emplacement mentionné dans le texte. Enfin, dans la Section 2.3, nous abordons la tâche de la prédiction sémantique de l'emplacement, que nous adressons dans nos travaux de recherche.

2.1 *Prédiction de l'emplacement du contenu généré par l'utilisateur*

Le premier axe de recherche consiste à prédire l'emplacement du contenu généré par l'utilisateur. La plupart de ces travaux portent plus particulièrement sur le réseau social Twitter, qui a gagné en popularité pour communiquer, partager des idées et diffuser des publicités (Kwak *et al.*, 2010; Teevan *et al.*, 2011). La finalité de cette tâche consiste à estimer l'emplacement géographique des contenus publiés en ligne, qu'ils soient géotaggés ou non. Par exemple, dans la Figure 2.12, l'objectif de cette tâche serait de déterminer l'emplacement du tweet, c.-à-d. son lieu d'émission (*tweet location*) ou le lieu de résidence de l'utilisateur (*home location*). Toutefois, concernant Twitter, il a été rapporté qu'entre 1% et 4% des tweets contiennent un géotag explicite (Hecht *et al.*, 2011; Graham *et al.*, 2014; Ryoo et Moon, 2014). De ce fait, inférer le géotag ou l'emplacement d'où ces contenus ont été publiés a fait l'objet de nombreuses études (Li *et al.*, 2011a; Lee *et al.*, 2014; Ajao *et al.*, 2015; Chong et Lim, 2018; Hoang et Mothe, 2018; Zheng *et al.*, 2018), permettant ainsi de mieux comprendre leur contexte, et dresser un portrait plus complet de la mobilité des utilisateurs.

Deux niveaux de granularité des emplacements ont été étudiés dans la littérature. Le premier, connu sous le nom de granularité grossière de l'emplacement, vise à géolocaliser les publications en fournissant une estimation des coordonnées

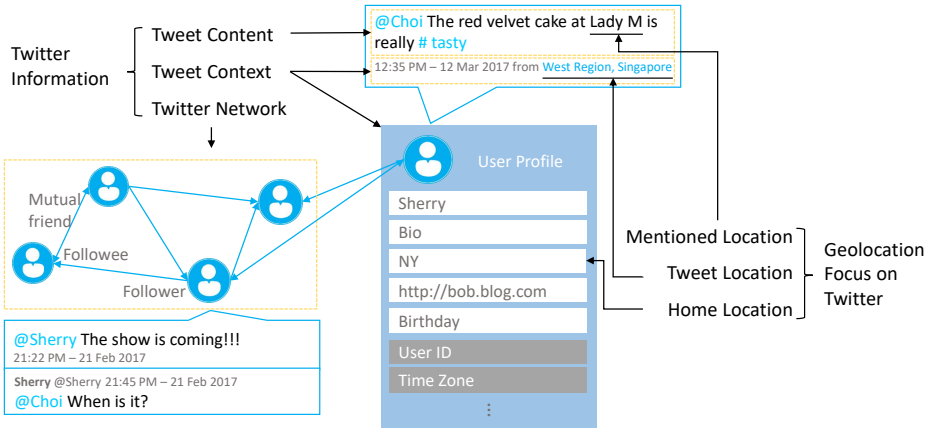


Figure 2.12 – Illustration du contenu d'un tweet (*tweet content*), du contexte (*tweet context*) et du réseau Twitter (*Twitter network*), ainsi que trois types de lieux : le lieu de résidence (*home location*), l'emplacement d'où le tweet a été publié (*tweet location*), et l'emplacement mentionné (*mentioned location*) (Zheng *et al.*, 2018).

GPS ou un emplacement discret grossier, c.-à-d. au niveau d'une grille, d'une ville ou d'une région (Ahmed *et al.*, 2013; Lee *et al.*, 2014; Ajao *et al.*, 2015). Le second niveau concerne la localisation précise, qui cherche à géolocaliser les publications en déduisant une estimation discrète du lieu (Li *et al.*, 2011a; Ji *et al.*, 2016; Zheng *et al.*, 2018). Dans l'ensemble, les méthodes proposées s'appuient sur le contenu des publications et/ou le contexte.

2.1.1 *Inférence de l'emplacement à partir du contenu*

L'approche la plus courante pour déterminer le lieu d'émission d'un géotexte à partir de son contenu consiste généralement à *(1)* identifier des mots locaux, c.-à-d. des mots qui montrent une forte empreinte spatiale, à l'aide de méthodes non-supervisées s'appuyant sur des mesures statistiques directement calculées sur les données, avec par exemple des estimations de densités (Laere *et al.*, 2014) ou des calculs d'IDF localement dépendants (Han *et al.*, 2012), ou à l'aide de méthodes supervisées en considérant la tâche comme un problème de classification (Cheng *et al.*, 2010) ; *(2)* modéliser les distributions locales des mots à l'aide de modèles de mélange Gaussien (GMM) pour obtenir des distributions lissées de l'utilisation des mots (Priedhorsky *et al.*, 2014; Flatow *et al.*, 2015; Chong et Lim, 2017a,b). Ce sont des approches dites centrées sur les mots. Par exemple, Priedhorsky *et al.* (2014) utilisent des modèles de mélange Gaussien (GMM) pour prédire les coordonnées des tweets. Comme l'information textuelle disponible dans un tweet est pauvre, les auteurs ont proposé de modéliser l'utilisation spatiale des mots et des n-grammes. D'après leurs résultats, lorsque les modèles considèrent des

n-grammes de mots, même rares, les performances des modèles de prédiction sont améliorées. Flatow *et al.* (2015) ont aussi modélisé l'utilisation des n-grammes spatiaux avec un modèle Gaussien. Dans le même esprit que les mots locaux, les auteurs préfèrent utiliser des n-grammes géospécifiques, c.-à-d. ceux dont les tweets sont principalement situés dans une petite région sur la carte. Chong et Lim (2017a,b) ont quant à eux appliqué un modèle d'apprentissage d'ordonnancement qui encode le contenu du tweet par une estimation lissée de la probabilité qu'un mot apparaisse dans un lieu.

D'autres travaux adoptent une approche différente, qui donne un rôle plus important aux emplacements (Kinsella *et al.*, 2011; Li *et al.*, 2011a; Ozdikis *et al.*, 2019). Ce sont des approches dites centrées sur la localisation. Kinsella *et al.* (2011) traitent les tweets et les emplacements avec des modèles de langue lissés par la formule de Dirichlet. L'ordonnancement des emplacements candidats s'effectue à partir des probabilités qu'un modèle de langue associé à un emplacement génère le tweet, autrement dit, ils calculent la divergence de Kullback-Leibler entre les modèles d'un tweet et d'un emplacement. Li *et al.* (2011a) se sont aussi appuyés sur ce calcul de divergence, en complétant les modèles de langue des régions pauvres en information avec de l'information issue de pages web. Dans leurs travaux, Ozdikis *et al.* (2019) ont proposé de prédire l'emplacement des tweets en analysant la distribution géographique des textes issus de tweets à l'aide d'estimation de densités (KDE). Leur KDE adapté localement permet d'identifier les bigrammes qui présent un écart de distribution significatif par rapport à des modèles d'unigrammes sous-jacents.

2.1.2 *Inférence de l'emplacement à partir du contexte*

Nous l'avons évoqué en introduction de cette section, les publications sur les RSNs sont généralement courtes et bruitées (vocabulaire non standard, erreurs typographiques, etc.), ce qui rend difficile la prédiction de leur emplacement. Pour enrichir cette information, certains travaux utilisent des données contextuelles, telles que le réseau d'amitié (Sadilek *et al.*, 2012; Chong et Lim, 2017b; Bakerman *et al.*, 2018), l'information temporelle (Li *et al.*, 2011a; Yuan *et al.*, 2013; Dredze *et al.*, 2016), l'information spatiale (Schulz *et al.*, 2013) ou un mélange de ces deux dernières (Chong et Lim, 2017a).

Le travaux de Sadilek *et al.* (2012) exploitent en temps réel les emplacements des amis de l'utilisateur et l'historique des lieux qu'il a visité. Un réseau bayésien dynamique est entraîné à partir des séquences de visite de chaque utilisateur, avec l'emplacement de ses amis, l'heure de la journée et le jour de la semaine comme caractéristiques. Bakerman *et al.* (2018) modélisent conjointement le contenu des

tweets sous forme d'unigrammes et les informations du réseau à l'aide de modèles de mélange gaussien, pour estimer efficacement l'emplacement des utilisateurs.

Comme l'ont montré de précédents travaux (Mahmud *et al.*, 2014), l'heure de publication des contenus en ligne est un bon indicateur de l'emplacement des utilisateurs. Un horodatage peut être instructif si suffisamment de données historiques sont fournies pour des lieux. En effet, l'historique des publications peut suggérer qu'un club a tendance à être actif la nuit, alors qu'un parc aura tendance à recevoir plus de publications le weekend. S'inspirant de ces constats, Li *et al.* (2011a) ont utilisé les distributions des dates de publication de tweets dans des lieux, selon trois échelles de périodes différentes : le jour, la semaine et le mois. Étant donné un tweet et sa date de publication, les probabilités des trois distributions sont combinées linéairement pour recommander des lieux. Dredze *et al.* (2016) se servent quant à eux du fuseau horaire et l'heure de publication du tweet comme caractéristiques d'un classifieur, pour prédire la ville ou le pays associé au tweet. Ils ont constaté que les modèles temporels cycliques ont des effets sur les résultats des prédictions.

Schulz *et al.* (2013) se sont plutôt focalisés sur l'information spatiale, en collectant les indicateurs de localisation des tweets à partir des profils d'utilisateurs, tels que les lieux de résidence, les sites web et fuseaux horaires déclarés par les utilisateurs dans leur profil, ainsi que les lieux explicitement mentionnés dans les tweets. En interrogeant des bases de données géographiques, ces indicateurs sont associés à des régions administratives sous forme de polygones. Finalement, ces derniers sont regroupés pour produire une distribution spatiale des emplacements possibles du tweet. Enfin, Chong et Lim (2017a) ont proposé de combiner les informations contextuelles en considérant l'activité des lieux (c.-à-d. l'affluence) et l'historique des lieux visités par les utilisateurs, pour aider à prédire l'emplacement des tweets. Ils ont étudié les périodes d'activités des lieux afin d'estimer à l'aide d'un KDE, la probabilité qu'un lieu soit populaire à une période donnée.

2.2 *Prédiction de l'emplacement mentionné*

Le deuxième axe de recherche porte sur la prédiction de l'emplacement mentionné dans le texte. Celui-ci vise à associer des parties de textes issues des publications en ligne avec des entités spatiales généralement répertoriées dans des bases de connaissances. En reprenant l'exemple illustré dans la Figure 2.12, ce deuxième axe de recherche aurait pour objectif de détecter l'emplacement nommé « *Lady M* » et de l'associer avec son entité spatiale correspondante, le *Lady M* de Singapour, une boutique de gâteau. Cette association permet d'augmenter le contenu des publications avec du contexte géographique ouvrant la voie à une multitude d'ap-

plications. En effet, sur les RSNs, les utilisateurs publient régulièrement des messages pour commenter des lieux d'intérêts, tels que des restaurants, des centres commerciaux ou des parcs. De plus, lors de catastrophes naturelles ou d'événements, les réseaux sociaux sont aussi largement utilisés pour diffuser l'information au près des utilisateurs. En plus d'attacher un géotag à leurs publications, les utilisateurs peuvent révéler les zones touchées en les mentionnant directement. La reconnaissance des lieux mentionnés est donc une étape cruciale pour recueillir des informations sur les utilisateurs et les événements (Lingad *et al.*, 2013; Imran *et al.*, 2015).

La prédiction de l'emplacement mentionné s'effectue en deux étapes : *(1)* la reconnaissance d'entités nommées, qui consiste à extraire les morceaux de textes qui mentionnent des lieux ; et *(2)* la désambiguïsation, qui met en correspondance les mentions de lieux avec les bonnes entrées dans une base de connaissance. Les problèmes de la reconnaissance d'entités nommées et de la désambiguïsation ont largement été étudiés pour des documents classiques et bien écrits, tels que des articles de presses ou des documents Wikipedia (Shaalan, 2014). Il est connu que la variabilité et l'ambiguïté des mentions d'entités sont deux difficultés majeures inhérentes à cette tâche. Néanmoins, avec les publications sur les RSNs, ces deux aspects sont rendus plus difficiles à appréhender, du fait de la nature bruitée et courte des publications.

2.2.1 *Reconnaissance de l'emplacement mentionné*

Pour déterminer l'emplacement mentionné dans les publications, la première étape consiste à reconnaître les parties de textes qui mentionnent des lieux, c'est la reconnaissance d'entités nommées (NER). Traditionnellement, les modèles développés pour ce type de tâche utilisent des algorithmes d'apprentissage tels les champs aléatoires conditionnels, ou *Conditional Random Fields* (CRFs) (Lafferty *et al.*, 2001) s'appuyant sur des caractéristiques linguistiques (p. ex. étiquettes morpho-syntaxiques). Des modèles tels que *StanfordNER* ou *OpenNLP* ont obtenu de très bonnes performances pour reconnaître les entités dans des textes formels (Ratinov et Roth, 2009). Néanmoins, Gelernter et Mushegian (2011) ont remarqué que l'utilisation de ces modèles sur les textes issus des RSNs n'ont pas permis de détecter avec précision les entités, dont les noms de lieux, en particulier lorsqu'elles étaient abrégées de manière inhabituelle ou que les majuscules sur les noms de lieux étaient absentes.

Plusieurs travaux de recherche (Ritter *et al.*, 2011; Liu *et al.*, 2011; Li *et al.*, 2012; Liu *et al.*, 2013; Lingad *et al.*, 2013) ont pallié ces limites. Ritter *et al.* (2011) ont proposé de reprendre le processus pour la NER en l'adaptant aux tweets. Pour cela, ils ont utilisé la classification de Brown (Brown *et al.*, 1992) afin d'identifier

des groupes de variations des mots (p. ex. « at » et « @ »), et un second classifieur pour déterminer si chaque lettre en capitale dans un tweet est informative. De même, Liu *et al.* (2011, 2013) ont proposé un modèle de normalisation de tweets permettant de corriger les mots informels avant d'effectuer la NER. Les auteurs ont ensuite entraîné un classifieur s'appuyant sur la méthode des k plus proches voisins pour alimenter le modèle de NER avec des informations globales, c.-à-d. comment le mot est étiqueté dans d'autres documents. Lingad *et al.* (2013) ont quant à eux réentraîné quatre outils de NER (*StanfordNER*, *OpenNLP*, *TwitterNLP* et *Yahoo ! Placemaker*) avec des tweets relatifs à des catastrophes.

Hormis les approches détaillées ci-dessus, qui font de la reconnaissance d'entités générales, d'autres travaux se caractérisent par la reconnaissance d'entités géographiques exclusivement, et l'utilisation de bases de données géographiques (ou *gazetteers*), tels que *Geonames* (Zhang et Gelernter, 2014; Malmasi et Dras, 2015) ou *Foursquare* (Li et Sun, 2014, 2017). Zhang et Gelernter (2014) s'appuient sur un outil de reconnaissance qu'ils ont développé. Ils utilisent conjointement un analyseur syntaxique de localisation reposant sur un index géographique, un outil de reconnaissance établi sur le CRF, et un second analyseur syntaxique de rue/immeuble construit à partir de règles. Contrairement au travail précédent, Malmasi et Dras (2015) n'utilisent pas de CRF dans leur outil de reconnaissance des mentions de lieux. Ils se sont orientés vers un analyseur de dépendances pour identifier toutes les phrases nominatives, et effectuent un appariement approximatif avec les données contenues dans *Geonames*. Leurs critères d'appariement prennent en compte les structures des adresses et des POIs. Li et Sun (2014, 2017) ont quant à eux remarqué que les utilisateurs de Twitter mentionnent souvent des lieux par des abréviations. Ils ont donc choisi d'étendre leur index géographique avec les noms partiels des POIs fréquemment rencontrés dans les publications.

2.2.2 *Désambiguïsation de l'emplacement mentionné*

Une fois les mentions de lieux reconnues, nous pouvons commencer la désambiguïsation, c.-à-d. associer correctement ces mentions aux entrées d'une base de données géographiques. La difficulté réside dans le fait que différents lieux peuvent avoir les mêmes noms, et ce, à différents niveaux de granularité. Par exemple, le terme « Paris » peut se référer à la capitale de la France, ou à l'une des trente villes qui portent le même nom à travers le Monde. À un niveau plus précis, celui du POI, les chaînes de restaurants peuvent avoir de nombreuses succursales dans une même ville.

Traditionnellement, les modèles désambiguïsent une mention à la fois (Milne et Witten, 2008). Pour exploiter les dépendances entre les mentions, des approches de désambiguïsation par paire (Kulkarni *et al.*, 2009) et globale (Hoffart *et al.*, 2011)

sont proposées. Ces approches supposent que les décisions de désambiguïsation pour les mentions multiples doivent être cohérentes. Ainsi, Zhang et Gelernter (2014) proposent de considérer la structure hiérarchique des lieux pour aider à la désambiguïsation. Les candidats potentiels sont ordonnés à l'aide d'un classifieur linéaire, le séparateur à vaste marge (SVM). Li et Sun (2014) optent pour une cohérence de désambiguïsation, non pas au niveau du tweet, mais au niveau de l'utilisateur. Ils supposent que les lieux mentionnés dans les tweets d'un utilisateur se trouvent généralement dans sa ville de résidence. Ils commencent donc par identifier la ville de résidence de l'utilisateur en agrégeant les lieux candidats déterminés à partir des mentions, puis affinent ces candidats avec la ville de résidence. Ji *et al.* (2016) ont étudié la désambiguïsation collective des mentions de POIs dans les tweets. Leur mesure de cohérence s'appuie sur la distance moyenne entre les POIs candidats sélectionnés pour les mentions reconnues.

Dans les approches conventionnelles, la désambiguïsation des lieux s'appuie sur les résultats de l'étape de reconnaissance des entités nommées. Si ces derniers sont erronés, par exemple avec des limites inexactes, la désambiguïsation peut échouer en raison de l'incapacité à trouver des candidats potentiels dans la base de donnés géographique. Pour palier cela, des auteurs (Guo *et al.*, 2013; Ji *et al.*, 2016) suggèrent de permettre à l'information de circuler entre les deux composants, de reconnaissance et de désambiguïsation. Guo *et al.* (2013) utilisent des SVMs pour optimiser conjointement la reconnaissance des mentions et leur désambiguïsation. De même, Ji *et al.* (2016) considèrent conjointement les caractéristiques des deux composants et déterminent à l'aide de l'algorithme de recherche en faisceau (Zhang et Clark, 2008), la meilleure combinaison pour prédire correctement les lieux mentionnés.

Comme pour la prédiction de l'emplacement de l'utilisateur (Section 2.1), des informations contextuelles peuvent être explorer pour désambiguïser les mentions de lieux (Fang et Chang, 2014; Han *et al.*, 2018). Par exemple, Fang et Chang (2014) ont considéré conjointement les géotags et les dates de publication des tweets pour désambiguïser les mentions. Han *et al.* (2018) ont proposé une approche non-supervisée, s'appuyant sur un réseau bayésien pour modéliser les relations entre les géotags des tweets et les lieux mentionnés.

2.3 *Prédiction sémantique de l'emplacement*

Le troisième axe de recherche porte sur la prédiction sémantique de l'emplacement, que nous abordons dans nos travaux. Cette tâche consiste à apparier des publications, géotaggées ou non, à des objets spatiaux sémantiquement liés, généralement représentés par des points d'intérêts (POIs). Autrement dit, en réalisant

cette tâche, nous tentons de répondre à la question : « *de quel lieu parle le message ?* ». Jusqu'ici, les travaux présentés faisaient l'hypothèse que si une publication parle, c.-à-d. mentionne implicitement ou explicitement un lieu, il est probable qu'elle soit publiée depuis celui-ci. Cependant, les utilisateurs pourraient parler d'un lieu qu'ils ont visités auparavant alors qu'ils se trouvent actuellement ailleurs. Ainsi, les lieux sémantiques, et l'emplacement de la publication représentée par son géotag, peuvent ne pas toujours coïncider.

Bien qu'apparaissant proche des tâches détaillées précédemment, le problème de prédiction sémantique de l'emplacement est sensiblement différent. Dans une tâche d'appariement d'entités nommées, seuls des fragments de textes sont mis en correspondance avec des données structurées, alors qu'ici nous nous intéressons à l'ensemble du contenu. De plus, elle diffère d'une tâche de RI ad-hoc puisque l'appariement implique un objet non structuré (p. ex. un tweet) et un objet structuré (p. ex. un POI). En outre, un et seul objet spatial pertinent doit correspondre au contenu de la publication.

La tâche de prédiction sémantique de l'emplacement est pertinente pour un grand nombre d'applications, telles que la recommandation de POIs (Zhang et Chow, 2015), le résumé (Nguyen *et al.*, 2015) ou la recherche géographique (Magdy *et al.*, 2014), qui ont le potentiel de fournir des services personnalisés aux utilisateurs des RSNs. Toutefois, l'étude de l'état-de-l'art révèle que très peu de travaux se sont intéressés à ce sujet, que ce soit pour l'appariement d'objets non-géotaggés, que nous détaillons dans la Section 2.3.1 ou l'appariement d'objets géotaggés, détaillé quant à lui dans la Section 2.3.2.

2.3.1 *Appariement d'objets non-géotaggés*

La première catégorie de travaux, portée par Dalvi *et al.* (2009a,b), aborde le problème d'associer des textes non-géotaggés, à savoir des critiques utilisateurs, à des objets spatiaux pertinents, ici des POIs. En d'autres termes, étant donné une remarque publiée par un utilisateur, il faut identifier le POI dont il est question.

Dans leur premier travail, Dalvi *et al.* (2009a) ont proposé un modèle d'appariement qui utilise un processus de génération multinomial analogue aux modèles de langue communs utilisés en RI ad-hoc (Hiemstra et Kraaij, 1999). Le processus sous-jacent s'appuie uniquement sur le contenu textuel et utilise un modèle de mélange (Équation 2.6) composé d'un modèle de langue pour la génération de critiques, qui intègre la description des POIs, et un modèle de langue de critiques générique.

$$P(r|e) = Z(r) \prod_{w \in r} ((1-\alpha)P(w) + \alpha P_e(w)) \tag{2.6}$$

où $Z(r)$ est un terme de normalisation dépendant de la longueur de la critique, $P_e(w)$ correspond à la probabilité que le mot w soit choisi dans l'objet e (c.-à-d. le modèle de langue pour la génération de critiques) et $P(w)$ correspond à la probabilité que le mot w soit sélectionné en fonction de sa distribution P (c.-à-d. le modèle de langue générique). Compte tenu du manque de données annotées, une normalisation sur la longueur du POI est appliquée au schéma de pondération des termes TfIdf pour estimer les paramètres du modèle. Les auteurs ont introduit un facteur d'amortissement qui tient compte de la brièveté des critiques. Les résultats de l'évaluation expérimentale ont montré qu'un niveau de performance raisonnable peut être atteint sur des modèles de référence dérivés du TfIdf. Néanmoins, d'après les résultats, il semble que les longues critiques soient plus difficiles à apparier.

Les auteurs ont poursuivi leur travail en introduisant un modèle de traduction pourvu d'une méthode d'estimation des paramètres plus robuste (Dalvi *et al.*, 2009b). Le modèle est capable de pallier l'inadéquation du vocabulaire entre les critiques et les objets spatiaux en améliorant le processus de génération avec des probabilités de traduction mot à mot. Ces probabilités sont estimées à l'aide de l'algorithme espérance-maximisation (EM) appliqué sur un ensemble de données d'apprentissage contenant des critiques de POIs associées à des attributs (nom, ville, cuisine). Le processus commence par choisir un attribut (indépendamment de l'objet), puis sélectionne un mot dans l'attribut et produit une traduction de ce mot selon le modèle de traduction dépendant de l'attribut. Formellement, la probabilité qu'une critique r soit générée à partir d'un objet e est définie par :

$$P(r|e) = Z(r) \prod_{w \in r} P(w|e) \tag{2.7}$$

où $Z(r)$ est une constante de normalisation qui dépend de la longueur de la critique r. La probabilité $P(w|e)$ qu'un mot w soit généré à partir d'un objet e est donnée par :

$$P(w|e) = \sum_k \alpha_k \sum_{u \in e_k} \beta_k(u|e) \cdot t_k(w|u) \tag{2.8}$$

où $k \in K$ est un ensemble d'attributs associés à l'objet e, $u \in U$ est l'ensemble des mots possibles dans les attributs des objets et $w \in V$ le vocabulaire des critiques. α_k sont les probabilités de choisir un attribut k. $t_k(\cdot|u)$ est la distribution sur V telle que la probabilité qu'un mot w soit traduit à partir de u est donnée par $t_k(w|u)$. Un mot u est tiré de l'ensemble e_k avec la probabilité proportionnelle $\beta_k(u|e)$. Les résultats de l'évaluation ont montré que le modèle de traduction dépasse les méthodes traditionnelles s'appuyant sur le TfIdf mais aussi des méthodes plus élaborées impliquant des modèles de mélange.

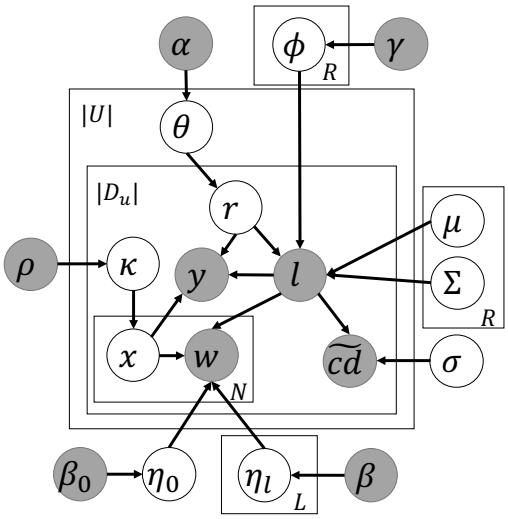


Figure 2.13 – Architecture du modèle supervisé bayésien (sBM) (Zhao *et al.*, 2016).

2.3.2 *Appariement d'objets géotaggés*

La seconde catégorie de travaux tente d'associer des géotextes, à savoir des tweets géotaggés, avec des points d'intérêts. Zhao *et al.* (2016) ont proposé sBM, un modèle bayésien supervisé pour associer les tweets géotaggés aux POIs, dont l'architecture est présentée dans la Figure 2.13. Contrairement aux approches de Dalvi *et al.* (2009a,b), les auteurs ont pris en compte les dimensions spatiales et le comportement des utilisateurs en plus du contenu textuel. Le modèle proposé est ainsi capable de saisir les intérêts des utilisateurs dans des régions latentes. Les coordonnées agissent comme des filtres spatiaux pour exclure les POIs situés trop loin du tweet, les comportements des utilisateurs fournissent les intérêts de l'utilisateur face à certains POIs ou catégories de POIs permettant de réduire l'espace de recherche, et le contenu textuel aide à identifier le véritable POIs parmi les POIs candidats restants.

Le processus de génération du modèle repose donc sur une distribution multinomiale combinée des tweets au travers les intérêts des utilisateurs, estimée par la probabilité de choisir un POI, et une distribution bivariée de la popularité du POI dans sa région correspondante. De plus, le modèle est capable de distinguer conjointement les tweets potentiellement reliés à un POI, c.-à-d. ceux qui discutent et/ou mentionnent de manière formelle ou non un POI, de ceux qui ne le sont pas, et d'associer les premiers au POI correspondant.

3 Discussion

Depuis l'introduction de la RIG dans les années 2000 (Baeza-Yates et Ribeiro-Neto, 1999) et les travaux qui ont été réalisés, l'importance de la localisation pour de nombreuses tâches liées à la recherche et à la récupération de textes non structurés est devenue de plus en plus évidente, non seulement grâce aux systèmes commerciaux qui se sont efforcés d'intégrer l'information géographique, mais aussi grâce à la recherche académique qui a souligné son importance. Nous l'avons discuté dans ce chapitre, la qualité d'un SRIG repose sur son aptitude à déterminer et valoriser la portée géographique des documents et des requêtes, qui reste aujourd'hui encore l'un des plus importants défis de ce domaine (Leidner et Lieberman, 2011). Parmi les défis les plus spécifiques en matière de géoréférencement figurent le géocodage efficace à différents niveaux de granularité (c.-à-d. valoriser efficacement les données à l'échelle locale et mondiale) et des moyens plus efficaces de traiter le langage spatial, y compris en tant qu'expressions spatiales (p. ex. « 10 km au nord de Toulouse »).

Par ailleurs, l'essor de la connectivité des utilisateurs, notamment grâce à l'utilisation des smartphones et la prolifération des réseaux sociaux numériques, tels que Twitter, Facebook ou Instagram, ont ouvert la voie à de nouveaux services, notamment axés sur la géolocalisation. De fait, depuis quelques années, de nombreuses applications ont été développées pour le traitement des données géolocalisées issues des réseaux sociaux. Parmi les applications possibles, dans le cadre de nos travaux de recherche, nous nous intéressons à l'association entre les géotextes et les objets physiques, et plus particulièrement au problème de prédiction sémantique de l'emplacement. Cette tâche se révèle utile pour de nombreuses applications telles que la gestion de crises (Imran *et al.*, 2015; Kumar et Singh, 2019), l'assistance touristique (Deveaud *et al.*, 2015; Yin *et al.*, 2015; Bothorel *et al.*, 2018), ou le résumé spatio-temporel (Rakesh *et al.*, 2013; Mallela *et al.*, 2017).

Comme détaillé dans la Section 2, jusqu'à présent, de nombreux travaux de recherche se sont concentrés sur la prédiction de l'emplacement des utilisateurs (Li *et al.*, 2011a; Ajao *et al.*, 2015; Zheng *et al.*, 2018) ou l'extraction et la désambiguïsation d'entités nommées (Ritter *et al.*, 2011; Liu *et al.*, 2011; Lingad *et al.*, 2013). La tâche de prédiction sémantique de l'emplacement a quant à elle été très peu étudiée par le passé (Dalvi *et al.*, 2009a,b; Zhao *et al.*, 2016) et demeure un défi pour plusieurs raisons qui conduisent généralement à de faibles taux de rappel et de précision :

1. la courte longueur des géotextes (p. ex. 280 caractères pour un tweet) et l'utilisation fréquente de mots non conventionnels (p. ex. abréviations, erreurs orthographiques, etc.) ;

2. l'ambiguïté des termes et l'inadéquation du vocabulaire entre les descriptions de deux objets géotextuels (p. ex. entre un tweet et un POI) dues à la rareté des données ;
3. un géotexte axé sur un POI peut mentionner un autre POI périphérique (p. ex. le tweet « *Superbe vue depuis la* ***Tour Eiffel*** *! Le restaurant vaut le détour ! Maintenant direction l'****Arc de Triomphe*** *!* ») ;
4. les utilisateurs peuvent publier des géotextes au sujet d'un POI alors qu'ils se trouvent loin de celui-ci ; et même si le géotexte est émis à proximité du POI, les coordonnées géographiques fournies par les dispositifs GPS sont souvent peu fiables (notamment dans les zones denses) et donc, le POI le plus proche du lieu de l'utilisateur n'est pas nécessairement le bon POI.

Quelques travaux ont tenté de pallier les difficultés énoncées ci-dessus (Dalvi *et al.*, 2009a,b, 2012; Zhao *et al.*, 2016). Ces approches portent notamment sur l'élaboration de modèles de langue spécifiques pour représenter les POIs et les géotextes (Dalvi *et al.*, 2009a,b). Ces modèles combinent modèles de langue génériques et des modèles reposant sur la distribution des mots dans les géotextes. Toutefois, ils se révèlent peu performants lorsqu'il y a une inadéquation de vocabulaire entre les sources (p. ex. la description d'un POI et les commentaires des utilisateurs). Des alternatives ont donc été proposées, avec le recours aux modèles de traduction (Dalvi *et al.*, 2012), ou plus récemment, avec des modèles bayésiens supervisés qui utilisent un processus génératif capable de capturer les relations entre les données spatiales et les données textuelles (Zhao *et al.*, 2016). La principale difficulté de ces modèles est de savoir comment formaliser et estimer à la fois les probabilités conditionnelles des entités à partir des textes, et les probabilités de traduction entre les mots.

Enfin, dans le contexte plus général de la RI, de nouvelles approches s'appuyant sur les réseaux de neurones ont été proposées, notamment pour pallier les problèmes inhérents à l'appariement de contenus hétérogènes et la discordance du vocabulaire (Onal *et al.*, 2017). Ces approches permettent d'encoder les relations syntaxiques et sémantiques des mots. Nous proposons d'explorer cette piste dans nos travaux de recherche.

4 Conclusion

Dans ce chapitre, nous avons commencé par introduire les notions de base relatives à la RIG en détaillant les concepts abordés dans ce manuscrit. Nous avons ensuite discuté les différentes approches de l'état-de-l'art pour déterminer la porté géographique des géotextes sous l'angle de la prédiction de l'emplacement. Nous

avons vu que ces travaux pouvaient se regrouper en trois catégories : la prédiction de l'emplacement du contenu généré par l'utilisateur, la prédiction des emplacements mentionnés dans le contenu, et enfin, la prédiction sémantique de l'emplacement qui a retenu notre attention. Toutes ces approches ont pour objectif de faire le pont entre le monde virtuel et le monde physique.

La prédiction sémantique de l'emplacement, peu étudiée jusqu'ici, a montré des performances limitées face aux spécificités des géotextes issus des réseaux sociaux (textes courts, vocabulaire non conventionnel). Dans le cadre de nos travaux de recherche, nous allons donc nous attaquer à ce problème en tentant de pallier les principales difficultés évoquées jusqu'à présent. Confortés par les performances des approches neuronales pour traiter des tâches de RI, nous avons choisi de les exploiter pour effectuer la prédiction sémantique de l'emplacement. Nous détaillons ces approches dans le chapitre suivant.

3

RÉSEAUX DE NEURONES POUR LA REPRÉSENTATION DISTRIBUÉE ET L'APPARIEMENT DE TEXTES ET DE GÉOTEXTES

Introduction

Les avancées dans l'apprentissage automatique ont fait apparaître un ensemble de méthodes non linéaires, connues sous le nom d'apprentissage structuré profond, ou plus simplement apprentissage profond (Deng et Yu, 2014). Ces nouvelles méthodes s'appuient sur différentes architectures neuronales, composées de plusieurs couches de traitement, pour apprendre des représentations des données avec plusieurs niveaux d'abstraction (LeCun *et al.*, 2015).

Les méthodes d'apprentissage profond se sont immiscées dans de nombreux domaines de l'intelligence artificielle, comme la vision par ordinateur (Krizhevsky *et al.*, 2012), la reconnaissance de la parole (Graves *et al.*, 2013) et la traduction automatique (Sutskever *et al.*, 2014). Bien que les réseaux de neurones artificiels aient été introduits dès 1943 par McCulloch et Pitts (1943), c'est notamment la démocratisation des processeurs graphiques (*Graphic Processing Units* ou GPU) (Nickolls *et al.*, 2008) qui a encouragé et accéléré la recherche dans le domaine de l'apprentissage profond. Leur structure hautement parallèle les rend plus efficaces que les processeurs classiques (*Central Processing Unit* ou CPU), pour des algorithmes qui traitent de grands blocs de données en parallèle tels que les réseaux de neurones.

Face au succès des approches neuronales dans la vision par ordinateur, les recherches en RI ont donné lieu à des modèles neuronaux pour l'apprentissage de représentations distribuées de mots (Bengio *et al.*, 2003; Pennington *et al.*, 2014; Mikolov *et al.*, 2013a,b), de phrases ou documents (Le et Mikolov, 2014; Hill *et al.*, 2016). Ces représentations distribuées, aussi appelées plongements lexicaux (ou *word embeddings*) sont des représentations vectorielles denses de valeurs réelles qui encodent la sémantique des mots en s'appuyant sur leur contexte. En capturant ainsi

la mesure dans laquelle les mots apparaissent dans des contextes similaires, les plongements lexicaux sont capables de représenter la similarité sémantique et syntaxique, dans la mesure où les représentations des mots similaires seront proches les unes des autres dans l'espace vectoriel. Pourtant, certains travaux (Iacobacci *et al.*, 2015; Yamada *et al.*, 2016; Nguyen *et al.*, 2018; Tamine *et al.*, 2019) ont montré que les approches neuronales ne sont pas suffisantes pour capturer l'ensemble des sémantiques, dont la sémantique relationnelle (p. ex. synonymie, homonymie ou polysémie), perdant ainsi de l'information. Pour améliorer la représentation sémantique des plongements lexicaux, différentes approches ont ainsi été proposées. Celles-ci injectent de nouvelles connaissances dans les modèles d'apprentissage des représentations sous formes de contraintes avec l'idée que les connaissances portées par des ressources externes doivent permettre de pallier les problèmes inhérents à la sémantique relationnelle.

L'exploitation de la sémantique distributionnelle se fait en RI avec l'élaboration de modèle spécifiques à des fins d'ordonnancement de documents, permettant ainsi d'améliorer les modèles de l'état de l'art. Nous identifions deux lignes de travaux, ceux qui exploitent directement les plongements lexicaux dans des modèles classiques de RI (Zuccon *et al.*, 2015; Mitra *et al.*, 2016; Nalisnick *et al.*, 2016), et ceux qui les exploitent via l'apprentissage de modèles d'appariement (Li et Lu, 2016; Onal *et al.*, 2017; Guo *et al.*, 2019) pour différentes applications d'appariement de textes (p. ex. recherche ad-hoc, réponses aux questions, classification de texte). Ces nouvelles approches consistent à apprendre la pertinence de paires document-requête à partir de représentations distribuées, en utilisant des architectures neuronales profondes (c.-à-d. avec plusieurs couches cachées).

Dans ce chapitre, nous commençons par présenter dans la Section 1 les concepts de base des réseaux de neurones et de l'apprentissage profond. Nous y introduisons les définitions clés et détaillons les architectures couramment utilisées en RI. Enfin, nous décrivons le processus d'apprentissage et les problématiques connexes. Ensuite, dans la Section 2, nous présentons les principaux travaux de recherche liés à l'apprentissage de représentations distribuées des textes et des géotextes, ainsi que leurs applications en RI. Nous continuons dans la Section 3 en détaillant les travaux de l'état-de-l'art qui utilisent les réseaux de neurones profonds pour apprendre la pertinence de paires document-requête. Enfin, dans la Section 4, nous discutons de quelques problématiques liées à l'utilisation des plongements lexicaux et des réseaux de neurones dans la RI.

1 Réseaux de neurones et apprentissage profond : concepts préliminaires

Les réseaux de neurones artificiels sont inspirés des systèmes nerveux biologiques qui constituent le cerveau humain et réalisent simplement de nombreuses applications telles que la reconnaissance de formes, le traitement du signal ou la mémorisation. C'est à partir de l'hypothèse que le comportement intelligent humain est le résultat de la structure et des éléments de bases du système nerveux central (que sont les neurones), qu'ont été développés les réseaux de neurones artificiels. Les réseaux de neurones artificiels sont donc un moyen de modéliser le mécanisme d'apprentissage et de traitement de l'information qui se produit dans le cerveau humain.

L'objectif des réseaux de neurones artificiels est d'apprendre un modèle qui permet d'encoder toute les informations du monde réel (p. ex. images, sons, textes) en un vecteur numérique et de les traiter, comme le ferait un réseau de neurones biologique pour répondre à un stimulus. Les réseaux de neurones artificiels peuvent être appliqués aux sujets qui requièrent de l'apprentissage automatique lorsque le problème est d'apprendre un alignement complexe entre l'espace d'entrée et l'espace de sortie. Un réseau de neurones commence, à l'aide de données d'entrée, par apprendre les paramètres du réseau grâce à une fonction objectif qui détermine une erreur d'apprentissage. C'est la phase de propagation avant (ou *feedforward*). Ensuite, le réseau propage cette erreur en arrière pour corriger les paramètres. C'est la phase de rétropropagation du gradient (ou *backpropagation*).

Dans les sections suivantes, nous rappelons les concepts de base des réseaux de neurones (Section 1.1 et Section 1.2) et présentons quelques architectures fondamentales couramment utilisées en RI (Section 1.3). Enfin, nous détaillons l'algorithme d'entraînement (Section 1.4) et revenons sur quelques limites connues des réseaux de neurones (Section 1.5).

1.1 *Concepts préliminaires*

1.1.1 *Neurone formel*

Un réseau de neurones est composé de nœuds de calcul connectés entre eux par des liens dirigés et pondérés. Les nœuds représentent les *neurones* et les liens pondérés sont le *poids* des connexions synaptiques reliant les neurones, aussi appelés *poids synaptiques*. Un neurone est donc un processeur primaire qui permet

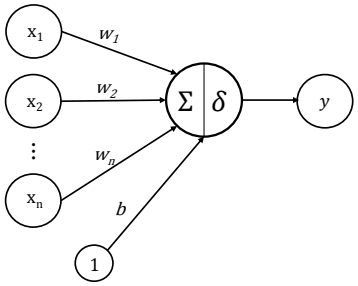


Figure 3.1 – Exemple d'un neurone formel : le perceptron.

de combiner les potentiels des signaux synaptiques qu'il reçoit et de transmettre l'information via une fonction de transfert, de préférence non linéaire.

Le modèle d'un neurone est un modèle mathématique qui reçoit l'information sous la forme d'un ensemble de signaux d'entrées numériques. Ces signaux sont ensuite intégrés, via des fonctions de combinaison et d'activation, avec un ensemble de paramètres libres et entraînables, incluant des poids de connexion w_i et un biais b, pour produire un message sous la forme d'un signal numérique unique. Le neurone formel le plus répandu, introduit par Rosenblatt (1958), est le *perceptron*. Son architecture est présentée en Figure 3.1. Il s'agit d'un neurone formel muni d'une règle d'apprentissage qui permet de déterminer automatiquement les poids synaptiques de manière à séparer un problème d'apprentissage supervisé. Le perceptron se compose de trois parties essentielles, qui transforment des signaux entrants $(x_1, \ldots, x_n)$ en une valeur de sortie y :

— un ensemble de *paramètres libres* θ (Section 1.1.2) qui consiste en un vecteur de poids $(w_1, \ldots, w_n)$ et un biais b ;
— une *fonction de combinaison* Σ (Section 1.1.3) qui combine les signaux entrants avec les paramètres libres pour produire une valeur appelée l'*état interne* ;
— une *fonction d'activation* δ (Section 1.1.4) qui, à partir de l'état interne du neurone, produit une valeur de sortie y.

1.1.2 *Paramètres libres*

À chaque neurone est associé un ensemble de paramètres appelés *paramètres libres*. Ces paramètres permettent au neurone d'être entraîné pour accomplir une tâche. L'ensemble des paramètres libres θ est défini par :

$$\theta = (b, \mathbf{w}) \in \mathbb{R} \times \mathbb{R}^n \tag{3.1}$$

où $\mathbf{w} = (w_1, \ldots, w_n)$ est le vecteur des poids synaptiques associés au vecteur des entrées $\mathbf{x}$ de taille n, et b le biais. Le biais est souvent représenté par un poids sy-

naptique θ_0 relié à une entrée additionnelle imaginaire x_0 fixée à 1. Les paramètres libres sont communément appelés *paramètres du modèle*.

1.1.3 *Fonction de combinaison*

Chaque neurone artificiel comprend une fonction de combinaison Σ utilisée pour calculer l'état interne du neurone. L'objectif de cette fonction est d'agréger l'élément d'entrée $\mathbf{x}$ avec le vecteur des poids synaptiques $\mathbf{w}$. Cette fonction peut être formalisée comme une fonction vecteur-à-scalaire :

$$\Sigma(\mathbf{x};\theta) = \sum_{i=0}^{n} \theta_i x_i = \sum_{i=1}^{n} w_i x_i + b \tag{3.2}$$

1.1.4 *Fonction d'activation*

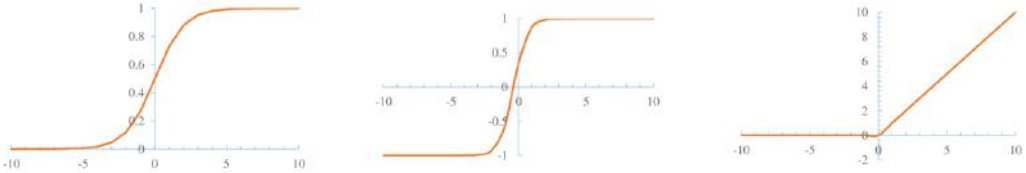
La fonction d'activation, aussi appelée fonction de transfert, notée δ, permet d'introduire une non-linéarité dans le fonctionnement du neurone et de contrôler la propagation de l'information. Elle calcule la sortie y du neurone à partir de la combinaison δ.

$$\begin{aligned} &y = f(\mathbf{x}) \\ &f(\mathbf{x}) = \delta(\Sigma(\mathbf{x},\theta)) = \delta\left(\sum_{i=1}^{n} w_i x_i + b\right) \end{aligned} \tag{3.3}$$

En pratique, plusieurs fonctions d'activation sont utilisées, selon l'objectif de l'application. Chaque fonction d'activation requiert une valeur unique en paramètre et effectue une opération mathématique sur celle-ci pour la faire correspondre dans un nouvel intervalle. Les fonctions les plus utilisées sont la fonction *sigmoïde* (ou *logistique*), la fonction *tangente hyperbolique* (*tanh*) et la fonction *unité de rectification linéaire* (*ReLU*). Leurs équations et leurs courbes correspondantes sont respectivement présentées dans le Tableau 3.1 et la Figure 3.2.

Fonction	**Équation**	**Dérivée**
Sigmoïde	$\delta(x) = \frac{1}{1+e^{-x}}$	$\delta'(x) = \delta(x)(1-\delta(x)$
Tanh	$\delta(x) = \frac{2}{1+e^{-2x}} - 1$	$\delta'(x) = 1 - \delta(x)^2$
ReLU	$\delta(x) = \begin{cases} 0 & \text{pour} \quad x < 0 \\ x & \text{pour} \quad x \geq 0 \end{cases}$	$\delta'(x) = \begin{cases} 0 & \text{pour} \quad x < 0 \\ 1 & \text{pour} \quad x \geq 0 \end{cases}$

Tableau 3.1 – Équations et dérivées des fonctions d'activation courantes.



(a) Sigmoïde (b) Tanh (c) ReLU

Figure 3.2 – Courbes décrivant le comportement des trois des fonctions d'activation courantes.

1.2 *Réseau de neurones artificiels*

Bien qu'un seul neurone puisse apprendre et résoudre quelques problèmes de classification, il ne peut pas modéliser les liens complexes entre les données et résoudre des tâches plus élaborées. La puissance de calcul neuronal provient de la connexion de nombreux neurones, au sein d'une architecture de réseau. Les réseaux de neurones sont constitués d'une succession de graphes hautement connectés, où les nœuds sont des neurones artificiels fonctionnant en parallèle, et les arrêtes sont les connexions du réseau. Comme nous l'avons évoqué dans la section précédente, chaque neurone est un processeur élémentaire qui calcule une valeur de sortie à partir des informations reçues en entrée. Les connexions entre les différents nœuds sont pondérées et constituent le réseau, et peuvent varier d'un modèle à l'autre.

La Figure 3.3 présente un perceptron multicouche (*multilayer perceptron*, MLP), le réseau de neurones le plus répandu. Le perceptron multicouche est un réseau de neurones à propagation avant ou acyclique (*feedforward network*) composé de couches successives orientées. Les signaux circulent de couche en couche, depuis l'entrée, via les couches cachées, vers la couche de sortie uniquement. Étant donné un ensemble de signaux d'entrée, l'information est propagée de couche en couche pour calculer les valeurs de sortie. Une couche est un ensemble de neurones n'ayant pas de connexions entre eux. Il existe des architectures récurrentes où l'entrée d'un neurone peut être sa propre sortie. Ces architectures sont présentées dans la Section 1.3 avec d'autres architectures populaires.

Le perceptron multicouche, dont l'architecture est illustrée dans la Figure 3.3, se compose d'une séquence de c couches cachées notée $(l^{(1)}, \ldots, l^{(c)})$ où $l^{(i)}$ désigne la i^{e} couche cachée, et d'une couche de sortie $l^{(c+1)}$, de telle sorte que les neurones d'une couche ne sont connectés qu'aux neurones de la couche suivante. La couche d'entrée $l^{(0)}$ se compose de n entrées et correspond au signal d'entrée $\mathbf{x}$. Chaque couche cachée $l^{(i)} \in (l^{(1)}, \ldots, l^{(c)})$ contient un ensemble de n_i états intermédiaires cachés notés $u_j^{(i)}$. Un neurone $u_j^{(i)}$ dans la couche $l^{(i)}$ reçoit $n_{(i-1)}$ signaux d'entrée

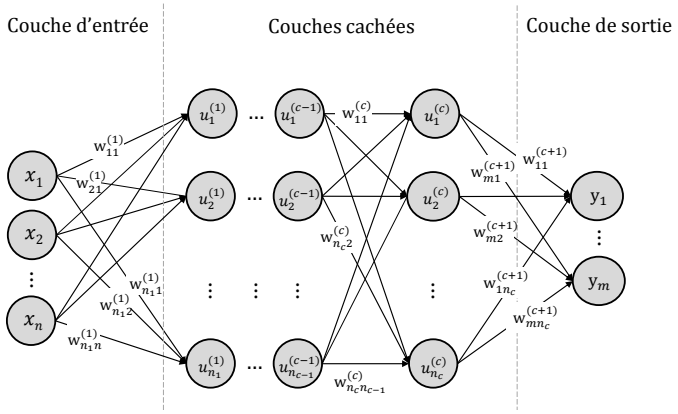


Figure 3.3 – Architecture d'un perceptron multicouche.

venant de tous les neurones de la couche précédente. Chaque connexion entre le neurone $u_j^{(i)}$ et le neurone $u_k^{(i-1)}$ de la couche précédente $l^{(i-1)}$ est pondéré par le poids $w_{jk}^{(i)}$. Ainsi, la valeur d'activation (sortie) du neurone $u_j^{(i)}$ est calculée par :

$$u_j^{(i)} = \delta^{(i)} \left(\sum_{k=1}^{n_{i-1}} w_{jk}^{(i)} u_k^{(i-1)} \right) \quad i = 1, \ldots, c \; ; \; j = 1, \ldots, n_i \tag{3.4}$$

où n_{i-1} est le nombre de neurones de la couche précédente et $\delta^{(i)}$ est une fonction d'activation. Pour la première couche cachée, c.-à-d. pour $i = 1$, $u_k^{(0)}$ représente la k^{e} signal d'entrée. En appliquant le même traitement aux neurones de la dernière couche cachée $l^{(c)}$, nous calculons les valeurs de sortie y_j comme :

$$y_j = u_j^{(c+1)} = \delta^{(c+1)} \left(\sum_{k=1}^{n_c} w_{jk}^{(c+1)} u_k^{(c)} \right) \quad j = 1, \ldots, m \tag{3.5}$$

Notons que la taille de la couche de sortie dépend des objectifs de l'application. Par exemple, si le réseau de neurones est utilisé dans une tâche de classification, la couche de sortie contiendra plusieurs nœuds, chacun se référent au label de la classe correspondante. Dans le cas d'une tâche de régression, la couche de sortie ne contiendra qu'un seul nœud qui correspondra à un seul score.

L'objectif de chaque couche d'un réseau de neurones (excepté la couche d'entrée) est d'approximer une fonction $\mathbf{f}^*$, à partir d'une fonction de transformation f (Équation 3.3) appliquée sur chaque nœud de la couche. Le processus d'entraînement corrige les paramètres libres θ pour une meilleure approximation de la prédiction via plusieurs itérations d'apprentissage. Ainsi, la fonction de transfor-

mation approximative d'une couche d'un perceptron multicouche peut être définie comme le produit matriciel suivant :

$$\begin{aligned} \mathbf{u}_0 &= \mathbf{x} \\ \mathbf{u}_i &= \mathbf{f}^*(\mathbf{u}_{i-1}) = \delta^{(i)}(\mathbf{w}_i \cdot \mathbf{u}_{i-1} + b) \end{aligned} \tag{3.6}$$

où $\mathbf{w}$ et b sont respectivement la matrice des poids synaptiques et le vecteur de biais correspondant à une couche dans le réseau de neurones. De même, la sortie $\mathbf{y}$ du perceptron multicouche peut être formulée comme suit :

$$\mathbf{y} = \delta^{(c+1)}(\mathbf{w}_{c+1} \cdot \mathbf{u}_c + b) \tag{3.7}$$

1.3 *Architectures populaires en recherche d'information*

Différentes architectures de réseaux de neurones ont été proposées pour répondre à différentes tâches. Dans cette section, nous présentons les architectures principalement utilisées en RI.

1.3.1 *Réseau de neurones à convolution (CNN)*

En apprentissage automatique, un réseau de neurones à convolution ou réseau de neurones convolutifs (*Convolutional Neural networks*, CNN) est un type de réseau de neurones artificiels acycliques, dans lequel le motif de connexion entre les neurones est inspiré du cortex visuel des animaux. Les neurones de cette région du cerveau sont arrangés de sorte qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuel. Ainsi, les réseaux de neurones à convolution consistent en un empilage de perceptrons multicouches, dont le but est de prétraiter de petites quantités d'informations. Les premiers réseaux de neurones à convolution ont été introduits à la fin des années 1980 par Denker *et al.* (1988) et LeCun *et al.* (1989) pour la reconnaissance de caractères manuscrits. La Figure 3.4 illustre un exemple de réseau de neurones à convolution, LeNet-5, proposé par LeCun *et al.* (1998). Depuis, ils ont été utilisés dans de nombreuses applications notamment pour la reconnaissance d'image (Lawrence *et al.*, 1997; Ciresan *et al.*, 2012), de vidéo (Ji *et al.*, 2013; Karpathy *et al.*, 2014) ou audio (Dieleman et Schrauwen, 2014; Ravanelli et Bengio, 2018), avec une grande variété d'architectures, plus ou moins profondes. Les réseaux de neurones convolutifs ont également été explorés pour le traitement automatique du langage naturel (TALN) où ils ont obtenu d'excellents résultats dans des tâches telles que l'analyse sémantique (Grefenstette *et al.*, 2014), la modélisation de phrases (Kalchbrenner *et al.*, 2014) ou la classification de phrases (Kim, 2014), et plus récemment pour résoudre des tâches orientées RI, comme nous le verrons dans la Section 3.

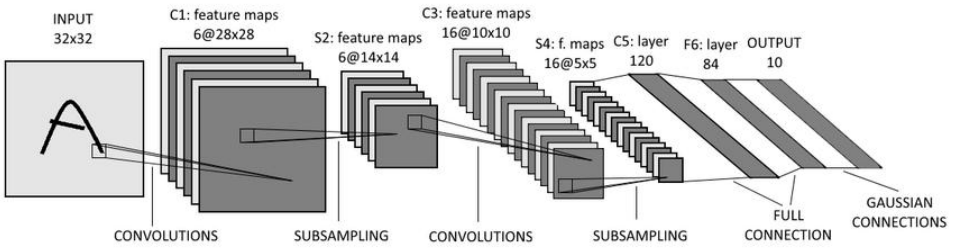


Figure 3.4 – Exemple de réseau de neurones à convolution LeNet-5 pour la reconnaissance de caractères manuscrits (LeCun *et al.*, 1998).

La popularité des réseaux de neurones à convolution est due à plusieurs facteurs, notamment leur aptitude à traiter des signaux de tailles variables mais aussi leur architecture hiérarchique qui permet un traitement des données à plusieurs niveaux. Les principales caractéristiques d'un réseau de neurones à convolution sont les suivantes :

— **superposition des neurones** : les couches d'un CNN peuvent être en trois dimensions : *largeur*, *hauteur* et *profondeur*, où chaque neurone est relié à seulement une petite région de la couche qui le précède. Cette région est appelée *champ récepteur* ;
— **connectivité locale** : grâce au champ récepteur qui limite le nombre d'entrées du neurone, les CNNs assurent que les *filtres* produisent la réponse la plus forte à un motif d'entrée spatialement localisé, ce qui conduit à une représentation parcimonieuse de l'entrée. Une telle représentation réduit le nombre de paramètres à estimer, permettant ainsi une estimation plus robuste ;
— **poids partagés** : dans les CNNs, les paramètres de filtrage d'un neurone sont identiques pour tous les autres neurones d'un même noyau. Ce paramétrage est défini dans une *carte de caractéristique* ;
— **invariance à la translation** : comme tous les neurones d'un même noyau (filtre) sont identiques, le motif détecté par ce noyau est indépendant de localisation spatiale dans l'image.

L'architecture d'un réseau de neurones à convolution est formée par un empilement de couches de traitement comprenant généralement des couches de convolution et des couches de *pooling* :

1. **couche de convolution** : la couche de convolution permet de traiter les données d'entrée. Elle consiste à multiplier une matrice par une autre matrice, appelée **filtre de convolution**. Le filtre de convolution parcourt toute la matrice d'entrée de manière incrémentale et génère une nouvelle matrice, appelée **carte de caractéristiques** (ou ***feature map***), constituée des résultats du produit de convolution. Celui-ci agit donc comme un extracteur de carac-

téristiques. Appliqués au texte, les filtres de convolution reviennent ainsi à considérer des n-grammes de mots ;

2. **couche de *pooling* (ou mise en commun)** : la couche de *pooling* permet de sous-échantillonner la sortie de l'opération de convolution et de ne garder que les caractéristiques les plus importantes. La méthode la plus utilisée est le *max pooling*. Elle consiste à réduire la dimension de la matrice en ne conservant que les valeurs les plus grandes.

La forme la plus commune d'une architecture de réseau de neurones à convolution empile quelques couches de convolutions suivies d'une couche de *pooling*, et répète ce schéma jusqu'à ce que l'entrée soit réduite dans un espace d'une taille suffisamment petite. Après quelques convolutions, il est fréquent de placer des couches entièrement connectées de type perceptron qui permettent le raisonnement de haut niveau.

1.3.2 *Réseau de neurones récurrents (RNN)*

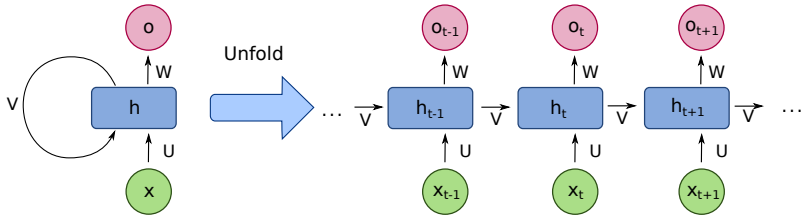


Figure 3.5 – Schéma d'un réseau de neurones récurrents à une unité reliant l'entrée et la sortie du réseau. (©*Wikimedia*)

Les réseaux de neurones récurrents (RNN) sont issus des travaux de Rumelhart *et al.* (1986). Ce sont des réseaux de neurones dans lesquels l'information peut se propager dans les deux sens, y compris des couches profondes aux premières couches. La Figure 3.5 illustre un exemple de réseau de neurones récurrents à une unité (ou neurone) reliant l'entrée et la sortie du réseau. Dépliés, RNN sont comparables à des réseaux de neurones classiques organisés en couches successives. Chaque neurone d'une couche donné est relié par une connexion dirigée à tous les autres neurones de la couche successive suivante. Chaque neurone possède une activation à valeur réelle qui varie dans le temps. Chaque synapse possède un poids modifiable. Les neurones sont soit *(1)* des neurones d'entrée, qui reçoivent des données extérieures au réseau ; *(2)* des neurones de sortie, qui donnent des résultats ; *(3)* des neurones cachés, qui modifient les données en cours de route de l'entrée à la sortie. Les RNN sont particulièrement adaptés pour modéliser des données d'entrée de taille variable de différentes applications, telles que la

reconnaissance de la parole (Graves *et al.*, 2013; Anumanchipalli *et al.*, 2019) et le traitement de texte avec le résumé automatique ou la traduction automatique (Kalchbrenner et Blunsom, 2013; Nallapati *et al.*, 2017).

Les techniques d'entraînement des réseaux de neurones récurrents sont les mêmes que pour les réseaux acycliques (Section 1.4.2), néanmoins ils se heurtent au problème de disparition du gradient pour apprendre à mémoriser des évènements passés (Hochreiter *et al.*, 2001). Des architectures particulières, tels que les réseaux récurrent à mémoire court-terme et long-terme (*Long Short-Term Memory*, LSTM) et leurs variantes comme l'architecture *Gated Reccurrent Unit* (GRU) (Cho *et al.*, 2014) répondent à ce problème (Hochreiter et Schmidhuber, 1997).

1.3.3 *Transformer*

Le *transformer* est un modèle d'apprentissage profond introduit par Vaswani *et al.* (2017). Un *transformer* est conçu de la même manière qu'un réseau de neurones récurrents reposant uniquement sur un mécanisme d'attention (Bahdanau *et al.*, 2015) et de simple réseaux de neurones acycliques, permettant de traiter des séquences de textes ordonnées. Cependant, contrairement aux réseaux de neurones récurrents, les *transformers* n'exigent pas que la séquence soit traitée dans l'ordre. Ainsi, si les données d'entrée sont du langage naturel, le *transformer* n'a pas besoin de traiter le début d'une phrase avant de traiter la fin. Grâce à cette caractéristique, les *transformers* permettent une parallélisation beaucoup plus importante que les RNN lors la phase d'apprentissage.

Le *transformer* se compose de deux éléments principaux : un ensemble de codeurs (*encoders*) reliés les uns aux autres, et un ensemble de décodeurs (*decoders*) reliés les uns aux autres. La fonction de chaque codeur est de traiter ses vecteurs d'entrée pour générer des *codages*, qui contiennent des informations sur les parties des entrées qui sont pertinentes les unes par rapport aux autres. Chaque décodeur fait le contraire, en récupérant et traitant tous les codages en utilisant les informations contextuelles qu'ils contiennent, pour générer une séquence de sortie. Pour réaliser ces opérations, chaque codeur et décodeur utilise un mécanisme d'attention qui, pour chaque entrée, évalue la pertinence de toutes les entrées et en extrait des informations lors de la génération de la sortie. Chaque décodeur dispose également d'un mécanisme d'attention supplémentaire qui tire des informations des sorties des décodeurs précédents, avant que le décodeur ne tire des informations des codages. Les encodeurs et les décodeurs disposent tous deux d'un réseau de neurones acyclique final pour le traitement supplémentaire des sorties. L'architecture générale du *transformer* est présentée dans la Figure 3.6.

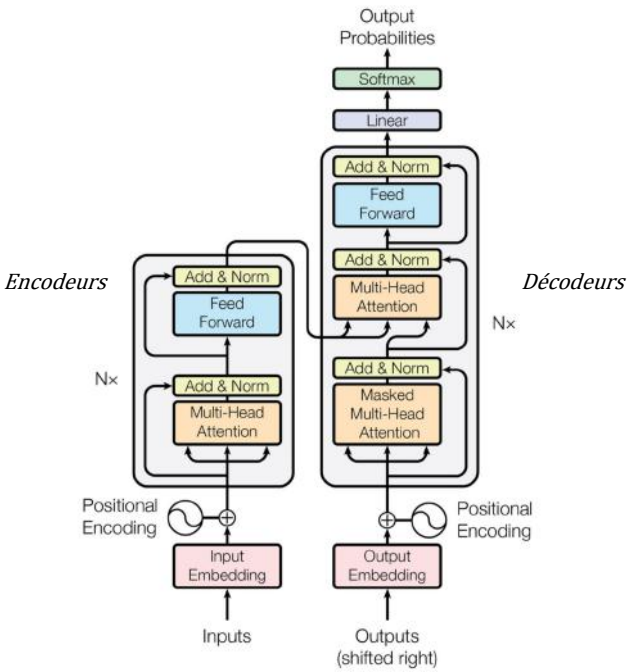


Figure 3.6 – Architecture générale d'un *transformer* (Vaswani *et al.*, 2017).

Depuis leur introduction, les *transformers* sont devenus les éléments de base de la plupart des architectures neuronales de l'état-de-l'art pour le TALN et la RI (Radford *et al.*, 2019; Adiwardana *et al.*, 2020), remplaçant peu à peu les réseaux neuronaux récurrents à portes tels que les LSTM ou les GRU. Comme l'architecture des *transformers* facilite une plus grande parallélisation lors de la phase d'apprentissage, il est aujourd'hui possible d'apprendre les paramètres des modèles sur un volume conséquent de données. Cela a conduit au développement de systèmes préformés tels que le modèle *Bidirectional Encoder Representations from Transformers*, ou plus simplement BERT (Devlin *et al.*, 2019), qui a été entraîné avec d'énormes quantités de données avant d'être publié et réutilisé dans de nombreuses tâches de RI et de TALN (Radford *et al.*, 2019; Zhang *et al.*, 2019).

1.4 *Algorithmes d'apprentissage des modèles neuronaux*

Comme n'importe quel algorithme d'apprentissage automatique, les réseaux de neurones doivent être entraînés pour répondre efficacement à la problématique posée. Pour cela, nous faisons appel à une fonction de coût qui permettra de calculer les paramètres libres optimaux du réseau. Selon les objectifs de l'application, l'apprentissage d'un réseau de neurones peut être *supervisé*, quand les étiquettes des données d'apprentissage sont connues et étiquetées ; *semi-supervisé* lorsque les

données d'apprentissage sont peu étiquetées, ou les étiquettes imprécises ou bruitées ; ou *non supervisé* lorsque les données ne sont pas étiquetées. Dans le cadre de nos travaux, nous nous intéressons principalement à l'apprentissage supervisé. La procédure utilisée pour effectuer le processus d'apprentissage d'un réseau de neurones s'appelle l'algorithme d'entraînement, ou algorithme d'apprentissage. Les étapes d'optimisation finissent par aboutir à une configuration stable au sein du réseau de neurones. L'algorithme le plus répandu pour entraîner un réseau de neurones est la rétropropagation du gradient via une descente de gradient.

Dans les sections suivantes, nous détaillons le rôle de la fonction de coût (Section 1.4.1), ainsi que l'algorithme de rétropropagation du gradient (Section 1.4.2) pour l'entraînement des réseaux de neurones.

1.4.1 *Fonction de coût*

Afin d'entraîner un réseau de neurones, c.-à-d. déterminer les valeurs des paramètres libres, une *fonction de coût*, parfois appelée *fonction objectif* ou *fonction de perte* (*loss function*), est définie pour mesurer l'erreur d'apprentissage à l'égard des résultats. À partir de cette erreur, le réseau de neurones s'entraîne en mettant à jour ses paramètres libres dans l'objectif de minimiser l'erreur. Pour cela, des mises à jour de poids sont effectuées continuellement afin de réduire l'erreur d'apprentissage, jusqu'à ce qu'un modèle suffisamment performant soit trouvé, c.-à-d. avec un taux d'erreur minimum, ou qu'un nombre prédéfini d'itérations d'entraînement soit atteint. Des fonctions de coût différentes donneront donc des erreurs d'apprentissage différentes pour la même prédiction et auront donc un effet non-négligeable sur la performance du modèle. Différentes fonctions de coût sont utilisées pour traiter différents types de tâches de régression ou de classification. Les plus communes sont présentées dans le Tableau 3.2.

Type d'application	**Fonction**	**Equation**
Régression	Erreur quadratique (*square error*)	$(y-\widehat{y})^2$
	Erreur absolue (*absolute error*)	$\lvert y-\widehat{y}\rvert$
Classification	Perte au carré (*square loss*)	$(1-y\widehat{y})^2$
	Marge Maximale (*Hinge loss*)	$\max\{1-y\widehat{y},0\}$
	Erreur logistique (*logistic loss*)	$\frac{1}{\ln 2}(1+e^{-y\widehat{y}})$
	Entropie croisée (*cross entropy*)	$-y\ln(\widehat{y})-(1-y)\ln(1-\widehat{y})$

Tableau 3.2 – Liste des fonctions objectifs courantes.

Dans le cas d'un apprentissage supervisé, l'erreur est mesurée par la différence entre les valeurs observées des données (étiquettes) et les valeurs calculées par

le modèle. En pratique, les erreurs d'apprentissage sont estimées pour tous les échantillons de l'ensemble d'apprentissage, et l'erreur d'apprentissage considérée est appelée *erreur moyenne*. Cette valeur est calculée en moyennant les différentes erreurs observées au travers les échantillons du jeu de données.

1.4.2 *Rétropropagation du gradient*

L'algorithme utilisé pour effectuer l'apprentissage des paramètres d'un réseau de neurones s'appelle l'algorithme d'entraînement (ou algorithme d'apprentissage). Son objectif est de minimiser une fonction de coût $\mathcal{L}$ en corrigeant les paramètres libres θ du réseau, à partir d'exemples d'apprentissage.

1.4.2.1 *Descente de gradient*

La descente de gradient (*gradient backpropagation*) est l'un des algorithmes les plus populaires pour l'optimisation des réseaux de neurones selon la méthode de rétropropagation. La descente de gradient est un moyen de minimiser la fonction de coût $\mathcal{L}(\theta)$ en mettant à jour les paramètres dans le sens inverse du gradient $\nabla\mathcal{L}$ par rapport aux paramètres libres θ. Le taux d'apprentissage α détermine la grandeur des pas à réaliser pour atteindre une valeur d'erreur minimum locale ou globale. Ainsi, la valeur des paramètres θ à l'itération t est calculée par :

$$\theta_t = \theta_{t-1} - \alpha \cdot \nabla\mathcal{L}(\theta) \tag{3.8}$$

Formellement, soit les poids $\mathbf{w}$ préalablement initialisés avec des valeurs aléatoires et un ensemble de données d'apprentissage $\mathbf{x}$. Chaque échantillon possède ses valeurs cibles (c.-à-d. étiquettes) qui sont celles que le réseau de neurones doit prédire lorsqu'on lui présente le même échantillon. L'algorithme général de la rétropropagation du gradient, dans le cadre d'un apprentissage supervisé, suit les étapes suivantes :

1. **sélection** (*input*) : soit un échantillon x que l'on présente à l'entrée du réseau de neurones et y la sortie recherchée ;
2. **propagation** (*output*) : le signal est propagé en avant dans les couches du réseau tel que $x_j^{(n)} = \delta^{(n)}(h_j^{(n)}) = \delta^{(n)}(\sum_k w_{jk}^{(n)} x_k^{(n-1)})$. Lorsque la propagation vers l'avant est terminée, nous obtenons la sortie $\widehat{y}$;
3. **calcul de l'erreur** (*loss*) : l'erreur entre la sortie du réseau $\widehat{y}$ et la sortie attendue y est calculée selon une fonction de coût $\mathcal{L}$: $e_i^{sortie} = \mathcal{L}(y, \widehat{y}) = \delta'^{(n-1)}(h_i^{sortie})(y_i - \widehat{y}_i)$;
4. **rétropropagation de l'erreur** (*backpropagation*) : l'erreur est propagée vers l'arrière : $e_j^{(n-1)} = g'^{(n-1)}(h_j^{(n-1)}) \sum_i w_{ij}^{(n)} e_i^{(n)}$;

5. **mise à jour** (*update*) : les poids du réseau sont mis à jour dans toutes les couches : $w_{ij}^{(l)} = w_{ij}^{(l)} - \alpha e_i^{(l)} x_j^{(l-1)}$ où α représente le taux d'apprentissage.

La réalisation complète de cette procédure sur un échantillon (ou un ensemble d'échantillons), c.-à-d. sélectionner un échantillon, propager les valeurs, calculer et rétropropager l'erreur puis mettre à jour les paramètres, est appelée une *itération*. Une période pour exécuter cet algorithme sur tous les échantillons du jeu de données d'apprentissage est appelée une *époque*. Le processus d'apprentissage complet peut ainsi contenir plusieurs époques.

1.4.2.2 *Variantes de la descente de gradient*

Il existe trois variantes de la descente du gradient, qui diffèrent par la quantité d'échantillons utilisés pour calculer le gradient de la fonction de coût $\mathcal{L}$. En fonction de cette quantité, il y a un compromis entre la qualité de la mise à jour des paramètres libres et le temps nécessaire pour effectuer cette mise à jour. Les trois variantes sont les suivantes :

— **descente de gradient par lots (GD)** : le gradient de la fonction de coût est calculé sur l'ensemble des échantillons pour effectuer une seule mise à jour par époque. La descente de gradient par lots peut être très lente, voire impossible pour des paquets de données qui ne tiennent pas en mémoire. De plus, elle ne permet pas non plus de mettre à jour le modèle « en ligne », c'est à dire avec de nouveaux échantillons ;
— **descente de gradient stochastique (SGD)** : contrairement à la variante GD, la descente de gradient stochastique ne calcule le gradient que d'un seul échantillon, tiré aléatoirement, à chaque itération. Ainsi, à chaque itération, les paramètres du modèle sont mis à jour. Cette variante est généralement beaucoup plus rapide et peut être utilisée pour un apprentissage en ligne ;
— **descente de gradient par mini-lots (MGD)** : combinaison des deux approches précédentes, la descente du gradient par mini-lots met à jour les paramètres du modèle pour des sous-ensembles (ou mini-lots) de données. Autrement dit, une itération est réalisée sur un mini-lot d'échantillons de taille k. La MGD permet de réduire la variance des mises à jour des paramètres (par rapport au SGD), conduisant à une convergence plus stable.

1.4.2.3 *Optimisation de la descente de gradient*

Certains travaux (Ruder, 2016) ont montré que la descente de gradient ne garantit pas une meilleure convergence, et pose ainsi quelques défis à relever :
— le choix du taux d'apprentissage est primordial : un taux trop élevé peut empêcher la fonction de coût de converger vers un minimum ou même diverger, tandis qu'un taux trop faible conduit à une convergence extrêmement lente ;

— utiliser les données d'apprentissage dans le même ordre pour toutes les époques peut biaiser l'algorithme d'apprentissage. Il est recommandé de les mélanger aléatoirement après chaque époque ;
— lors de la minimisation de fonctions de coût non-convexes, une grande difficulté est d'éviter les minimums locaux sous-optimaux. Dauphin *et al.* (2014) ont mis en évidence la problématique des points-selles, c.-à-d. des points où une dimension est inclinée vers le haut et une autre vers le bas.

Pour relever ces défis, plusieurs algorithmes d'optimisation ont été proposés et sont aujourd'hui largement adoptés. Les méthodes les plus répandues sont AdaGrad (Duchi *et al.*, 2011), AdaDelta (Zeiler, 2012) et Adam (Kingma et Ba, 2015).

1.5 *Surapprentissage et régularisation*

En apprentissage automatique, le surapprentissage (ou sur-ajustement) est généralement provoqué par un mauvais dimensionnement de l'architecture utilisée pour classifier ou faire une régression. De par sa trop grande capacité à capturer des informations, une structure dans une situation de surapprentissage n'arrivera pas à généraliser les caractéristiques des données. Elle se comporte alors comme une table contenant tous les échantillons utilisés lors de l'apprentissage (données d'apprentissage) et perd ses pouvoirs de prédiction sur de nouveaux échantillons (données de validation).

Pendant l'étape d'apprentissage du réseau de neurones, les valeurs poids augmentent en taille afin de modéliser les spécificités des données d'entraînement. Les poids importants ont tendance à provoquer des transitions brusques dans les fonctions des nœuds (transformation et activation) et donc de grands changements dans la sortie pour de petits changements dans les entrées (Reed et Marks, 1998). Autrement dit, avec des poids importants, le réseau devient instable. Pour éviter les situations de surapprentissage et les mauvaises performances lors de la phase de prédiction, il convient de modifier l'algorithme d'apprentissage afin d'encourager le réseau à maintenir des poids faibles, et à pénaliser les poids élevés. Ce processus est appelé la régularisation des poids. Traditionnellement, la régularisation est effectuée en ajoutant un terme additionnel à la fonction de coût de l'algorithme d'apprentissage. Les deux approches principalement utilisées sont les régularisations $L1$ et $L2$ (Ng, 2004).

D'autres techniques de régularisation ont été proposées dans la littérature, telles que l'arrêt anticipé (*early stopping*), l'abandon (*dropout*), la normalisation des lots (*batch normalization*) ou la dégradation des pondérations (*weight decay*) (Bishop *et al.*, 1995). Dans ce qui suit, nous décrivons les trois méthodes de régularisations largement utilisées dans différentes applications d'apprentissage automatique :

— **arrêt anticipé** (Yao *et al.*, 2007) : lors de l'utilisation de cette méthode, les jeux de données sont divisés en trois sous-ensembles : entraînement, validation et test. L'erreur d'apprentissage sur l'ensemble de validation est surveillée pendant le processus d'apprentissage. Lorsque l'erreur augmente pour un nombre spécifique d'itérations consécutives, l'apprentissage est arrêté et les poids correspondant à l'erreur minimum sont renvoyés ;
— **abandon** (Hinton *et al.*, 2012) : cette méthode consiste à omettre aléatoirement, avec une probabilité définie, une partie des détecteurs de caractéristiques (nœuds) sur chaque donnée d'apprentissage. L'abandon vise à éviter les co-adaptations complexes des différents neurones sur les données d'apprentissage. Hinton *et al.* (2012) ont réalisé une étude empirique évaluant plusieurs taux d'abandon dans différentes couches d'un réseau de neurones pour la classification des images, et a montré qu'une valeur d'abandon entre $0,2$ et $0,5$ permet de réduire fortement les erreurs de classification par rapport aux différentes méthodes existantes ;
— **normalisation des lots** (Ioffe et Szegedy, 2015) : la normalisation des données est directement intégrée à l'architecture du modèle. Elle s'effectue pour chaque mini-lots d'entraînement. Son objectif est d'améliorer l'apprentissage et de réduire l'impact des changements de distribution des fonctions activations du réseau. Ainsi, la fonction de coût converge plus rapidement.

2 Représentations distribuées de textes et de géotextes

Depuis leur introduction par Salton *et al.* (1975) dans les années 1970, les modèles vectoriels ont largement été utilisés en RI. Cependant, de nombreuses lacunes ont été pointées lors de l'utilisation de la représentation classique en sac de mots (*bag of words*) dans différentes tâches d'appariement de textes, notamment la grande dimension des vecteurs, les représentations très éparses et l'inadéquation du vocabulaire (Wallach, 2006; Kao et Poteet, 2007; Croft *et al.*, 2009). Ces limites ont encouragé la recherche scientifique sur le développement de représentations denses capables de saisir la sémantique d'un texte ainsi que les informations contextuelles (Yu et Dredze, 2014; Iacobacci *et al.*, 2015; Nguyen *et al.*, 2018).

Depuis, les modèles vectoriels ont commencé à être utilisé pour représenter la sémantique distributionnelle (Rieger, 1991). Devant les résultats prometteurs, différentes approches ont été explorées pour estimer les représentations continues des mots afin de contourner les limites des sacs de mots. Ces premières approches s'appuient sur le statistiques des cooccurrences des mots au travers des matrices

mot-contexte. Les premiers travaux de recherche pour produire des plongements lexicaux par comptage sont ceux de Deerwester *et al.* (1990), avec le modèle *Latent Semantic Analysis* (LSA) qui applique une décomposition en valeurs singulières sur la matrice de cooccurrence terme-document. D'autres travaux, tels que *Hyperspace Analog to Language* (HLA) (Lund et Burgess, 1996), *Correlated Occurrence Analogue to Lexical Semantics* (COALS) (Rohde *et al.*, 2006) ou *Hellinger PCA* (HPCA) (Lebret et Collobert, 2014) ont suivi, en s'appuyant sur des matrices de cooccurrence mot à mot. Pour calculer ces représentations, d'autres approches utilisent des réseaux de neurones, ce sont les modèles de langues neuronaux. Ces approches, considérées comme une variété de modèles fondés sur la sémantique distributionnelle, ont montré leur efficacité sur les tâches d'analogie des mots et de relations sémantiques par rapport aux modèles traditionnels s'appuyant sur les statistiques (Baroni *et al.*, 2014). Bengio *et al.* (2003) furent les premiers à proposer un modèle de langue neuronal en introduisant l'idée d'apprendre simultanément un modèle de langue qui prédit un mot en fonction de son contexte, ainsi que sa représentation. Cette représentation est appelée plongement lexical, représentation distribuée ou *word embedding*. Depuis, cette idée a été adoptée par de nombreuses études. Les modèles de représentations les plus connus, *Word2Vec* (Mikolov *et al.*, 2013a,b), *GloVe* (Pennington *et al.*, 2014), ont largement été utilisés dans divers domaines de recherche, et notamment en RI et TALN. Le succès des plongements lexicaux a également donné lieu à des travaux sur l'apprentissage de représentations pour de plus grandes unités textuelles, comme les paragraphes et les documents (Le et Mikolov, 2014), ou plus récemment, pour l'apprentissage de représentations d'objets, tels que des évènements (Hong *et al.*, 2017) ou des POIs (Feng *et al.*, 2017; Yan *et al.*, 2017).

Dans les sections suivantes, nous détaillons les principaux travaux liés à l'apprentissage de représentations distribuées de textes ainsi que leurs différents niveaux de granularité (Section 2.1). Ces travaux sont présentés en deux catégories, à savoir l'apprentissage de représentations depuis les textes d'un corpus, et l'apprentissage de représentations combinant la sémantique distributionnelle venant d'un corpus et la sémantique relationnelle recensée dans les ressources sémantiques. Enfin, nous abordons dans la Section 2.2 les travaux plus récents pour la représentation de géotextes.

2.1 *Représentations distribuées de textes*

2.1.1 *Représentations distribuées des mots*

Les premiers modèles de langue neuronaux n'avaient pas pour objectif premier d'apprendre la représentation distribuée des mots. Cependant, les expérimenta-

tions ont démontré que la couche composante des représentations, qui aborde le problème de dimensionnalité des vecteurs de termes en entrée, fournit des représentations distribuées efficaces pour la RI. Ces approches, s'appuyant sur le contexte local, consistent à apprendre les représentations des mots à partir d'une fenêtre d'occurrence, appelée fenêtre contextuelle. Le modèle de langue neuronal, *Neural Network Language Model* (NNLM), proposé par Bengio *et al.* (2003) est à l'origine des méthodes contextuelles pour l'apprentissage de représentations de mots. Il s'agit d'un modèle de langue probabiliste dans lequel les probabilités des mots sont calculées à l'aide d'une architecture neuronale. Le modèle NNLM apprend simultanément les représentations distribuées (ou plongements lexicaux) des mots d'entrée, et la fonction de probabilité pour les fenêtres contextuelles correspondantes.

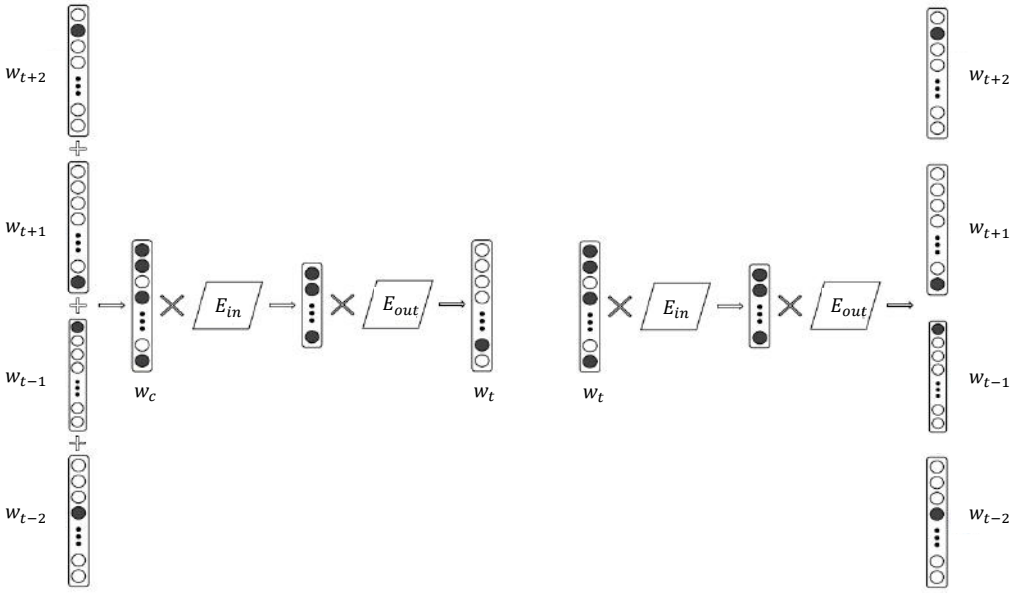
Pour rappel, un modèle de langue traditionnel calcule la probabilité d'obtenir un ensemble de mots $P(w_1, \ldots, w_m)$. Les modèles de langues probabilistes approximent quant à eux la probabilité $P(w_t|w_1, w_{(t-1)})$ en considérant seulement un contexte réduit de taille n mots qui précède w_t. Dans les modèles de langue neuronaux, la probabilité $P(w_t|c)$ d'un mot $w_t \in \mathcal{V}$ qui suit le contexte $c =< w_1, \ldots, w_{|c|} >$, c.-à-d. une séquence de mots qui précède le mot w_t, est calculée par un réseau de neurones. Autrement dit, le réseau de neurones calcule la probabilité conditionnelle $P(w_t|c)$ d'un mot w_t à partir d'un contexte c. Le modèle de langue neuronal est entraîné pour optimiser une fonction de coût $\mathcal{L}$ pour tous les mots w_t dans le texte T d'un corpus et de leurs fenêtres contextuelles correspondantes, comme définie dans l'Équation 3.9.

$$\mathcal{L}(\theta) = \sum_{(w_t,c)\in T} \log P(w_t|c;\theta) \tag{3.9}$$

où θ représente les paramètres libres du réseau de neurones utilisés pour calculer la probabilité P.

Les résultats obtenus par le modèle NNLM ont poussé Collobert *et al.* (2011) à utiliser une fenêtre contextuelle symétrique autour du mot, afin d'apprendre son vecteur de représentation, plutôt que de se contenter de prédire les mots contextuels précédents. Dans leur modèle, Collobert *et al.* (2011) utilisent une fenêtre contextuelle symétrique $c =< w_{t-n+1}, \ldots, w_t, \ldots w_{t+n-1} >$. Ainsi, le contexte d'un mot w_t fait référence aux n mots qui le précèdent et n qui le suivent.

Devant le succès des modèles de langues neuronaux, Mikolov *et al.* (2013a,b) ont proposé un modèle, appelé *Word2Vec*, pour le calcul des représentations distribuées des mots. Plus particulièrement, deux configurations ont été adoptées, le *Continuous Bag-of-Words* (CBOW) et le *Skip-Gram*, qui suivent tous deux l'architecture du modèle NNLM. Néanmoins, Mikolov *et al.* (2013a) ont adapté plusieurs méthodes pour améliorer l'efficacité de l'apprentissage ainsi que la qualité des re-



(a) CBOW (b) Skip-Gram

Figure 3.7 – Architecture des configurations du *Word2Vec* : (a) CBOW et (b) Skip-Gram (Mikolov *et al.*, 2013a,b).

présentations des mots. L'architecture des deux configurations est illustrée dans la Figure 3.7. Le modèle CBOW (Figure 3.7a) se rapproche du modèle de Collobert *et al.* (2011), et est entraîné pour prédire un mot w_t en tenant compte des mots dans son contexte symétrique $c =< w_{t-n+1}, \ldots, w_t, \ldots w_{t+n-1} >$. Les mots du contexte c sont d'abord agrégés via une somme ou une moyenne avant d'être envoyés dans la couche cachée du réseau de neurones. A contrario, le modèle *Skip-Gram* (Figure 3.7b) est entraîné pour prédire les mots du contexte symétrique c, à partir d'un mot central w_t. Dans ces deux configurations, pour chaque paire (w_t, c), le score neuronal $s_\theta(w_t, c)$ est calculé par le produit entre la représentation d'entrée du contexte c et la représentation de sortie du mot à prédire w_t :

$$s_\theta(w_t, c) = e_{out}(w_t) \cdot e_{in}(c) = (\overrightarrow{w_t} \cdot E_{out}) \cdot (\overrightarrow{c} \cdot E_{in}) \tag{3.10}$$

où E_{in}, $E_{out} \in \mathbb{R}^{|\mathcal{V}| \times dim}$ sont les poids synaptiques associés à la couche cachée de taille dim, et qui constituent l'espace de représentation des plongements lexicaux. $\overrightarrow{c}$ et $\overrightarrow{w_t}$ sont respectivement les vecteurs de représentation du contexte et du mot courant, avec $\{\overrightarrow{c}, \overrightarrow{w_t}\} \in \mathbb{R}^{|\mathcal{V}|}$. Dans la configuration CBOW, w_t est le mot central à prédire et c son contexte de k mots avant et k mots après. Les vecteurs associés aux représentations distribuées des mots du contexte sont agrégés en un seul vecteur $\overrightarrow{c} = \sum_{w \in c} \overrightarrow{w}$ envoyé dans le réseau de neurones. Dans le cas du modèle *Skip-Gram*, l'objectif est de prédire un mot w_j qui appartient au contexte c du mot central w_t. Ainsi, pour chaque mot central w_t, le réseau de neurones itère $2 \times k$

fois avec la même entrée (c.-à-d. le mot central w_t) pour prédire tous les mots $w_j \in \{w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+4}\}$ de la fenêtre contextuelle. Chaque mot w_j prédit devrait être du même contexte sémantique que les mots prédits précédemment. L'entraînement des modèles CBOW et Skip-Gram est plus efficace que celle du modèle de langue neuronale (Bengio *et al.*, 2003) grâce à l'utilisation d'une fonction exponentielle normalisée hiérarchique (*hiearchical softmax*) qui repose sur une architecture arborescente binaire (*Huffman Tree*) (Mikolov *et al.*, 2013a) ou d'un échantillonnage négatif (*negative samplig*) qui donne de meilleures performances (Mikolov *et al.*, 2013b).

Confortés par les performances des représentations de mots entraînées à partir du contexte pour accomplir des tâche de RI et de TALN, certains auteurs se sont intéressés à l'adaptation du modèle *Word2Vec* pour résoudre d'autres tâches. Nous retrouvons notamment des modèles de représentation multilingue (Vulic et Moens, 2015; Coulmance *et al.*, 2015), le modèle *Dual Embedding Space Model* (Mitra *et al.*, 2016) qui utilise les deux espaces de représentations distribuées (l'équivalent de E_{in} et E_{out} sur la Figure 3.7) pour l'ordonnancement de documents ou encore l'approche proposée par Zamani et Croft (2017), qui consiste à apprendre des représentations de mots à partir des informations de pertinence des documents. Enfin, parmi les modèles les plus récents pour la représentation des mots en tenant compte du contexte, nous pouvons citer les articles de Bojanowski *et al.* (2017) et Joulin *et al.* (2017) comme source de l'application *FastText*. Ils ont suggéré d'améliorer le modèle *Skip-Gram* non pas en calculant des représentations de mots, mais des représentations distribuées de n-grammes, qui peuvent-être composés pour former des mots. Cette hypothèse se justifie par le fait que les langues qui dépendent fortement de la morphologie et de la composition des mots ont des informations encodées dans les parties de mots elles-mêmes, qui peuvent être utilisées pour aider à généraliser des mots nouveaux.

D'autres approches ont été proposées pour construire des plongements lexicaux, non pas en entraînant un algorithme qui prédit un mot étant donné un contexte, mais en tirant parti des matrices de cooccurrences comptées globalement dans le corpus. Ce sont les modèles fondés sur le comptage au travers des matrices mot-contexte, tels que LSA (Deerwester *et al.*, 1990), HLA (Lund et Burgess, 1996), COALS (Rohde *et al.*, 2006) ou HPCA (Lebret et Collobert, 2014). Toutefois, ces différentes approches ont largement été dépassées par les modèles de langue neuronaux de Mikolov *et al.* (2013a) (et dérivés). Plus récemment, Pennington *et al.* (2014) ont proposé les représentations *GloVe*, des vecteurs de représentations globales. Les vecteurs sont entraînés pour s'adapter à la matrice de cooccurrence glo-

bale en combinant le contexte global et le contexte local lors de l'apprentissage des représentations des mots. Ce processus est mis en avant par l'équation suivante :

$$\mathcal{L} = \sum_{i,j}^{\mathcal{V}} f(x_{ij})(e(w_i) \cdot e(w_j) + b_i + b_j - \log x_{ij})^2 \tag{3.11}$$

où x_{ij}, $e(w_i)$ et $e(w_j)$ sont respectivement le nombre de cooccurrences des mots w_i et w_j, et les représentations distribuées des mots w_i et w_j. b_i et b_j sont les valeurs de biais associées aux représentations $e(w_i)$ et $e(w_j)$. $\mathcal{V}$ est le vocabulaire du corpus et $f(x)$ est une fonction de poids définie par :

$$f(x) = \begin{cases} (x/x_{\max})^\alpha & \text{si} \quad x \leq x_{\max} \\ 1 & \text{sinon} \end{cases} \tag{3.12}$$

2.1.2 *Représentations distribuées des phrases*

Devant les performances des représentations distribuées des mots dans de nombreuses tâches de RI et de TALN, certains auteurs ont proposé de construire des représentations distribuées pour des unités textuelles plus longues comme les phrases, les paragraphes ou les documents entiers. L'approche la plus simple, est d'utiliser la somme ou la moyenne des représentations des mots composants les textes (Weston *et al.*, 2014; Yin *et al.*, 2015). Cependant, une telle agrégation ne tient pas compte de l'ordre des mots, et la simple moyenne traite tous les mots avec la même importance, même si certains travaux ont considéré des moyennes pondérées (Vulic et Moens, 2015).

Pour surmonter cette limite, plusieurs méthodes alternatives ont été proposées pour apprendre des représentations distribuées de ces unités textuelles (ou plus simplement, textes). Nous pouvons séparer les travaux en deux catégories (Nguyen, 2018), selon l'approche utilisée pour générer les représentations des textes : *(1)* la méthode *agrégée* (Kenter *et al.*, 2016; Hill *et al.*, 2016; Arora *et al.*, 2017), où un réseau de neurones calcule le vecteur de représentation d'un texte à partir de l'agrégation des vecteurs de représentation des mots qui composent le texte ; *(2) non-agrégée* (Le et Mikolov, 2014; Kiros *et al.*, 2015; Zamani et Croft, 2016), où le modèle obtient directement une représentation du texte, sans utiliser les représentations distribuées des mots qui le composent.

2.1.2.1 *Représentations distribuées agrégées*

La première approche, c.-à-d. par agrégation, utilise une fonction linéaire qui combine les représentations des mots d'une séquence pour construire son vecteur de représentation correspondant. L'approche la plus simple est d'effectuer

une simple moyenne des représentations distribuées (Vulic et Moens, 2015; Yin *et al.*, 2015; Zheng et Callan, 2015). Formellement, soit une séquence de mots $s = \{w_1, \ldots, w_n\}$, sa représentation distribuée $\mathbf{s}$ est calculée par :

$$\mathbf{s} = \frac{1}{\sum_i \alpha_i} \sum_{w_i \in s} \alpha_i \times \mathbf{w}_i \tag{3.13}$$

avec α_i le poids du mot w_i et $\mathbf{w_i}$ son vecteur de représentation.

Certains modèles s'appuient sur l'Équation 3.13 pour calculer la représentation latente d'un texte (Kenter *et al.*, 2016; Hill *et al.*, 2016; Arora *et al.*, 2017). Par exemple, Kenter *et al.* (2016) ont proposé le modèle *Siamese CBOW* qui consiste à apprendre les représentations des mots et à les agréger à l'aide d'une fonction (moyenne), pour construire la représentation d'un texte. Au cours de la phase d'apprentissage, le modèle optimise les plongements lexicaux pour mieux calculer le vecteur de représentation du texte. Le modèle *Siamese CBOW* s'appuie sur des techniques d'apprentissage supervisée pour apprendre à prédire les phrases qui se succèdent. Hill *et al.* (2016) ont introduit le modèle *Sequential Denoising Auto-Encoder* (SDAE), construit à partir d'une architecture LSTM. Le modèle est entraîné pour optimiser les représentations des mots. La représentation du texte est finalement calculée en agrégeant les vecteurs à l'aide d'une combinaison linéaire. Arora *et al.* (2017) ont quant à eux proposé une méthode qui se passe des réseaux de neurones pour apprendre les représentations des textes. Leur approche consiste à appliquer une moyenne pondérée sur les vecteurs distribués des mots d'une séquence (Équation 3.13). La représentation moyenne est ensuite modifiée à l'aide d'une analyse en composantes principales (ACP) ou d'une décomposition en valeurs singulières (DVS) pour réduire l'espace de représentation des textes.

Comme nous l'avons évoqué en introduction de cette section, effectuer une simple moyenne de vecteurs revient à accorder la même importance à tous les mots du texte. Bien que certains travaux (Vulic et Moens, 2015; Zheng et Callan, 2015) aient considéré d'appliquer des pondérations, Zamani et Croft (2016) ont remarqué que l'agrégation de vecteurs de mots issus d'une longue séquence pouvait entraîner des représentations imprécises du contenu sémantique. Les liens tels que les dépendances et les similarités entre les mots, phrases ou paragraphes d'un texte ne sont pas pris en compte. Pour dépasser ces limitations, des modèles de représentations distribuées non-agrégées ont été proposés.

2.1.2.2 *Représentations distribuées non-agrégées*

Contrairement à l'approche précédente, les modèles de représentations non-agrégées n'utilisent pas de fonctions d'agrégation. Ces derniers construisent directement une représentation latente du texte sans utiliser les représentations des mots composants.

L'approche la plus populaire pour construire la représentation latente d'un texte est le modèle *ParagraphVector* de Le et Mikolov (2014), une extension des *Word2Vec* de Mikolov *et al.* (2013a). Il est composé de deux modèles, à savoir *Distributed Memory* (PV-DM) et *Distributed Bag of Words* (PV-DBOW), dont les architectures sont illustrées dans la Figure 3.8. Le modèle *ParagraphVector* considère une séquence de mots telle qu'une phrase, un paragraphe ou un document complet, comme une unité atomique, plutôt qu'une combinaison des représentations des mots qui le compose.

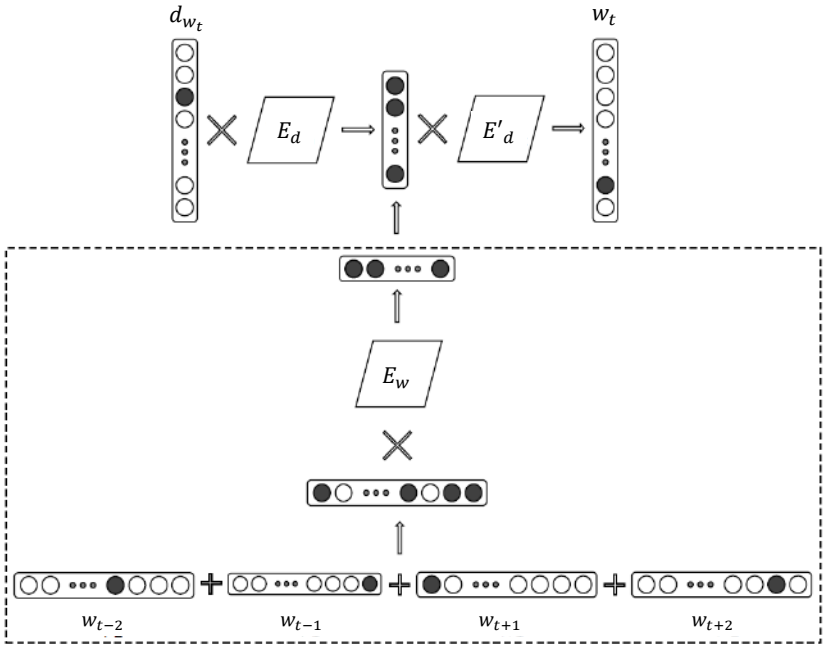


Figure 3.8 – Architecture du modèle *ParagraphVector* (Le et Mikolov, 2014). Le modèle PV-DBOW est entraîné à prédire un mot en tenant compte du document qui contient ce mot. Dans la variante PV-DM (zone en pointillée), les mots voisins sont aussi fournis comme données d'entrée. *(©Mitra et Craswell (2018))*

Le modèle PV-DBOW adopte la même architecture que celle du *Skip-Gram*, où la donnée d'entrée est un paragraphe plutôt qu'un mot. La matrice des représentations de mots E_w est remplacée par la matrice des représentations des documents E_d. Le réseau est entraîné pour prédire un mot compte tenu du paragraphe qui contient ce mot. Le vecteur de contexte c est le vecteur de représentation du document e_d, le score neuronal $s_\theta(w, c)$ est estimé par :

$$s_\theta(w, c) = e_w \cdot e_d \tag{3.14}$$

Le modèle PV-DM présente quant à lui une architecture identique à celle du CBOW, mais étendue avec un paragraphe dans la couche d'entrée, et une matrice de représentation de documents E_d. Ce modèle est entraîné pour prédire un

mot en tenant compte du contexte d'entrée composé des mots voisins et le paragraphe qui les contient. Le vecteur de contexte c est constitué de la représentation du document e_d et la combinaison (somme ou concaténation) des représentations des mots voisins dans la fenêtre de k mots (comme pour le CBOW classique). Le score neuronal $s_\theta(w, c)$ est obtenu par :

$$s_\theta(w, c) = e_w \cdot e_c = e_w \cdot (e_d \oplus e_{t-k} \oplus \cdots \oplus e_{t-1} \oplus e_{t+1} \oplus \cdots \oplus e_{t+k}) \tag{3.15}$$

où $\oplus$ est l'opérateur de combinaison des vecteurs de représentation. Ces deux modèles étant des extensions des architectures du *Word2Vec*, ils sont aussi optimisés à l'aide d'une fonction exponentielle normalisée hiérarchique ou d'un échantillonnage négatif. Une représentation de chaque paragraphe de la collection d'entraînement est apprise par le modèle. La représentation d'un nouveau paragraphe est obtenue par une étape d'inférence supplémentaire.

En suivant le succès des encodeurs-décodeurs (Kalchbrenner et Blunsom, 2013; Cho *et al.*, 2014), Kiros *et al.* (2015) ont introduit le modèle *Skip-Thought* qui conserve le principe de la composition sémantique[1] lors de l'apprentissage des vecteurs latents des textes. Le modèle consiste en un encodeur qui produit un vecteur de représentation du texte source, et d'un décodeur qui prédit séquentiellement les mots des phrases adjacentes. L'hypothèse sous-jacente est que, dans le contenu du document, tout ce qui conduit à une meilleure reconstruction des phrases voisines est essentiel à la représentation de la phrase. Un autre modèle, *Quick Thought* (QT) proposé par Logeswaran et Lee (2018), utilise une architecture d'encodeur-décodeur pour apprendre des représentations de textes. Plus spécifiquement, le modèle apprend les représentations des phrases tout en apprenant à prédire le contexte dans lequel elles apparaissent. L'encodeur construit la représentation latente de la phrase d'entrée tandis que le décodeur essaie de trouver les représentations distribuées des mots qui la compose.

2.1.3 *Apprentissage augmenté par des ressources externes*

Indéniablement, les plongements lexicaux sont largement utilisés pour traiter les différentes tâches, telles que la traduction automatique (Kalchbrenner et Blunsom, 2013), la désambiguïsation lexicale (Iacobacci *et al.*, 2016), le résumé automatique (Nallapati *et al.*, 2017) ou l'annotation sémantique (Moreno *et al.*, 2017). Néanmoins, les représentations sont très sensibles à la fenêtre contextuelle choisie (Levy et Goldberg, 2014), qui varie selon la collection. De ce fait, en fonction de la taille de la fenêtre, les significations sémantiques de certains mots peuvent ne pas

1. Le principe de compositionnalité sémantique est « *le principe selon lequel la signification d'une expression complexe est définie par les significations des expressions la composant, et par les règles employées pour les combiner* » (Pelletier, 1994).

être correctement représentées par les vecteurs. Une autre limite des plongements lexicaux est qu'ils ne permettent pas de lever le problème de polysémie, puisque les différents sens d'un même mot sont regroupés en une seule et même représentation (Iacobacci *et al.*, 2015). Par exemple, la représentation du mot « vivre » ne distingue pas la différence entre « exister » (c.-à-d. posséder une réalité) et « habiter » (c.-à-d. avoir son domicile quelque part). Par ailleurs, un apprentissage uniquement fondé sur la sémantique distributionnelle aura tendance à fusionner les différentes relations sémantiques. Par exemple, Mrksic *et al.* (2016) ont montré qu'il était difficile de distinguer les synonymes des antonymes dans les espaces de représentations.

Pour résoudre ces différents problèmes, plusieurs approches ont été proposées, principalement en exploitant des ressources sémantiques lors de l'apprentissage des plongements lexicaux. L'intuition de ces travaux est qu'injecter de la connaissance, au travers de concepts ou d'entités et de leurs relations (p. ex. synonyme, antonyme), permettrait de pallier le problème de polysémie et de corriger (ou régulariser) les plongements lexicaux. Une autre approche consiste à représenter des relations entre des paires de mots qui sont inventoriées dans une ressource (p. ex. Wikipedia, WordNet), mais pour lesquelles leurs contextes sont peu fréquents dans le corpus. Ainsi, Yu et Dredze (2014) ont proposé d'incorporer l'information externe dans les espaces vectoriels en rapprochant les représentations des mots similaires, c'est ce que l'on appelle la régularisation. Celle-ci peut s'effectuer à différentes étapes de l'apprentissage. Nous classons les travaux en deux catégories, *en ligne* et *hors-ligne*, déterminées par l'étape à laquelle la connaissance externe est injectée dans les représentations.

2.1.3.1 *Apprentissage en ligne des représentations de textes*

La régularisation dite *en ligne* exploite la connaissance issue des ressources externes pendant la phase d'apprentissage des plongements lexicaux. Ces modèles modifient l'objectif original de l'apprentissage distributionnel en y intégrant les contraintes issues des ressources externes. Nous distinguons deux types d'approchez : une première qui propose d'améliorer la lisibilité des représentations des mots en utilisant les relations entre les mots dans une ressource externe (Xu *et al.*, 2014; Yu et Dredze, 2014; Nguyen *et al.*, 2017) avec pour intuition que des mots liés via des relations sémantiques sont supposés avoir des représentations proches dans l'espace latent ; une seconde qui s'intéresse à un apprentissage conjoint des éléments d'un corpus (les mots) et des éléments des ressources sémantiques (les concepts) (Cheng *et al.*, 2015; Iacobacci *et al.*, 2015; Yamada *et al.*, 2016; Mancini *et al.*, 2017; Nguyen *et al.*, 2018), afin de mieux discriminer le sens des mots et donc résoudre le problème de polysémie.

Dans la première approche, les auteurs intègrent la connaissance dans le processus d'apprentissage des modèles tels que CBOW ou *Skip-Gram*, en modifiant la fonction objectif avec un terme de régularisation. Ainsi, Yu et Dredze (2014) utilisent des ressources externes comme une source d'évidence pour la prédiction d'un mot en fonction de son contexte. Ils étendent la fonction objectif du modèle CBOW en injectant la connaissance préalable des synonymes à partir de ressources sémantiques telles que *Paraphrase Database* ou *WordNet*. En s'appuyant sur une ressource, leur modèle apprend les représentations latentes de manière à prédire un mot à partir d'un autre mot connecté. De la même manière, Xu *et al.* (2014) ont proposé le modèle *RC-NET* qui exploite des connaissances relationnelles (R) et catégorielles (C) afin de produire de meilleures plongements lexicaux. La connaissance relationnelle (p. ex. *est-un*, *partie-de*, *enfant-de*) encode la relation entre les entités, permettant ainsi de différencier les paires de mots avec des relations analogiques ; la connaissance catégorielle (p. ex. le sexe, l'emplacement) encode les attributs et les propriétés des entités, selon lesquelles des mots similaires peuvent être regroupés dans les catégories significatives. Par ailleurs, les connaissances catégorielles encodent les attributs ou propriétés des mots, à partir desquels il est possible de regrouper des mots similaires en fonction de leurs attributs. Ainsi, Xu *et al.* (2014) suggèrent que les représentations des mots qui appartiennent à la même catégorie doivent être proches l'une de l'autre. Comparé au modèle *Skip-Gram*, *RC-NET* présente des améliorations significatives en termes de qualité des représentations, sur les tâches d'analogie, de similarité de mots et de prédiction de catégorie (Mikolov *et al.*, 2013a; Finkelstein *et al.*, 2001). Enfin, Nguyen *et al.* (2017) ont proposé un modèle neuronal, *HyperVec*, permettant l'apprentissage de représentations hiérarchiques qui discriminent l'hyperonyme des autres relations et distinguent l'hyperonyme de l'hyponyme dans une paire de relations. *HyperVec* étend le modèle *Skip-Gram* en ajoutant deux fonctions objectifs pour apprendre les représentations hiérarchiques pour l'hyponyme.

Dans la seconde approche, les auteurs s'intéressent à l'apprentissage conjoint des représentations des mots d'un corpus et des éléments des ressources sémantiques. Cet apprentissage permet de mieux discriminer le sens des mots, et par conséquent, aide à résoudre le problème de polysémie. Par exemple, Iacobacci *et al.* (2015) utilisent le logiciel Babelfy (Moro *et al.*, 2014), un algorithme de désambiguïsation pour obtenir les sens des mots dans le corpus Wikipedia, puis révisent la fonction objectif du CBOW en identifiant des paires concepts-mots dans un contexte donné pour apprendre les représentations des mots et leur signification dans le même espace latent. Cheng *et al.* (2015) proposent quant à eux d'estimer, dans la phase d'apprentissage des représentations, la probabilité d'associer un concept à un mot dans la fenêtre de contexte. Leur modèle étend le *Skip-Gram* en identifiant les paires mots-concepts dans un contexte donné, en effectuant l'entraînement conjoint de leurs représentations latentes. De ce fait, les représentations

des mots et leur sens sont apprises dans le même espace latent. Dans le même esprit, de représenter dans un espace vectoriel partagé les représentations des mots et des concepts, Yamada *et al.* (2016) ont proposé des extensions du modèle *Skip-Gram* pour désambiguïser des entités nommées. Le modèle *KB-Graph* permet d'apprendre la similarité des entités en utilisant les relations issues de ressources sémantiques, tandis que le modèle *anchor-context* aligne les vecteurs de représentation de telle sorte que les mots et les entités similaires se situent à proximité les uns des autres dans l'espace vectoriel. Enfin, plus récemment Mancini *et al.* (2017) ont introduit un modèle d'apprentissage conjoint des représentations des mots et des concepts en exploitant les connaissances issues des ressources sémantiques. Le modèle *SW2V* s'appuie sur l'architecture du modèle CBOW en ajoutant aux couches d'entrée et de sortie, des sens de mots. Le modèle exploite ainsi la relation intrinsèque entre les mots et les sens. Leur intuition est que, comme un mot est un symbole d'un sens sous-jacent, la mise à jour de la représentation du mot devrait produire une mise à jour de ce sens spécifique, et vice-versa. En appliquant un algorithme d'identification des sens fondé sur *WordNet*, un mot donné peut avoir zéro, un ou plusieurs sens. Chaque mot cible prend comme contexte à la fois les mots qui l'entourent (dans la fenêtre contextuelle), et tous les sens associés à ces mots. Entraîné sur le corpus Wikipédia, leur modèle est en mesure de construire un espace vectoriel des mots et des sens sémantiquement cohérents.

2.1.3.2 *Apprentissage hors-ligne des représentations de textes*

La régularisation dite *hors-ligne* diffère de l'approche précédente par l'utilisation des ressources hors de l'étape d'apprentissage des plongements lexicaux (Faruqui *et al.*, 2015; Mrksic *et al.*, 2016, 2017; Vulic *et al.*, 2017, 2018; Vulic et Mrksic, 2018). Cette approche de correction a posteriori, appelée *retrofitting*, corrige (ou affine) des représentations distribuées préalablement calculées à partir de n'importe quel modèle (modèles de langue neuronaux, matrices de cooccurrences, etc.) pour satisfaire les contraintes des ressources externes. En d'autres termes, l'idée principale de la correction a posteriori est de rapprocher des mots qui sont reliés par une relation définie dans une ressource sémantique donnée, en corrigeant leurs représentations distribuées pré-entraînées, comme l'illustre la Figure 3.9. Comme l'un de nos travaux de recherche exploite ce type d'approche, nous nous y attardons dans la suite de cette section.

Les travaux de Faruqui *et al.* (2015) sont une première introduction de correction a posteriori, qui propose une méthode pour corriger les représentations dans l'espace vectoriel à l'aide des informations relationnelles issues des lexiques sémantiques, en encourageant les mots connectés à avoir des représentations vectorielles similaires. De plus, aucune hypothèse n'est posée quant à l'algorithme utilisé pour construire les plongements lexicaux, rendant ainsi la méthode de correction com-

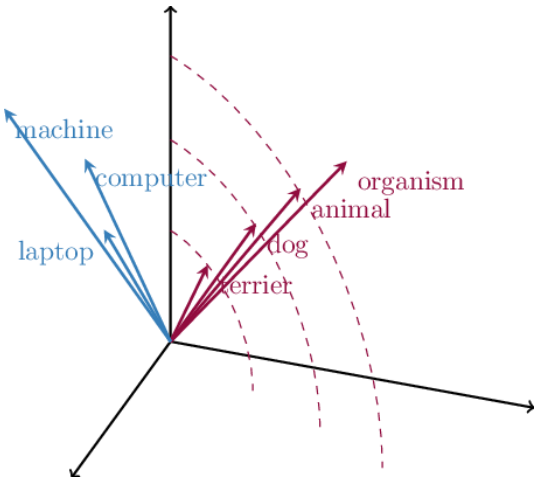


Figure 3.9 – Illustration de l'espace vectoriel transformé par Vulic et Mrksic (2018). Le modèle contrôle la disposition des vecteurs dans l'espace vectoriel en mettant l'accent sur les similarités des paires d'hyponymie et en imposant un ordre d'hyponymie.

plètement indépendante des algorithmes d'apprentissage des plongements lexicaux. La méthode proposée par Faruqui *et al.* (2015) encourage les nouvelles représentations à être similaires aux représentations des mots reliés dans la ressource externe et similaires à leurs représentations purement distribuées (c.-à-d. représentations d'origine). Formellement, soit $\mathcal{V} = \{w_1, \ldots, w_n\}$ le vocabulaire, Ω une ressource qui encode les relations sémantiques entre les mots du vocabulaire, et $\mathcal{R}$ un ensemble de types de relation. La ressource Ω est représentée par un graphe non-orienté $(\mathcal{V}, \mathcal{R})$ avec un nœud pour chaque mot et des arrêtes $(w_i, w_j) \in \mathcal{R} \subseteq \mathcal{V} \times \mathcal{V}$ qui indiquent une relation sémantique entre deux mots (w_i, w_j). Étant donnée la matrice $\widehat{\mathbf{W}}$, la collection des plongements lexicaux $\widehat{\mathbf{w}}_i \in \mathbb{R}^d$ (pour chaque mot $w_i \in \mathcal{V}$), entraînée par n'importe quel modèle, le but de la correction a posteriori est d'apprendre un ensemble de nouvelles représentations $\mathbf{W} = \{\mathbf{w_1}, \ldots, \mathbf{w_n}\}$ qui contiennent les informations qui encondent conjointement les connaissances de la sémantique distributionnelle à partir des corpus de textes et la structure de la ressource sémantique externe. En utilisant la distance euclidienne comme distance sémantique entre deux représentations de mots, les auteurs ont défini la fonction objectif à minimiser suivante :

$$\Psi(\mathbf{W}) = \sum_{i=1}^{|\mathcal{V}|} \left[\alpha_i ||\mathbf{w_i} - \widehat{\mathbf{w}}_i||^2 + \sum_{(i,j)\in\mathcal{R}} \beta_{ij} ||\mathbf{w}_i - \mathbf{w}_j||^2 \right] \tag{3.16}$$

où α_i et β_{ij} sont des valeurs qui contrôlent les forces relatives de la combinaison. Faruqui *et al.* (2015) ont expérimenté leur modèle sur différentes représentations vectorielles pré-entraînées (*GloVe*, *Skip-Gram*, etc.) avec des ressources lexicales telles

que *PPDB*, *WordNet* et *FrameNet*. Les résultats de l'évaluation expérimentale sur différentes tâches (similarité de mots, relations syntaxiques) ont montré que cette méthode améliore significativement la qualité des plongements lexicaux et aussi qu'elle surpasse les méthodes de correction en ligne telles que Yu et Dredze (2014) et Xu *et al.* (2014).

Dans la même lignée, Mrksic *et al.* (2016) ont proposé la méthode *counter-fitting* qui injecte les contraintes d'antonymie et de synonymie dans l'espace de représentation vectorielle afin de renforcer la capacité des vecteurs à évaluer la similarité sémantique. Leur modèle s'appuie sur le principe d'attraction et de répulsion : l'idée est de rapprocher les représentations des paires de mots synonymes et d'éloigner les représentations des paires de mots antonymes, tout en conservant la sémantique distributionnelle préalablement apprise. Formellement, soit $\mathcal{V} = \{w_1, \ldots, w_n\}$ le vocabulaire, et $\widehat{\mathbf{W}} = \{\widehat{\mathbf{w_1}}, \ldots, \widehat{\mathbf{w_n}}\}$ l'ensemble des représentations distribuées. L'objectif du modèle est d'obtenir un ensemble de nouvelles représentations $\mathbf{W} = \{\mathbf{w_1}, \ldots, \mathbf{w_n}\}$ augmentées par un ensemble de contraintes d'antonymie et de synonymie respectivement notés $\mathcal{A}$ et $\mathcal{S}$, et qui contiennent des paires de mots (w_i, w_j) reliées par la relation correspondante. La fonction de coût utilisée pour corriger les vecteurs de mots se décompose en trois parties : *AR*, *SA* et *VSP*. Le premier terme, *AR*, sert à éloigner les vecteurs des mots antonymes les uns des autres dans l'espace vectoriel transformé $\mathbf{W}$. Il est défini par :

$$AR(\mathbf{W}) = \sum_{(i,j)\in\mathcal{A}} \tau(1 - d(\mathbf{w_i}, \mathbf{w_j})) \tag{3.17}$$

où $d(\mathbf{w_i}, \mathbf{w_j} = 1 - cos(\mathbf{w_i}, \mathbf{w_j})$ est la distance dérivée de la similarité du cosinus et $\tau(x) = \max(0, x)$ impose une marge maximale sur la valeur du coût. Le deuxième terme, *SA*, cherche à rapprocher les vecteurs de mots synonymes. Il est défini par :

$$SA(\mathbf{W}) = \sum_{(i,j)\in\mathcal{S}} \tau(d(\mathbf{w_i}, \mathbf{w_j})) \tag{3.18}$$

Le dernier terme, *VSP*, permet de rapprocher le plus possible l'espace vectoriel transformé de l'espace vectoriel original afin de préserver les informations sémantiques contenues dans le vecteur original. Sa formulation est la suivante :

$$VSP(\mathbf{W}, \widehat{\mathbf{W}}) = \sum_{i=1}^{|\mathcal{V}|} \sum_{j\in\mathcal{N}(i)} \tau(d(\mathbf{w_i}, \mathbf{w_j}) - d(\widehat{\mathbf{w_i}}, \widehat{\mathbf{w_j}})) \tag{3.19}$$

où $N(i)$ désigne l'ensemble des mots situés dans un rayon ρ autour du vecteur du i^{e} mot dans l'espace vectoriel d'origine. La fonction objectif finale est donnée par la somme pondérée des trois termes :

$$\mathcal{L}(\mathbf{W}, \widehat{\mathbf{W}}) = k_1 AR(\mathbf{W}) + k_2 SA(\mathbf{W}) + K_3 VSP(\mathbf{W}, \widehat{\mathbf{W}}) \tag{3.20}$$

où $k_1, k_2, k_3 \geq 0$ sont les hyperparamètres qui contrôlent l'importance relative de chaque terme. Les auteurs ont montré au travers une évaluation expérimentale que leur modèle corrige efficacement les plongements lexicaux, permettant ainsi d'améliorer les performances sur des tâches de similarité sémantique. De plus, la séparation des représentations des paires antonymes contribue à améliorer substantiellement les performances.

La méthode *counter-fitting* introduite par Mrksic *et al.* (2016) a été la pierre angulaire à de nombreux travaux de recherche. En effet, devant les bonnes performances de cette méthode, Mrksic *et al.* (2017) ont proposé un nouvel algorithme, baptisé *Attract-Repel* (« attirer-repousser ») pour améliorer la qualité sémantique des plongements lexicaux en injectant des contraintes extraites des ressources lexicales. Ce dernier diffère de la méthode *counter-fitting* sur plusieurs aspects : ils utilisent des contraintes mono- et multilingues pour l'apprentissage des représentations ; ils ajustent les espaces vectoriels en mettant à jour à la fois les représentations des paires synonymes (ou antonymes) et celles de leurs exemples négatifs [2] ; ils s'appuient sur une régularisation L2 qui tire chaque nouveau vecteur vers sa représentation distribuée initiale. Vulic *et al.* (2017) se sont quant à eux inspirés du modèle *Attract-Repel* (Mrksic *et al.*, 2017) pour proposer une procédure de régularisation s'appuyant sur la morphologie pour améliorer la qualité des plongements lexicaux. Enfin, Vulic et Mrksic (2018) ont utilisé l'idée d'attirer et repousser les vecteurs de mots pour rapprocher les paires hyponymie-hypernymie dans l'espace euclidien. Pour ce faire, ils ont modifié la fonction objectif du modèle *Attract-Repel* (Équation 3.20) en ajoutant le terme *LE* (*Lexical Entailment*) pour mettre en relief la distance hiérarchique de l'implication lexicale. Contrairement à la similarité symétrique, l'implication lexicale impose une distance asymétrique qui encode un ordre hiérarchique entre les concepts.

De par leur conception, les méthodes décrites précédemment ne font que mettre à jour les vecteurs des mots apparaissant dans les ressources externes, laissant intactes les représentations de tous les autres mots. De ce fait, Vulic *et al.* (2018) réutilisent le modèle *Attract-Repel* introduit par Mrksic *et al.* (2017) mais en l'étendant aux mots absents des ressources externes. Leur approche, appelée *post-specialisation* et illustrée dans la Figure 3.10, préserve les connaissances linguistiques des mots rencontrés dans la ressource, et propage ensuite le signal aux autres mots du vocabulaire pour améliorer leur représentation. Concrètement, ils commencent par appliquer le modèle *Attract-Repel* pour corriger les représentations des mots contenus dans les ressources externes, puis, à l'aide d'un réseau de neurones, ils apprennent à prédire les vecteurs corrigés à partir de leurs homo-

2. Les exemples négatifs sont utilisés pour forcer les paires synonymes à être plus proches les unes des autres que de leur exemple négatif respectif, et pour forcer les paires antonymes à être plus éloignées l'une de l'autre que de leur exemple négatif.

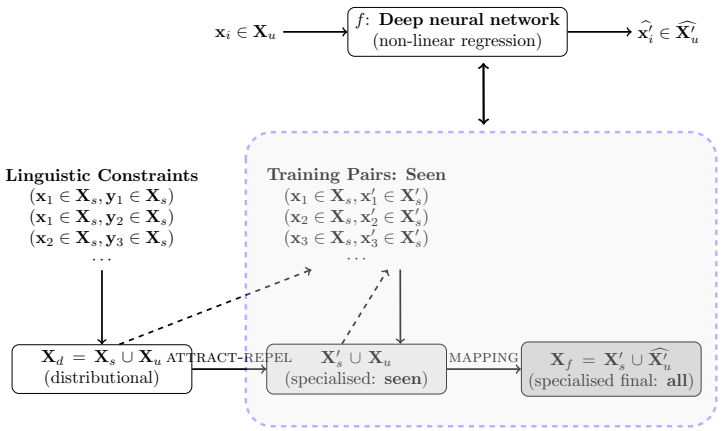


Figure 3.10 – Illustration de l'approche *post-specialisation* : le sous-espace $\mathbf{X}_s$ de l'espace vectoriel initial $\mathbf{X}_d = \mathbf{X}_s \cup \mathbf{X}_u$ est d'abord spécialisé/corrigé par le modèle *Attract-Repel* pour obtenir le sous-espace transformé $\mathbf{X}'_s$. Les mots présents dans l'ensemble des contraintes linguistiques d'entrée se voient maintenant attribuer différentes représentations dans $\mathbf{X}_s$ (le vecteur distribué original) et $\mathbf{X}'_s$ (le vecteur corrigé). Ils sont ainsi utilisés comme exemples d'apprentissage pour apprendre une fonction de *mapping*. Cette fonction est ensuite appliquée à tous les vecteurs de mots $\mathbf{x}_i \in \mathbf{X}_u$ représentant les mots non vus dans les contraintes pour produire le sous-espace corrigé $\widehat{\mathbf{X}'_u}$. L'espace final est $\mathbf{X}_f = \mathbf{X}'_s \cup \widehat{\mathbf{X}'_u}$ et contient les représentations transformées de tous les mots de l'espace initial $\mathbf{X}_d$. (©Vulic et Mrksic (2018))

logues d'origine. Enfin, ils utilisent la fonction apprise pour obtenir les représentations distribuées manquantes. Cette approche leur a permis de réaliser un gain de performance considérable par rapport aux précédents modèles de régularisation, tant sur des tâches de similarité intrinsèque que dans des tâches de TALN (simplification de textes et suivi de dialogue).

2.2 *Représentations distribuées de géotextes augmentées par les contextes spatiaux*

Nous l'avons évoqué dans la Section 2, les plongements lexicaux ont été le point de départ de nombreux de travaux de recherche sur l'appariement de textes. Les nouvelles approches proposées s'appuient ainsi sur la sémantique des mots et des documents, au travers des vecteurs de représentation qui capturent la sémantique distributionnelle. Cependant, plusieurs travaux ont montré l'existence de langages sensibles à la localisation qui sont suivis par une variation de mots et de sujets en fonction des contextes géospatiaux (Backstrom *et al.*, 2008; Han *et al.*, 2012; Laere *et al.*, 2014). En étudiant les distributions des occurrences des mots, Laere *et al.*

(2014) ont par exemple remarqué que les distributions de certains mots divergent de la distribution générale de la collection dans certaines régions, révélant ainsi des spécificités locales. Han *et al.* (2012) ont quant à eux découvert, par le biais de mesures d'entropies s'appuyant sur la fréquence des termes, que les mots pouvaient être révélateurs de localisation. Enfin, grâce à un *topic modelling*, Eisenstein *et al.* (2010) ont montré que les sujets avaient des variantes lexicales régionales. L'intégration de ces spécificités locales semble donc essentielle pour résoudre des tâches de RIG.

La prolifération croissante des réseaux sociaux s'appuyant sur la localisation a entraîné la création de larges volumes données, aujourd'hui exploitées dans de nombreuses tâches de RIG, notamment pour la recommandation de POIs. Plusieurs travaux récents se sont ainsi focalisés sur la représentation d'objets issus des réseaux sociaux dans un espace de faible dimension en utilisant conjointement diverses informations telles que la localisation, la temporalité et le contenu textuel. La plupart des recherches récentes sur l'apprentissage de représentations de géotextes s'appuient sur les techniques d'apprentissage des plongements lexicaux utilisées dans les NNLM, et notamment sur les modèles *Skip-Gram* et CBOW (Mikolov *et al.*, 2013a,b). Les travaux peuvent être regroupés en deux catégories, une première qui considère l'influence du contexte géographique sur les représentations des mots, une seconde qui modélise directement les représentations des géotextes.

2.2.1 *Représentations distribuées des mots*

Une première catégorie de travaux s'est concentrée sur la modélisation cartographique entre les mots et les sujets à l'aide des plongements lexicaux (Cocos et Callison-Burch, 2017; Zhang *et al.*, 2017b).

Dans une première tentative d'étudier l'impact du contexte géographique sur la sémantique des mots, Cocos et Callison-Burch (2017) ont exploité l'idée d'inférer des représentations vectorielles régionales de mots, de telle sorte que les mots spatialement proches aient des significations similaires. Pour cela, ils adaptent le modèle *Skip-Gram* (Mikolov *et al.*, 2013b) pour déterminer les représentations des mots issus d'une collection de tweets géolocalisés, en utilisant des contextes géographiques dérivés de Google Places et OpenStreetMap, décrivant le type des lieux (p. ex. théâtre, restaurant, université) situés autour des tweets, comme illustré dans la Figure 3.11. Plus spécifiquement, pour chaque tweet de taille n, ils récupèrent tous les objets situés dans un rayon r autour du tweet et énumèrent la liste de taille m des *tags* associés à ces objets. Par exemple, le tweet illustré dans la Figure 3.11 contient, dans un rayon de 30 mètres, $m = 10$ objets (p. ex. `point7728`, `line575`, `poly1903`) associés à différents tags géographiques (p. ex. `route:bus`,

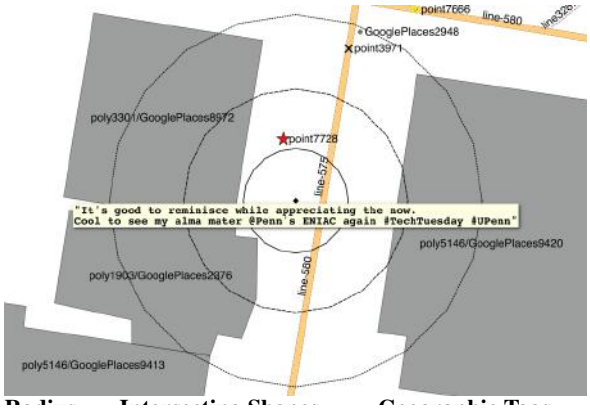


Figure 3.11 – Enrichissement d'un tweet avec des contextes géographiques situés à des rayons D croissants (Cocos et Callison-Burch, 2017).

`highway:tertiary`, `building:yes`). Dans leur modèle, chaque mot du tweet partage le même contexte. Ainsi, le contexte utilisé comme entrée du *Skip-Gram*, généré à partir du tweet d'exemple, est une liste de dimension $m \times n$ de paires *(mots, contexte)* : $[(\text{it's}, \texttt{route:bus}), (\text{good}, \texttt{route:bus}), \ldots, (\#\text{UPenn}, \texttt{poi:marker})]$. L'évaluation intrinsèque des plongements lexicaux ainsi créés a montré que le contexte spatial permet d'encoder efficacement des informations sur la relation sémantique. Par ailleurs, d'après les résultats de l'évaluation extrinsèque, bien que le contexte géospatial ne soit pas aussi riche en sémantique que le contexte textuel, il fournit des informations pertinentes sur la relation sémantique, qui peuvent être complémentaires dans le cadre d'un modèle multimodal.

Zhang *et al.* (2017b) ont abordé le problème du glissement lexical entre les régions, en proposant un modèle permettant aux utilisateurs de faire des recherches à partir d'exemples analogiques. Par exemple, un utilisateur américain recherchant des informations sur la « *NASA Japonaise* », devrait recevoir des informations sur l'équivalent de la NASA au Japon, c.-à-d. *JAXA*. Les auteurs proposent de transformer l'espace vectoriel sous différentes distributions thématiques pour générer un *mapping* entre différents contextes géographiques.

Plus formellement, soit deux espaces vectoriels (c.-à-d. des plongements lexicaux) entraînés sur des collections différentes (Figure 3.12) : un espace de base noté $S^b = \{w_1^b, \ldots, w_m^b\}$ ($w_i^b \in \mathcal{V}^b$) à partir duquel les requêtes sont sélectionnées, et un espace cible $S^t = \{w_1^t, \ldots, w_m^t\}$ ($w_i^t \in \mathcal{V}^t$) où la réponse doit être recherchée. L'objectif est de déterminer un objet w^t (p. ex. JAXA) qui est contextuellement similaire à l'objet w^b (p. ex. NASA). Pour cela, Zhang *et al.* (2017b) proposent de calculer une matrice de transformation permettant de passer d'un espace à l'autre. En supposant que nous disposions de *termes ancres*, c.-à-d. des termes pour lesquels nous connaissons un équivalent dans les deux espaces vectoriels,

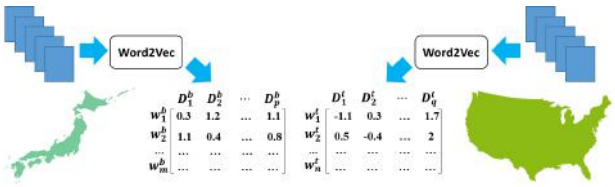


Figure 3.12 – Entraînement des plongements lexicaux sur des collections différentes (Zhang *et al.*, 2017b).

notés $\{(x_1^b, x_1^t), \ldots, (x_u^b, x_u^t)\}$ où x_i^b est l'ancre dans un espace (p. ex. Japon) et x_i^t son équivalent dans le second espace (p. ex. USA). La matrice de transformation $\mathbf{M}$ est construite en minimisant la différence entre $\mathbf{M}\mathbf{x_i^b}$ et $\mathbf{x_i^t}$:

$$\mathbf{M} = \arg\min_{M} \sum_{i=1}^{u} \left\| \mathbf{M}\mathbf{x_i^b} - \mathbf{x_i^t} \right\|_2^2 + \gamma \left\| \mathbf{M} \right\|_2^2 \tag{3.21}$$

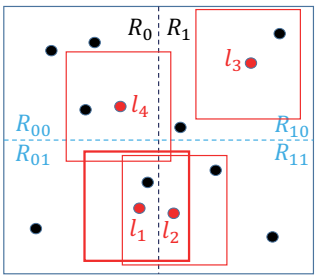
L'approche proposée se concentre plutôt sur les aspects linguistiques, tandis que les aspects géographiques ne sont pas directement considérés.

2.2.2 *Représentations distribuées des géotextes*

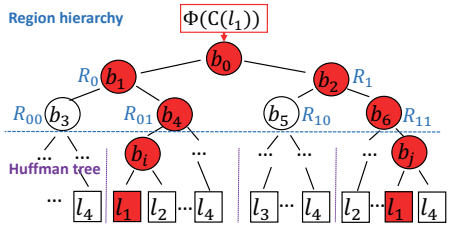
La seconde catégorie de travaux s'est directement focalisée sur la représentations de géotextes en tenant compte des aspects sémantiques, géographiques et temporels.

Une première famille de travaux (Feng *et al.*, 2017; Zhao *et al.*, 2017; Yan *et al.*, 2017) s'est intéressée à la proximité géographique et l'influence temporelle (succession de *check-ins*) pour représenter directement les POIs. Feng *et al.* (2017) ont proposé *POI2Vec*, un modèle d'apprentissage des représentations latentes des POIs, permettant d'intégrer l'influence géographique. Les représentations sont déterminées par l'architecture CBOW, qui consiste à prédire un item, ici un POI, étant donné son contexte, ici les POIs visités auparavant par un utilisateur. Feng *et al.* (2017) ont remplacé l'arbre binaire hiérarchique (*Huffman tree*) classiquement employé, par un arbre s'appuyant sur la distance géographique entre les POIs, afin de refléter l'influence géographique. Comme l'illustre la Figure 3.13, les POIs sont divisés en une hiérarchie de régions binaires, de sorte que les POIs les plus proches soient plus susceptibles d'être regroupés dans la même région. Ainsi, la probabilité d'observer un POI l étant donné son contexte $C(l)$ est définie comme :

$$P(l|C(l)) = \prod_{path_k \in P(l)} P(path_k) \times P(l|C(l))^{path_k} \tag{3.22}$$



(a) Illustration de construction de l'arbre binaire de POIs.



(b) Arbre binaire généré par le modèle.

Figure 3.13 – Illustration des étapes de génération de l'arbre binaire avec le modèle POI2Vec (Feng *et al.*, 2017).

avec $P(l|C(l))^{path_k}$ la probabilité d'observer le POI l le long du chemin $path_k$ et $P(path_k)$ la probabilité de ce chemin.

Zhao *et al.* (2017) ont développé le modèle *Geo-Temporal Sequential Embedding Rank* (*Geo-Teaser*) qui capture les informations contextuelles des POIs à partir des séquences d'enregistrements des utilisateurs. Leur modèle s'appuie sur l'architecture *Skip-Gram*, où chaque POI est traité comme un mot, et les *check-ins* d'un utilisateur comme le contexte. Ces travaux ont donné d'excellents résultats pour la recommandation de POIs et la prédiction des futures visites.

Yan *et al.* (2017) se sont plutôt attachés à la représentation des catégories des POIs avec leur modèle *Place2Vec*. Leur modèle s'appuie sur l'hypothèse qu'un POI peut être catégorisé par ses voisins. Ils ont ainsi adapté le modèle *Skip-Gram* pour prédire les POIs contextuels étant donné la catégorie centrale. Contrairement aux travaux précédents, ils n'utilisent pas de données temporelles pour apprendre leurs représentations. Dans leur approche, Yan *et al.* (2017) proposent d'approximer la distribution de probabilité des types de lieux en utilisant l'entropie croisée pour mesurer la différence entre la probabilité apprise $\widehat{y}$ et la probabilité réelle y :

$$D(\widehat{y}, y) = -y_c \log(\widehat{y}_c) \tag{3.23}$$

$\widehat{y}_c$ est la probabilité prédite des types de POIs du contexte, et y_c est la probabilité réelle. $\widehat{y}_c$ peut être défini comme :

$$\widehat{y}_c = P(t_1, t_2, \ldots, t_m | t_c) \tag{3.24}$$

où $t_1, t_2, \ldots, t_m$ sont les types de lieux du contexte et t_c est le type de lieu central.

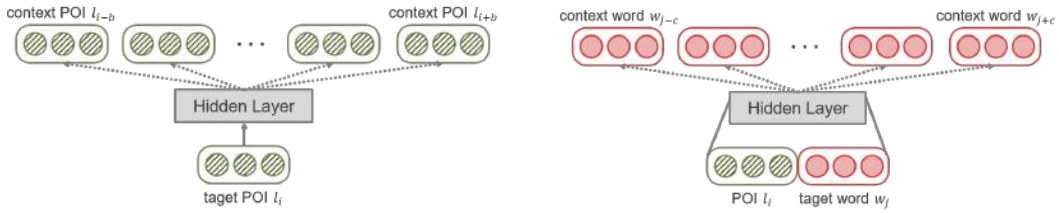
D'autres auteurs (Xie *et al.*, 2016; Zhang *et al.*, 2017a; Hang *et al.*, 2018; Chang *et al.*, 2018; Hao *et al.*, 2019) se sont plutôt tournés vers l'apprentissage de représentations multimodales, pour représenter dans un même espace les différentes caractéristiques des objets géotextuels.

Zhang *et al.* (2017a) ont proposé *TrioVecEvent*, une méthode s'appuyant sur les représentations distribuées pour la détection des événements locaux, à partir de flux de tweets géolocalisés. Le modèle *TrioVecEvent* représente dans un même espace latent, la localisation sous forme de régions, la temporalité divisée en heure, et le texte, tout en préservant leurs corrélations. Ces représentations multimodales permettent non seulement de saisir les similitudes sémantiques entre les tweets, mais aussi de révéler les mots clés typiques des différentes régions et heures. L'apprentissage des représentations s'inspire du modèle CBOW, en prédisant une unité (région, heure ou mot-clé) étant donné son contexte (Équation 3.25). La fonction objectif de leur modèle est optimisée à l'aide de l'échantillonnage négatif, sur toutes les unités en même temps. Formellement, étant donné un tweet d, pour une unité $i \in d$ de type X (région, heure, mot-clé), v_i la représentation distribuée de l'unité i, les auteurs modélisent la probabilité d'observer i comme :

$$P(i|d_i) = \exp(sim(i, d_i)) / \sum_{J \in X} \exp(sim(j, d_i) \tag{3.25}$$

où d_i est l'ensemble de toutes les unités présentes dans d excepté i, et $sim(i, d_i)$ est un score de similarité. Dans le même esprit, Hao *et al.* (2019) ont représenté dans un même espace, la temporalité divisée en parties de la semaine et en heures, l'emplacement sous la forme d'un identifiant unique, et le texte.
Chang *et al.* (2018) ont quant à eux proposé le modèle *Content-Aware hierarchical POI Embedding* (*CAPE*) qui capture conjointement l'influence géographique des POIs via les *check-ins* successifs d'utilisateurs, ainsi que les caractéristiques des POIs, pour la tâche de recommandation successive de POIs. Comme illustré dans la Figure 3.14, le modèle *CAPE* se compose de deux couches : la couche de contexte des *check-ins* qui permet de représenter les POIs à l'aide des *check-ins* séquentiels d'utilisateurs, en forçant les POIs d'une séquence à être proche dans l'espace vectoriel, et la couche de contexte du texte, où les représentations des POIs sont entraînées pour capturer les caractéristiques d'un POI à partir des mots qui le décrivent. La fonction objectif du modèle *CAPE* s'appuie sur la combinaison de deux architectures *Skip-Gram* (une par couche), combinant ainsi le contexte des *check-ins* et le contenu textuel. De fait, le modèle *CAPE* apprend simultanément l'influence géographique et les caractéristiques sémantiques des POIs.

Enfin, d'autres travaux (Xie *et al.*, 2016; Hang *et al.*, 2018) ont proposé d'apprendre des représentations multimodales à partir de graphes pour la recommandation de POIs. Plus spécifiquement, Xie *et al.* (2016) ont développé *Graph-Embedding* (*GE*), un modèle d'apprentissage de représentations distribuées de POIs s'appuyant sur un ensemble de graphes. Leur modèle capture conjointement l'effet séquentiel (séquence des visites des POIs), l'influence géographique (proximité spatiale des POIs), l'effet cyclique temporel (fréquence des visites d'un POIs dans une période) et l'effet sémantique (mots en communs entre les POIs) en les intégrant dans un espace latent partagé. L'apprentissage des représentations s'appuie



(a) Illustration de la couche des *check-ins*.

(b) Illustration de la couche de texte.

Figure 3.14 – Illustration des couches *check-ins* et *texte* du modèle CAPE (Feng *et al.*, 2017).

sur le modèle *Skip-Gram* (Mikolov *et al.*, 2013b), appliqué aux graphes. Plus précisément, étant donné un graphe bipartie $G = (\mathcal{V}_A \cap \mathcal{V}_B, \varepsilon)$, où $\mathcal{V}_A$ et $\mathcal{V}_B$ sont deux ensembles disjoints de sommets et ε l'ensemble des arêtes. La probabilité conditionnelle qu'un sommet $v_j \in \mathcal{V}_B$ soit généré par le sommet $v_i \in \mathcal{V}_A$ est définie par :

$$P(v_j|v_i) = \frac{\exp(\mathbf{v_j} \cdot \mathbf{v_i})}{\sum_{v_k} \exp(\mathbf{v_k} \cdot \mathbf{v_i})} \tag{3.26}$$

où $\mathbf{v_x}$ est le vecteur de représentation du sommet v_x. Finalement, étant donnés différentes instances (c.-à-d. un nœud) et leurs contextes (c.-à-d. ses nœuds voisins), l'objectif est de minimiser la fonction de coût permettant de prédire le contexte à partir de la représentation distribuée de l'instance.

$$O = -\sum_{e_{i,j} \in \varepsilon} \alpha_{ij} \log P(v_j|v_i) \tag{3.27}$$

Dans le même esprit, Hang *et al.* (2018), avec le modèle *Embedding for Dense Heterogeneous Graphs* (*EDHG*), ont choisi de représenter les corrélations entre les utilisateurs, les POIs et les activités. Bien qu'ils utilisent la même fonction objectif que Xie *et al.* (2016), ils ont ajusté l'approche d'échantillonnage négatif pour mieux tenir compte des caractéristiques des graphes.

3 Réseaux de neurones profonds pour l'appariement de textes

Nous l'avons vu dans la section précédente, les réseaux de neurones ont largement fait leurs preuves pour l'apprentissage de représentations distribuées de mots (Mikolov *et al.*, 2013a,b; Pennington *et al.*, 2014), de phrases (Le et Mikolov, 2014; Kiros *et al.*, 2015) ou de géotextes (Feng *et al.*, 2017; Hao *et al.*, 2019), qui sont ensuite utilisés efficacement dans des tâches d'appariement (Mitra *et al.*, 2016; Shen

et al., 2014b; Yan *et al.*, 2017). D'autres travaux, s'appuyant aussi sur les réseaux de neurones, ont été proposés pour modéliser de bout-en-bout des tâches d'appariement de textes. Ces modèles ont largement été abordés dans de multiples tutoriels et études comparatives (Li et Lu, 2016; Onal *et al.*, 2017; Guo *et al.*, 2019), où les modèles proposés utilisent les réseaux de neurones afin d'extraire automatiquement les différents signaux d'appariement à partir des plongements lexicaux, afin de mieux les combiner. Ces modèles sont utilisés dans diverses tâches d'appariement de textes, telles que :

— *la recherche ad-hoc* : recherche documentaire (Guo *et al.*, 2016; Mitra *et al.*, 2017; Fan *et al.*, 2018) et recherche de passages (Schneider *et al.*, 2018; Nogueira et Cho, 2019) ;
— *le système de questions-réponses* : sélection de phrases (Yu et Dredze, 2014; Rao *et al.*, 2016; Wang *et al.*, 2016) et réponse aux questions communautaires (Yang *et al.*, 2016; Zhu *et al.*, 2019) ;
— *la classification* : classification de documents (Lai *et al.*, 2015; Yang *et al.*, 2016; Wang *et al.*, 2019), identification des paraphrases (Hu *et al.*, 2014; Yin *et al.*, 2015; Tan *et al.*, 2018)

Dans cette section, nous nous intéressons aux modèles neuronaux pour l'appariement sémantique de requêtes avec des documents. Ces derniers utilisent des architectures neuronales pour apprendre une fonction d'appariement : c'est l'apprentissage d'ordonnancement. Les modèles d'apprentissage d'ordonnancement ont pour objectif de classifier un élément (p. ex. une paire requête-document, tweet-POI) à partir d'un vecteur de caractéristiques $\vec{x} \in \mathbb{R}^n$. Le modèle d'ordonnancement $f : \vec{x} \mapsto \mathbb{R}$ est optimisé pour assigner un score d'appariement au vecteur $\vec{x}$ de telle sorte que, pour une requête donnée, les documents les plus pertinents soient mieux notés. L'apprentissage du modèle est dit « de bout-en-bout » (*end-to-end*) si les paramètres libres de f sont optimisés en une seule fois, et si le vecteur $\vec{x}$ consiste en des caractéristiques simples. Ce type de réseau de neurones peut apprendre conjointement les vecteurs de représentation et la similarité entre des textes donnés en entrée. Les différents modèles neuronaux proposés pour l'appariement de textes sont divisés en deux groupes principaux, à savoir les modèles *axés sur la représentation* et les modèles *axés sur l'interaction* (Guo *et al.*, 2016).

3.1 *Formulation unifiée des modèles d'ordonnancement*

Avant de décrire les différents groupes, nous donnons une formulation unifiée des modèles d'ordonnancement, adaptée de Guo *et al.* (2019), que nous utiliserons pour décrire les différents modèles d'appariement neuronaux. Nous considérons l'architecture décrite dans la Figure 3.15, où le modèle d'appariement se compose de deux parties : la partie *représentation* qui construit les représentations distri-

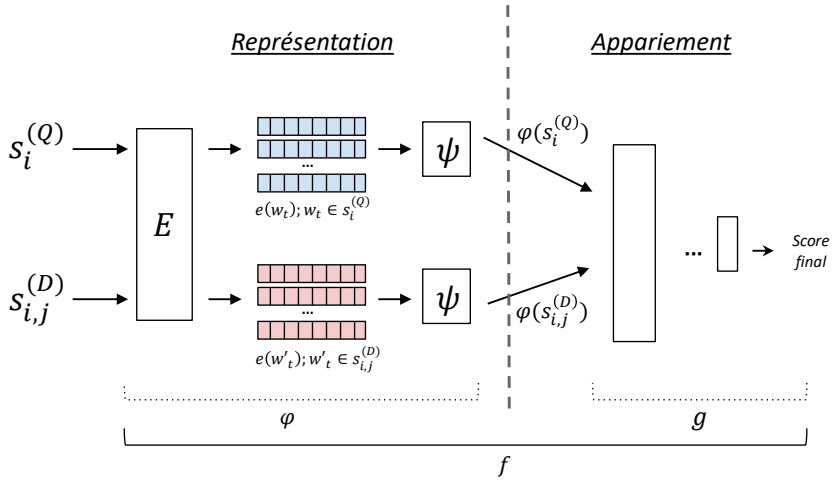


Figure 3.15 – Architecture générale décrivant un modèle neuronal unifié pour l'appariement de textes.

buées des séquences d'entrée, et la partie *appariement* qui compare les représentations construites dans la partie précédente.

Nous commençons par définir l'espace d'entrée d'un modèle neuronal pour l'appariement de textes, tel que $X = S^{(Q)} \cup S^{(D)}$, où $S^{(Q)}$ est l'espace des requêtes contenant la première séquence de la paire de séquences à apparier (p. ex. requête, question) ; $S^{(D)}$ est l'espace des documents contenant la deuxième séquence de la paire (p. ex. document, réponse). Dans l'espace d'entrée X, pour chaque séquence de requête $s_i^{(Q)} \in S^{(Q)}$, nous définissons $T_i = \left\{ s_{i,1}^{(D)}, \ldots, s_{i,n_i}^{(D)} \right\} \subseteq S^{(D)}$ un ensemble de n_i séquences. Soit $\mathbf{y}_i = \{y_{i,1}, \ldots, y_{i,n_i}\}$ un ensemble d'étiquettes associées à l'exemple d'entrée $s_i^{(Q)}$, de telle sorte que $y_{i,j}$ soit l'étiquette de pertinence correspondant à l'entrée $s_{i,j}^{(D)}$ de $s_i^{(Q)}$. L'objectif du modèle d'appariement neuronal est d'apprendre le modèle optimal f^* en minimisant un ensemble d'erreurs de prédiction. Ces erreurs sont calculées à l'aide d'une fonction de coût $\mathcal{L}$, mesurant la différence entre l'étiquette réelle $y_{i,j}$ et la valeur prédite $\widehat{y}_{i,j}$, à l'aide de l'équation suivante :

$$f^* = \arg\min \sum_i \sum_j \mathcal{L}\left(f; s_i^{(Q)}, s_{i,j}^{(D)}, \widehat{y}_{i,j}, y_{i,j}\right) \tag{3.28}$$

où l'étiquette de sortie $\widehat{y}_{i,j}$ est calculée par le modèle d'appariement f, qui peut être résumé par :

$$f = g\left(\varphi(s_i^{(Q)}), \varphi(s_{i,j}^{(D)}\right) = \widehat{y}_{i,j} \tag{3.29}$$

où φ est une fonction de représentation permettant d'extraire les caractéristiques d'un texte d'entrée, définie par :

$$\varphi(s) = \psi(e(w_1), \ldots, e(w_{|s|})) \tag{3.30}$$

avec ψ, une fonction qui combine les représentations distribuées des mots $e(w_t)$ d'une séquence d'entrée $s =< w_1, \ldots, w_{|s|} >$ et $e : \mathcal{V} \mapsto E$, une fonction de représentation qui associe chaque mot du vocabulaire à sa représentation distribuée. ψ peut être une fonction d'agrégation (Section 2.1.2.1) ou non (Section 2.1.2.2).

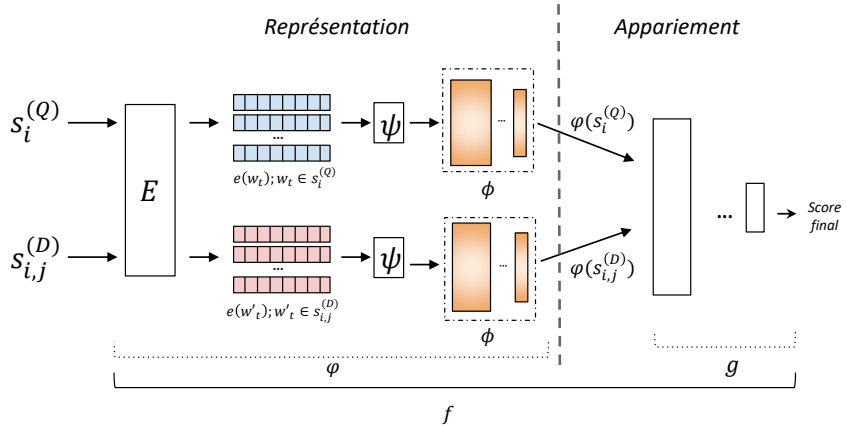
3.2 *Modèles axés sur la représentation*

La Figure 3.16a illustre l'architecture générale des modèles neuronaux axés sur la représentation. Ces derniers adoptent l'architecture d'un réseau de neurones de type siamois pour apprendre à apparier deux textes. Concrètement, les modèles neuronaux cherchent à apprendre une bonne représentation latente des séquences d'entrée, puis procèdent à un appariement entre les deux représentations. Dans cette approche, la fonction φ calcule les représentations distribuées $\varphi(s_i^{(Q)})$ et $\varphi(s_{i,j}^{(D)})$ correspondant aux séquences d'entrée $s_i^{(Q)}$ et $s_{i,j}^{(D)}$ comme définie par l'Équation 3.31 :

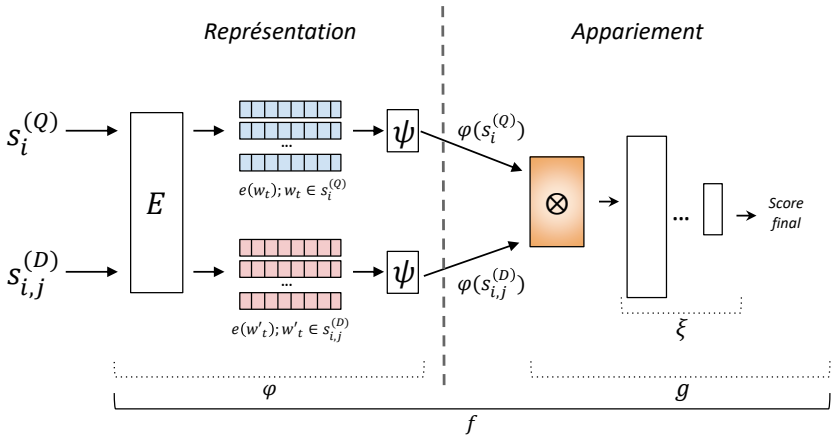
$$\varphi(s) = \phi(\psi(e(w_1), \ldots, e(w_{|s|}))) \tag{3.31}$$

où $e(w_t)$ est la représentation distribuée du mot w_t. Les représentations latentes $\phi(s_i^{(Q)})$ et $\phi(s_{i,j}^{(D)})$ sont ensuite envoyées à une fonction d'appariement g, qui peut être un réseau de neurones (p. ex. CNN, RNN) ou une simple fonction d'appariement (p. ex. similarité du cosinus). La fonction ϕ combine les caractéristiques de bas niveau calculées par la fonction ψ.

Huang *et al.* (2013) ont par exemple proposé *Deep Structured Semantic Model* (DSSM), un modèle sémantique profond pour la recherche ad-hoc sur le web. Le modèle DSSM se compose de deux branches symétriques dont les paramètres sont partagés pour les séquences d'entrée (requête et document). Les couches cachées du réseau utilisent des perceptrons multicouches successifs pour obtenir une représentation sémantique latente intermédiaire. S'appuyant sur l'architecture des modèles de représentation telle que décrite dans la Figure 3.16a, l'espace de représentation E du modèle DSSM contient un ensemble de vecteurs de représentation appris au niveau des lettres. En effet, compte tenu de la taille importante du vocabulaire, les auteurs ont proposé une méthode de hachage de mots qui transforme le vecteur de termes à haute dimension de la requête et du document, en un vecteur des trigrammes de lettres ayant une dimension réduite. Ce hachage permet également de traiter les termes hors du vocabulaire qui n'apparaissent pas



(a) Architecture axée sur la représentation.



(b) Architecture axée sur l'interaction.

Figure 3.16 – Architectures générales des modèles neuronaux montrant les différences entre les modèles axés sur la représentation (a) et les modèles axés sur l'interaction (b).

dans les données d'apprentissage. La fonction de représentation intermédiaire ψ correspond à une concaténation des différents vecteurs de représentation des n-grammes d'une séquence d'entrée. La fonction de représentation ϕ désigne quant à elle l'ensemble des couches de perceptrons multicouches successives utilisées pour calculer les représentations sémantiques des séquences d'entrée. Enfin, la couche d'appariement g correspond à la fonction de similarité du cosinus.

Tandis que le modèle DSSM utilise une succession de perceptrons multicouches pour apparier des documents, d'autres approches plus sophistiquées ont été explorées. Dans leurs travaux, Shen *et al.* (2014a,b) ont étendu le modèle DSSM en introduisant un réseau de neurones à convolution dans son architecture. Cette extension, appelée *Convolutional Latent Semantic Model* (CLSM) ou *Convolutional*

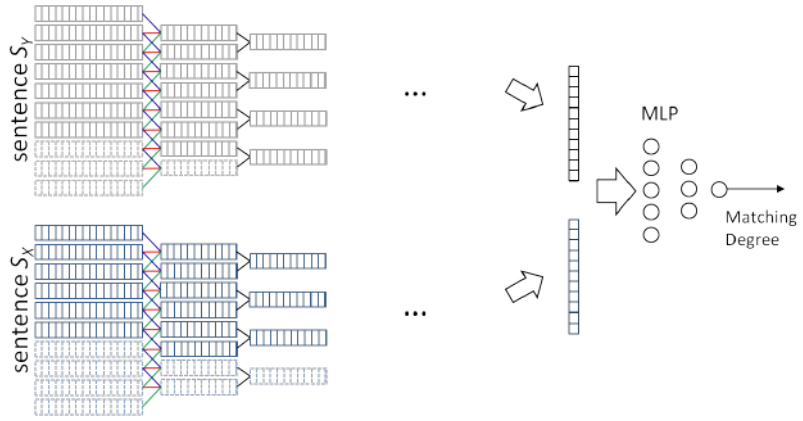


Figure 3.17 – Architecture du modèle d'appariement Arc-I (Hu *et al.*, 2014).

DSSM (C-DSSM), au lieu d'enchaîner de simples perceptrons multicouches dans la fonction de représentation ϕ, utilise une couche de convolution pour représenter les vecteurs initiaux calculées par ψ dans une nouvelle représentation s'appuyant sur la sémantique. La couche de convolution transforme les trigrammes des mots dans un vecteur latent. Le réseau intègre ensuite une couche de *max-pooling* pour extraire les caractéristiques locales saillantes afin de former un vecteur de représentation de taille fixe pour les séquences d'entrée. Dans d'autres travaux (Hu *et al.*, 2014; Severyn et Moschitti, 2015), la fonction de représentation ϕ est un CNN constitué de plusieurs couches convolutives et de plusieurs couches de *pooling*. L'approche Arc-I proposée par Hu *et al.* (2014), dont l'architecture est présentée dans la Figure 3.17, se compose de deux parties. La première est un modèle de représentation de phrases où la fonction ψ concatène les différents vecteurs de mots et la fonction ϕ calcule les représentations latentes des séquences à l'aide d'un CNN. La deuxième partie, qui effectue l'appariement, se compose d'un perceptron multicouche permettant de comparer les représentations des deux séquences d'entrée en calculant un score d'appariement. L'approche proposée par Severyn et Moschitti (2015) adopte la même architecture. Cependant, la deuxième partie du réseau de neurones, utilisée pour un problème de classification, reçoit, en plus des représentations latentes des séquences, un score de similarité intermédiaire calculé en utilisant le produit matriciel entre les vecteurs latents et une matrice de poids à entraîner, ainsi que quelques caractéristiques supplémentaires.

Enfin, d'autres approches ont utilisé des réseaux de neurones récurrents comme fonction de représentation ϕ. Par exemple, Sutskever *et al.* (2014) ont proposé un modèle qui apprend à prédire une phrase cible censée être similaire à une phrase donnée en entrée, représentée par les vecteurs distribuées des mots qui la composent, en utilisant l'architecture LSTM. Le modèle MV-LSTM de Wan *et al.* (2016a) utilise une couche bi-LSTM dans la fonction φ pour apprendre les représentations

à partir des positions des mots, pour chaque séquence d'entrée. Le modèle *Skip-thought* de Kiros *et al.* (2015) utilise quant à lui une couche GRU (Bahdanau *et al.*, 2015) pour apprendre la fonction de transformation séquence à vecteur φ, en prédisant les phrases avant et après la phrase actuellement traitée. Certains modèles s'appuyant sur la récurrence (Kamath *et al.*, 2019; Kim *et al.*, 2019) combinent des signaux de représentation calculés par les états cachés des réseaux récurrents avec des signaux extérieurs pour renforcer les représentations finales.

De plus, les couches récurrentes et convolutives peuvent être combinées dans une même fonction de représentation ϕ (Chen *et al.*, 2018; Zhou *et al.*, 2018), pour capturer les dépendances contextuelles avec les couches convolutives, en tenant compte de l'enchaînement des informations à l'aide de la couche récurrente.

3.3 *Modèles axés sur l'interaction*

Le deuxième groupe de modèles neuronaux pour l'appariement de textes sont les modèles axés sur l'interaction. Ces derniers essaient d'apprendre les différentes caractéristiques d'appariement étant données les représentations distribuées initiales des séquences d'entrée à apparier. Comme l'ont évoqué Guo *et al.* (2016), l'objectif de cette architecture est de construire les interactions entre la requête et le document assez tôt dans le modèle, pour extraire des signaux d'appariement de bas niveau. L'architecture générale des modèles d'interaction, illustrée dans la Figure 3.16b, se compose de deux parties. Une partie *représentation* commence par calculer les représentations des séquences d'entrée, en combinant les vecteurs de mots dans la fonction ψ. Dans la partie *appariement*, la couche $\otimes$ désigne une fonction qui extrait les caractéristiques des interactions au niveau des mots, et la fonction ξ fait référence aux couches suivantes qui calculent les caractéristiques de pertinence à des niveaux d'abstraction supérieurs. Ainsi, la fonction d'appariement g peut être définie comme :

$$g\left(\varphi(s_i^{(Q)}), \varphi(s_{i,j}^{(D)})\right) = \xi(M) \tag{3.32}$$

où M est une matrice d'interaction au niveau des mots calculée par l'Équation 3.33 et qui varie en fonction des différents modèles de l'état-de-l'art. ξ est une fonction d'interaction appliquée aux niveau des mots, comme illustrée dans la Figure 3.16b.

$$M = \otimes\left(\varphi(s_i^{(Q)}), \varphi(s_{i,j}^{(D)})\right) \tag{3.33}$$

avec $\otimes$, une fonction d'appariement (p. ex. similarité du cosinus).

Différents types de réseaux de neurones peuvent être utilisés pour apprendre la fonction d'interaction ξ (Figure 3.16b). Dans le cas d'un perceptron multicouche

(Guo *et al.*, 2016; Xiong *et al.*, 2017), l'objectif est d'apprendre les différentes interactions entre les caractéristiques d'appariement au niveau des mots. Dans le cas d'un réseau de neurones à convolution (Hu *et al.*, 2014; Pang *et al.*, 2016), l'objectif est d'apprendre des caractéristiques contextuelles. Ainsi, une fenêtre contextuelle glissante $c^{Q\times D}$ est définie et parcourt la matrice d'interaction M, correspondant à une combinaison de deux fenêtres glissantes, $c^{(Q)}$ et $c^{(D)}$, considérées simultanément dans les séquences $s_i^{(Q)}$ et $s_{i,j}^{(D)}$. Enfin, dans le cas d'un réseau de neurones récurrents (Fan *et al.*, 2018; Wan *et al.*, 2016b), l'objectif est d'apprendre des caractéristiques séquentielles et structurées. Quelle que soit la fonction ξ, les séquences d'entrée peuvent être représentées sous différentes formes, incluant des vecteurs classiques de mots (Mitra *et al.*, 2017), des plongements lexicaux pré-entraînés (Hu *et al.*, 2014) ou des représentations apprises en même temps que les paramètres du modèle (Pang *et al.*, 2016).

L'un des modèles neuronaux qui utilise un perceptron multicouche pour apprendre la fonction ξ est le modèle *DeepMatch* proposé par Lu et Li (2013). Dans leur approche, Lu et Li (2013) appliquent le perceptron sur la matrice d'interaction M, construite à partir des cooccurrences des mots. Le modèle *DeepMatch* considère une décision hiérarchique pour l'appariement à différents niveaux d'abstraction. Les décisions locales, qui saisissent l'interaction entre des mots sémantiquement liés, sont combinées à travers les différentes couches hiérarchiques du réseau de neurones, pour apprendre la décision d'appariement global. Pour entraîner leur modèle, les auteurs utilisent des triplets de séquences $(s_i^{(Q)}, s_{i,j}^{(D+)}, s_{i,j}^{(D-)})$, où $s_{i,j}^{(D)+}, s_{i,j}^{(D)-} \in T_i$, et $s_i^{(Q)}$ est plus similaire à $s_{i,j}^{(D)+}$ qu'à $s_{i,j}^{(D)-}$. L'objectif d'apprentissage est de maximiser la valeur similarité calculée par g (Équation 3.32) de la paire positive et de minimiser la similarité de la paire négative, en utilisant la fonction de coût de type *hinge-loss* suivante :

$$\mathcal{L}\left(s_i^{(Q)}, s_{i,j}^{(D+)}, s_{i,j}^{(D-)}\right) = \max\left(0, \varepsilon + g\left(\varphi(s_i^{(Q)}), \varphi(s_{i,j}^{(D)+})\right) - g\left(\varphi(s_i^{(Q)}), \varphi(s_{i,j}^{(D)-})\right)\right) \tag{3.34}$$

avec ε un paramètre qui contrôle la marge d'apprentissage.

Dans leurs travaux, Guo *et al.* (2016) ont proposé *Deep Relevance Matching Model* (DRMM), l'un des premiers modèles neuronaux à montrer une amélioration significative par rapport aux modèles de RI traditionnels sur des collections TREC en considérant le texte intégral. Les auteurs ont montré que les méthodes d'appariement neuronaux traditionnellement utilisés en TALN pour un appariement sémantique ne sont pas bien adaptés à la recherche ad-hoc, qui concerne l'appariement de pertinence. En effet, selon eux, il existe trois principales différences entre l'appariement sémantique et l'appariement de pertinence :

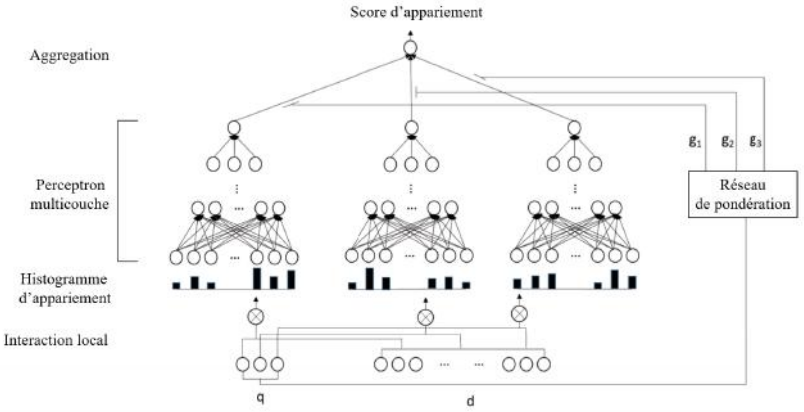


Figure 3.18 – Architecture du modèle d'appariement DRMM (Guo *et al.*, 2016).

— l'appariement sémantique recherche la similarité sémantique entre les termes, tandis que l'appariement de pertinence met davantage l'accent sur la correspondance exacte ;
— l'appariement sémantique s'intéresse souvent à la façon dont la composition et la grammaire déterminent le sens, tandis que dans l'appariement de pertinence, l'importance des termes de la requête est plus important que la grammaire car les requêtes sont généralement courtes et constituées de mots-clés ;
— l'appariement sémantique compare deux textes entiers dans leur intégralité, alors que l'appariement de pertinence ne compare que des parties de documents.

S'appuyant sur ces intuitions, les auteurs ont développé le modèle DRMM, dont l'architecture générale est présentée dans la Figure 3.18. Le réseau de neurones commence par calculer les interactions mot à mot entre les séquences d'entrée, en utilisant la fonction d'appariement $\otimes$ (Équation 3.33) qui calcule une similarité du cosinus. La matrice d'interaction M est ensuite transformée en des histogrammes d'interactions, qui représentent les niveaux de similarité entre les termes de la requête et les termes du document. Un histogramme d'appariement regroupe les interactions locales en fonction de la force du signal plutôt que de leur position dans les séquences. Comme pour le modèle d'appariement (Mitra *et al.*, 2016; Shen *et al.*, 2014b; Yan *et al.*, 2017) de Lu et Li (2013), un perceptron multicouche est appliqué sur la matrice d'interaction M pour capturer les représentations latentes des interactions. Dans le même esprit, Yang *et al.* (2016) ont proposé le modèle *Attention-Based Neural Matching Model* (ANMM) pour la tâche de question-réponse. Leur approche s'appuie sur un système de pondération partagée pour combiner différents signaux d'appariement et intègre l'apprentissage de l'importance des termes des questions à l'aide d'un réseau d'attention.

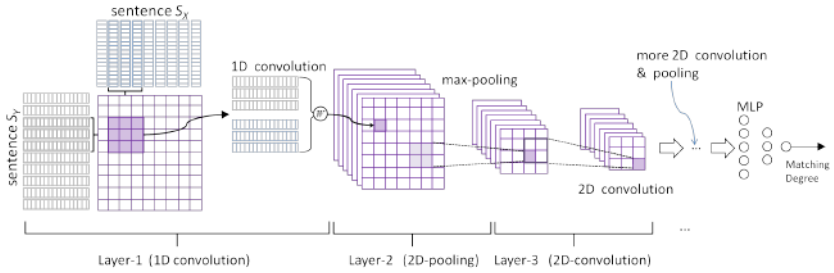


Figure 3.19 – Architecture du modèle d'appariement Arc-II (Hu *et al.*, 2014).

D'autres approches définissent la fonction d'interaction ξ en utilisant des réseaux de neurones à convolution, comme le modèle Arc-II de Hu *et al.* (2014), dont l'architecture est présentée dans la Figure 3.19. Ce dernier utilise une matrice d'interaction M construite à l'aide d'un opérateur matriciel (ici une multiplication) entre les représentations de la requête et du document d'entrée. L'intuition est de capturer les interactions entre les mots au niveau de l'espace latent plutôt que les cooccurrences dans le corpus, comme l'ont fait Lu et Li (2013). Un réseau de neurones à convolution est ensuite appliqué sur cette matrice pour apprendre des motifs d'appariement. Celui-ci permet de capturer et préserver l'ordre des caractéristiques locales dans la matrice d'interaction. Les auteurs suggèrent qu'une telle architecture peut capturer la composition successive des phrases ainsi qu'extraire et fusionner des motifs d'appariement entre les phrases.
Dans leurs travaux, Mitra *et al.* (2017) ont souligné l'importance de l'appariement lexical dans les modèles neuronaux pour la RI. Ils ont montré que les modèles axés sur la représentation ont tendance à ne pas être performants lorsqu'ils rencontrent des mots rares. Ils ont aussi argumenté que la recherche sur le web nécessite à la fois un appariement exact et inexact (c.-à-d. appariement sémantique). En effet, ce type de recherche implique l'utilisation de termes rares dans certaines requêtes, qui ne sont pas nécessairement utilisés dans la collection d'apprentissage des représentations latentes. À partir de ces constats, Mitra *et al.* (2017) ont proposé l'architecture Duet, qui s'appuie sur l'appariement lexical et sémantique. Ce modèle se compose de deux parties parallèles : un réseau pour l'appariement exact qui apprend les signaux d'appariement entre deux textes, et un réseau pour l'appariement sémantique qui s'appuie sur la représentation latente des textes. Le score d'appariement final est calculé en additionnant les scores obtenus par les deux sous-réseaux qui composent le modèle Duet.

Dans d'autres cas, la fonction d'appariement g utilise principalement un réseau de neurones récurrents. Par exemple, dans leurs travaux, Wan *et al.* (2016b) ont proposé le modèle Match-SRNN pour un appariement s'appuyant sur la position des mots dans les séquences d'entrée. Dans le modèle, compte tenu des séquences d'entrée $s_i^{(Q)}$ et $s_{i,j}^{(D)}$, une matrice d'appariement au niveau des mots est calcu-

lée à l'aide de la fonction $\otimes$ (Équation 3.33). L'interaction est calculée comme la combinaison du préfixe et du mot courant dans chaque séquence. La matrice d'interaction est envoyée à la fonction d'interaction g qui est composée d'un réseau de neurones récurrents suivi d'un perceptron multicouche, pour calculer le score final d'appariement.
Une autre architecture récurrente est proposée par Fan *et al.* (2018). Les auteurs ont mis en évidence que les principales limites des modèles neuronaux qui traitent des documents en entier sont qu'ils mettent en concurrence des documents longs et courts. Au contraire, les modèles s'appuyant sur les paragraphes (ou passages) exploitent des stratégies d'agrégation simplifiées de signaux locaux mais ne peuvent pas déterminer efficacement des motifs de pertinence complexes. Pour pallier ces limitations, les auteurs ont proposé le modèle *Hierarchical Neural Matching* (HiNT), qui se compose de deux composants superposés. La couche d'appariement locale utilise une architecture récurrente pour calculer un modèle d'appariement au niveau du passage, et la couche de décision globale utilise une deuxième architecture récurrente pour calculer les interactions entre les différents signaux locaux pour déterminer les caractéristiques de pertinence au niveau du document.

4 Discussion

Nous l'avons vu dans ce chapitre, l'introduction des représentations distribuées des mots, des séquences et des documents a impacté de nombreux travaux portés sur l'appariement de textes. Ces représentations ont amélioré les résultats lorsqu'elles sont utilisés dans des modèles classiques (Zheng et Callan, 2015; Guo *et al.*, 2016), mais leurs performances sont limitées. Certaines études (Iacobacci *et al.*, 2015) ont évalué l'impact des plongements lexicaux dans des tâches d'appariement de textes. Ces études ont montré que les plongements lexicaux ne permettent pas de résoudre les problèmes d'ambiguïté lexicale, telle que la polysémie, comme tous les sens d'un même mot sont représentés dans un seul vecteur, quel que soit leur contexte d'utilisation. Nous l'avons discuté précédemment, des premières approches (Xu *et al.*, 2014; Faruqui *et al.*, 2015), s'appuyant sur des ressources sémantiques externes, ont été proposées pour pallier ces problèmes.

Des modèles plus récents (MacAvaney *et al.*, 2019; Dai et Callan, 2019) considèrent des représentations contextuelles des mots, comme ELMo (Peters *et al.*, 2018) et BERT (Devlin *et al.*, 2019). Néanmoins, ces représentations sont construites à partir d'architectures spécifiques à une tâche qui incluent les représentations préentraînées comme des caractéristiques additionnelles (Peters *et al.*, 2018) ou qui requièrent un réglage très précis des hyperparamètres (Devlin *et al.*, 2019) dans le corpus cible pour accomplir la tâche en question. Les analyses réalisées par Ten-

ney *et al.* (2019) et Qiao *et al.* (2019) ont montré quelques limites liées à l'utilisation de ces représentations. Ils ont notamment montré que les représentations contextuelles surpassent les représentations classiques dans les tâches syntaxiques par rapport aux tâches sémantiques, ce qui suggère que ces représentations encodent davantage la syntaxe que la sémantique de haut niveau. De plus, bien que les représentations contextuelles de MacAvaney *et al.* (2019) ont permis d'améliorer significativement les performances des modèles d'ordonnancement, elles présentent un coût d'inférence considérable lorsqu'elles sont incorporées dans un modèle de recherche de documents. Ainsi, les plongements lexicaux traditionnels, tels que *GloVe* (Pennington *et al.*, 2014) ou *Word2Vec* (Mikolov *et al.*, 2013a,b) s'avèrent plus efficaces en terme de temps, que des représentations plus élaborées comme BERT (Devlin *et al.*, 2019) ou ELMo (Peters *et al.*, 2018).

Les représentations des séquences sont aussi limitées en terme de performance. L'exploitation de ces représentations avec les modèles classiques (Ai *et al.*, 2016; Mitra *et al.*, 2016; Nalisnick *et al.*, 2016) a parfois contribué à améliorer les résultats. Toutefois, les caractéristiques latentes des vecteurs de phrases ne sont pas explicites et ne renvoient pas à des informations concrètes sur l'ensemble des données, contrairement aux signaux TfIdf classiques, où chaque caractéristique se réfère à un aspect statistique significatif sur les mots et les documents de la collection. Ai *et al.* (2016) ont par exemple analysé l'utilité du modèle ParagraphVector (Le et Mikolov, 2014) dans des modèles classiques de RI. Ils ont montré que ce modèle de représentation de séquences n'est pas adapté pour représenter des documents longs, car il a tendance à surapprendre les documents courts, ce qui conduit à les privilégier lorsque les représentations sont utilisées dans un modèle de RI.

Les modèles détaillés dans ce chapitre se sont principalement focalisés sur l'apprentissage ou la correction de plongements lexicaux à partir de données contextuelles textuelles, c.-à-d. des mots situés autour d'un mot central, ou des mots issus de ressources externes. Hors, comme nous l'avons détaillé dans le Chapitre 2, lorsque nous traitons avec des géotextes, le contenu géographique est aussi vecteur d'information permettant contextualiser le contenu. C'est d'ailleurs un élément essentiel dans le cadre de développement d'applications de RIG comme nous l'avons vu dans la Section 2.2. Selon les régions géographiques, les mots peuvent véhiculer des informations différents, selon leur contexte local d'utilisation. Avec les différents modèles de l'état-de-l'art , ces spécificités locales ne peuvent être capturées efficacement, puisque d'une part toutes les variantes sémantiques d'un mot sont regroupées dans une seule et même représentation, et d'autre part, parce que les modèles ne tiennent pas compte du contexte géographique lors de l'apprentissage des représentations. De ce fait, l'utilisation de plongements lexicaux traditionnels dans des tâches de RIG conduit à perdre une information cruciale, se traduisant par des performances dégradées.

Concernant l'utilisation des réseaux de neurones pour l'appariement de géotextes, notre motivation est guidée par les enseignements issus des travaux que nous avons détaillés dans la Section 3 de ce chapitre. Ces derniers ont montré que les approches s'appuyant sur les réseaux de neurones sont très efficaces pour l'appariement de textes. Les architectures neuronales permettent de s'attaquer avec succès aux problèmes de rareté des données et d'inadéquation du vocabulaire, qui sont quelques-uns des problèmes fondamentaux rencontrés dans des tâches de géoréférencement de contenus non structurés (Section 2). Cependant, nous identifions deux enjeux majeurs : les données et les modèles. Les données requises dans les applications d'appariement de textes sont généralement : un ensemble de requêtes, un ensemble de documents et des jugements de pertinence explicites (p. ex. jugements humains) ou implicites (p. ex. clics). Ces trois éléments sont essentiels pour concevoir des modèles d'appariement performants. Il est aussi utile de distinguer les modèles qui se concentrent sur l'ordonnancement de documents longs et ceux conçus pour l'appariement de textes courts. Les challenges de ces deux approches sont différents, et les réseaux de neurones doivent être modélisés en conséquence (Cohen *et al.*, 2016). Le deuxième enjeu est lié à l'adaptabilité des modèles neuronaux. En effet, les structures des modèles profonds conçus pour l'appariement de textes sont souvent optimisées pour fonctionner sur des jeux de données spécifiques (Cohen *et al.*, 2016). De plus, les tâches de RI portent sur des textes de longueur variable, passant de séquences très courtes (quelques mots) à des documents longs. Ce point rend difficile l'adaptation directe d'un modèle conçu pour apparier des textes courts, à la recherche de documents ad-hoc. De fait, les modèles d'appariement diffèrent selon la tâche à accomplir.

Conscient des limites des modèles neuronaux, mais aussi des spécificités des tâches de RIG qui requièrent la prise en compte de l'aspect géographique, nous constatons que les modèles de l'état-de-l'art détaillés dans ce chapitre, bien qu'efficaces dans diverses tâches de RI, sont peu adaptés à notre problématique. Tout d'abord, d'après les analyses de Guo *et al.* (2016), les modèles axés sur l'interaction semblent plus performants sur des tâches de RI que les modèles axés sur la représentation, puisqu'ils s'appuient sur un appariement de pertinence. Cependant, les travaux de cette lignée se limitent à l'apprentissage de caractéristiques d'appariement local mot à mot. Hors, la tâche de géoréférencement de contenus, et plus généralement, les tâches de RIG nécessitent de tenir compte à la fois de l'information textuelle, mais aussi de l'information spatiale. De par leur architecture, les modèles neuronaux traditionnellement utilisés en RI ne permettent pas d'exploiter conjointement les régularités spatiales et textuelles des géotextes, et se concentrent uniquement sur les signaux de pertinences locales mot à mot. De fait, il convient proposer une architecture permettant de capturer conjointement les correspondances locales, mais aussi les signaux d'appariement qui exploitent les caractéristiques sémantiques et géographiques des géotextes.

5 Conclusion

Dans ce chapitre, nous avons commencé par introduire les concepts de base des réseaux de neurones et de l'apprentissage profond, en présentant quelques architectures populaires en RI. La disponibilité et l'accessibilité des données ont encouragé les chercheurs à développer de nouvelles applications.

Nous avons ensuite discuté différents modèles utilisés pour représenter des mots, des séquences, ou des objets géotextuels. Nous avons vu comment les différentes approches ont évolué afin de mieux représenter les textes et les géotextes. En particulier, les plongements lexicaux ont largement été utilisés pour tirer parti des informations de correspondance sémantique. Différents modèles de représentations de séquences, de mots et de documents ont ainsi été proposés. Tous ces modèles ont pour objectif de mieux représenter les informations sémantiques dans une séquence de mots. Des modèles plus spécifiques ont aussi été proposés pour représenter des géotextes, en tenant compte des caractéristiques spatiales et temporelles des géotextes dans le cadre de tâches de RIG.

Cependant, ces représentations ont montré des performances limitées lorsqu'elles sont utilisées avec des modèles d'appariement classiques. Des méthodes plus puissantes sont donc nécessaires pour construire des représentations textuelles permettant de mieux réaliser l'appariement. Ces modèles sont conçus pour tirer parti des capacités des réseaux de neurones à calculer et combiner des signaux de pertinence. Nous avons ensuite défini un modèle unifié d'appariement neuronal que nous avons utilisé pour décrire les différentes architectures adoptées par les modèles de l'état-de-l'art. Enfin, nous avons présenté différentes architectures populaires pour l'appariement de textes.

Dans cette thèse, nous nous intéressons particulièrement à la prédiction de l'emplacement des documents, et plus spécifiquement à l'appariement et la représentation d'objets géotextuels. Dans cette optique, nous présentons dans ce manuscrit deux contributions principales :

— deux méthodes de correction a posteriori de plongements lexicaux augmentés par des ressources géographiques. Ces méthodes exploitent les répartitions spatiales des mots pour identifier des relations sémantiques locales entre mots ainsi que des sens locaux de mots ;
— un modèle d'appariement d'objets géotextuels s'appuyant sur une architecture neuronale. Ce modèle exploite les représentations sémantiques latentes des tweets et des POIs via des interactions locales mot à mot, ainsi que des caractéristiques sémantiques et spatiales pour résoudre la tâche de prédiction sémantique de l'emplacement ;

En l'état courant de nos connaissances, aucune approche neuronale n'a été développée pour résoudre la tâche de prédiction sémantique de l'emplacement. De plus, notre travail est la première tentative de correction spatiale de plongements lexicaux et de discrimination des sens locaux des mots à l'aide de répartitions spatiales.

Partie II

CONTRIBUTIONS

4

RÉGULARISATION SPATIALE DE PLONGEMENTS LEXICAUX

Introduction

Dans le Chapitre 3, nous avons introduit un état-de-l'art des méthodes d'apprentissage de représentations distribuées de mots et de documents ainsi que leurs principales limites, avec notamment la représentation des polysèmes. En effet, tous les sens d'un mot sont représentés dans un seul vecteur. Les approches détaillées dans la Section 2.1.3 pallient ce problème en enrichissant les représentations avec des ressources sémantiques externes issues par exemple de WordNet ou de Wikipedia. Ces dernières permettent ainsi de modéliser, dans les plongements lexicaux, la sémantique relationnelle telle que la synonymie, l'hypernonymie ou encore la polysémie. Toutefois, aucune de ces approches ne tient compte des contextes géographiques des mots, un facteur pourtant essentiel en RIG, comme nous l'avons vu dans le Chapitre 2. Néanmoins, quelques travaux se sont intéressés à la modélisation cartographique des mots, soit indirectement en calculant des plongements lexicaux pour chaque région (p. ex. par pays), soit directement lors de l'apprentissage des représentations en utilisant un contexte géographique.

Nous présentons dans ce chapitre notre contribution pour l'apprentissage de plongements lexicaux augmentés par des connaissances issues des répartitions spatiales des mots. Notre contribution se distingue des approches de régularisation de l'état-de-l'art (Section 2.1.3 du Chapitre 3) par la prise en compte de caractéristiques spatiales pour l'élaboration des représentations et de la discrimination des sens locaux des mots. Notre objectif est donc de révéler d'éventuelles relations sémantiques locales entre les mots ainsi que la multiplicité de leurs sens. Pour cela, nous proposons une stratégie de régularisation a posteriori permettant de corriger les plongements lexicaux à l'aide de connaissances spatiales. Nous menons également plusieurs expérimentations pour analyser et évaluer l'hypothèse proposée ainsi que la qualité des représentations corrigées.

L'organisation de ce chapitre est la suivante. Nous introduisons en Section 1 les principales motivations de notre contribution ainsi que notre hypothèse de re-

cherche. Nous présentons ensuite dans la Section 2 la problématique et les définitions qui vont guider nos travaux de recherche. Dans la Section 3, nous détaillons la méthodologie proposée pour la régularisation a posteriori des plongements lexicaux à l'aide de connaissances spatiales. L'évaluation expérimentale est présentée dans la Section 4. Enfin, nous concluons ce chapitre dans la Section 5.

1 Contexte et motivations

Au cours des dernières décennies, la création de contenus géoréférencés, également connus sous le nom de géotextes, a fortement augmenté. L'interaction entre le texte et la localisation géographique soulève d'importantes questions de recherche sous-jacentes à l'appariement d'objets géotextuels, qui est l'étape clef de diverses tâches de RIG telles que l'interrogation de géotextes (Zhang *et al.*, 2014), la mention de lieux (Han *et al.*, 2018) ou la prédiction sémantique de l'emplacement (Zhao *et al.*, 2016). Les solutions existantes reposent principalement sur l'utilisation d'une combinaison d'éléments textuels et spatiaux pour construire des représentations efficaces d'objets (Zhang *et al.*, 2014) ou pour définir des modèles efficaces d'appariement objet-objet (Dalvi *et al.*, 2009a).

En parallèle, de nombreux travaux de recherche se sont intéressés à la sémantique distributionnelle, via des plongements lexicaux, pour résoudre des tâches de RI traditionnelles, notamment depuis l'émergence d'algorithmes populaires tels que *Word2Vec* (Mikolov *et al.*, 2013a). La sémantique distributionnelle s'appuie sur la théorie selon laquelle des mots sémantiquement similaires se retrouvent dans les mêmes contextes textuels. Cependant, son utilisation présente quelques inconvénients. En effet, les plongements lexicaux ne permettent pas de lever le problème de polysémie puisque les différents sens d'un même mot sont regroupés en un seul vecteur (Iacobacci *et al.*, 2015). Par ailleurs, des travaux ont montré que les mots suivent traditionnellement des schémas géographiques d'utilisation (Eisenstein *et al.*, 2010; Han *et al.*, 2012; Laere *et al.*, 2014; Ozdikis *et al.*, 2019). Par exemple, il est plus probable d'entendre des conversations sur la nourriture dans et autour d'un restaurant qu'à proximité d'une salle de cinéma. Toutefois, cette spécificité géographique n'est pas prise en compte lors de la construction des plongements lexicaux. Quelques travaux se sont dont penchés sur l'impact du contexte géographique sur la sémantique des mots (Cocos et Callison-Burch, 2017) et sur le problème de glissement lexical entre les régions (Zhang *et al.*, 2017b). Ces travaux nous conduisent à l'observation suivante :

Observation 1. Les sens de certains mots diffèrent selon les régions où ils sont utilisés. Par exemple, le mot *dinosaure* peut se référer à un animal préhistorique ou à une chaîne de restaurants dans le contexte local de la ville de New York.

Exemple 4.1 (*Différents sens locaux du mot* « football »)**.** Prenons l'exemple du mot *football*. En fonction de la région dans laquelle le mot est utilisé, par exemple, en Europe, aux États-Unis ou en Australie, il ne désigne pas le même sport. En Europe, le terme *football* désigne un sport qui se joue au pied, avec un ballon rond. Dans son contexte local, nous retrouvons les termes tels que « *FIFA* », « *penalty* », ou « *corner* ». Aux États-Unis, le terme *football* désigne un sport qui se joue à la main avec un ballon ovale. Son contexte local diffère du précédent, puisqu'il contient des termes tels que « *touchdown* », « *quaterback* » ou « *shoulder pad* ». Enfin, concernant le terme *football* en Australie, il désigne un sport se jouant avec un ballon ovale, dans un terrain ovale composé de quatre poteaux. Son contexte local est composé des termes « *handpass* », « *drop punt* » et « *torpedo* ».

(a) Football (Europe) (b) Football (USA) (c) Football (Australie)

Figure 4.1 – Illustration des différents sens locaux du mot « *football* ».

Comme corollaire à cette observation, nous posons l'hypothèse suivante :

Hypothèse 1. Un mot peut véhiculer des sens locaux différents selon la zone géographique où il est spatialement dense. En pratique, cela se traduirait par des zones particulièrement denses dans la répartition spatiale d'un mot.

L'objectif de notre première contribution est d'améliorer la représentation des géotextes afin de les intégrer dans des tâches de RIG. De ce fait, nous nous intéressons à l'apprentissage de représentations de mots, éléments essentiels à la construction des représentations de géotextes. Notre proposition se distingue des approches de régularisation a posteriori de l'état-de-l'art par la prise en compte du contexte spatial pour corriger les plongements lexicaux et de la discrimination des sens locaux des mots à partir de répartitions spatiales. Nous présentons dans ce chapitre deux méthodes pour la détection des sens locaux des mots, l'une via une méthode de *clustering*, et l'autre via une méthode probabiliste, ainsi qu'une méthode de régularisation a posteriori (ou *retrofitting*) comme moyen d'affiner des représentations de mots préalablement apprises. De plus, nous menons une évaluation expérimentale approfondie et des analyses qualitatives afin de valider empiriquement notre hypothèse de recherche et d'évaluer l'efficacité des plongements lexicaux spatiaux que nous proposons d'intégrer dans une tâche de similarité d'objets géotextuels et dans une tâche de prédiction sémantique de l'emplacement.

2 Problématiques et définitions

Notre contribution repose sur la régularisation de plongements lexicaux à partir de connaissances spatiales. Dans la suite, nous commençons par détailler les concepts et définitions qui sont exploitées dans ce manuscrit. Nous définissons ensuite formellement le principe de régularisation spatiale.

2.1 *Concepts et définitions*

2.1.1 *Objets géotextuels*

Définition 4.1 (Géotexte). Un géotexte o est un objet textuel géotaggé. Le géotag est représenté par une paire de coordonnées (*latitude, longitude*) se référant à un emplacement géographique l, noté $o.l$. Chaque objet o est décrit par un ensemble d'attributs textuels, valorisés ou non selon la nature de l'objet, incluant par exemple un nom ou une description. Nous adoptons une représentation vectorielle de l'objet o établie sur la concaténation de ses attributs textuels $o = [w_1^{(o)}, \ldots, w_n^{(o)}]$, où chaque mot $w_i^{(o)}$ est issu d'un vocabulaire $\mathcal{V}$.

Dans le cadre de nos travaux, nous utilisons deux types de géotextes, les *points d'intérêts* (POI) et les *tweets géotaggés*, que nous définissons par la suite.

Définition 4.2 (Point d'intérêt, POI). Un point d'intérêt (POI) p est un géotexte physique qui référence un emplacement spécifique que les gens peuvent trouver intéressant et/ou utile. Un POI peut être une attraction, un hôtel, un restaurant, un distributeur de billets, un magasin, etc. Comme tout géotexte, un POI est décrit par ses coordonnées géographiques qui référencent un emplacement l noté $p.l$, et possède un ensemble d'attributs textuels incluant son nom $p.nom$, une description $p.desc$ et des commentaires/critiques d'utilisateurs $p.rw$. La représentation vectorielle d'un POI est définie par $p = [w_1^{(p)}, \ldots, w_m^{(p)}]$. L'ensemble $\mathcal{P} = \{p_1, p_2, p_3, \ldots\}$ constitue une base de données de POIs. $\mathcal{P}_l^r = \{p \in \mathcal{P} \mid dist(p.l, l) < d\}$ est l'ensemble des POIs situés dans un rayon r de l'emplacement l.

Exemple 4.2 (*Point d'intérêt, POI*)**.** La Figure 4.2 présente un POI décrivant la *Tour Eiffel* de Paris, tel qu'il apparaît sur le réseau social Foursquare [1] (Figure 4.2a) et sous sa forme structurée (Figure 4.2b). À partir des attributs textuels de l'objet (nom, description et commentaires des utilisateurs), nous obtenons sa représentation vectorielle $p = [\text{'}tour\text{'}, \text{'}eiffel\text{'}, \ldots \text{'}exposition\text{'}, \text{'}universelle\text{'}, \ldots, \text{'}moins\text{'}, \text{'}chargée\text{'}]$.

1. https://fr.foursquare.com/v/tour-eiffel/51a2445e5019c80b56934c75

(a) Représentation visuelle.

```
{ "nom" : "Tour Eiffel",
  "des" : "Dominant Paris du haut de ses 324
       mètres, la Tour Eiffel fut construite
       de 1887 à 1889 pour l'exposition
       universelle de 1889, Aujourd'hui, elle
       est le symbole de la France et le
       monument le plus visité au monde.",
  "l" : [2.2944, 48.8583],
  "cr" : ["L'entrée EST de la tour Eiffel se
          révèle souvent moins chargée."]
}
```

(b) Représentation structurée.

Figure 4.2 – Exemple d'un POI issu de Foursquare.

Définition 4.3 (**Tweet géotaggé**)**.** Un tweet géotaggé t est un géotexte qui fait référence à un message publié sur le réseau social Twitter contenant l'emplacement géographique de l'utilisateur lorsque le tweet a été posté. Un tweet est décrit par ses coordonnées géographiques notées $t.l$ et un court texte (280 caractères) dont la représentation vectorielle est notée $t = [w_1^{(t)}, \ldots, w_n^{(t)}]$. Un tweet géotaggé peut être lié à un POI s'il cible et/ou mentionne un POI spécifique. L'ensemble $\mathcal{T} = \{t_1, t_2, t_3, \ldots\}$ constitue une base de données de tweets.

Exemple 4.3 (*Tweet géotaggé*)**.** La Figure 4.3 présente un tweet géotaggé tel qu'il apparaît sur le réseau social Twitter (Figure 4.3a) et sous sa forme structurée (Figure 4.3b). À partir de l'attribut textuel de l'objet, nous obtenons sa représentation vectorielle $t = [\text{'toute'}, \text{'première'}, \ldots \text{'juste'}, \text{'ici'}]$.

(a) Représentation visuelle.

```
{ "texte" : "Toute première sortie du guépard
          au ZooParc, aujourd'hui ! On vous
          en dit plus juste ici",
  "l" : [1.3473, -47.2500]
}
```

(b) Représentation structurée.

Figure 4.3 – Exemple d'un tweet géotaggé.

2.1.2 *Distance géographique et répartition spatiale*

Définition 4.4 (Distance géographique entre géotextes). La distance d entre deux géotextes o_i et o_j, est la distance géographique entre les emplacements $o_i.l$ et $o_j.l$. Formellement, $d(o_i, o_j) = dist(o_i.l, o_j.l)$. De même, la distance géographique entre un géotexte o_i et un emplacement géographique l quelconque est notée $d(o_i, l) = dist(o_i.l, l)$.

Définition 4.5 (Répartition spatiale d'un mot). L'ensemble $O_i = \{o_{i,1}, o_{i,2}, \ldots\}$, associé au mot w_i, correspond à l'ensemble des géotextes $o_{i,k}$ citant le mot w_i. La répartition spatiale d'un mot w_i, notée S_i, est donc représentée par l'ensemble des emplacements $o_{i,k}.l$ des objets $o_{i,k} \in O_i$. Formellement, $S_i = \{o_{i,1}.l, o_{i,2}.l, \ldots\}$. Un exemple de répartition spatiale d'un mot est illustré dans la Figure 4.4. Chaque point rouge représente l'emplacement géographique d'une occurrence du mot w_i. Tous les emplacements des occurrences de w_i (situés à l'intérieur du cadre gris) constituent la répartition spatiale du mot w_i.

Figure 4.4 – Illustration de la répartition spatiale d'un mot et de son centre géographique.

Définition 4.6 (Centre géographique d'un mot). Soit S_i la répartition spatiale du mot w_i. Le centre géographique $\mathcal{B}$ du mot w_i, noté $\mathcal{B}(w_i)$ ou plus simplement $\mathcal{B}_i$, est l'emplacement médian géographique de l'ensemble des emplacements contenus dans S_i. Le centre géographique $\mathcal{B}_i$ est calculé comme suit :

1. pour chaque emplacement $o_{i,j}.l \in S_i$, les coordonnées (φ_j, λ_j), exprimées en degrés, sont converties en radians ;

$$\varphi'_j = \varphi_j \times \frac{\pi}{180}; \qquad \lambda'_j = \lambda_j \times \frac{\pi}{180} \tag{4.1}$$

2. les coordonnées sont ensuite converties dans un système de coordonnées cartésiennes ;

$$x_j = \cos(\lambda'_j) \times \cos(\varphi'_j); \quad y_j = \cos(\lambda'_j) \times \sin(\varphi'_j); \quad z_j = \sin(\lambda'_j) \tag{4.2}$$

3. nous calculons la moyenne des coordonnées x, y et z ;

$$X = \frac{1}{|S_i|}\sum_{j=0}^{|S_i|} x_j; \quad Y = \frac{1}{|S_i|}\sum_{j=0}^{|S_i|} y_j; \quad Z = \frac{1}{|S_i|}\sum_{j=0}^{|S_i|} z_j \tag{4.3}$$

4. les coordonnées moyennes X, Y et Z sont converties en latitude φ et longitude λ (en degrés) ;

$$\varphi = \text{atan2}(Z, \sqrt{X^2 + Y^2}) \times \frac{180}{\pi}; \qquad \lambda = \text{atan2}(Y, X) \times \frac{180}{\pi} \tag{4.4}$$

Reprenons la répartition spatiale du mot w_i illustrée dans la Figure 4.4. En utilisant l'emplacement géographique de chaque occurrence (points rouges), nous pouvons déterminer le centre géographique du mot w_i. Celui-ci est matérialisé par le repère rouge.

Définition 4.7 (Distance géographique entre mots). La distance d entre les mots w_i et w_j qualifie la distance géographique entre leur répartition spatiale S_i et S_j. La distance entre les mots w_i et w_j, notée $d(w_i, w_j)$ est telle que :

$$d(w_i, w_j) = dist(\Phi(S_i), \Phi(S_j)) \tag{4.5}$$

où $\Phi(S_i)$ est une fonction qui agrège la répartition spatiale de w_i en une localisation. Par la suite, $\Phi(S_i)$ calculera le centre géographique $\mathcal{B}_i$ de S_i. La distance géographique entre un géotexte o_j et un mot w_i est quant à elle définie par :

$$d(w_i, o_j) = dist(\Phi(S_i), o_j.l) = dist(\mathcal{B}_i, o_j.l) \tag{4.6}$$

2.2 *Définition du problème*

À partir de l'Observation 1 et de l'Hypothèse 1 formulées dans la Section 1, nous définissons la problématique abordée dans cette contribution. Nous supposons que les signaux issus des répartitions spatiales des mots pourraient contribuer à la construction des plongements lexicaux. Pour cela, nous proposons d'injecter la ressource spatiale dans le processus d'apprentissage des représentations (Faruqui *et al.*, 2015; Vulic et Mrksic, 2018). Cependant, les plongements lexicaux, même corrigés ne permettent pas de distinguer complètement les différents sens d'un même mot, puisqu'ils sont regroupés en un seul vecteur (Iacobacci *et al.*, 2015; Mancini *et al.*, 2017). En d'autres termes, à chaque mot, ne correspond qu'une seule représentation, c.-à-d. un seul vecteur. Ainsi, en tenant compte de l'Hypothèse 1, nous souhaitons construire, pour chaque mot, un ensemble de représentations basées sur ses statistiques d'occurrence au sein des objets géotextuels associés.

Formellement, soit un ensemble de représentations distribuées de mots $\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{V}|}\}$, où $\mathbf{w}_i \in \mathbb{R}^k$ est le vecteur k-dimensionnel construit pour un

mot cible $w_i \in \mathcal{V}$ en utilisant un modèle de langue neuronal standard, p. ex. le modèle *Skip-Gram* de Mikolov *et al.* (2013b). Notre objectif est de déterminer pour chaque mot w_i, l'ensemble des représentations des mots spatiaux associés $\mathbf{W}_i^s = \{\mathbf{w}_{i,1}^s, \ldots, \mathbf{w}_{i,j}^s, \ldots, \mathbf{w}_{i,n_i}^s\}$. Chaque vecteur de mot spatial $\mathbf{w}_{i,j}^s$, issu d'une représentation distribuée $\mathbf{w}_i$ initiale, désigne la représentation distributionnelle localisée du mot w_i sur une zone spatiale dense, et n_i est le nombre de sens localisés distincts du mot w_i provenant de sa répartition spatiale sur les géotextes O_i auquel il appartient (Définition 4.5).

Les problématiques inhérentes à notre contribution sont donc *(1)* de définir le nombre de sens localisés n_i pour chaque mot w_i, et *(2)* d'injecter les connaissances spatiales dans les représentations distribuées des mots. Dans les sections suivantes, nous proposons des solutions pour résoudre ces deux problématiques.

3 Apprentissage a posteriori des plongements lexicaux spatiaux

Nous présentons dans cette section notre contribution permettant de construire les représentations distribuées des mots augmentées par des ressources spatiales. Un aperçu général de la solution est d'abord introduit dans la Section 3.1. Nous détaillons ensuite dans la Section 3.2 les différentes méthodes adoptées pour déterminer les différents sens locaux des mots. Enfin, nous développons dans la Section 3.3 l'algorithme mis en œuvre pour la régularisation a posteriori des plongements lexicaux.

3.1 *Aperçu général de la solution*

Nous proposons dans cette contribution une nouvelle méthode de régularisation a posteriori permettant de construire des plongements lexicaux spatiaux. Une vue d'ensemble de notre méthode est présentée par l'Algorithme 1.

L'apprentissage des plongements lexicaux spatiaux se décompose en deux parties principales. La première partie de l'algorithme, dénommée *détection des sens locaux* (lignes 1 à 4), consiste à identifier les « empreintes spatiales » de chacun des mots, c.-à-d. les zones spatiales où ils sont denses. Pour cela, nous commençons par calculer les répartitions spatiales S_i des mots du vocabulaire $\mathcal{V}$ à l'aide de la fonction `ExtractionObjets` (ligne 3), puis identifions leurs différents sens locaux à l'aide de la fonction `PartitionnementSpatial` (ligne 4). Dans le cadre de cette

contribution, nous envisageons deux méthodes pour identifier les sens locaux : *(1)* l'une via une méthode de *clustering*, en exploitant l'algorithme des k-moyennes (MacQueen *et al.*, 1967) ; *(2)* et l'autre via une méthode probabiliste, en utilisant la méthode de Parzen-Rosenblatt (Parzen, 1962). Cette première partie est décrite dans la Section 3.2.
La deuxième partie de l'algorithme, dénommée *plongements lexicaux spatiaux* (lignes 5 à 12) permet de corriger les plongements lexicaux préalablement construits en tenant compte des répartitions spatiales des mots. Pour affiner les représentations des mots en tenant compte de leur contexte spatial, nous proposons d'utiliser les représentations des mots proches spatialement, déterminés par la fonction `Voisins` (ligne 9), ainsi que les représentations des mots distants spatialement, déterminés quant à eux par la fonction `Distants` (ligne 10). L'étape de régularisation, opérée par la fonction `Régularisation` (ligne 11), s'effectue en tenant compte des mots contenus dans les deux ensembles. Cette deuxième partie est décrite dans la Section 3.3.

Algorithme 1 : Construction des plongements lexicaux spatiaux.

Entrées : Vocabulaire $\mathcal{V}$; Ensemble des représentations distribuées de mots $\mathbf{W} = \{\mathbf{w}_1, \ldots ; \mathbf{w}_{|\mathcal{V}|}\}$; Ensemble des objets géotextuels O
Sorties : Ensemble des représentations distribuées de mots $\mathbf{W}^s = \{\mathbf{w}^s_{1,1}, \ldots, \mathbf{w}^s_{1,n_1}, \ldots, \mathbf{w}^s_{|\mathcal{V}|,1}, \ldots, \mathbf{w}^s_{|\mathcal{V}|,n_k}\}$

1 `# Détection des sens locaux (Section 3.2)`
2 **pour** $i \in \{1, .., |\mathcal{V}|\}$ **faire**
3 $\quad S_i$ = `ExtractionObjets`(w_i, O)
4 $\quad$ `PartitionnementSpatial`(S_i)
5 `# Plongements lexicaux spatiaux (Section 3.3)`
6 **répéter**
7 $\quad$ **pour** $i \in \{1, .., |\mathcal{V}|\}$ **faire**
8 $\quad\quad$ **pour** $j \in \{1, .., n_i\}$ **faire**
9 $\quad\quad\quad W^+_{i,j}$ = `Voisins`$(w^s_{i,j})$
10 $\quad\quad\quad W^-_{i,j}$ = `Distants`$(w^s_{i,j})$
11 $\quad\quad\quad \mathbf{w}^s_{ij}$ = `Régularisation`$(\mathbf{w_i}, W^+_{i,j}, W^-_{i,j})$
12 **jusqu'à** *Convergence*;

3.2 *Détection des sens locaux des mots*

Notre objectif est de construire des plongements lexicaux augmentés par la connaissance spatiale. Néanmoins, avant d'obtenir ces nouvelles représentations, nous souhaitons déterminer les différentes spécificités locales des mots, comme

évoqué dans l'Hypothèse 1. Pour cela, nous commençons par déterminer les répartitions spatiales S_i des mots du vocabulaire $\mathcal{V}$, à l'aide de la fonction `PartitionnementSpatial`, que nous spécifions à l'aide de la Définition 4.5. Nous proposons ensuite de partitionner les répartitions spatiales S_i pour déterminer leurs « empreintes spatiales », c.-à-d. les zones où ils sont denses. Nous envisageons deux méthodes : l'une via un partitionnement spatial des données à l'aide d'une méthode de *clustering* (algorithme des k-moyennes) que nous décrivons dans la Section 3.2.1 ; l'autre via une méthode probabiliste (méthode de Parzen-Rosenblatt), que nous détaillons dans la Section 3.2.2.

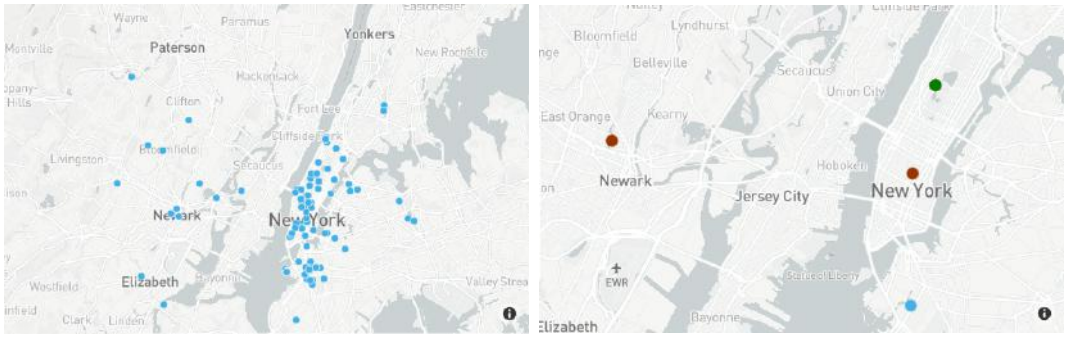
3.2.1 *Méthode de clustering : algorithme des* k*-moyennes*

Dans cette première méthode, nous proposons d'utiliser une méthode de *clustering*. Pour chaque mot w_i du vocabulaire $\mathcal{V}$, nous commençons par déterminer sa répartition spatiale S_i (Algorithme 1, ligne 3). Pour identifier les zones spatiales denses du mot w_i, nous effectuons un partitionnement spatial à l'aide de la méthode des k-moyennes (MacQueen *et al.*, 1967) (Algorithme 1, ligne 4). Formellement, étant donné un ensemble de points $o_{i,1}.l, o_{i,2}.l, \cdots \in S_i$, nous cherchons à partitionner les points en n_i ensembles $E_i = \{E_{i,1}, E_{i,2}, \ldots, E_{i,n_i}\}$ en minimisant la distance entre les points à l'intérieur de chaque partition :

$$\underset{E_i}{arg\,min} \sum_{j=1}^{n_i} \sum_{o_{i,l} \in E_{i,j}} ||o_{i,l} - \mathcal{B}_{i,j}||^2 \tag{4.7}$$

où $\mathcal{B}_{i,j}$ est le centre géographique (ou barycentre) calculé pour l'ensemble $E_{i,j}$ (Définition 4.6). Les n_i groupes spatiaux sont représentés par leurs barycentres $\mathcal{B}_i = \{\mathcal{B}_{i,1}, \cdots, \mathcal{B}_{i,n_i}\}$, où $\mathcal{B}_{i,j}$ est le j-ième barycentre du mot w_i et n_i le nombre optimal de groupes pour le mot w_i. Chaque barycentre $\mathcal{B}_{i,j}$ peut être vu comme un représentant spatial de la zone qui donne lieu à un sens local du mot w_i, qui sera ensuite représenté par le vecteur distribué $\mathbf{w}^s_{i,j}$, que nous détaillons dans la Section 3.3.

Exemple 4.4 (*Partitionnement spatial en* k*-moyennes du mot dinosaure*)**.** Un exemple de partitionnement spatial est présenté en Figure 4.5 pour le mot *dinosaure* dans la ville de New York et ses environs. Chaque point sur la Figure 4.5a correspond à une occurrence du mot *dinosaure*. L'application de l'algorithme de partitionnement nous permet d'obtenir quatre barycentres, chacun révélant une spécificité locale du mot *dinosaure*, comme le révèle Figure 4.5b. Les points marrons sont plutôt associés à une chaîne de restauration populaire à New York, *Dinosaur Bar-B-Que*, qui possède notamment des établissements dans le quartier de Manhattan et la ville de Newark, tandis que le point vert est plutôt associé au *American Museum of Natural History*, un musée consacré, entre autres, aux dinosaures.



(a) Répartition spatiale. (b) Barycentres des sens locaux.

Figure 4.5 – Exemple du partitionnement en k-moyennes appliqué au mot *dinosaure* dans la région de New York.

Dans cette méthode, le nombre de groupes spatiaux n_i considérés pour chaque mot, c.-à-d. le nombre de sens locaux, dépend d'une valeur fixée. Cette dernière peut être identique pour tous les mots ou différente. Nous pensons qu'utiliser une valeur de n_i identique pour tous les mots serait fortement discutable. En effet, certains mots du vocabulaire ne présentent raisonnablement qu'un seul sens, comme les mots vides par exemple, tandis que d'autres peuvent présenter une multitude de spécificités locales, comme le mot *football*. Ainsi, nous proposons de déterminer, pour chaque mot w_i, le nombre n_i optimal à l'aide du coefficient de silhouette (Rousseeuw, 1987). La qualité de cette méthode repose donc de façon critique sur le nombre de groupes spatiaux à considérer pour chaque mot. Cette limite nous conduit à proposer une autre méthode, où le nombre de sens locaux ne sera pas fixé manuellement, mais automatiquement à l'aide de mesures statistiques.

3.2.2 *Méthode probabiliste : méthode de Parzen-Rosenblatt*

Dans cette variante, nous adoptons une méthode probabiliste pour déterminer les sens locaux des mots. Dans la littérature, de précédents travaux ont déjà utilisé les contextes spatiaux des mots pour déterminer ceux qui sont spécifiques à une région ou pertinent dans un contexte spatial donné (Laere *et al.*, 2014; Ozdikis *et al.*, 2019). Ils reposent principalement sur l'utilisation de la méthode de Parzen-Rosenblatt, aussi appelée méthode d'estimation par noyau (KDE) (Parzen, 1962). Notre objectif ici est d'introduire un ensemble de mesures, dérivées du KDE, permettant d'identifier les mots ayant une forte empreinte spatiale, c.-à-d. spécifiques à une ou plusieurs régions (Section 3.2.2.1) et de différencier leurs sens locaux (Section 3.2.2.2).

3.2.2.1 *Répartition géographique des mots*

Dans la lignée des travaux de Laere *et al.* (2014), nous proposons de calculer pour chaque mot w_i la distribution de probabilités de ses emplacements, notée $p_{KDE}(\mathcal{G}|w_i)$, où les lieux d'apparitions des mots sont considérés comme les cellules d'une grille $\mathcal{G}$ divisée en grilles de 200×200 mètres. La probabilité $p_{KDE}(\mathcal{G}|w_i)$ est obtenue en utilisant l'estimation par noyau définie par :

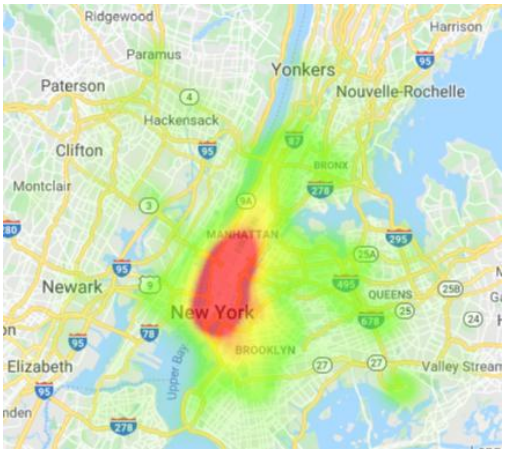
$$q(\mathcal{G}|w_i) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} K \frac{(w_i - w_{i,g})}{\theta} \tag{4.8}$$

où K est le noyau Gaussien et θ (en degrés latitude/longitude) la fenêtre qui régit le degré de lissage de l'estimation. De même, à partir de tous les emplacements géographiques des mots issus d'un ensemble de géotextes, nous calculons la distribution générale $p_{KDE}(\mathcal{G})$.

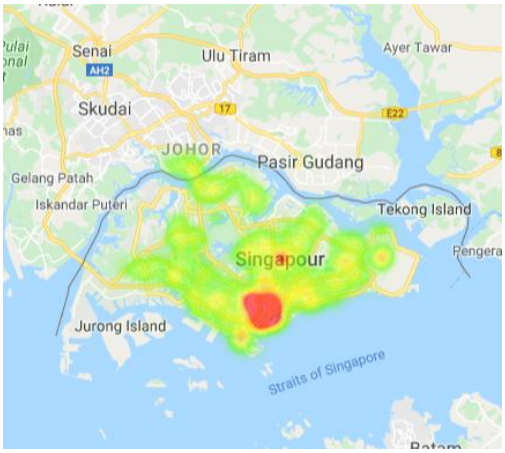
Exemple 4.5 (*Visualisation des densités de probabilités calculées*)**.** La Figure 4.6 présente les cartes de chaleur des densités de probabilités calculées sur la distribution générale $p_{KDE}(\mathcal{G})$ dans les environs de New York (Figure 4.6a) et de Singapour (Figure 4.6b), sur le mot *dinosaure* (Figure 4.6c) et sur le mot *marynabay* (Figure 4.6d). La distribution générale (Figures 4.6a et 4.6b) présente deux pics principaux dans les quartiers de Manhattan à New York et de Downton Core à Singapore. A contrario, la distribution du mot *dinosaure* (Figure 4.6c) montre plusieurs pics, répartis dans différents quartiers de New York, indiquant ainsi la présence d'ambiguïtés lexicales. Enfin, la distribution du mot *marinabay* (Figure 4.6d) présente surtout des pics de concentration autour de la Marina Bay de Singapour.

À partir des distributions de probabilités calculées, nous allons maintenant étudier la dispersion géographique des mots à l'aide de la mesure d'entropie. L'intuition est la suivante : plus les occurrences d'un mot sont fréquemment regroupées autour de quelques emplacements, plus il est probable que ce mot ait une signification spécifique dans ces emplacements. Ainsi, les mots regroupés autour de peu d'emplacements devraient être favorisés. Autrement dit, plus la distribution de probabilités $p_{KDE}(\mathcal{G}|w_i)$ est concentrée autour de quelques pics, plus l'entropie de cette distribution sera faible, et plus le mot w_i sera pertinent. Toutefois, lors de l'estimation de $p_{KDE}(\mathcal{G}|w_i)$, le nombre total d'occurrence du mot w_i n'a pas été pris en compte. De fait, un mot qui n'apparaîtrait qu'une seule fois dans la collection aura une distribution de probabilités avec une valeur d'entropie de 0. Pour faire face à ce problème, Laere *et al.* (2014) ont proposé de lisser les probabilités $p_{KDE}(\mathcal{G}|w_i)$ en fonction du nombre total d'occurrences du mot w_i. Ainsi, en utilisant un lissage Bayésien combiné à une loi de probabilités a priori de Dirichlet, nous obtenons :

$$p_{Dir}(g|w_i) = \frac{p_{KDE}(g|w_i) \times N_i + \mu \times p_{KDE}(g)}{N_i + \mu} \tag{4.9}$$

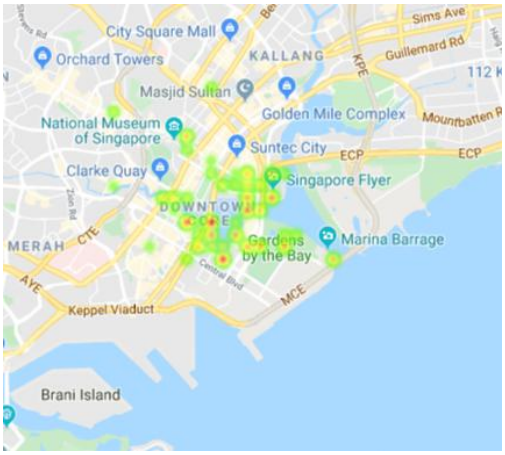


(a) Distribution générale $p_{KDE}(\mathcal{G})$ dans les environs de New York.



(b) Distribution générale $p_{KDE}(\mathcal{G})$ dans les environs de Singapour.

(c) Distribution du mot *dinosaure* dans les environs de New York.



(d) Distribution du mot *marinabay* dans les environs de Singapour.

Figure 4.6 – Carte de chaleur des densités calculées pour la distribution générale dans les environs de New York (a) et Singapour (b), le mot *dinosaure* (c) et le mot *marinabay* (d), en utilisant la méthode d'estimation par noyau.

avec N_i, le nombre d'occurrences du mot w_i, et $\mu \in [0, +\infty]$ un paramètre qui contrôle le nombre d'occurrences que nous devrions considérer pour abandonner l'idée que les occurrences de w_i suivent la distribution générale. Notons que si nous considérons de faibles valeurs de μ, nous sélectionnerions des mots plus rares (c.-à-d. ayant de faibles occurrences). Après l'étape de lissage, nous utilisons la mesure d'entropie pour ordonnancer les mots.

$$H(\mathcal{G}|w_i) = -\sum_{g \in \mathcal{G}} p_{Dir}(g|w_i) \times \log(p_{Dir}(g|w_i)) \tag{4.10}$$

3.2.2.2 *Identification des différents sens locaux*

L'étape précédente nous a permis de calculer, pour chaque mot, sa répartition géographique via une mesure d'entropie. Partant des résultats obtenus, nous sélectionnons un sous-ensemble de mots ayant des valeurs d'entropies les plus faibles (c.-à-d. inférieur à un seuil fixé) pour lesquels nous allons identifier les différents sens locaux. Ce sous-ensemble de mots est noté $\mathcal{V}' \subset \mathcal{V}$. Dans le cadre de nos expérimentations, nous avons retenu les mots ayant une valeur d'entropie inférieure à deux écarts-types par rapport à la moyenne (c.-à-d. $H(\mathcal{G}|w_i) < \mu_H - 2 \times \sigma_H$). Les différentes étapes de réalisation sont présentées dans l'Algorithme 2 et illustrées dans la Figure 4.7.

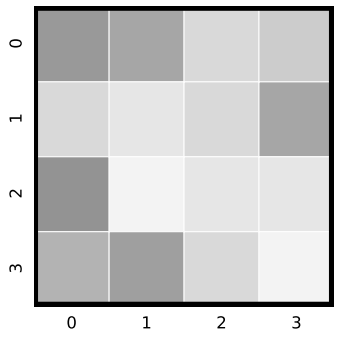
Algorithme 2 : Identification des sens locaux des mots.

Entrées : Sous-ensemble du vocabulaire $\mathcal{V}'$; Probabilités lissées $p_{Dir}(\mathcal{G}|w)$
Sorties : Sens locaux des mots

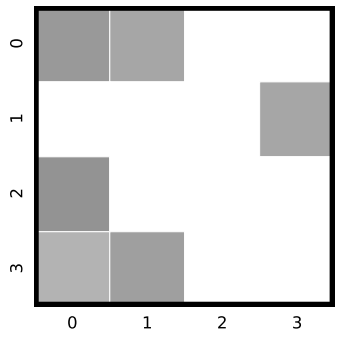
1 **pour** $i \in \{1, .., |\mathcal{V}'|\}$ **faire**
2 $\quad G_i$ = `SélectionTopGrilles`$(p_{Dir}(\mathcal{G}|w_i))$
3 $\quad G_i^l$ = `RegroupementSpatial`(G_i)
4 $\quad G_i^s$ = `RegroupementSémantique`(G_i^s)

Dans un premier temps, pour chaque mot $w_i \in \mathcal{V}'$, nous sélectionnons, à l'aide de la fonction `SélectionTopGrilles`, les grilles les plus pertinentes, c.-à-d. les grilles $g \in \mathcal{G}$ ayant les probabilités $p_{Dir}(g|w_i)$ les plus élevées (Algorithme 2, ligne 2). Concrètement, dans le cadre de nos expérimentations, nous n'avons conservé que les grilles ayant une probabilité incluse dans l'intervalle de probabilité qui suit le neuvième décile divisant la loi de probabilité $p_{Dir}(g|w_i)$. Ainsi, chacune des grilles retenues peut être vue comme une spécificité locale du mot w_i. À titre d'exemple, dans la Figure 4.7, à partir de l'ensemble des distributions de probabilités associées au mot w_i (Figure 4.7a), l'étape de sélection permet de ne retenir que les six grilles les plus pertinentes (Figure 4.7b).

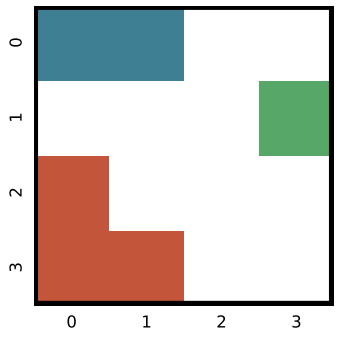
Dans un deuxième temps, nous regroupons les grilles $G_{i,j} \in G_i$ ($j = 1, \ldots, |G_i|$) adjacentes à l'aide de la fonction `RegroupementSpatial` (Algorithme 2, ligne 3), afin de former un nouvel ensemble de grilles, noté G_i^l. Cette étape est portée par l'intuition suivante : si les sens locaux d'un même mot sont proches spatialement, alors ils ont un sens proche. Dans notre cas, deux grilles sont considérées comme proches spatialement si elles sont contiguës. Cette intuition rejoint l'Hypothèse 1 formulée dans la Section 1 et nous permet de réduire le nombre de sens locaux (associés aux grilles G_i). En reprenant l'exemple de la Figure 4.7, nous réduisons ainsi le nombre de sens locaux à trois (Figure 4.7c).



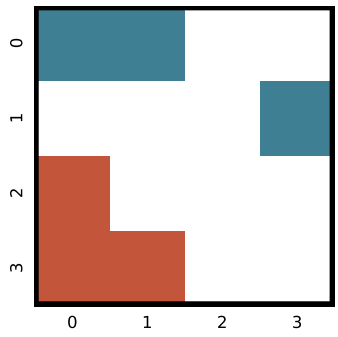
(a) Distribution des probabilités $p_{Dir}(\mathcal{G}|w_i)$.



(b) Sélection des grilles G_i pertinentes.



(c) Regroupement spatial des grilles G_i^l.



(d) Regroupement sémantique des grilles G_i^s.

Figure 4.7 – Exemple d'identification des sens locaux du mot w_i à partir des distributions spatiales $p_{Dir}(\mathcal{G}|w_i)$.

Enfin, dans un troisième temps, à partir des grilles regroupées spatialement G_i^l, nous effectuons un regroupement sémantique à l'aide de la fonction `RegroupementSémantique` pour obtenir l'ensemble des grilles G_i^s (Algorithme 2, ligne 4). L'idée sous-jacente est que tous les sens locaux restants ne sont pas nécessairement différents. Par exemple, avec les étapes précédentes, nous pourrions avoir déterminé et regroupé plusieurs zones géographiques, c.-à-d. des grilles, où le terme « *subway* » est dense. Cependant, parmi toutes ces grilles, certaines peuvent faire allusion à la chaîne de restauration rapide *Subway*, tandis que d'autres peuvent faire allusion aux transports en commun (*subway* est la dénomination du *métro* à New York). Dès lors, il nous semble pertinent de regrouper les grilles ayant un champ lexical proche. Formellement, pour chaque grille $G_{i,j}^l$ ($j \in 1, \ldots, |G_i^l|$), nous récupérons l'ensemble des termes w mentionnés par les géotextes o publiés dans les grilles de l'ensemble $G_{i,j}^l$, ainsi que les IDF associés (calculés sur l'ensemble de la collection). Cet ensemble est noté $X_{i,j}$. Nous calcu-

lons ensuite les similarités grille-à-grille à l'aide du coefficient de corrélation des rangs de Spearman :

$$sim(G^l_{i,j}, G^l_{i,k}) = \rho(rg_{X_{i,j}}, rg_{X_{i,k}}) \tag{4.11}$$

où $rg_{X_{i,j}}$ et $rg_{X_{i,k}}$ sont les variables de rang calculées à partir des données $X_{i,j}$ et $X_{i,k}$. Les ensembles $G^l_{i,j}$ et $G^l_{i,k}$ sont considérés comme similaires si leur similarité dépasse un seuil fixé à $0,5$. Pour conclure avec l'exemple de Figure 4.7, à partir des ensembles rouge, vert et bleu (Figure 4.7c), nous calculons les similarités deux à deux, ce qui nous conduit à ne retenir que deux ensembles de grilles (bleu et rouge), correspondant à deux sens locaux du mot w_i (Figure 4.7d).

3.3 *Régularisation a posteriori des plongements lexicaux*

Dans la section précédente, nous avons déterminé les différentes spécificités locales des mots, c.-à-d. leurs sens locaux. Nous abordons à présent l'étape principale de construction des plongements lexicaux augmentés par la connaissance spatiale. Nous commençons par détailler dans la Section 3.3.1 la méthode proposée pour corriger les représentations distribuées des mots préalablement entraînées (*retrofitting*). Enfin, dans la Section 3.3.2, nous détaillons l'intégration des contraintes spatiales dans le modèle de régularisation.

3.3.1 *Correction des plongements lexicaux*

La construction des représentations distribuées spatiales d'un mot w_i, notées $\mathbf{w}^s_{i,j}$, s'appuie sur un processus de régularisation a posteriori du vecteur distribué $\mathbf{w}_i \in \mathbf{W}$ en considérant à la fois les mots voisins $W^+_{i,j}$ et les mots éloignés $W^-_{i,j}$ des sens locaux $w^s_{i,j}$. Dans la section suivante nous détaillons les concepts de mots voisins et éloignés.

Notre objectif est d'apprendre l'ensemble des représentations distribuées spatiales des mots $\mathbf{W}^s$, autrement dit, de corriger les plongements lexicaux $\mathbf{W}$ préalablement entraînés (Algorithme 1, ligne 5). Nous contraignons que le vecteur de représentation $\mathbf{w}^s_{i,j}$ soit :

1. géométriquement proche (selon une métrique de distance), c.-à-d. sémantiquement relié, de la représentation distribuée du mot $\mathbf{w_i}$;
2. géométriquement proche de ses voisins spatiaux $W^+_{i,j}$;
3. géométriquement non proche des mots spatialement distants $W^-_{i,j}$.

Pour répondre à notre objectif de rapprocher sémantiquement les mots proches spatialement et d'éloigner sémantiquement les mots distants spatialement, nous proposons de minimiser la fonction objectif suivante :

$$\Psi(\mathbf{W}^s) = \sum_{i=1}^{|\mathcal{V}|} \sum_{j=1}^{n_i} \left[\alpha \; d(\mathbf{w}^s_{i,j}, \mathbf{w}_i) + \beta \sum_{w_k \in W^+_{i,j}} d(\mathbf{w}^s_{i,j}, \mathbf{w}^s_k) + \gamma \sum_{w_k \in W^-_{i,j}} 1 - d(\mathbf{w}^s_{i,j}, \mathbf{w}^s_k) \right] \tag{4.12}$$

$$d(\mathbf{w_i}, \mathbf{w_j}) = 1 - sim(\mathbf{w_i}, \mathbf{w_j}) \tag{4.13}$$

$$sim(\mathbf{w_i}, \mathbf{w_j}) = \frac{\mathbf{w_i} \cdot \mathbf{w_j}}{||\mathbf{w_i}|| \cdot ||\mathbf{w_j}||} \tag{4.14}$$

où $\mathbf{w_i} \in \mathbf{W}$ est la représentation distribuée du mot w_i, issue d'une matrice de plongements lexicaux préalablement entraînée ; $\mathbf{w^s_{i,j}} \in \mathbf{W^s}$ est la représentation distribuée corrigée d'un des sens locaux du mot w_i ; $d(\mathbf{w_i})$ (Équation 4.13) est une mesure de distance dérivée de la mesure de similarité du cosinus ; $sim(\mathbf{w_i}, \mathbf{w_j})$ (Équation 4.14) est la similarité du cosinus ; $W^+_{i,j}$ et $W^-_{i,j}$ sont respectivement les ensembles des mots voisins et distants du mot local $w^s_{i,j}$ et seront définis dans la Section 3.3.2 ; α, β, $\gamma \geq 0$ sont des hyperparamètres qui contrôlent l'importance relative de chaque terme. La fonction objectif Ψ étant différentiable, nous la minimisons par une descente de gradient. Par la suite, $\mathbf{W}^s_{km}$ désignera les plongements lexicaux spatiaux dont les sens locaux $w^s_{i,j}$ ont été déterminés par la méthode de *clustering* (k-moyennes). De même, $\mathbf{W}^s_{kde}$ désignera les plongements lexicaux des sens locaux issus du partitionnement probabiliste (KDE).

3.3.2 *Intégration des contraintes spatiales*

Pour intégrer les contraintes spatiales dans le modèle de régularisation Ψ (Équation 4.12), nous devons déterminer les ensembles des mots voisins $W^+_{i,j}$ et des mots distants $W^-_{i,j}$ pour chacun des sens locaux $w^s_{i,j}$ (Algorithme 1, lignes 9 et 10). Deux solutions sont envisagées : l'une adaptée aux sens locaux déterminés par la méthode de *clustering* (Section 3.2.1), l'autre adaptée aux sens locaux déterminés par la méthode probabiliste (Section 3.2.2). Ces deux solutions sont décrites dans les sections suivantes en détaillant les fonctions `Voisins` (Algorithme 1, ligne 9) et `Distants` (Algorithme 1, ligne 10) utilisées dans l'Algorithme 1.

3.3.2.1 *Méthode de clustering : recherche des mots voisins et des mots distants*

L'Algorithme 3 détaille les fonctions $\texttt{Voisins}_{km}$ et $\texttt{Distants}_{km}$ utilisées pour la recherche des mots voisins et distants pour les sens locaux déterminés selon la méthode de *clustering*. La Figure 4.8 illustre les différentes étapes de l'algorithme.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Algorithme 3 : Recherche des ensembles de mots voisins et distants (*k*-moyennes).

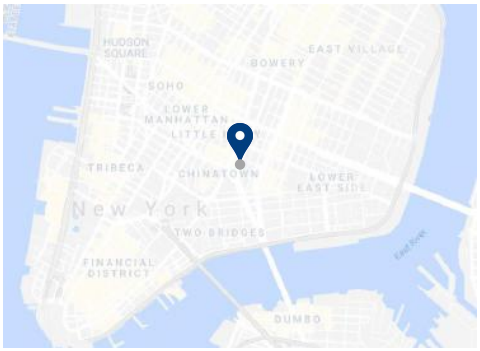
Entrées : Sens local $w^s_{i,j}$, Vocabulaire $\mathcal{V}$; Barycentre $\mathcal{B}_{i,j}$; Ensemble des barycentres $\mathcal{B}$; Rayon des mots proches r^+ ; Rayon des mots distants r^-

Sorties : Ensemble des mots voisins $W^+_{i,j}$; Ensemble des mots distants $W^-_{i,j}$

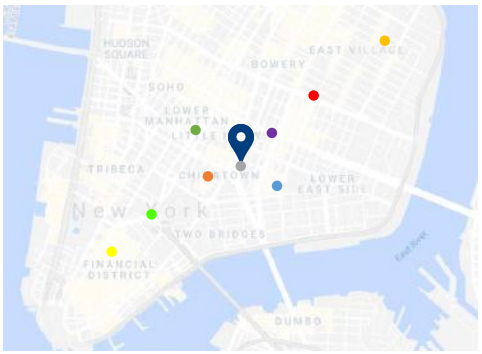
1 **Fonction** $\texttt{Voisins}_{km}(w^s_{i,j}, \mathcal{V}, \mathcal{B}_{i,j}, \mathcal{B}, r^+)$
2 $\quad \mathcal{V}_{candidats} = \mathcal{V} \setminus w_i$
3 $\quad W^+_{i,j} = \{\}$
4 $\quad$ **pour** $k \in \{1, .., |\mathcal{V}_{candidats}|\}$ **faire**
5 $\quad\quad$ **pour** $l \in \{1, .., n_k\}$ **faire**
6 $\quad\quad\quad$ **si** $dist(\mathcal{B}_{i,j}, \mathcal{B}_{k,l}) < r^+$ **alors**
7 $\quad\quad\quad\quad W^+_{i,j} = W^+_{i,j} \cup w^s_{k,l}$
8 $\quad$ **retourner** $W^+_{i,j}$
9
10 **Fonction** $\texttt{Distants}_{km}(w^s_{i,j}, \mathcal{V}, \mathcal{B}_{i,j}, \mathcal{B}, r^-)$
11 $\quad W^+_{i,j} = \text{Voisins}_{km}(w^s_{i,j}, \mathcal{V}, \mathcal{B}_{i,j}, \mathcal{B}, r^-)$
12 $\quad W^-_{i,j} = \mathcal{V} \setminus W^+_{i,j}$
13 $\quad$ **retourner** $W^-_{i,j}$

Considérons le sens local $w^s_{i,j}$, représenté par son barycentre $\mathcal{B}_{i,j}$ (Définition 4.6), comme illustré dans la Figure 4.8a (point gris surmonté d'un marqueur bleu). Pour rechercher l'ensemble des mots proches $W^+_{i,j}$, à l'aide de la fonction $\texttt{Voisins}_{km}$, nous commençons par déterminer $\mathcal{V}_{candidats}$, l'ensemble des mots candidats (Algorithme 3, ligne 2). Dans notre exemple, cet ensemble est composé de tous les points colorés, c.-à-d. les barycentres des mots, présents sur la carte de la Figure 4.8b, le point gris étant exclu. Nous parcourons ensuite, pour chaque mot w_k candidat (Algorithme 3, ligne 4), ses sens locaux $w^s_{k,l}$ représentés par les barycentres $\mathcal{B}_{k,l}$ (Algorithme 3, ligne 5). Dans le cas où la distance spatiale séparant le barycentre $\mathcal{B}_{i,j}$ du barycentre $\mathcal{B}_{k,l}$ est inférieure à un seuil r^+ (Algorithme 3, lignes 6), le mot $w^s_{k,l}$ associé à ce barycentre est ajouté à l'ensemble $W^+_{i,j}$ (Algorithme 3, ligne 7). En reprenant l'exemple de la Figure 4.8, l'ensemble des mots proches du sens local $w^s_{i,j}$ sont tous les points, c.-à-d. tous les barycentres, situés dans un rayon r^+ autour de $w^s_{i,j}$. Ce rayon est matérialisé par la zone verte dans la Figure 4.8c.

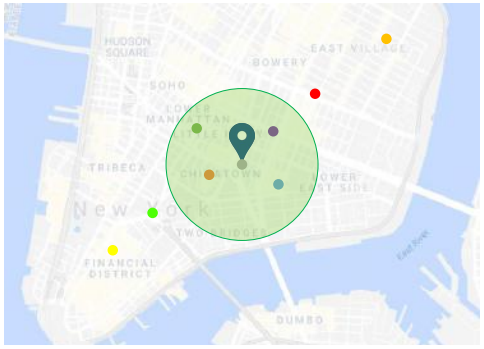
De même, pour rechercher l'ensemble des mots distants $W^-_{i,j}$, à l'aide de la fonction $\texttt{Distants}_{km}$, nous commençons par déterminer l'ensemble des mots voisins situés dans un rayon r^- autour du barycentre $\mathcal{B}_{i,j}$ (Algorithme 3, ligne 11). Ce rayon est matérialisé par la zone rouge dans la Figure 4.8d. L'ensemble des mots



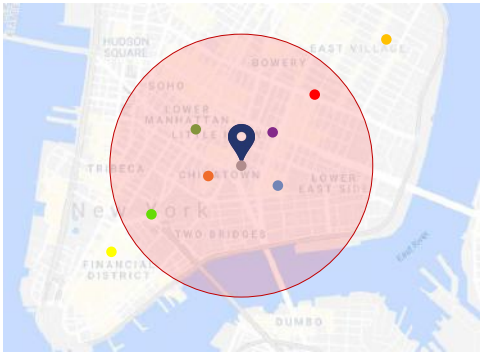
(a) Sélection d'un mot $w^s_{i,j}$.



(b) Sélection des mots candidats $\mathcal{V}_{candidats}$.



(c) Sélection des mots situés dans un rayon r^+ autour du mot $w^s_{i,j}$.



(d) Exclusion des mots situés dans un rayon r^- autour du mot $w^s_{i,j}$.

Figure 4.8 – Exemple de sélection des mots proches et des mots distants du mot $w^s_{i,j}$ issu du partitionnement en k-moyennes. Chaque point de couleur correspond au barycentre $\mathcal{B}_{i,j}$ d'un sens local $w^s_{i,j}$.

distants est donc constitué des mots qui ne sont pas contenus dans l'ensemble des mots voisins (Algorithme 3, ligne 12).

Il convient de noter que les mots compris entre les rayons r^+ et r^- autour du barycentre ne sont pas considérés pour régulariser les représentations distribuées des mots puisque considérés comme non discriminants.

3.3.2.2 *Méthode probabiliste : recherche des mots voisins et des mots distants*

L'Algorithme 4 détaille les fonctions $\texttt{Voisins}_{kde}$ et $\texttt{Distants}_{kde}$ utilisées pour la recherche des mots voisins et distants pour les sens locaux déterminés selon la méthode probabiliste. La Figure 4.9 illustre les différentes étapes de l'algorithme.

Considérons le sens local $w^s_{i,j}$, représenté par l'ensemble des grilles $\mathcal{G}(w^s_{i,j})$ dans lesquels il est mentionné, comme illustré dans la Figure 4.9a (grilles grisées). Pour

Algorithme 4 : Recherche des ensembles de mots voisins et distants (KDE).

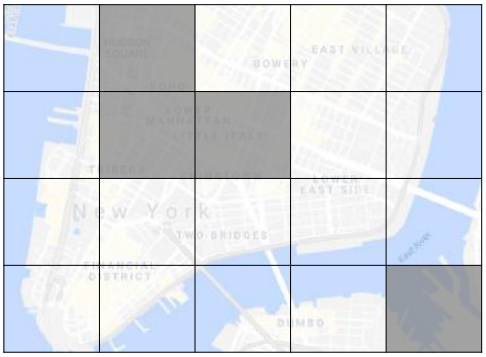
Entrées : Sens local $w^s_{i,j}$; Ensemble des grilles $\mathcal{G}$; Vocabulaire $\mathcal{V}$

Sorties : Ensemble des mots voisins $W^+_{i,j}$; Ensemble des mots distants $W^-_{i,j}$

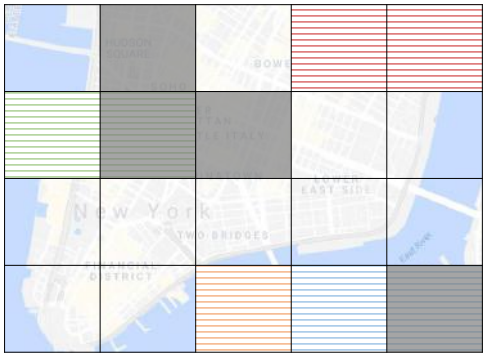
1 **Fonction** `Voisins`$_{kde}(w^s_{i,j}, \mathcal{V})$
2 $\quad \mathcal{V}_{candidats} = \mathcal{V} \setminus w_i$
3 $\quad W^+_{i,j} = \{\}$
4 $\quad$ **pour** $k \in \{1, .., |\mathcal{V}_{candidats}|\}$ **faire**
5 $\quad\quad$ **pour** $l \in \{1, .., n_k\}$ **faire**
6 $\quad\quad\quad$ **si** $\mathcal{G}(w^s_{k,l}) \cap \mathcal{G}(w^s_{i,j})$ **alors**
7 $\quad\quad\quad\quad W^+_{i,j} = W^+_{i,j} \cup w^s_{k,l}$
8 $\quad$ **retourner** $W^+_{i,j}$
9
10 **Fonction** `Distants`$_{kde}(w^s_{i,j}, \mathcal{V})$
11 $\quad W^+_{i,j} = \text{Voisins}(w^s_{i,j}, \mathcal{V})$
12 $\quad W^-_{i,j} = \mathcal{V} \setminus W^+_{i,j}$
13 $\quad$ **retourner** $W^-_{i,j}$

rechercher l'ensemble des mots proches $W^+_{i,j}$ du sens local $w^s_{i,j}$, à l'aide de la fonction `Voisins`$_{kde}$, nous commençons par déterminer $\mathcal{V}_{candidats}$, l'ensemble des mots candidats (Algorithme 4, ligne 2). Dans notre exemple, cet ensemble est composé de toutes les grilles avec des rayures horizontales présentes sur la carte de la Figure 4.9b. Nous parcourons ensuite, pour chaque mot w_k candidat (Algorithme 4, ligne 4), ses différents sens locaux $w^s_{k,l}$, représentés par l'ensemble des grilles $\mathcal{G}(w^s_{k,l})$ (Algorithme 4, ligne 5). Dans le cas où il y a une intersection entre les ensembles $\mathcal{G}(w^s_{k,l})$ et $\mathcal{G}(w^s_{i,j})$ (Algorithme 4, lignes 6), le mot $w^s_{k,l}$ est ajouté à l'ensemble $W^+_{i,j}$ (ligne 7). En reprenant l'exemple de la Figure 4.9, l'ensemble des mots proches du sens local $w^s_{i,j}$ sont tous ceux dont les grilles s'entrecroisent avec les grilles grisées, c.-à-d. les grilles vertes et bleues sur la Figure 4.9c.

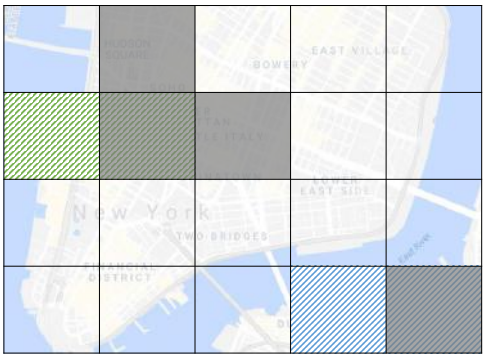
De même, pour rechercher l'ensemble des mots distants $W^-_{i,j}$, nous commençons par déterminer l'ensemble des mots voisins (Algorithme 4, ligne 11). L'ensemble des mots distants est donc constitué des mots qui ne sont pas contenus dans l'ensemble des mots voisins (Algorithme 4, ligne 12). Dans notre exemple, cet ensemble est représenté par les grilles orange et rouges sur la Figure 4.9d.



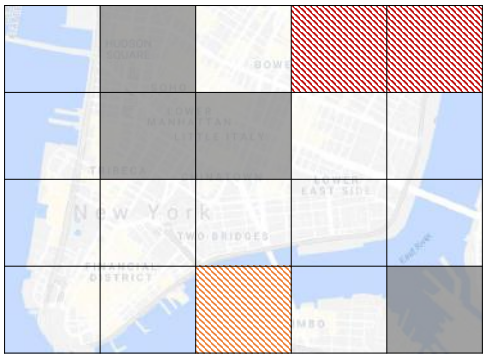
(a) Sélection d'un mot $w^s_{i,j}$ et des grilles associées.



(b) Sélection des mots candidats $\mathcal{V}_{candidats}$ et leurs grilles respectives.



(c) Sélection des mots dont les grilles s'entrecoupent avec celles du mot $w^s_{i,j}$.



(d) Exclusion des mots dont les grilles s'entrecoupent avec celles du mot $w^s_{i,j}$.

Figure 4.9 – Exemple de sélection des mots proches et des mots distants du mot $w^s_{i,j}$ issu du partitionnement probabiliste. Chaque grille de couleur correspond à l'empreinte géographique d'un sens local $w^s_{i,j}$.

4 Évaluation expérimentale

Afin d'évaluer la qualité de nos plongements lexicaux spatiaux, nous avons mis en place un protocole d'évaluation comparative s'appuyant sur des tâches d'évaluation intrinsèques et extrinsèques. La question de recherche de notre évaluation expérimentale est triple :

QR1 – Peut-on confirmer empiriquement l'Hypothèse 1 (Section 1) ?

QR2 – Dans quelle mesure les plongements lexicaux spatiaux permettent-ils de saisir la sémantique d'un texte ?

QR3 – Quel est l'apport des plongements lexicaux spatiaux quant à leur capacité à saisir les signaux de pertinence pour résoudre la tâche de prédiction sémantique de l'emplacement ?

Nous proposons de répondre à ces trois questions de recherche dans les sections suivantes. Nous commençons dans la Section 4.1 par répondre à **QR1** en validant l'Hypothèse 1. Pour répondre à **QR2**, nous menons, dans la Section 4.2, une évaluation intrinsèque des plongements lexicaux spatiaux sur une tâche de similarités de POIs. Enfin, nous répondons dans la Section 4.3 à **QR3** en réalisant une évaluation extrinsèque des représentations à l'aide de la tâche de prédiction sémantique de l'emplacement.

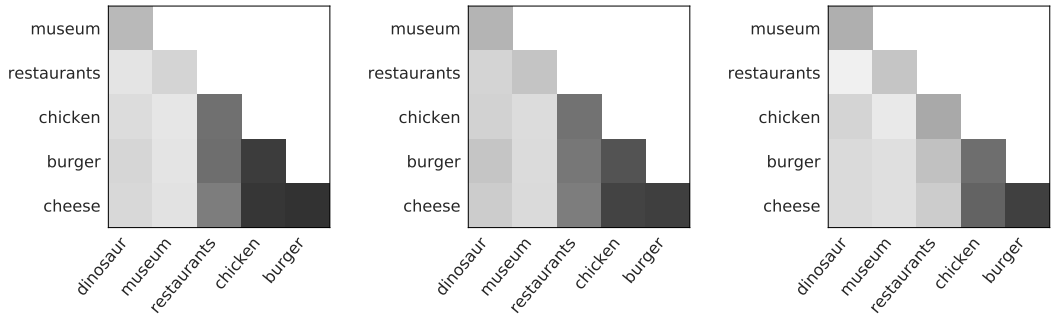
4.1 *Retour sur l'Hypothèse 1 (QR1)*

Nous commençons notre évaluation expérimentale en revenant sur l'Hypothèse 1. Pour rappel, nous supposions que les mots pouvaient véhiculer des sens locaux différents, selon différentes zones. Partant de cette hypothèse, nous avons proposé deux méthodes pour déterminer les sens locaux des mots. Pour vérifier le bien-fondé de cette hypothèse, nous construisons les cartes thermiques (ou *heat map*) des valeurs de similarité du cosinus entre les représentations distribuées d'un échantillon de mots. Ces dernières devraient nous permettre de visualiser les différents niveaux de similarité entre les mots en fonction des régions. Afin de représenter les sens locaux des mots, nous disposons d'une collection de géotextes (Section 4.2.1.1) que nous divisons en deux sous-ensembles distincts, déterminés selon la ville dont ils sont originaires (c.-à-d. à New York ou à Singapour). Pour chaque mot de chaque ville, nous déterminons sa répartition spatiale représentée par un unique barycentre $\mathcal{B}_i$ correspondant à sa zone d'apparition moyenne. Enfin, pour chaque paire de mots (w_i, w_j), les similarités du cosinus sont atténuées par un facteur spatial $f_s(w_i, w_j)$ qui permet de tenir compte de leur proximité spatiale. Formellement, $f_s(w_i, w_j)$ est définie par :

$$f_s(w_i, w_j) = \exp\{-\frac{d(\mathcal{B}_i, \mathcal{B}_j) - \mu}{\sigma}\} \tag{4.15}$$

avec $d(\mathcal{B}_i, \mathcal{B}_j)$ la distance Haversine entre les barycentres de w_i et w_j et μ (resp. σ) la distance moyenne (resp. l'écart type) entre toutes les paires de mots d'un sous-ensemble. Les résultats de cette analyse sont présentés dans la Figure 4.10. La Figure 4.10a montre la carte thermique non-pondérée (c.-à-d. similarité simple) tandis que les Figures 4.10b et 4.10c montrent respectivement les cartes thermiques pondérées par le facteur spatial pour les villes de New York et Singapour. Plus la cellule est sombre, plus la paire de mots est similaire (c.-à-d. possède une valeur de similarité du cosinus élevée).

Comme nous pouvons par exemple le constater dans la Figure 4.10, la cellule (*restaurants*, *dinosaur*) est plus sombre sur la Figure 4.10b que sur la Figure 4.10a



(a) SIM PL (b) SIM PL-A (NY) (c) SIM PL-A (SG)

Figure 4.10 – Similarité du cosinus des plongements lexicaux (SIM PL) traditionnels (a), amortis (SIM PL-A) par les distances des barycentres des mots dans les sous-ensemble de New York (NY) (b) et de Singapour (SG) (c).

tandis qu'elle est plus claire sur la Figure 4.10c que sur la Figure 4.10a. D'une manière générale, il n'y a pas de raison évidente pour que les mots *restaurants* et *dinosaur* soient liés l'un à l'autre, comme le montre la similarité de leurs plongements lexicaux dans la Figure 4.10a. Cependant, nous avons noté l'existence d'une chaîne de restaurants à New York, appelée *Dinosaur Bar-B-Que*, ce qui conduit à une forte proportion d'objets géotextuels où les deux termes cooccurrent. Cela conduit ainsi à une relation sémantique locale plus forte, pour cette paire de mots, dans la ville de New York, comme le révèle la Figure 4.10b. Par ailleurs, l'observation conjointe de la Figure 4.10a et de ses variations spatiales dans la Figure 4.10b et la Figure 4.10c fournit quelques indices sur le bien fondé de l'Hypothèse 1. En effet, nous pouvons constater que les mots *dinosaur* et *museum* sont similaires quel que soit l'endroit. En reliant cette observation à la précédente, nous pouvons en déduire que le mot *dinosaur* pourrait se référer tant à *museum* qu'à *restaurant* dans la ville de New York, comme le révèle la forte similarité avec des mots tels que *burger* et *cheese* dans la Figure 4.10b qui est nettement moins prononcée dans la Figure 4.10c. Ce raisonnement vaut également pour d'autres paires de mots telles que *burger* et *restaurants*.

En résumé, cette analyse qualitative démontre que *(1)* la similarité sémantique entre deux mots dépend de la distance spatiale entre les sources de ces mots (c.-à-d. que plus les sources sont proches, plus les termes sont sémantiquement reliés) et *(2)* que le mot peut avoir des sens distincts selon sa répartition spatiale au travers des objets où il occurre. Cela valide l'Hypothèse 1 et motive la nécessité d'apprendre des représentations distribuées distinctes pour un mot donné en fonction de la zone géographique des objets qui le référencent.

4.2 *Évaluation intrinsèque : similarité de POIs (QR2)*

L'objectif de cette première évaluation expérimentale est de mesurer la qualité des plongements lexicaux spatiaux quant à leur capacité à saisir la sémantique sous-jacente d'un texte. À cet égard, nous réalisons une tâche de similarité d'objets géotextuels, dérivée de la tâche de *proximité sémantique des phrases* proposée dans le cadre de l'évaluation SentEval (Conneau et Kiela, 2018). Cette tâche consiste à quantifier, dans quelle mesure, la similarité entre deux géotextes représentés dans un espace latent, peut se rapprocher des scores de similarités notés par un humain. La qualité des représentations est mesurée par la corrélation de Spearman entre les scores annotés manuellement et la similarité des représentations apprises par nos modèles.

Plus précisément, soit une collection contenant des paires d'objets géotextuels (o_i, o_j), dans le cas présent, des POIs, pour lesquelles un score de similarité a été attribué par des assesseurs humains. Pour chaque objet géotextuel $o = [w_1^{(o)}, \cdots, w_m^{(o)}]$ de chacune des paires, nous commençons par déterminer sa représentation distribuée $\widehat{o}$ en agrégeant les plongements lexicaux $\mathbf{w}_k$ associés aux mots $w_k^{(o)} \in o$ (Équation 5.14). Nous calculons ensuite la similarité du cosinus entre les vecteurs de représentation associés aux paires d'objets (o_i, o_j), puis reportons le coefficient de Spearman, noté ρ_{sim}, calculé entre le classement déterminé par nos représentations, et le classement obtenu grâce aux scores attribués par les experts humains.

Dans les sections suivantes, nous commençons par définir le cadre expérimental de notre évaluation dans la Section 4.2.1 puis nous discutons les résultats obtenus dans la Section 4.2.2.

4.2.1 *Cadre expérimental*

Dans la suite, nous décrivons le protocole d'évaluation permettant de répondre à **QR2**. Nous commençons par présenter dans la Section 4.2.1.1 les jeux de données utilisés pour l'apprentissage et l'évaluation intrinsèque des représentations. Nous continuons en présentant les scénarios d'évaluation dans la Section 4.2.1.2 et les modèles de référence dans la Section 4.2.1.3. Enfin, dans la Section 4.2.1.4, nous donnons des détails sur l'implémentation de nos modèles et ceux de référence, avec notamment le réglage des hyperparamètres.

4.2.1.1 *Jeux de données*

Dans cette contribution, nous nous attachons à corriger des plongements lexicaux et à évaluer leur qualité. De ce fait, nous avons utilisé plusieurs collections de données, que nous décrivons dans la suite.

Pour construire les répartitions spatiales des mots du vocabulaire et régulariser les plongements lexicaux, nous avons utilisé la collection de tweets géotaggés publiée par Zhao *et al.* (2016) et une collection de POIs et de critiques d'utilisateurs publiés sur le réseau social Foursquare. Concernant la collection de Zhao *et al.* (2016), cette dernière se compose d'environs 647 000 tweets géotaggés publiés dans les villes de New York et de Singapour entre septembre 2010 et janvier 2015 et collectés via l'API Twitter. La collection de POIs est quant à elle constituée de 804 465 POIs et de 2, 183 millions critiques d'utilisateurs. Par ailleurs, chaque POI est affilié à une catégorie (p. ex. restaurant chinois, piscine, etc.). L'ensemble des catégories possibles est organisé selon une hiérarchie (p. ex. Art et Divertissement est la catégorie parente des catégories Cinéma, Piscine, etc.) définie par Foursquare [2].

Pour réaliser l'évaluation intrinsèque et donc mesurer la qualité des plongements lexicaux spatiaux à l'aide d'une tâche de similarité de POIs, nous avons constitué notre propre collection selon le protocole suivant. Nous avons commencé par sélectionner aléatoirement un échantillon de 500 POIs dans notre collection de POIs détaillée ci-dessus. Pour chacun des POIs de cet échantillon, nous avons récupéré un ensemble de 20 POIs candidats, dont la première moitié (c.-à-d. 10 POIs) est proche spatialement selon la distance Haversine, et l'autre moitié est proche sémantiquement selon la mesure BM25. Notre collection se compose donc de 10 000 paires de POIs *(POI, POI candidat)* que nous avons fait annoter manuellement par des juges humains.

Figure 4.11 – Aperçu de la tâche d'évaluation sur Amazon Mechanical Turk.

2. https://developer.foursquare.com/docs/build-with-foursquare/categories/

Plus spécifiquement, nous avons réalisé une tâche d'annotation sur la plateforme de production participative (*crowdsoursing)* Amazon Mechanical Turk. L'objectif de cette tâche était de déterminer dans quelle mesure les deux POIs issus d'une paire *(POI, POI candidat)* sont similaires. Pour cela, comme illustré dans la Figure 4.11, nous avons présenté aux annotateurs les informations relatives aux POIs, à savoir leur nom, leur description (si elle est disponible) et cinq avis publiés par les utilisateurs. Nous leur avons ensuite proposé d'évaluer, selon eux, la similarité en choisissant d'attribuer une note comprise entre 1 (extrêmement dissimilaire) et 4 (extrêmement similaire). Pour les guider, nous avons proposé quelques exemples permettant de qualifier la similarité :

— deux POIs sont *extrêmement similaires (4)* s'ils partagent exactement les mêmes propriétés. Une similarité extrême caractérise deux POIs appartenant à la même catégorie ou au même type (p. ex. restaurant chinois). Voici quelques exemples de POIs extrêmement similaires : deux restaurants vietnamiens, le Palais de l'Élysée et la Maison Blanche ;
— deux POIs sont *très similaires (3)* s'ils partagent la plupart de leurs propriétés. Une similarité forte caractérise deux POIs appartenant à la même catégorie ou au même type. Voici quelques exemples de POIs très similaires : un stade de football et un stade de baseball sont deux stades, un muséum d'histoire naturelle et un musée d'art sont deux musées ;
— deux POIs sont *légèrement similaires (2)* s'ils partagent certaines propriétés. Une similarité légère caractérise deux POIs n'appartenant pas à la même catégorie mais partageant certaines propriétés. Voici quelques exemples de POIs légèrement similaires : un casino et une salle d'arcade sont deux lieux de divertissement, les montagnes et les parcs nationaux sont deux lieux naturels ;
— deux POIs sont *extrêmement dissimilaires (1)* s'ils ne partagent aucune propriété. Une dissimilarité caractérise deux POIs n'appartenant pas à la même catégorie et ne partageant aucune propriété. Voici quelques exemples de POI dissemblables : un restaurant vietnamien et un terrain de baseball, un cinéma et une gare.

Cette tâche a été rémunérée 0,01$ par question. Pour plus de fiabilité, chaque paire a été évaluée par trois juges, et le score final de pertinence est choisi comme la moyenne des scores donnés par les trois annotateurs.

À l'issue de cette tâche d'annotation, nous avons construit une vérité terrain qui nous permettra par la suite d'évaluer la qualité des similarités calculées par les scénarios proposés (Section 4.2.1.2) et les modèles de référence (Section 4.2.1.3). Pour comparer les similarités, nous calculerons le coefficient de corrélation de Spearman entre les similarités de la vérité terrain et celles obtenus par les différents modèles.

4.2.1.2 *Scénarios d'évaluation*

Dans le cadre de cette évaluation expérimentale, nous évaluons trois configurations, une première notée ρ-$\mathbf{W}$ utilisant les plongements lexicaux traditionnels, une deuxième notée ρ-$\mathbf{W}^s_{km}$ utilisant les plongements lexicaux spatiaux déterminés à l'aide de la méthode de *clustering*, et une troisième notée ρ-$\mathbf{W}^s_{kde}$ utilisant les plongements lexicaux spatiaux déterminés à l'aide de la méthode probabiliste. Pour les scénarios adoptant les représentations $\mathbf{W}^s_{km}$, nous utilisons le sens local $w^s_{i,j}$ le plus proche en minimisant la distance Haversine entre l'emplacement du géotexte et le barycentre du mot $\mathcal{B}_{i,j}$. Pour les scénarios adoptant les représentations $\mathbf{W}^s_{kde}$, nous utilisons, s'il existe, le sens local $w^s_{i,j}$ présent dans la grille associée au géotexte considéré. Ces différentes configurations sont résumées dans le Tableau 4.1.

Scénarios	**Commentaires**
ρ_{sim}-$\mathbf{W}$	Plongements lexicaux traditionnels
ρ_{sim}-$\mathbf{W}^s_{km}$	Plongements lexicaux spatiaux (*k-moyennes*)
ρ_{sim}-$\mathbf{W}^s_{kde}$	Plongements lexicaux spatiaux (*kde*)

Tableau 4.1 – Configuration des scénarios utilisés pour l'évaluation intrinsèque.

4.2.1.3 *Modèles de référence*

Pour répondre à **QR2** et évaluer la qualité intrinsèque de nos plongements lexicaux spatiaux $\mathbf{W}^s_{km}$ et $\mathbf{W}^s_{kde}$, nous comparons les résultats des scénarios ρ_{sim}-$\mathbf{W}^s_{km}$ et ρ_{sim}-$\mathbf{W}^s_{kde}$ avec des modèles de référence. Ces modèles sont classés en quatre catégories : *(1)* proximité textuelle (BM25, TfIdf) ; *(2)* proximité sémantique neuronale (Doc2Vec) ; *(3)* proximité hiérarchique (SIM_{WP}, SIM_{LC}). Les caractéristiques de ces modèles sont les suivantes :

— **TfIdf** (Salton et McGill, 1984) : modèle d'appariement traditionnel ;
— **BM25** (Robertson et Jones, 1976) : modèle probabiliste de référence, largement utilisé en RI textuelle ;
— **Doc2Vec** (Le et Mikolov, 2014) : modèle neuronal de représentation de documents dans un espace latent ;
— **SIM_{WP}** (Wu et Palmer, 1994) : similarité sémantique de Wu et Palmer (1994) s'appuyant sur la proximité hiérarchique des catégories associées aux objets considérés. Son équation est la suivante :

$$\mathbf{SIM}_{WP}(p_1, p_2) = \frac{2N_3}{N_1 + N_2 + 2N_3} \tag{4.16}$$

t_1 et t_2 sont respectivement les catégories associées aux POIs p_1 et p_2. t_{lcs} est défini comme la superclasse commune aux catégories t_1 et t_2. N_1 est le

chemin le plus court entre t_1 et t_2. N_2 esy le chemin le plus court entre t_2 et t_{lcs}. N_3 est le chemin le plus court entre t_{lcs} et la racine.

— **SIM**$_{LC}$ (Leacock et Chodorow, 1998) : similarité sémantique de Leacock et Chodorow (1998) s'appuyant sur la profondeur hiérarchique entre les catégories des objets considérés. Son équation est la suivante :

$$\textbf{SIM}_{LC}(p_1, p_2) = -\log\left(\frac{N}{2D}\right) \tag{4.17}$$

où D est la profondeur maximale de la taxonomie et N est le chemin le plus courts entre t_1 et t_2.

4.2.1.4 *Détails de mise en œuvre*

L'objet de cette contribution étant de régulariser des plongements lexicaux à l'aide de répartitions spatiales, il convient de disposer de vecteurs pré-entraînés (Section 3.3.1). La matrice des plongements lexicaux **W**, utilisée par notre modèle de régularisation Ψ, mais aussi par nos scénarios (ρ_{sim}-**W**, CM-**W** et CE-**W**) est obtenue en utilisant l'implémentation du modèle *Skip-Gram* (Mikolov *et al.*, 2013b) de la librairie Python *gensim* (Rehurek et Sojka, 2010).

Le vocabulaire de la collection se compose initialement de 3,2 millions de mots, qui apparaissent au moins 3 fois dans le corpus. La fenêtre contextuelle a été fixée à 5 mots. Les vecteurs de taille 300 ont été entraînés à partir d'un corpus de documents issus de Wikipedia anglais[3] et d'un corpus de géotextes incluant des tweets, des POIs et des critiques d'utilisateurs.

Dans le cadre de cette contribution, nous n'effectuons le partitionnement spatial (Section 3.2) que sur les mots qui occurrent dans les collections décrites dans la Section 4.2.1.1, ce qui représente 238 369 mots w_i distincts. Avec la méthode de *clustering*, nous avons déterminé 630 732 sens locaux. En excluant les mots n'ayant qu'une seule empreinte spatiale (c.-à-d. un seul sens local $w^s_{i,j}$), soit 166 139 (69,7%) mots w_i, nous comptons environ 6,43 sens locaux $w^s_{i,j}$ par mots. Concernant la méthode probabiliste, la mesure d'entropie $H(\mathcal{G}|w_i)$ (Équation 4.10) nous a permis de conserver 35 730 mots w_i pour lesquels nous avons déterminé 435 286 sens locaux, soit environ 12,18 sens locaux par mots.

Notre modèle de régularisation est implémenté en utilisant l'API *keras* (Chollet *et al.*). Les hyperparamètres par défaut utilisés pour les expérimentations sont les suivants. Nous minimisons la fonction objectif Ψ par une descente du gradient, en utilisant l'algorithme d'optimisation Adam (Kingma et Ba, 2015). Nous adoptons la version mini-lots pour accélérer le processus de correction. Les hyperparamètres

3. https://dumps.wikimedia.org

α, β et γ sont fixés à 1, ce qui permet d'accorder une importance égale aux différents paramètres de le fonction de coût.

Concernant l'intégration des contraintes spatiales dans le modèle de régularisation avec la méthode de *clustering*, nous avons fixé respectivement les rayons r^+ et r^- à 100 et 500 mètres pour la recherche des mots voisins et distants.

4.2.2 *Résultats*

Nous présentons dans cette section les résultats de notre évaluation de la qualité des plongements lexicaux spatiaux au travers d'une tâche de similarité d'objets géotextuels. Cette tâche de similarité nous permet de mesurer l'aptitude des plongements lexicaux à saisir la sémantique des textes. Les résultats empiriques de l'évaluation, en terme de corrélation de Spearman, discutés dans cette section, sont résumés dans le Tableau 4.2.

Modèle	**Corrélation de Spearman**
Plongements lexicaux spatiaux	
ρ_{sim}-$\mathbf{W}^s_{kde}$	0,426
ρ_{sim}-$\mathbf{W}^s_{km}$	0,402
Plongements lexicaux traditionnels	
ρ_{sim}-$\mathbf{W}$	0,383
Modèles de référence	
TfIdf	0,199
BM25	0,178
Doc2Vec	0,213
$\mathbf{SIM}_{WP}$	0,372
$\mathbf{SIM}_{LC}$	0,342

Tableau 4.2 – Comparaison de l'efficacité des plongements lexicaux spatiaux et des modèles de référence sur la tâche de similarité des POIs (corrélation de Spearman ρ).

Les résultats présentés montrent clairement que l'utilisation des plongements lexicaux sont bénéfiques pour la tâche de similarité des objets géotextuels. En effet, les modèles ρ_{sim}-x, avec une corrélation de Spearman variant de $0,383$ à $0,426$, obtiennent toujours de meilleurs résultats que les modèles de référence, ayant une corrélation variant de $0,178$ à $0,372$. Plus spécifiquement, nous faisons les observations suivantes.

Similarité textuelle. Les modèles s'appuyant sur la proximité textuelle qui utilisent donc une similarité terme à terme exacte (BM25 et TfIdf), n'obtiennent pas de bonnes performances : il y a une très faible corrélation avec les jugements humains. Le coefficient de corrélation de Spearman des modèles BM25 et TfIdf atteint seulement des valeurs de $0,178$ et $0,199$. Ces résultats prouvent qu'il y a une forte discordance du vocabulaire ente les géotextes. De fait, nous devons nous tourner vers d'autres approches pour résoudre la tâche de similarité.

Similarité hiérarchique. Les modèles de l'état-de-l'art utilisant la proximité hiérarchique des catégories des POIs ($\mathbf{SIM}_{WP}$ et $\mathbf{SIM}_{LC}$) comme critère de similarité obtiennent des résultats prometteurs, avec une corrélation de $0,372$ (resp. $0,342$) pour le modèle $\mathbf{SIM}_{WP}$ (resp. $\mathbf{SIM}_{LC}$), sans toutefois dépasser les modèles utilisant les plongements lexicaux. Toutefois ces résultats sont surprenants car les humains utilisent généralement des informations beaucoup plus riches que les catégories (et leur hiérarchie) pour analyser la similarité des POIs. Il semble donc évident que des POIs partageant la même hiérarchie, c.-à-d. ayant plus ou moins les mêmes catégories et sous-catégories, sont proches sémantiquement, et partagent des motifs communs.

Similarité sémantique. Finalement, les meilleures performances sont obtenues avec les plongements lexicaux traditionnels et spatiaux. Plus spécifiquement, l'utilisation des représentations distribuées traditionnelles $\mathbf{W}$ pour représenter les géotextes a permis d'améliorer l'ordonnancement des POIs, puisque nous obtenons une corrélation de Spearman de $0,383$, contre $0,372$ pour le modèle $\mathbf{SIM}_{LC}$. Ce premier résultat met en avant l'importance de l'aspect sémantique pour résoudre la tâche de similarité. Cet aspect sémantique peut être capturé de plusieurs façons. Dans cette évaluation expérimentale, nous avons testé deux approches : l'une via l'agrégation de plongements lexicaux (ρ_{sim}-$\mathbf{W}$, ρ_{sim}-$\mathbf{W}^s_{kde}$ et ρ_{sim}-$\mathbf{W}^s_{km}$), l'autre via l'apprentissage des représentations des géotextes dans un espace latent (Doc2Vec). En comparant les performances des modèles ρ_{sim}-x et Doc2Vec, nous remarquons que la qualité des résultats dépend fortement de la méthode utilisée pour représenter les géotextes. En effet, le coefficient de corrélation du modèle Doc2Vec n'atteint qu'une valeur de $0,213$, ce qui est moins efficace que les modèles de similarité hiérarchique. Enfin, concernant les plongements lexicaux spatiaux $\mathbf{W}^s_{kde}$ et $\mathbf{W}^s_{km}$, nous remarquons dans le Tableau 4.2, qu'ils surpassent eux aussi très nettement les modèles de référence, mais aussi le modèle de représentation utilisant les plongements lexicaux traditionnels, en se rapprochant des jugements d'experts humains. La corrélation de Spearman des modèles ρ_{sim}-$\mathbf{W}^s_{kde}$ et ρ_{sim}-$\mathbf{W}^s_{km}$ sont respectivement de $0,426$ et $0,402$ contre $0,383$ pour le modèle traditionnel ρ_{sim}-$\mathbf{W}$. De fait, la régularisation des plongements lexicaux, via l'intégration de contraintes spatiales, permet très clairement d'améliorer les représentations latentes des objets géotextuels. Toutefois, nous pouvons noter une différence de qualité entre les

plongements lexicaux issus de la méthode par *clustering* ($\mathbf{W}^s_{km}$) et ceux issus de la méthode probabiliste ($\mathbf{W}^s_{kde}$). Nous y reviendrons en détail dans la Section 4.3.2.

En résumé, les résultats indiquent que la méthode utilisée pour représenter les géotextes est primordiale pour résoudre notre tâche de similarité. Une similarité s'appuyant uniquement sur la proximité textuelle ou la proximité hiérarchique des POIs se révèle moins performante qu'une similarité fondée sur la proximité sémantique, au travers de l'agrégation de plongements lexicaux. Enfin, les résultats montrent clairement l'intérêt de prendre en compte des contraintes spatiales pour construire les représentations latentes des géotextes.

4.3 *Évaluation extrinsèque : prédiction sémantique de l'emplacement (QR3)*

L'objectif de cette seconde évaluation est de mesurer l'efficacité des représentations des géotextes dans une tâche d'appariement, quant à leur capacité à saisir les signaux de pertinence. Pour mener cette évaluation extrinsèque, nous considérons la tâche de prédiction sémantique de l'emplacement (Section 2.3 du Chapitre 2). Cette tâche consiste à apparier des géotextes, ici des tweets, à des objets spatiaux sémantiquement liés représentés par des POIs.

Nous adoptons deux méthodes d'évaluation impliquant l'utilisation de plongements lexicaux : le *réordonnancement de POIs* et l'*expansion de tweets*.

Réordonnancement de POIs

Cette méthode vise à améliorer le score de pertinence d'un document, ici un géotexte représenté par un POI, avec un score supplémentaire, issu des plongements lexicaux appris. Pour cela, nous adaptons le modèle CLASS proposé par Zhao *et al.* (2016), défini par l'Équation 4.18, qui combine initialement un score d'appariement s'appuyant sur la distance, avec un modèle de langue. Notre choix est motivé par le fait qu'il comporte une composante d'appariement des mots ($\prod_{w \in t} P(w|p)$) qui permet aisément d'injecter des représentations distribuées d'objets géotextuels dans le calcul du score d'appariement.

$$Score(t, p) \propto exp\left\{-\frac{d(t.l, p.l)^2}{2\sigma^2}\right\} \times \prod_{w \in t} P(w|p) \tag{4.18}$$

Plus précisément, pour un objet géotextuel $o = [w_1^{(o)}, \cdots, w_m^{(o)}]$ donné, p. ex. un tweet t ou un POI p, nous commençons par déterminer sa représentation distribuée $\widehat{o}$ (Équation 3.13). Nous révisons ensuite le calcul du score d'appariement en

remplaçant le score donné par le modèle de langue ($\prod_{w \in t} P(w|p)$) par la similarité du cosinus de la paire de représentations distribuées $(\widehat{t}, \widehat{p})$:

$$Score(t, p) \propto exp\left\{-\frac{d(t.l, p.l)^2}{2\sigma^2}\right\} \times sim(\widehat{t}, \widehat{p}) \tag{4.19}$$

où σ^2 est la variance empirique de la distance d et $\widehat{t}$ et $\widehat{p}$ sont respectivement les représentations distribuées du tweet t et du POI p. Par la suite, $Score(t, p)$ sera noté CM.

Expansion de tweets

Cette méthode consiste à réécrire le tweet, en exploitant la proximité de ses mots dans l'espace des plongements lexicaux. L'intuition est que la pertinence pourrait être déterminée en calculant les similarités entre les représentations des tweets d'un côté et les représentations des mots de l'autre. Nous suivons l'approche d'expansion de requête proposée par Zamani et Croft (2016). Celle-ci consiste à réécrire le tweet en exploitant la proximité des mots dans l'espace des représentations distribuées.

$$Score(t, p) \propto exp\left\{-\frac{d(t.l, p.l)^2}{2\sigma^2}\right\} \times \prod_{w \in t^*} P(w|p) \tag{4.20}$$

avec t^* le tweet étendu. Par la suite, $Score(t, p)$ sera noté CE. Les mots étendus sont sélectionnés comme suit :

1. nous déterminons un ensemble de mots candidats $V \in \mathcal{V}$, obtenu à partir des mots contenus dans une liste ordonnée de POIs classés par le modèle Class pour le tweet t ;
2. nous calculons une distribution probabiliste en utilisant l'espace sémantique du tweet. Formellement, la probabilité de chaque terme w, étant donné la représentation distribuée $\widehat{t}$ du tweet, est calculée par :

$$p(w|\widehat{t}) = \frac{sim(\mathbf{w}, \widehat{t})}{\sum_{w' \in V} sim(w', \widehat{t})} \quad ; \tag{4.21}$$

3. nous considérons les top m termes ayant la probabilité $p(w|\widehat{t})$ la plus élevée, pour étendre le tweet.

Pour évaluer la qualité extrinsèque des plongements lexicaux spatiaux, nous réalisons donc une tâche de prédiction sémantique de l'emplacement selon les deux approches décrites ci-dessus. De ce fait, pour chaque tweet t de la collection d'évaluation, les 200 POIs les plus proches spatialement sont sélectionnés et retenus comme candidats potentiels (Acc@200 = 100%). L'ordonnancement via les modèles de référence et scénarios s'appuie donc sur cette liste.

Dans les sections suivantes, nous commençons par définir le cadre expérimental de notre évaluation dans la Section 4.3.1 puis nous discutons les résultats obtenus dans la Section 4.3.2.

4.3.1 *Cadre expérimental*

Dans la suite, nous décrivons le protocole d'évaluation permettant de répondre à **QR3**. Nous commençons par présenter dans la Section 4.3.1.1 les jeux de données utilisés pour l'évaluation extrinsèque des représentations. Nous continuons en présentant les scénarios d'évaluation dans la Section 4.3.1.2 et les modèles de référence dans la Section 4.3.1.3. Enfin, dans la Section 4.3.1.4, nous détaillons les mesures utilisées pour évaluer la qualité de nos plongements lexicaux spatiaux.

4.3.1.1 *Jeux de données*

Pour réaliser l'évaluation extrinsèque, et donc mesurer la qualité des plongements lexicaux spatiaux à l'aide d'une tâche de prédiction sémantique de l'emplacement, nous avons utilisé la collection de tweets géotaggés publiée par Zhao *et al.* (2016) et décrite dans la Section 4.2.1.1.

L'objectif de notre tâche d'évaluation étant d'associer des POIs avec des tweets, nous n'avons conservé qu'un échantillon de tweets sémantiquement reliés à un POI. Pour constituer cette collection, Zhao *et al.* (2016) ont effectué une annotation permettant de déterminer si les tweets étaient liés à des POIs ou non. D'une part, grâce à l'aide de trois annotateurs, et d'autre part en recherchant des *check-in tweets*. Les *check-in tweets* sont des tweets publiés depuis l'application Foursquare lorsqu'un utilisateur effectue le *check-in* d'un POI. Ils se présentent généralement sous la forme « XXX (@ YYY @ ZZZ) » où « XXX » est le texte rédigé par l'utilisateur, et « (@ YYY @ ZZZ) » la partie relative à Foursquare, où « YYY » contient le nom du POI et « ZZZ » son emplacement géographique (quartier, ville, région, etc.). Il convient de noter que par la suite, la partie relative à Foursquare a été retirée des *check-in tweets*.

Finalement, puisque la tâche d'évaluation repose sur des tweets liés à des POIs, nous avons écarté les tweets non pertinents (c.-à-d. non liés à des POIs) et sélectionné un échantillon de tweets pour constituer l'ensemble d'évaluation. Ce dernier se compose de 7 364 tweets sémantiquement reliés à un POI.

4.3.1.2 *Scénarios d'évaluation*

Dans le cadre de cette évaluation expérimentale, et pour chacune des tâches décrites ci-dessus, nous évaluons trois configurations, une première utilisant les

plongements lexicaux traditionnels (notée x-$\mathbf{W}$), une deuxième utilisant les plongements lexicaux spatiaux déterminés à l'aide de la méthode de *clustering* (notée x-$\mathbf{W}^s_{km}$), et une troisième utilisant les plongements lexicaux spatiaux déterminés à l'aide de la méthode probabiliste (notée x-$\mathbf{W}^s_{kde}$). Pour les scénarios adoptant les représentations $\mathbf{W}^s_{km}$, nous utilisons le sens local $w^s_{i,j}$ le plus proche en minimisant la distance Haversine entre l'emplacement du géotexte et le barycentre du mot $\mathcal{B}_{i,j}$. Pour les scénarios adoptant les représentations $\mathbf{W}^s_{kde}$, nous utilisons, s'il existe, le sens local $w^s_{i,j}$ présent dans la grille associée au géotexte considéré. Ces différentes configurations sont résumées dans le Tableau 4.3.

Tâche	**Scénarios**	**Commentaires**
Réordonnancement de POIs	CM-$\mathbf{W}$	Plongements lexicaux
	CM-$\mathbf{W}^s_{km}$	Plongements lexicaux spatiaux (*k-moyennes*)
	CM-$\mathbf{W}^s_{kde}$	Plongements lexicaux spatiaux (*KDE*)
Expansion de tweets	CE-$\mathbf{W}$	Plongements lexicaux traditionnels
	CE-$\mathbf{W}^s_{km}$	Plongements lexicaux spatiaux (*k-moyennes*)
	CE-$\mathbf{W}^s_{kde}$	Plongements lexicaux spatiaux (*KDE*)

Tableau 4.3 – Configuration des scénarios utilisés pour l'évaluation extrinsèque.

4.3.1.3 *Modèles de référence*

Pour répondre à **QR3** et évaluer la qualité extrinsèque de nos plongements lexicaux spatiaux $\mathbf{W}^s_{km}$ et $\mathbf{W}^s_{kde}$, nous comparons les résultats de nos scénarios avec des modèles de référence. Ces modèles sont classés en trois catégories : *(1)* proximité spatiale (Dist) ; *(2)* appariement textuel (BM25) ; *(3)* une combinaison des deux (Class). Les caractéristiques de ces modèles sont les suivantes :

— **Dist** (De Smith et Goodchild, 2007) : modèle d'ordonnancement spatial ;
— **BM25** (Robertson et Jones, 1976) : modèle probabiliste de référence, largement utilisé en RI textuelle ;
— **Class** (Zhao *et al.*, 2016) : modèle d'ordonnancement de POIs qui combine la distance spatiale avec un modèle de langue.

4.3.1.4 *Mesures d'évaluation*

Chaque scénario proposé, nous permet de calculer pour un tweet t et un POI p donné, un score d'appariement $Score(t, p)$. Ainsi, pour chaque tweet, nous calculons les scores d'appariement entre le tweet t et un ensemble de POIs candidats.

Pour évaluer l'efficacité des modèles, nous ordonnons pour chaque tweet, les scores d'appariement calculés pour ses POIs candidats. Ensuite, nous comparons les résultats obtenus avec la vérité-terrain en utilisant deux mesures : le rang réciproque moyen (ou *mean reciprocal kank*), noté *MRR* (Craswell, 2009), et la précision (ou *accuracy*), notée *Acc@k* (Powers, 2011).

La *MRR* est calculée comme suit :

$$MRR = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \frac{1}{rang_i} \tag{4.22}$$

où $rang_i$ est la position du POI pertinent pour le i^{e} tweet dans la liste ordonnée des POIs candidats ; $\mathcal{T}$ est l'ensemble des tweets de la collection de test.

La définition de l'*Acc@k* est la suivante :

$$Acc@k = \frac{|t \in \mathcal{T} : t.l^* \in L_k(t)|}{|\mathcal{T}|} \tag{4.23}$$

où $t.l^*$ est le POI pertinent pour le tweet t ; $L_k(t)$ sont les tops-k POIs candidats issus de la liste ordonnée calculée pour le tweet t. Le résultat est considéré comme correct si le POI pertinent est dans les tops-k POIs de la liste. Étant donnée la tâche d'évaluation considérée, à savoir la prédiction sémantique de l'emplacement, il convient de noter que de faibles valeurs de k sont particulièrement considérées.

4.3.2 *Résultats*

Le troisième et dernier objectif de cette évaluation expérimentale est de mesurer l'efficacité des plongements lexicaux spatiaux dans une tâche d'appariement. Plus précisément, nous analysons la qualité des représentations régularisées via la tâche de prédiction sémantique de l'emplacement. Le Tableau 4.4 résume les résultats empiriques obtenus en termes de *MRR* et d'*Acc@k* ($k = 1, 5, 10$). Nous présentons les valeurs obtenues pour chacunes des mesures, ainsi que les taux d'amélioration relatifs (%Chg).

Dans l'ensemble, nous remarquons à partir du Tableau 4.4, que les scénarios impliquant l'appariement avec les représentations distribuées traditionnelles (CM-$\mathbf{W}$ et CE-$\mathbf{W}$) et régularisées (CM-$\mathbf{W}^s_{kde}$, CE-$\mathbf{W}^s_{kde}$, CM-$\mathbf{W}^s_{km}$ et CE-$\mathbf{W}^s_{km}$) dépassent largement les modèles de référence. Le modèle CE-$\mathbf{W}^s_{kde}$ affiche de meilleurs résultats en termes de *MRR* (0, 618) avec des accroissements relatifs compris entre 20, 23% et 46, 10% par rapport aux modèles Dist, Bm25 et Class. Plus précisément, ce scénario permet un appariement tweet-POI plus efficace : plus de 53% des tweets sont associés à leur POI correspondant dès les premiers résultats (c.-à-d. *Acc*@1), contre 43% pour le modèle Dist. Les résultats sont sensiblement identiques pour le scénario CE-$\mathbf{W}^s_{km}$, avec une *MRR* de 0, 604, soit des taux d'amélioration relatifs compris

	MRR		Acc@1		Acc@5		Acc@10	
Model	**Valeur**	**%Chg**	**Valeur**	**%Chg**	**Valeur**	**%Chg**	**Valeur**	**%Chg**
Scénarios s'appuyant sur les représentations régularisées (KDE) $\mathbf{W}^s_{kde}$								
CM-W$^s_{kde}$	0,589	- +4,92% †	0,507	- +5,52% †	0,694	- +4,61% †	0,730	- +9,73% †
CE-W$^s_{kde}$	0,618	-4,69% * -	0,535	-5,23% * -	0,726	-4,41% * -	0,801	-8,86% * -
Scénarios s'appuyant sur les représentations régularisées (k-moyennes) $\mathbf{W}^s_{km}$								
CM-W$^s_{km}$	0,577	+2,08% * +7,11% †	0,489	+3,68% * +9,41% †	0,675	+2,81% * +7,56% †	0,717	+1,81% * +11,72% †
CE-W$^s_{km}$	0,604	-2,48% * +2,32% †	0,515	-1,55% * +3,88% †	0,698	-0,57% +4,01% †	0,775	-5,81% * +3,35% †
Scénarios s'appuyant sur les représentations traditionnelles								
CM-W	0,521	+13,05% * +18,62% †	0,413	+22,76% * +29,54% †	0,640	+8,44% * +13,44% †	0,706	+3,40% * +13,46% †
CE-W	0,563	+4,62% * +9,77% †	0,470	+7,87% * +13,83% †	0,659	+5,31% * +10,17% †	0,702	+3,99% * +14,10% †
Modèles de référence								
DIST	0,514	+14,59% * +20,23% †	0,430	+17,91% * +24,42% †	0,605	+14,71% * +20,00% †	0,686	+6,41% * +16,76% †
BM25	0,423	+39,24% * +46,10% †	0,307	+65,15% * +74,27% †	0,668	+3,89% * +8,68% †	0,831	-12,15% * -3,61% †
CLASS	0,585	+16,17% * +21,89% †	0,501	+26,43% * +33,42% †	0,681	+11,22% * +16,35% †	0,698	+4,58% * +14,76% †

Tableau 4.4 – Comparaison des performances des modèles **CM-W**$^s_{kde}$ et **CE-W**$^s_{kde}$ par rapport aux différents scénarios et modèles de référence. La différence significative par rapport au modèle **CM-W**$^s_{kde}$ est déterminée par le test t de Welch ($* : p < 0,01$). De même, la différence significative par rapport au modèle **CE-W**$^s_{kde}$ est déterminée par le test t de Welch ($† : p < 0,01$).

entre 14,59% et 39,24% par rapport aux modèles de référence. De même, plus de 51% des tweets sont correctement associés à leur POI correspondant dès les premiers résultats (c.-à-d. *Acc*@1). Enfin, nous observons que l'injection de représentations distribuées (traditionnelles ou régularisées) permet d'améliorer l'efficacité du modèle CLASS. En effet, la *MRR* des scénarios CE-**W**, CE-**W**$^s_{km}$ et CE-**W**$^s_{kde}$ augmente respectivement de 11,05%, 19,13% et 21,89% par rapport au modèle CLASS ($p < 0,01$). Ces premières observations confirment ainsi l'effet positif de la régularisation spatiale pour la construction des objets géotextuels utilisés pour résoudre la tâche de prédiction sémantique de l'emplacement.

En examinant plus particulièrement les scénarios impliquant des représentations distribuées régularisées CM-$\mathbf{W}^s_{kde}$ et CE-$\mathbf{W}^s_{kde}$, nous pouvons remarquer que le scénario CE-$\mathbf{W}^s_{kde}$ améliore la MRR de $4,92\%$ et l'Acc@1 de $5,52\%$ comparé au scénario CM-$\mathbf{W}^s_{kde}$. L'Acc@1 (resp. MRR) est passée de $0,507$ (resp. $0,589$) à $0,535$ (resp. $0,618$). De même, en analysant les scénarios CM-$\mathbf{W}^s_{km}$ et CE-$\mathbf{W}^s_{km}$, nous remarquons que le scénario CE-$\mathbf{W}^s_{km}$ améliore la MRR de $4,68\%$ et la mesure Acc@1 de $5,32\%$ comparé au scénario CM-$\mathbf{W}^s_{km}$. Ces résultats pourraient s'expliquer par l'approche utilisée pour injecter les plongements lexicaux spatiaux. Tandis que dans les scénarios CE-$\mathbf{W}^s_{kde}$ et CE-$\mathbf{W}^s_{km}$ les vecteurs sont utilisés de façon indépendante pour étendre la description du tweet avant l'étape d'appariement, ils sont plutôt agrégés dans le scénario CM-$\mathbf{W}^s_{kde}$ et CM-$\mathbf{W}^s_{km}$ pour construire des représentations de tweets et de lieux d'intérêts pondérées par l'IDF, ce qui génère un biais de représentation. Cette observation nous montre clairement l'impact positif de l'utilisation intrinsèque des plongements lexicaux augmentés par les ressources spatiales.

En comparant maintenant le modèle CE-$\mathbf{W}^s_{kde}$ impliquant les plongements lexicaux spatiaux issus de la méthode probabilistes avec le modèle CE-$\mathbf{W}^s_{km}$ dont les plongements lexicaux spatiaux ont été déterminés à l'aide de la méthode de *clustering*, nous remarquons que le modèle CE-$\mathbf{W}^s_{kde}$ dépasse le modèle CE-$\mathbf{W}^s_{km}$, quels que soient les mesures considérées. Plus spécifiquement, avec le modèle CE-$\mathbf{W}^s_{km}$, $51,5\%$ des tweets sont correctement associés à leur POI contre $53,5\%$ pour le modèle CE-$\mathbf{W}^s_{kde}$, lorsque nous considérons les résultats dans le top-1 (c.-à-d. Acc@1). Cela représente une amélioration très significative de $3,88\%$. En considérant des valeurs de k plus élevées (p. ex. $k = 10$), plus de 8 tweets sur 10 sont correctement associés à leur POI lorsque nous utilisons les plongements lexicaux $\mathbf{W}^s_{kde}$, contre un peu plus de 7 tweets pour les plongements lexicaux $\mathbf{W}^s_{km}$. Ces observations rejoignent celles formulées dans la Section 4.2.2 lors de l'évaluation intrinsèque des plongements lexicaux : les représentations distribuées $\mathbf{W}^s_{kde}$ sont de meilleure qualité que les représentations distribuées $\mathbf{W}^s_{km}$. La méthode employée pour déterminer les sens locaux des mots influence donc grandement la qualité des plongements lexicaux corrigés, et donc la qualité de représentation des objets géotextuels. Cet écart de performance peut notamment s'expliquer par les spécificités de l'algorithme des k-moyennes, qui rend cette méthode moins robuste que la méthode d'estimation par noyau. Les principaux inconvénients du partitionnement en k-moyennes sont notamment la sélection du nombre de classes k à retenir, sa dépendance aux valeurs initiales comme l'ont discuté Celebi *et al.* (2013) ou encore sa sensibilité aux valeurs aberrantes, qui ont tendance à attirer les centroïdes ou à être considérées comme des groupes à part entière. Ces principaux obstacles sont résolus par la méthode d'estimation par noyaux, nous permettant ainsi d'obtenir de meilleurs partitionnements.

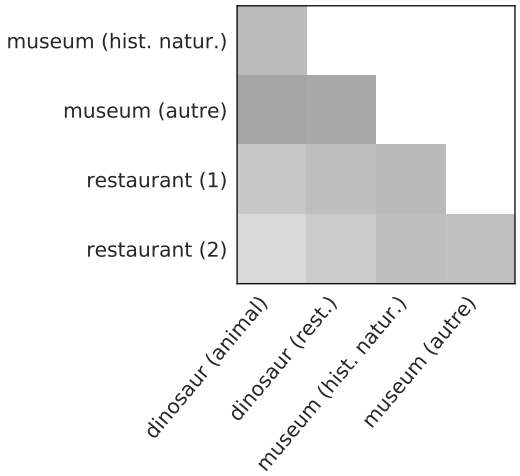


Figure 4.12 – Similarité du cosinus des représentations distribuées régularisées, pour les mots *dinosaur* (animal et chaîne de restauration), *museum* (musée d'histoire naturelle et un autre musée) et *restaurant* (deux types de restaurants).

Pour conclure l'analyse de la qualité des représentations distribuées régularisées des mots, nous illustrons dans la Figure 4.12, la carte thermique des similarités entre un sous-ensemble des mots issus de la Figure 4.10. Plus précisément, nous déclinons dans cette figure les sens locaux et régularisés que nous avons identifié grâce à l'Algorithme 1 et la méthode de *clustering*. Par exemple, nous avons identifié deux sens distincts du mot *museum* selon sa répartition spatiale. Nous pouvons constater, que le sens du mot *dinosaur (rest.)* est plus proche des représentations des restaurants (*restaurant (1) et (2)*) que ne l'est celui de *dinosaur (animal)*. Ceci montre bien qualitativement le rapprochement entre le résultat de la régularisation proposée et notre hypothèse initialement énoncée.

5 Bilan

Nous avons présenté dans ce chapitre notre contribution portant sur l'apprentissage de plongements lexicaux augmentés par des connaissances issues des répartitions spatiales des mots. Dans un premier temps, nous avons détaillé les limites des plongements lexicaux et l'intérêt de calculer des plongements lexicaux sensibles à la dimension spatiale.

C'est dans cette optique que nous avons proposé une méthode générale pour l'apprentissage a posteriori de plongements lexicaux, en intégrant des contraintes spatiales. Contrairement aux modèles de régularisation traditionnels, où chaque

représentation distribuée est corrigée par un ensemble d'éléments (p. ex. mots, documents, concepts) pour obtenir une nouvelle représentation, nous avons d'abord déterminé, pour chaque mot, ses empreintes spatiales (c.-à-d. sens locaux) pour lesquels nous avons associé des représentations distribuées initiales que nous avons ensuite corrigées. Les principales caractéristiques de notre méthode sont les suivantes. Pour détecter les sens locaux des mots, nous avons proposé deux méthodes de partitionnement, l'une via une méthode de *clustering* à l'aide de l'algorithme des k-moyennes, l'autre via une méthode probabiliste avec la méthode de Parzen-Rosenblatt (KDE). Nous avons ensuite régularisé les plongements lexicaux en tirant profit des répartitions spatiales des mots voisins et distants.

Nous avons mené une évaluation expérimentale pour mesurer la qualité de nos plongements lexicaux spatiaux, selon plusieurs aspects. Plus précisément, nous avons commencé par évaluer la qualité des représentations au moyen d'une tâche de similarité de géotextes. Les résultats de l'analyse ont montré l'intérêt des plongements lexicaux spatiaux pour construire les représentations latentes des géotextes permettant ainsi de résoudre efficacement la tâche de similarité. Nous avons ensuite évalué l'effet de l'utilisation des représentations régularisées dans la tâche de prédiction sémantique de l'emplacement. Les résultats montrent que la méthode proposée permet d'améliorer significativement les performances de recherche comparativement à des modèles de l'état-de-l'art. Par ailleurs, nous avons aussi étudié l'effet des différentes méthodes de partitionnement (probabiliste et *clustering*) sur la qualité des prédictions.

Exploitant l'hypothèse de la variation des représentations des mots et des géotextes selon la dimension spatiale, nous proposons dans le chapitre suivant, une architecture neuronale dite de bout-en-bout permettant de résoudre efficacement la tâche de prédiction sémantique de l'emplacement, en tenant compte des caractéristiques textuelles et spatiales des géotextes.

5

MODÈLE NEURONAL POUR LA PRÉDICTION SÉMANTIQUE DE L'EMPLACEMENT

Introduction

Nous présentons dans ce chapitre notre deuxième contribution qui aborde le problème de géoréférencement des documents. Plus spécifiquement, nous traitons la tâche de prédiction sémantique de l'emplacement à partir de données sociales. Nous l'avons vu dans le Chapitre 2, cette tâche reste encore aujourd'hui un défi, notamment à cause de la nature bruitée des géotextes qui rend inefficace les modèles d'appariement traditionnels s'appuyant sur un appariement exact des termes (p. ex. TfIdf, BM25). Les approches détaillées dans la Section 2.3 pallient ce problème en élaborant des modèles de langue spécifiques pour représenter les tweets et les POIs. Toutefois, la principale difficulté de ces approches est de savoir comment formaliser et estimer les probabilités conditionnelles des géotextes à partir des textes, et les probabilités de traduction entre les mots.

Confortés par les performances des approches neuronales pour la représentation distribuée de mots et l'appariement de textes et de géotextes, tels que détaillés dans le Chapitre 3, nous proposons dans cette contribution un modèle axé sur l'interaction pour la résoudre la tâche de prédiction sémantique de l'emplacement. L'objectif de notre modèle est de pallier les lacunes des modèles d'appariement classiques pour résoudre ce type de tâche, en utilisant conjointement les aspects spatiaux et thématiques des documents.

L'organisation de ce chapitre est la suivante. Nous détaillons en Section 1 le contexte de cette contribution et notre positionnement. Nous formalisons dans la Section 2 la tâche d'annotation. Dans la Section 3, nous donnons un aperçu général du modèle proposé et décrivons ses différents composants. Le protocole d'évaluation de notre contribution est détaillé dans la Section 4. Nous présentons et discutons ensuite dans la Section 5 les performances obtenues par notre modèle

sur des jeux de données réels pour le valider expérimentalement. Enfin, nous concluons ce chapitre dans la Section 6.

1 Contexte et motivations

Au cours de la dernière décennie, les RSNs tels que *Facebook*, *Twitter* ou *Instagram* sont devenus des espaces populaires d'échanges permettant d'établir des liens sociaux et de partager de l'information textuelle, audio et vidéo. Avec la connectivité croissante des utilisateurs, de plus en plus de géotextes (c.-à-d. des messages associés à des coordonnées GPS, tels que des tweets et des photos géolocalisés) sont créés quotidiennement. Les RSNs aident à combler le fossé entre les utilisateurs de médias sociaux en ligne d'une part, et d'autre part, le monde physique qui comprend des lieux physiques du monde réel. Dans le cadre de cette contribution, nous étudions le problème de la prédiction sémantique de l'emplacement, qui consiste à déterminer, pour un géotexte donné, l'emplacement sur lequel il se focalise, c.-à-d. qui constitue le sujet abordé.

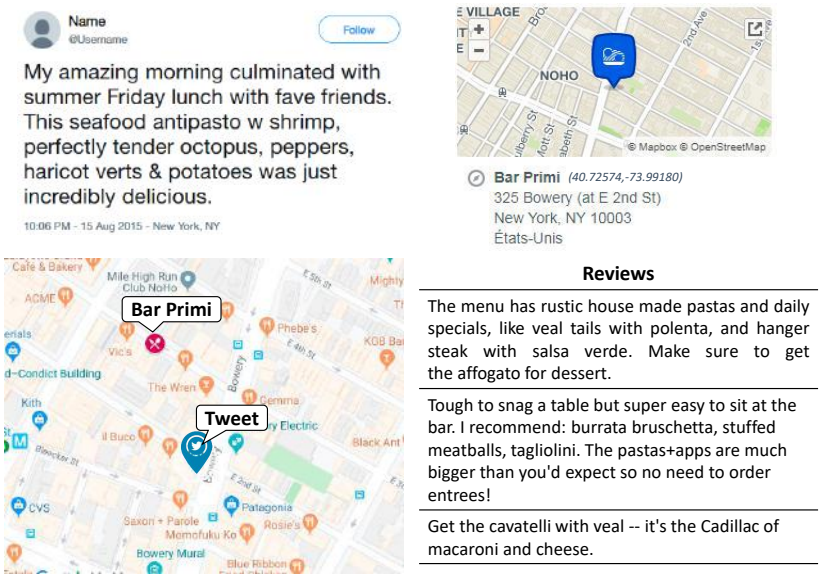


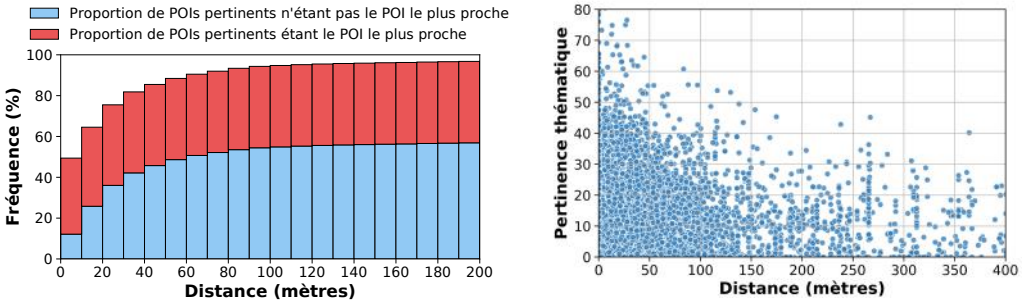
Figure 5.1 – Exemple d'un tweet associé à un POI et ses candidats potentiels.

Pour illustrer cette tâche, considérons le tweet géotaggé de la Figure 5.1 qui porte un commentaire sur le *Bar Primi*, un restaurant italien situé à New York City. Comme nous pouvons le remarquer, l'appariement textuel s'appuyant sur une cor-

respondance exacte des termes ne permettrait pas d'identifier le POI pertinent. En effet, le contenu textuel du tweet ne coïncide pas avec la description textuelle du POI pertinent, qui inclut son nom, son type ainsi que quelques critiques publiées par les utilisateurs de Foursquare. En outre, nous pouvons observer que l'emplacement de l'utilisateur au moment où il publie son tweet n'est pas plus utile, puisque l'emplacement du POI ne correspond pas à celui de l'utilisateur. Pire encore, il y a de nombreux POIs candidats non pertinents entre l'emplacement de l'utilisateur et le POI concerné. Tous ces éléments rendent la tâche de prédiction sémantique de l'emplacement encore plus difficile à aborder.

Dans la littérature, peu de travaux se sont intéressés à cette tâche, malgré son intérêt pour de nombreuses applications de la vie quotidienne telles que la recommandation d'événements (Yuan *et al.*, 2013; Yin *et al.*, 2015) ou de lieux d'intérêts (Deveaud *et al.*, 2015; Bothorel *et al.*, 2018), ou encore le résumé spatio-temporel (Rakesh *et al.*, 2013; Mallela *et al.*, 2017). Comme rapporté dans le Chapitre 2, cette tâche est fondamentalement différente des tâches de prédiction de l'emplacement du contenu (Section 2.1) et de l'emplacement mentionné dans le texte (Section 2.2), notamment pour les raisons suivantes : *(1)* l'emplacement d'émission du tweet peut être différent de l'emplacement sémantique sur lequel il se focalise ; *(2)* plusieurs lieux périphériques pourraient être mentionnés dans le tweet, mais au maximum un seul d'entre eux est le lieu principal. La plupart des approches proposées jusqu'à présent s'appuient sur des modèles de langues (Dalvi *et al.*, 2009a,b, 2012) ou des modèles bayésiens supervisés (Zhao *et al.*, 2016). Le principal inconvénient de ces approches réside dans la difficulté et le coût de l'estimation des probabilités conditionnelles de la pertinence thématique à travers les régions géographiques. Quelle que soit l'approche utilisée, il ressort clairement que les aspects spatiaux et thématiques sont deux facteurs à prendre en compte. Néanmoins, nous constatons qu'il peut exister des dépendances non linéaires complexes entre les deux, qui ne sont pas prises en compte par les approches de l'état-de-l'art mais qu'il conviendrait d'exploiter.

Pour nous en convaincre, et illustrer la complexité de la tâche, nous avons effectué une analyse préliminaire en utilisant la vérité terrain de couples tweet-POI pertinents contenus dans un jeu de données du monde réel publié par Zhao *et al.* (2016) et décrit dans la Section 4.2.1.1 du Chapitre 4. Cette analyse, illustrée dans la Figure 5.2, vise à montrer l'importance capitale du facteur spatial pour sélectionner les paires pertinentes, mais aussi son interaction avec le facteur thématique. La Figure 5.2a présente la distribution cumulée croissante de la distance séparant le tweet de son POI pertinent. La Figure 5.2b présente la distribution bivariée des scores de pertinence spatiale calculés selon la distance géographique entre le tweet et le POI pertinent (axe des x), et des scores de pertinence thématique calculée en utilisant le modèle BM25 (Robertson et Jones, 1976) (axe des y).



(a) Distribution de la distance

(b) Distribution bivariée des scores de pertinence thématique et spatiale

Figure 5.2 – Analyse des scores de pertinence thématique et spatiale en utilisant la vérité terrain de paires tweet-POI issues du monde réel (Zhao *et al.*, 2016).

Les résultats présentés dans la Figure 5.2a montrent clairement qu'il existe en effet une forte concentration de paires pertinentes (plus de 96%) ayant une distance inférieure à 100 mètres, suggérant que la distance joue un rôle crucial pour la prédiction sémantique de l'emplacement. Cependant, pour évaluer si le POI correspondant est systématiquement le POI le plus proche du tweet, nous avons divisé les paires pertinentes (t, p) en deux catégories : les paires telles que p est le POI le plus proche de t (en rouge) et les paires telles que p n'est pas le POI le plus proche de t (en bleu). Les résultats montrent que la stratégie naïve qui consiste à prédire le POI le plus proche serait inefficace puisqu'elle conduirait à une précision d'environ 40%. Autrement dit, sélectionner le POI le plus proche du tweet comme POI pertinent n'est pas une solution efficace. En étudiant la Figure 5.2b, nous remarquons que le facteur de pertinence thématique ne permet pas d'expliquer la pertinence tweet-POI pour près de 60% des paires. Pire encore, 16% des paires présentent un score de pertinence thématique nul. Au-delà des facteurs de pertinence individuels, il ressort également en étudiant la forme de la distribution, que les dimensions de pertinence spatiale et thématique sont indépendantes. Pour s'en assurer, nous avons calculé la corrélation de Pearson entre les paires situées dans la région inférieure ($\sigma = 0,013$, $p < 0,01$) ainsi ceux de la distribution globale ($\sigma = 0,010$, $p < 0,01$). Pour résumer, cette analyse a mis en évidence d'une part, l'apport mais aussi les limites de la distance pour l'appariement de tweets et de POIs, et d'autre part, les limites d'un appariement thématique en raison de la présence d'un fossé sémantique entre les tweets et les POIs.

Guidés par ces éléments, nous sommes convaincus qu'utiliser des approches neuronales pour réaliser l'appariement de tweets géotaggés avec des POIs permettrait de pallier les limites des modèles de l'état-de-l'art. Notre raisonnement est porté par les enseignements tirés des travaux antérieurs (Onal *et al.*, 2017) montrant que les approches s'appuyant sur les réseaux de neurones sont très efficaces

pour l'appariement des textes. Onal *et al.* (2017) ont montré que ces approches s'attaquent avec succès aux problèmes de rareté des données et de discordance de vocabulaire, qui sont les principaux obstacles rencontrés dans la tâche de prédiction sémantique de l'emplacement. Par ailleurs, pour exploiter conjointement les facteurs de distance et de pertinence thématique, nous proposons d'effectuer un appariement à deux niveaux de granularité, local et global. Cette démarche est motivée par les observations suivantes :

Dans le Chapitre 4, nous avons formulé l'Observation 1 en disant que les sens de certains mots diffèrent selon les régions où ils sont utilisés. Comme corollaire à cette observation, nous avançons l'hypothèse suivante :

Hypothèse 2. Les mots qui occurent dans des objets géotextuels spatialement proches ont tendance à avoir des significations similaires. De manière analogue, plus les mots sont spatialement proches, en ce qui concerne la distance entre leurs objets associés, plus leurs significations sont proches. Intuitivement, la force de la relation sémantique entre les mots « *football* » et « *coupe* » devrait être plus importante, selon le même sens sous-jacent du mot « *coupe* » (c.-à-d. une récompense), dans des localités autour de la France que dans d'autres localités moins impliquées dans la pratique du football (p. ex. les États-Unis).

En suivant cette hypothèse, nous intégrons à notre modèle neuronal une matrice d'interaction calculée à partir des paires de mots qui apparaissent dans des couples tweet-POI candidats. Par ailleurs, nous proposons de corriger cette matrice par des distributions de noyau [1] issues des cooccurrences spatiales des mots. Nous pensons que ces interactions de grain fin pourraient révéler des sens locaux qui favoriseraient l'apprentissage de la similarité entre les objets auxquels elles appartiennent. En ce qui concerne l'exemple ci-dessus et abstraction faite de la distance entre l'emplacement du tweet et l'emplacement du POI candidat, un tweet comme « *Rien de mieux qu'un bar pour regarder un match de coupe du monde de football !* » est plus susceptible d'être associé à un bar situé en Europe qu'aux États-Unis.

Observation 2 (*Niveau global*)**.** Comme l'a souligné notre analyse précédente, la distance et la proximité thématique entre les géotextes, ainsi que leur interaction, sont essentielles pour sélectionner les paires pertinentes. Nous pouvons par exemple voir qu'un tweet comme « *Sur les sentiers de la guerre d'indépendance ! #fortgreenepark* », lié au POI « *Prison Ship Martyrs Monument* », un mémorial à New York, a une pertinence thématique très faible car ils ne partagent pas de termes communs alors qu'ils sont sémantiquement liés. De même, le POI spatialement le plus proche du tweet « *Journée ensoleillée ! #rafraichissement #starbucks* » est « *Paesanos Pizza* », un restaurant italien, alors que ce tweet est lié à un *Starbucks*.

1. Une distribution de noyau est une représentation non paramétrique de la fonction de densité de probabilité d'une variable aléatoire. Elle est définie par une fonction et un paramètre de lissage qui contrôlent le lissage de la courbe de densité.

Pour compléter les similarités de bas niveaux au niveau des mots des géotextes, nous proposons également de capturer les signaux de correspondance sémantique en exploitant les géotextes à un niveau de gros grain, c.-à-d. au niveau global du tweet et du POI. Ainsi, nous cherchons à saisir les interactions globales de correspondance entre les tweets et les POIs en fonction de leur distance géographique et de leur proximité sémantique globale afin d'apprendre la fonction d'appariement sémantique des tweets et des POIs.

Ainsi, dans cette seconde contribution, nous proposons de concevoir un modèle d'interaction pour répondre à la tâche de prédiction sémantique de l'emplacement. À notre connaissance, il s'agit d'une des premières approches neuronales combinant des interactions globales et locales pour l'appariement de géotextes. Nous présentons dans ce chapitre le modèle **SGM** (*Spatially-aware Geotext Matching*) qui repose sur les contributions suivantes :

— un réseau de neurones s'appuyant sur l'interaction de structures non linéaires qui associent des tweets et des POIs en tenant compte des interactions spatiales et textuelles. Nous considérons conjointement les interactions d'appariement local (signaux de pertinence) et les interactions d'appariement global (signaux sémantiques) ;
— l'intégration d'un facteur d'amortissement sur les interactions entre les mots du tweet et du POI afin de discriminer les similarités sémantiques des paires de mots en fonction de leur répartition géographique.

De plus, nous menons une évaluation expérimentale approfondie et des analyses qualitatives afin d'étudier et d'évaluer l'influence de chacun des composants du modèle proposé.

2 Définition du problème

Dans le Chapitre 4, nous avons abordé la notion d'objet géotextuel (Section 2.1) en détaillant deux types de géotextes, les POIs (Définition 4.2) et les tweets géotaggés (Définition 4.3). À partir de ces notions, nous définissons formellement la notion de géotexte lié à un POI ainsi que la problématique abordée dans ce chapitre, la tâche de prédiction sémantique de l'emplacement.

Définition 5.1 (Géotexte lié à un POI). Un géotexte est lié à un POI s'il se concentre et/ou mentionne un POI spécifique, comme nous l'avons illustré dans la Figure 5.1. Cette association peut être explicitement fournie (p. ex. via des publications sur les RSNs géodépendants tels que Foursquare) ou automatiquement identifiée à l'aide d'algorithmes de classification (Zhao *et al.*, 2016).

Problème (Tâche de prédiction sémantique de l'emplacement). Étant donné un géotexte o qui se focalise sur un POI, la tâche de prédiction sémantique de l'emplacement consiste à identifier le POI p sur lequel le géotexte o se focalise. Dans cette contribution, nous instancions le géotexte o avec un tweet géotaggé t (Définition 4.3). Il convient de noter que la tâche qui consiste à déterminer si un géotexte est sémantiquement relié à un POI n'entre pas dans le cadre de cette contribution.

Formellement, la tâche de prédiction sémantique de l'emplacement permet d'identifier l'unique POI p^*, qui est le premier d'un classement de POIs candidats renvoyés par une fonction d'appariement sémantique : $p^* = \max_{p_i \in \mathcal{P}} Score(t, p_i)$ où $\mathcal{P}$ contient l'ensemble des POIs candidats. *Score* désigne une fonction Ψ permettant de calculer un score d'appariement entre le tweet t et le POI candidat p. Elle est définie par $Score(t, p) = \Psi(\Phi(t), \Phi(p))$, où Φ est une fonction qui associe chaque objet à un vecteur de représentation.

Comme indiqué en introduction de ce chapitre, nous adoptons l'architecture d'un réseau de neurones d'interactions. Contrairement aux travaux précédents (Lu et Li, 2013; Guo *et al.*, 2016), le modèle réalise un processus décisionnel hiérarchique à travers la matrice d'interaction locale et les interactions globales objet-objet (*tweet, POI*). En ce qui concerne notre modèle **SGM**, la fonction Φ associe chaque géotexte à une séquence de plongements lexicaux ainsi qu'à une représentation d'objet établie en fonction de ses caractéristiques (spatiales et textuelles), tandis que la fonction Ψ représente un réseau neuronal alimenté par une matrice d'interaction augmentée par des connaissances spatiales et par des propriétés d'interactions entre les géotextes.

3 Architecture du réseau de neurones

Nous présentons dans cette section notre contribution permettant de résoudre la tâche de prédiction sémantique de l'emplacement décrite dans la Section 2. Un aperçu général de l'approche est d'abord présenté dans la Section 3.1. Nous définissons formellement sa structure dans les Sections 3.2 à 3.4. Enfin, nous détaillons la fonction de coût dans la Section 3.5.

3.1 *Aperçu général de la solution*

Dans la section précédente, nous avons introduit la tâche de prédiction sémantique de l'emplacement. À partir des observations et de l'Hypothèse 2 formulées dans la Section 1, nous proposons un modèle neuronal axé sur les interactions,

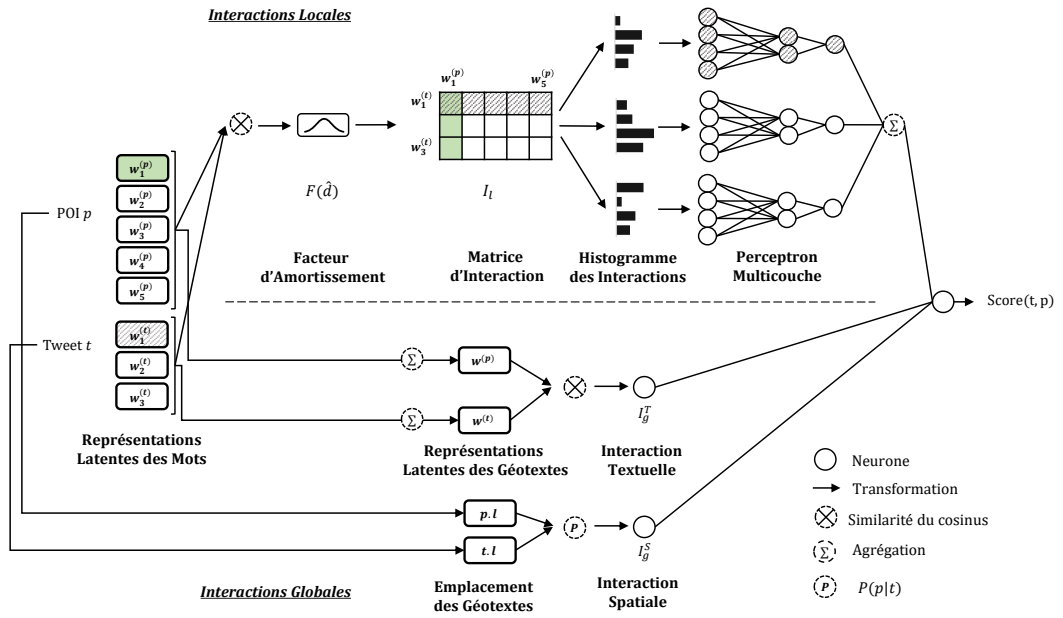


Figure 5.3 – Architecture du modèle **SGM**. La case verte correspond au 1er mot du POI p. La case hachurée correspond au 1er mot du tweet t.

appelé *Spatially-aware Geotext Matching* (**SGM**), conçu pour l'appariement d'objets géotextuels. Ce réseau, dit de « de bout en bout » (ou *end-to-end*), permettent d'apprendre la fonction d'ordonnancement des documents *Score* en utilisant comme vecteur d'entrée, des représentations de géotextes. Une vue d'ensemble de leur architecture est présentée dans la Figure 5.3.

Le modèle **SGM** est un réseau de neurones qui combine deux types d'interactions : *locales* et *globales*, qui permettent d'apprendre une fonction d'appariement de pertinence. Comme nous l'avons détaillé dans la Section 3.3 du Chapitre 3, les interactions locales sont conçues pour détecter des signaux d'appariements entre les termes (c.-à-d. effectuer un appariement lexical). La matrice d'interaction I_l est construite via des interactions locales mot à mot $I_l(w_i^{(t)}, w_j^{(p)})$ amorties par un facteur d'amortissement $F(d^*)$. À partir de cette matrice, nous apprenons les représentations latentes des interactions locales à l'aide d'histogrammes d'appariement. Les interactions globales sont quant à elles utilisées pour capturer d'autres signaux d'appariement au niveau de l'objet (tweet ou POI) grâce à des caractéristiques textuelles (I_g^T) et spatiales (I_g^S). Les interactions locales et globales sont ensuite combinées dans une couche cachée qui génère un score d'appariement. Par la suite, nous détaillons les deux branches de notre modèle : l'une pour modéliser les interactions locales, l'autre pour modéliser les interactions globales, ainsi que la fonction objectif permettant d'apprendre les paramètres des modèles.

3.2 *Modélisation des interactions locales*

La première branche de notre réseau de neurones permet d'apprendre les représentations latentes des interactions locales. Cette modélisation se déroule en trois étapes : *(1)* représenter les géotextes sous forme de vecteurs distribués de mots ; *(2)* calculer les interactions locales ; *(3)* apprendre les représentations latentes de ces interactions. Nous les décrivons dans les sections suivantes.

3.2.1 *Représentation des géotextes*

Pour satisfaire la tâche de prédiction sémantique de l'emplacement, notre modèle doit calculer un score d'appariement entre deux géotextes. Dans le cas présent, ces géotextes sont un tweet t et un POI candidat p. Pour rappel, nous avons introduit dans la Section 2.1 du Chapitre 4 ces deux objets géotextuels. Ainsi, le tweet t est défini par $t = [w_1^{(t)}, \ldots, w_n^{(t)}]$ et l'emplacement $t.l$. De même, le POI p est défini par $p = [w_1^{(p)}, \ldots, w_m^{(p)}]$ et l'emplacement $p.l$.

Dans un premier temps, pour chaque terme $w_i^{(t)}$ du tweet t, nous récupérons sa représentation distribuée (ou *embedding*) $\mathbf{w_i} \in \mathbb{R}^k$. Le vecteur $\mathbf{w_i}$ est issu d'une matrice de plongements lexicaux (ou matrice de *word embeddings*) $\mathbf{W_e} \in \mathbb{R}^{k \times |\mathcal{V}|}$ préalablement entraînée (Mikolov *et al.*, 2013b). $\mathcal{V}$ représente le vocabulaire de la collection (c.-à-d. les mots pour lesquels nous avons une représentation distribuée) et k la dimension des représentations. Les représentations distribuées des mots du tweet t sont regroupées dans une matrice noté $T_{1:n}$ et défini par :

$$T_{1:n} = \left[\mathbf{w_1}^{(t)}, \mathbf{w_2}^{(t)}, \ldots, \mathbf{w_n}^{(t)}\right] \tag{5.1}$$

où $\mathbf{w_i}^{(t)} \in \mathbb{R}^k$ est la représentation distribuée associée au mot $w_i^{(t)}$. Un exemple de matrice est présenté dans la Figure 5.4.

De même, nous construisons $P_{1:m}$, une matrice qui regroupe les représentations distribuées des mots $w_j^{(p)}$ du POI p.

$$P_{1:m} = \left[\mathbf{w_1}^{(p)}, \mathbf{w_2}^{(p)}, \ldots, \mathbf{w_m}^{(p)}\right] \tag{5.2}$$

Il convient de noter que la matrice de plongements lexicaux $\mathbf{W_e}$ est indépendante de notre modèle. Celle-ci a été préalablement entraînée sur un corpus de documents issus de Wikipedia et de géotextes (Section 4.4). Nous réutilisons les vecteurs distribués $\mathbf{w_i}$, sans les modifier, pour obtenir $T_{1:n}$ et $P_{1:m}$.

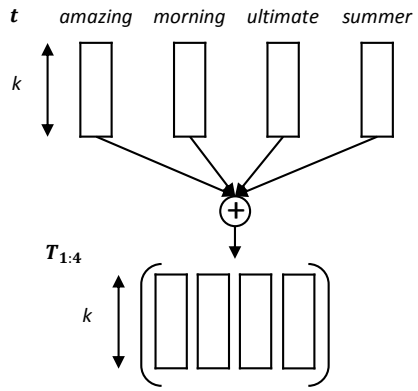


Figure 5.4 – Illustration de la matrice de représentation d'un tweet composé de quatre mots et de vecteurs distribués de dimension k.

3.2.2 *Matrice des interactions locales*

En utilisant les représentations des géotextes $T_{1:n}$ et $P_{1:m}$, nous construisons la matrice des interactions I_l, dont les composantes sont calculées à partir des paires de mots du tweet et du POI. En nous appuyant sur l'Hypothèse 2 (Section 1), nous supposons que les signaux d'appariement spatial entre les mots des tweets et des POIs peuvent contribuer à une estimation locale de leur similarité sémantique, ce qui conviendrait pour résoudre la tâche de prédiction sémantique de l'emplacement. En conséquence, nous calculons la matrice d'interaction locale mot à mot $I_l \in \mathbb{R}^{n \times m}$ à l'aide de la fonction suivante :

$$I_l(w_i^{(t)}, w_j^{(p)}) = sim(\mathbf{w}_i^{(t)}, \mathbf{w}_j^{(p)}) \times F(d^*) \tag{5.3}$$

où $F(d^*)$ est un facteur d'amortissement spatial et $sim(\mathbf{w}_i^{(t)}, \mathbf{w}_j^{(p)})$ une fonction mesurant la similarité entre les représentations distribuées des mots $\mathbf{w}_i^{(t)}$ et $\mathbf{w}_j^{(p)}$. Nous utilisons la similarité du cosinus, telle que définie dans l'Équation 5.4, comme mesure de similarité.

$$sim(\mathbf{w}_i^{(t)}, \mathbf{w}_j^{(p)}) = \frac{\mathbf{w}_i^{(t)} \cdot \mathbf{w}_j^{(p)}}{\|\mathbf{w}_i^{(t)}\| \, \|\mathbf{w}_j^{(p)}\|} \tag{5.4}$$

Le facteur d'amortissement spatial $F(d^*)$ permet d'intégrer la distance géographique dans le calcul de la similarité entre le terme $w_i^{(t)}$ du tweet et le terme $w_j^{(p)}$ du POI. Ce facteur est calculé à partir de la distance géographique entre l'emplacement $t.l$ du tweet t et l'emplacement du centre géographique $\mathcal{B}_j$ associée au terme $w_j^{(p)}$. Nous conjecturons que si $w_j^{(p)}$ est un terme caractéristique des régions autour

de l'emplacement du tweet, c.-à-d. qu'il est régulièrement mentionné par des géotextes dans cette zone, alors son centre géographique sera situé dans les environs du tweet. Ainsi, comme détaillé dans l'Hypothèse 2, le terme $w_j^{(p)}$ du POI aura un sens proche de celui du terme $w_i^{(t)}$. De fait, via le facteur d'amortissement $F(d^*)$, nous accorderons plus d'importance au terme $w_j^{(p)}$. Le facteur $F(d^*)$ est calculé avec la fonction du noyau Gaussien comme suit :

$$F(d^*) = \begin{cases} K(0) & \text{si} \quad d^* \leq 0 \\ \max(K(d^*), \alpha) & \text{sinon} \end{cases} \tag{5.5}$$

$$\text{avec} \quad K(d^*) = \frac{1}{\sqrt{2\pi}} \times \exp\left(-\frac{1}{2} \times d^{*^2}\right) \tag{5.6}$$

avec α un seuil permettant d'éviter les valeurs nulles et d la distance définie par $d = d(t.l, w_j^{(p)})$ (Équation 4.6), qui est la distance géographique entre l'emplacement du tweet et le centre géographique de $w_j^{(p)}$. Nous proposons d'utiliser la distance corrigée (centrée-réduite) $d^* = \frac{d-\mu}{\sigma}$, où μ et σ désignent respectivement la distance moyenne empirique et la variance empirique de la distance. La distance corrigée rend le modèle moins sensible aux grandes variation de distance.

En considérant les caractéristiques spatiales en plus des caractéristiques syntaxiques dans le calcul des interactions locales, notre modèle est optimisé pour la découverte de motifs sémantiques dans des objets spatiaux proches.

3.2.3 *Représentations latentes des interactions locales*

De la matrice d'interaction I_l contenant les signaux locaux d'appariement, déterminés avec les caractéristiques spatiales et syntaxiques des géotextes, notre modèle va apprendre des motifs d'appariement pour produire un score de pertinence. Toutefois, il convient de noter que la dimension de la matrice d'interaction I_l est variable et dépend de la longueur des descriptions textuelles du tweet t et du POI p. Cette contrainte, abordée par de précédents travaux, est résolue par trois principales stratégies : *(1)* l'histogramme d'appariement ; *(2)* les réseaux de neurones à convolution (CNN) ; *(3)* les réseaux de neurones récurrents (RNN) :

1. l'histogramme d'appariement (Guo *et al.*, 2016; Yang *et al.*, 2016; Fan *et al.*, 2018) est une stratégie assez récente, qui consiste à discrétiser les interactions locales dans un intervalle préalablement fixé, sans tenir compte de la position réelle des correspondances. Nous adoptons cette stratégie pour le modèle **SGM** et la détaillons dans la Section 3.2.3 ;

2. les réseaux de neurones à convolution, qui ont largement fait leurs preuves dans le traitement de l'image et de la parole (LeCun *et al.*, 1995) ainsi que dans diverses tâches de TALN (Collobert *et al.*, 2011; Kalchbrenner *et al.*, 2014; Shen *et al.*, 2014b), sont régulièrement utilisés en RI (Pang *et al.*, 2016; Mitra *et al.*, 2017; Dai *et al.*, 2018). Les CNN fixent généralement la longueur des documents en utilisant la technique de remplissage par zéro (*zero-padding*) qui consiste à compléter les documents avec des données nulles. Couplés à une stratégie de mise en commun (*pooling*), ils permettent d'obtenir un vecteur ou une matrice de taille fixe. Nous adoptons cette stratégie pour une variante du modèle **SGM**, utilisée comme modèle de référence Section 4.3 ;
3. enfin, les réseaux de neurones récurrents comme les LSTM (*Long short-term memory*) (Wan *et al.*, 2016a; Chen *et al.*, 2017) et leurs variantes GRU (*Gated Recurrent Unit*) (Wan *et al.*, 2016b; Tan *et al.*, 2018) sont aptes à travailler avec des données d'entrée de taille variable. Ces derniers ne sont pas abordés dans ce manuscrit.

Nous proposons de construire, pour chaque mot du tweet, les représentations latentes de ses interactions locales en utilisant un histogramme d'appariement. Nous utilisons ensuite un perceptron multicouche pour apprendre les motifs d'appariement et produire un score de pertinence pour chaque terme du tweet (en fonction de tous les termes du POI). Enfin, le score de pertinence des interactions locales est généré en agrégeant le score obtenu pour chaque terme du tweet.

Partant de la matrice d'interaction I_l, nous souhaitons déterminer des représentations latentes. Notre intuition est que les interactions locales $I_l(w_i^{(t)}, w_j^{(p)})$ joueraient des rôles différents en fonction de la puissance de leurs signaux d'appariement plutôt que de leurs positions dans les géotextes. Nous proposons donc d'adopter une représentation sous forme d'histogrammes d'appariement qui regroupent les interactions locales en fonction de la force de leurs signaux. Plus précisément, nous discrétisons l'intervalle des interactions locales dans un ensemble de classes de taille fixe, et agrégeons les interactions locales dans chaque classe. Par exemple, supposons que l'amplitude des classes soit fixée à $0,5$, nous obtenons quatre classes d'intervalle $\{[-1;-0,5), [-0,5;-0), [0;0,5), [0,5;1]\}$. Étant donné le mot « *burger* » issu d'un tweet, un POI représenté par les mots suivants *(hamburger, fromage, sandwich, couteau, bœuf, restaurant)*, et les interactions locales correspondantes calculées selon la similarité du cosinus $(1;0,6;0,3;-0,4;0,7;0,1)$. En agrégeant les valeurs par un simple comptage, nous obtenons l'histogramme suivant : $[0;1;1;2;2]$. Pour construire les histogrammes, nous explorons les méthodes suivantes :

— **histogramme par comptage (CH)** : il s'agit de la transformation la plus simple, qui utilise directement le nombre d'interactions locales dans chaque classe comme valeur de l'histogramme ;

— **histogramme par comptage logarithmique (LCH)** : cette transformation applique le logarithme sur la valeur de chaque classe pour d'une part, réduire la plage de valeurs et ne pas favoriser les géotextes les plus longs, et d'autre part, pour que le réseau apprenne plus facilement les relations multiplicatives (Burges *et al.*, 2005) ;
— **histogramme normalisé (NH)** : cette transformation normalise la valeur de chaque classe par le nombre total pour considérer des valeurs relatives plutôt que absolues ;
— **histogramme sommatif (SH)** : cette dernière transformation somme les valeurs de chaque classe pour considérer les valeurs réelles des différents niveaux d'interactions.

Pour chaque mot $w_i^{(t)}$ du tweet t, nous calculons son histogramme d'appariement en considérant toutes les interactions locales $I_l(w_i^{(t)}, w_j^{(p)})$ pour $j = 1, \dots, m$. À ce stade, la taille de la représentation devient fixe (déterminée par le nombre de classes). L'histogramme est ensuite transmis à un perceptron multicouche, qui projette les représentations dans un espace latent à l'aide de couches cachées. Formellement, supposons que nous avons les représentations $T_{1:n}$ et $P_{1:m}$ du tweet t et du POI p et la matrice d'interaction I_l, nous effectuons les transformations suivantes :

$$\mathbf{z}^{(0)} = I_l \tag{5.7}$$

$$\mathbf{z}_i^{(1)} = h(\mathbf{z}^{(0)}[i, *]) = h\left(I_l[i, *]\right), \quad i = 1, \dots, n \tag{5.8}$$

$$\mathbf{z}_i^{(l)} = \delta\left(\mathbf{W}^{(l)} \cdot \mathbf{z}_i^{(l-1)} + \mathbf{b}^{(l)}\right), \quad i = 1, \dots, n \ ; \ l = 1, \dots, L \tag{5.9}$$

$$s_{I_l} = \sum_{i=1}^{n} \mathbf{z}_i^{(L)} \tag{5.10}$$

où $\mathbf{z}^{(0)}$ est l'entrée du réseau de neurones ; $\mathbf{z}_i^{(l)}$ $(l = 1, \dots, L)$ désigne la l^{e} couche cachée du réseau traitant le i^{e} terme du tweet t.

Plus précisément, dans la première couche du réseau (c.-à-d. $\mathbf{z}_i^{(1)}$), nous construisons l'histogramme d'appariement d'un mot en utilisant la fonction de mappage h qui, à partir du vecteur des interactions locales $I_l[i, *]$ du mot w_i, calcule l'histogramme correspondant (Équation 5.8).
Ensuite, dans les couches cachées suivantes (c.-à-d. $\mathbf{z}_i^{(l)}$), nous projetons le vecteur de la couche précédente (c.-à-d. $\mathbf{z}_i^{(l-1)}$) dans un nouvel espace latent au moyen d'une transformation non-linéaire grâce à une matrice de poids $\mathbf{W}^{(l)}$ et un biais $\mathbf{b}^{(l)}$ (Équation 5.9). La non-linéarité de la transformation est assurée par la fonction d'activation δ. Les matrices de poids $\mathbf{W}^{(l)}$ et le biais $\mathbf{b}^{(l)}$ sont partagés entre les différents termes. Autrement dit, nous effectuons la même transformation pour tous les termes du tweet.

Enfin, en sortie du réseau de neurones, nous sommons les scores de pertinence de chaque terme, calculés dans la dernière couche cachée (c.-à-d. $\mathbf{z}_i^{(L)}$), pour déterminer le score de pertinence des interactions locales (Équation 5.10). Ce score sera ensuite associé aux interactions globales, que nous détaillons dans la Section 3.3.

3.3 *Modélisation des interactions globales*

En complément des interactions locales, qui permettent de capturer les signaux d'appariement exact entre les termes du tweet et du POI, nous proposons de capturer des correspondances sémantiques, par le biais d'interactions globales. Ces nouveau signaux d'appariement sont issus des caractéristiques spatiales et textuelles des géotextes. Nous détaillons dans la Section 3.3.1 l'interaction globale spatiale. L'interaction globale textuelle est quant à elle décrite dans la Section 3.3.2.

3.3.1 *Interaction globale spatiale*

L'objectif de l'interaction globale spatiale est de mesurer la force de l'interaction entre le tweet t et le POI p d'un point de vue spatial. Cette caractéristique nous paraît essentielle pour l'appariement d'objets géotextuels. En effet, des travaux antérieurs (Dalvi *et al.*, 2009a; Zhao *et al.*, 2016) ont montré la pertinence de la distance géographique entre l'emplacement de l'utilisateur et celui du POI pour résoudre ce type de tâche. Cependant, d'autres travaux (Shaw *et al.*, 2013; Bhattacharya *et al.*, 2015) ont révélé que les services de localisation tels que le GPS peuvent être imprécis (erreur médiane de 70 mètres), notamment lorsque les utilisateurs se trouvent à l'intérieur de bâtiments ou dans des canyons urbains [2]. Cela confirme les résultats de notre analyse préliminaire (Section 1), dans laquelle nous avions conclu que la distance entre les géotextes, bien que pertinente, ne peut pas être utilisée telle-quelle dans la tâche de prédiction sémantique de l'emplacement.

Dans cette optique, nous proposons de calculer l'interaction globale spatiale $I_g^S(t,p)$, aussi notée I_g^S, entre le tweet t et le POI p comme une similarité normalisée. Celle-ci est calculée à partir de la distance géographique entre l'emplacement du tweet et celui du POI, pondérée par la densité des lieux autour du tweet :

$$I_g^S(t,p) = \frac{P(p|t)}{\sum_{p_r \in \mathcal{P}_{p,l}^r} P(p_r|t)} \tag{5.11}$$

2. Un canyon urbain est une voie urbaine dont l'encaissement entre des bâtiments provoque des difficultés en matière d'environnement ou de radiocommunication.

où $\mathcal{P}^r_{p.l} \subset \mathcal{P}$ est l'ensemble des POIs situés dans un rayon r autour de l'emplacement du tweet $t.l$; $P(p|t)$ est la probabilité qu'un POI p soit spatialement pertinent pour le tweet t. Elle est calculée selon la formule suivante (Shaw *et al.*, 2013) :

$$P(p|t) \approx \left(\frac{r}{d(p.l, t.l) + r} \right)^4 \tag{5.12}$$

Le score $I^S_g(t, p)$ ainsi calculé sera combiné avec le score des interactions locales (Section 3.2) et celui de l'interaction globale textuelle (Section 3.3.2) afin de déterminer le score d'appariement $Score(t, p)$.

3.3.2 *Interaction globale textuelle*

L'objectif de l'interaction globale textuelle est de mesurer la force de l'interaction entre le tweet t et le POI p d'un point de vue sémantique. L'interaction textuelle vient donc compléter les similarités locales effectuées au niveau des plongements lexicaux (Section 3.2). Ce signal d'appariement est notamment essentiel pour s'attaquer aux discordances de vocabulaires susceptibles de se produire lors du calcul des interactions locales entre les plongements lexicaux (Mitra *et al.*, 2017). L'interaction globale textuelle $I^T_g(t, p)$, aussi notée I^T_g, entre le tweet t et le POI p est calculée par :

$$I^T_g(t, p) = sim(\widehat{t}, \widehat{p}) \tag{5.13}$$

où sim est la mesure de similarité du cosinus (Équation 5.4) ; $\widehat{t}$ (resp. $\widehat{p}$) est le vecteur distribué du tweet t (resp. POI p) construit comme la moyenne pondérée des représentations distribuées qui le compose, telle que définie par l'Équation 5.14. L'idée sous-jacente est que les mots importants, qui déterminent la plupart de la sémantique du texte, ont une plus grande contribution dans la construction du vecteur distribué du géotexte.

$$\widehat{x} = \frac{1}{|x|} \sum_{w_i \in x} \text{idf}_i \times \mathbf{w_i}, \quad x \in \{t, p\} \tag{5.14}$$

$$\text{idf}_i = \frac{|D|}{|\{d_j : w_i \in d_j\}|} \tag{5.15}$$

avec $\mathbf{w_i} \in \mathbf{W_e}$ la représentation distribuée associée au mot w_i (Section 3.2.1) et idf_i la fréquence inverse de document, définie par l'Équation 5.15, qui mesure l'importance du mot w_i dans l'ensemble du corpus. $|D|$ et $|\{d_j : w_i \in d_j\}|$ sont respectivement le nombre total de documents dans le corpus et le nombre de documents où le terme w_i apparaît.

Le score $I_g^T(t,p)$ ainsi calculé sera combiné avec le score des interactions locales (Section 3.2) et celui de l'interaction globale spatiale (Section 3.3.1) afin de déterminer le score d'appariement $Score(t,p)$.

3.4 *Calcul du score d'appariement*

À ce stade, nous avons calculé les scores des interactions locales s_{I_l} (Section 3.2) et des interactions globales spatiales I_g^S (Section 3.3.1) et textuelles I_g^T (Section 3.3.2). Ces trois scores de pertinence sont concaténés et transmis à un ultime perceptron multicouche pour obtenir le score d'appariement $Score(t,p)$:

$$\mathbf{z}^{(L+1)} = \left[s_{I_l}; I_g^S; I_g^T\right] \tag{5.16}$$

$$Score(t,p) = \delta\left(\mathbf{W}^{(L+1)} \cdot \mathbf{z}^{(L+1)} + \mathbf{b}^{(L+1)}\right) \tag{5.17}$$

où $\mathbf{z}^{(L+1)}$ dénote la $(L+1)^{\text{e}}$ couche cachée ; δ est la fonction d'activation qui permet d'assurer la non-linéarité de la transformation ; $\mathbf{W}^{(L+1)}$ et $\mathbf{b}^{(L+1)}$ sont respectivement la matrice de poids et le biais associés à la $(L+1)^{\text{e}}$ couche cachée du réseau.

3.5 *Apprentissage du modèle*

Le modèle **SGM** est un réseau de neurones acyclique faisant intervenir différents paramètres qu'il convient d'optimiser pour résoudre la tâche de prédiction sémantique de l'emplacement. Nous l'avons évoqué dans la Section 3.2.1, les représentations distribuées des mots ont été préalablement entraînées sur un corpus de documents. Ainsi, les seuls paramètres à apprendre sont les matrices de poids $\mathbf{W}$ et les vecteurs de biais $\mathbf{b}$.

La tâche de prédiction sémantique de l'emplacement étant assimilable à un problème d'ordonnancement de POIs candidats, nous optimisons les paramètres à l'aide d'une fonction de coût d'ordonnancement relatif. Cette fonction s'appuie sur la distance de similarité entre un couple (*tweet*, *POI*) pertinent, et des couples (*tweet*, *POI*) non pertinents. Pour ce faire, comme suggéré par Huang *et al.* (2013), nous construisons des échantillons de couples (*tweet*, *POI*) pour lesquels nous opposons, pour le tweet t, son POI pertinent p^+ avec R POIs non pertinents p_r^- ($r \in [\![1, R]\!]$). L'ensemble des POIs non pertinents est issu d'un regroupement de POIs spatialement (Définition 4.4) ou sémantiquement (BM25) proches du tweet t. Ainsi, durant la phase d'apprentissage, le modèle devrait maximiser le score d'appariement des paires pertinentes tout en minimisant celui des paires non per-

tinentes. Formellement, la distance Δ entre le score d'appariement du couple pertinent (t, p^+) et des couples non pertinents (t, p_r^-) est définie par :

$$\Delta = \sum_{r=1}^{R} \left[Score(t, p^+) - Score(t, p_r^-) \right] \tag{5.18}$$

où $Score(t, p)$ est le score d'appariement calculé par notre modèle. Pour maximiser la distance Δ et donc estimer les paramètres d'un modèle, nous minimisons la fonction de coût de type *hinge loss*, adaptée pour des tâches d'apprentissage d'ordonnancement :

$$L(\theta) = \max(0, \alpha - \Delta) \tag{5.19}$$

avec α la marge de l'amplitude de Δ et θ l'ensemble des paramètres du modèle considéré.

La fonction de coût étant différentiable, nous pouvons effectuer une descente de gradient stochastique (SGD) pour apprendre les paramètres. Nous utilisons la méthode Adam (Kingma et Ba, 2015), une variante du SGD, qui détermine des taux d'apprentissage adaptatifs pour chaque paramètre. Nous optons pour la version mini-lot de la descente du gradient pour accélérer le processus d'entraînement, comme détaillée dans la Section 1.4.2.2 du Chapitre 3.

4 Cadre expérimental

Afin d'évaluer la qualité de nos deux réseaux de neurones pour la prédiction sémantique de l'emplacement, nous avons mis en place un protocole d'évaluation. Celui-ci a pour objectif de répondre aux questions de recherche (**QR**) suivantes :

- **QR1 –** Quel est l'impact intrinsèque du facteur d'amortissement spatial $F(d^*)$ sur les similarités sémantiques des paires de mots ? Peut on confirmer empiriquement l'Hypothèse 2 ?
- **QR2 –** Quels sont les effets extrinsèques des composants (c.-à-d. facteur d'amortissement, histogramme d'appariement, interactions globales et locales) sur les performances des modèles ? Toutes les caractéristiques globales (spatiales et textuelles) contribuent-elles à la performance du modèle ?
- **QR3 –** Les modèles que nous proposons apportent-ils des améliorations significatives par rapport aux modèles de référence reconnus ?
- **QR4 –** Quels sont les hyperparamètres clés du modèle **SGM** et quel est leur impact sur la qualité des prédictions ?

Nous décrivons dans ce qui suit le protocole d'évaluation permettant de répondre aux quatre questions de recherche édictées ci-dessus. Nous commençons

par présenter dans la Section 4.1, les jeux de données utilisés pour l'apprentissage et l'évaluation des modèles. Nous détaillons ensuite les scénarios d'évaluation dans la Section 4.2 ainsi que les modèles de référence dans la Section 4.3. Enfin, dans la Section 4.4, nous donnons des détails sur l'implémentation de notre modèle et ceux de référence, avec notamment le réglage des hyperparamètres.

4.1 *Jeux de données*

Dans cette contribution, nous nous attachons à apparier des tweets géotaggés avec des POIs. De ce fait, pour entraîner le modèle **SGM**, nous avons réutilisé la collection de tweets géotaggés publiée par Zhao *et al.* (2016) et présentée dans la Section 4.2.1.1 du Chapitre 4. Dans le cadre de cette contribution, nous divisons la collection en deux ensembles de données dont les principales statistiques sont présentées dans le Tableau 5.1.

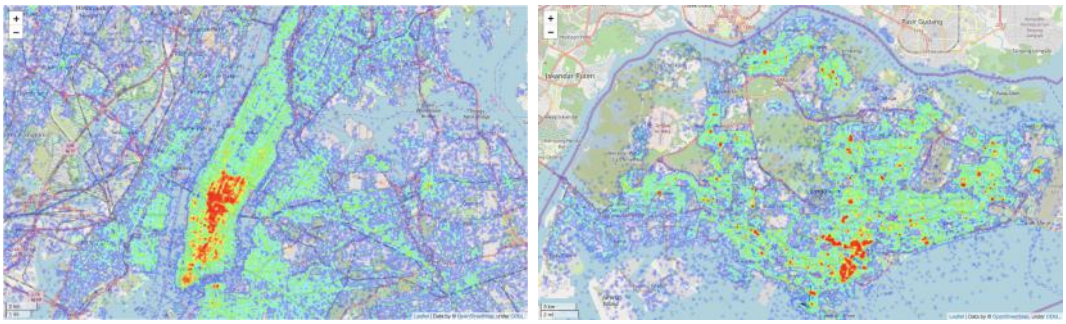
	NY	**SG**
Nb. de tweets	43 914	29 719
Nb. d'utilisateurs	1 200	10 189
Nb. de POIs	482 480	321 985
Nb. de critiques *(million)*	1,430	0,753
Densité de POIs/km²	616	446
Nb. moyen de tweets/utilisateur	36,59	2,91
Nb. moyen de tweets/POI	4,19	4,96
Nb. moyen de POIs visités/utilisateur	21,49	2,38
Nb. moyen de termes/tweet	5,97	5,44
Nb. moyen de termes/POI	50,91	28,38
Dist. moyenne Tweet-POI *(mètres)*	59,08	41,74
Écart-type dist. Tweet-POI *(mètres)*	389,18	379,99

Tableau 5.1 – Statistiques des jeux de données de NY et de SG.

Le premier ensemble de données, appelé *NY*, contient 210 000 tweets géotaggés postés entre septembre 2010 et janvier 2015 dans la ville de New York aux États-Unis. Le deuxième ensemble, appelé *SG*, est quant à lui constitué de 380 000 tweets publiés entre mars 2014 et août 2014, dans la cité-État Singapour. Comme évoqué dans la Section 4.3.1.1 du Chapitre 4, Zhao *et al.* (2016) ont déterminé si les tweets étaient liés à des POIs ou non. Puisque notre contribution repose sur des tweets liés à des POIs, nous avons écarté les tweets non pertinents (c.-à-d. non liés à des POIs). Ainsi, comme détaillé dans le Tableau 5.1, les caractéristiques des jeux de

données filtrés sont les suivantes : la collection de NY, se compose de 43 914 tweets liés à des POIs, tandis que la collection de SG en contient 29 719.

Les tweets des ensembles de données de NY et de SG sont liés à des POIs issus du monde réel. Nous avons donc collecté la liste des POIs localisés dans les villes de New York et de Singapour via l'API de Foursquare. Nous avons ainsi récupéré respectivement 482 480 POIs pour NY (soit environ 616 POIs par km²) et 321 985 POIs pour SG (soit environ 446 POIs par km²). De plus, pour enrichir l'information textuelle de chaque POI, nous avons récupéré un ensemble de critiques (ou *reviews*) publiées par les utilisateurs de Foursquare, soit environ 1,4 million de données supplémentaires pour les POIs de NY, et 753 000 pour les POIs de SG. Pour compléter ces statistiques, nous présentons dans Figure 5.5, la carte de chaleur de la concentration de POIs dans les villes de New York (Figure 5.5a) et de Singapour (Figure 5.5b).



(a) Jeu de données de NY (b) Jeu de données de SG

Figure 5.5 – Carte de chaleur des concentrations de POIs dans les villes de New York et de Singapour.

4.2 *Scénarios d'évaluation*

Pour répondre à **QR2** et mesurer l'effet de chaque composant du modèle **SGM**, nous envisageons plusieurs scénarios, résumés dans le Tableau 5.2. Nous avons décomposé l'évaluation en trois étapes :

1. dans la Section 5.2.1, nous évaluons les différentes méthodes proposées pour générer les histogrammes d'appariement. Cette analyse est effectuée à partir des modèles $\mathbf{SGM}_x$ ($x \in \{LCH, CH, NCH, SH\}$), c.-à-d. les modèles qui ne contiennent que les interactions locales, tout en faisant varier la fonction de calcul des histogrammes. En complément, nous comparons les histogrammes d'appariement avec une autre méthode s'appuyant sur un réseau de neurones à convolution, conçu dans le cadre du scénario $\mathbf{SGM}_{Conv}$;

Scénarios	F	Histogramme	$\mathbf{I_g^S}$	$\mathbf{I_g^T}$	Commentaires
SGM	✓	LCH	✓	✓	Modèle complet.
$\mathbf{SGM}_{LCH}$	✓	LCH	-	-	Évaluer l'effet des différents histogrammes d'appariement.
$\mathbf{SGM}_{CH}$	✓	CH	-	-	
$\mathbf{SGM}_{NCH}$	✓	NCH	-	-	
$\mathbf{SGM}_{SH}$	✓	SH	-	-	
$\mathbf{SGM}_{Conv}$	✓	*	-	-	
$\mathbf{SGM}_{I_l-F}$	-	LCH	✓	✓	Évaluer l'effet des interactions locales et du facteur d'amortissement $F(d^*)$.
$\mathbf{SGM}_{I_l}$	✓	LCH	-	-	
$\mathbf{SGM}_{\varnothing}$	-	LCH	-	-	
$\mathbf{SGM}_{I_g^T}$	✓	LCH	-	✓	Évaluer l'effet des interactions globales.
$\mathbf{SGM}_{I_g^S}$	✓	LCH	✓	-	

Le symbole « ✓ » indique que le scénario utilise le facteur d'amortissement ($F(d^*)$), l'interaction globale spatiale (I_g^S) ou textuelle (I_g^T). Le symbole « - » indique que le scénario n'utilise pas le composant. * Le scénario $\mathbf{SGM}_{Conv}$ n'utilise pas un histogramme d'appariement, mais un réseau de neurones à convolution pour calculer les représentations latentes des interactions locales.

Tableau 5.2 – Configuration des scénarios utilisés pour l'évaluation.

2. nous poursuivons l'évaluation, dans la Section 5.2.2, en mesurant l'efficacité du facteur d'amortissement spatial $F(d^*)$:
   a) sur le modèle complet **SGM**, en considérant le modèle $\mathbf{SGM}_{I_l-F}$ qui occulte le facteur d'amortissement $F(d^*)$ pour calculer la matrice d'interaction I_l ;
   b) sur le modèle tronqué $\mathbf{SGM}_{I_l}$ (c.-à-d. modèle sans les interactions locales) en considérant le modèle $\mathbf{SGM}_{\varnothing}$ qui ignore l'utilisation du facteur $F(d^*)$ et des interactions globales ;
3. enfin, dans la Section 5.2.3, nous évaluons la contribution de chaque interaction globale sur les performances du modèle complet **SGM** et sur celles du modèle tronqué $\mathbf{SGM}_{I_l}$. Pour cela, nous considérons d'une part, le modèle $\mathbf{SGM}_{I_g^S}$ qui n'utilise que l'interaction globale spatiale (en plus du facteur d'amortissement), et d'autre part, le modèle $\mathbf{SGM}_{I_g^T}$ qui n'utilise quant à lui, que l'interaction globale textuelle.

4.3 *Modèles de référence*

Modèles de référence	Information textuelle	Information spatiale	Appariement neuronal	Appariement de RDG
DIST (De Smith et Goodchild, 2007)		✓		
BM25 (Robertson et Jones, 1976)	✓			
TFIDF$^+$-D (Dalvi *et al.*, 2009a)	✓			
CLASS (Zhao *et al.*, 2016)	✓	✓		
SBM (Zhao *et al.*, 2016)	✓	✓		
ARC-I (Hu *et al.*, 2014)	✓		✓	
ARC-II (Hu *et al.*, 2014)	✓		✓	
DRMM (Guo *et al.*, 2016)	✓		✓	
ANMM (Yang *et al.*, 2016)	✓		✓	
DRMM+$F(d^*)$ (Guo *et al.*, 2016)*	✓	✓	✓	
ANMM+$F(d^*)$ (Yang *et al.*, 2016)*	✓	✓	✓	
WORD2VEC (Mikolov *et al.*, 2013b)	✓			✓
DISTILBERT (Sanh *et al.*, 2019)	✓			✓
SBERT$_{STS}$ (Reimers et Gurevych, 2019)	✓			✓
SBERT$_{X}$ (Reimers et Gurevych, 2019)	✓			✓

Le symbole « ✓ » indique que le modèle de référence utilise l'information textuelle ou spatiale, ainsi qu'un appariement neuronal ou de représentations distribuées de géotextes (RDG).
** Modèle de référence étendu avec le facteur d'amortissement $F(d^*)$ (Équation 5.5).*

Tableau 5.3 – Synthèse des modèles de référence utilisés pour évaluer la qualité de notre contribution pour la tâche de prédiction sémantique de l'emplacement.

Pour répondre à **QR3** et évaluer la qualité de notre contribution, nous comparons nos résultats avec des modèles d'appariement de référence. Ces modèles sont classés en cinq catégories : *(1)* appariement spatial (DIST) ; *(2)* appariement textuel (BM25, TFIDF$^+$-D) ; *(3)* mélange des approches textuelles et spatiales (CLASS, SBM) ; *(4)* appariement neuronal (ARC-I, ARC-II, DRMM, ANMM) ; et *(5)* appariement de représentations distribuées de géotextes (WORD2VEC, DISTILBERT, SBERT). Ces approches, résumées dans le Tableau 5.3, sont les suivantes :

1. **Appariement spatial**
   — **DIST** (De Smith et Goodchild, 2007) : cette méthode sélectionne comme POI pertinent, le POI géographiquement le plus proche de l'emplacement du tweet. Nous considérons ici la mesure de Haversine (Équation 2.2) ;
2. **Appariement textuel**
   — **BM25** (Robertson et Jones, 1976) : modèle probabiliste de référence, largement utilisé en RI textuelle ;

— **TfIdf$^+$-D** (Dalvi *et al.*, 2009a) : variante du modèle TfIdf qui accentue le poids des mots des POIs ;

3. **Appariement spatial et textuel**
   — **Class** (Zhao *et al.*, 2016) : modèle d'ordonnancement des POIs qui combine la distance géographique et la similarité thématique calculée à partir d'un modèle de langue ;
   — **sBM** (Zhao *et al.*, 2016) : modèle bayésien supervisé pour l'ordonnancement de POIs s'appuyant sur les caractéristiques sémantiques et spatiales des géotextes ;
4. **Appariement neuronal**
   — **Arc-I** (Hu *et al.*, 2014) : modèle d'appariement neuronal axé sur la représentation, avec utilisation de réseaux de neurones à convolution pour déterminer des motifs d'appariement ;
   — **Arc-II** (Hu *et al.*, 2014) : modèle d'appariement neuronal axé sur l'interaction, qui apprend des signaux d'appariement hiérarchiques à partir des interactions locales, via un réseau de neurones à convolution ;
   — **DRMM** (Guo *et al.*, 2016) : modèle d'appariement neuronal axé sur l'interaction, qui se focalise sur l'apprentissage de motifs d'appariement local hiérarchique au niveau du mot, couplé à un réseau de pondération (*term gating network*). Pour faire des comparaisons équitables, nous proposons également d'utiliser une variante de ce modèle, **DRMM**$+F(d^*)$, qui intègre le facteur d'amortissement spatial $F(d^*)$;
   — **aNMM** (Yang *et al.*, 2016) : modèle neuronal d'attention pour l'appariement, qui adopte un schéma de pondération partagée pour combiner les signaux d'appariement locaux. De même que pour le modèle **DRMM**, nous proposons le modèle **aNMM**$+F(d^*)$ qui intègre le facteur d'amortissement spatial $F(d^*)$.
5. **Appariement de représentations distribuées de géotextes**
   L'objectif de cette catégorie de modèles est de construire des représentations de géotextes (tweet et POIs) apprises à l'aide de différents modèles de l'état-de-l'art. La similarité du cosinus entre les représentations des tweets et celles des POIs candidats sert de fonction d'appariement.
   — **Word2Vec** (Mikolov *et al.*, 2013b). Plongements lexicaux pré-entraînés en utilisant le modèle *Skip-Gram*. Les représentations ont été apprises grâce à une collection d'articles de Wikipedia et de revues de POIs. Une moyenne pondérée par le TfIdf est calculée pour dériver les représentations des documents à partir des plongements lexicaux.
   — **DistilBERT** (Sanh *et al.*, 2019). Un modèle de *transformer* léger et rapide s'appuyant sur l'architecture BERT (Devlin *et al.*, 2019) pour produire des représentations distribuées d'entités lexicales (ou *tokens*) contextualisées. Les représentations des documents sont dérivées des représentations des entités lexicales en utilisant une stratégie de mise en commun (moyenne).

— **SBERT** (Reimers et Gurevych, 2019). Un modèle qui ajuste (*fine-tune*) BERT (ou des modèles similaires) à l'aide d'une structure de réseau siamois ou triplet pour produire des représentations de phrases sémantiquement pertinentes. Ces dernières peuvent être utilisées dans des tâches non supervisées telles que la similarité sémantique de textes. Nous proposons d'ajuster **DistilBERT** sur deux ensembles de données différents. Pour $\textbf{SBERT}_{STS}$, le modèle est entraîné sur les données NLI [3] puis ajusté sur la collection de données STS (Baudis et Sedivý, 2016). Pour $\textbf{SBERT}_{X}$ (où $X \in \{NY, SG\}$), le modèle est entraîné sur la collection NLI, puis ajusté sur la collection utilisée pour l'évaluation de notre tâche c.-à-d. X.

4.4 *Détails de mise en œuvre*

Dans cette section, nous détaillons la mise en œuvre de notre protocole d'évaluation, et donnons notamment des détails sur les hyperparamètres de notre modèle **SGM** et des scénarios, et sur ceux des modèles de référence.

4.4.1 *Représentations distribuées des mots*

Comme nous l'avons évoqué dans la Section 3.2.1, les représentations distribuées des mots sont indépendantes de notre modèle. La matrice de plongements lexicaux $\mathbf{W}_e$, utilisée par notre modèle, mais aussi par les modèles de référence (Arc-I, Arc-II, DRMM, DRMM+$F(d^*)$, aNMM, aNMM+$F(d^*)$ et Word2Vec) est obtenue en utilisant l'implémentation du modèle *Skip-Gram* (Mikolov *et al.*, 2013b) de la librairie Python *gensim* (Rehurek et Sojka, 2010).

Le vocabulaire de la collection se compose de 3,2 millions de mots, qui apparaissent au moins 3 fois dans le corpus. La fenêtre contextuelle a été fixée à 5 mots. Les vecteurs, de dimension 300, ont été entraînés à partir d'un corpus de documents issus de Wikipedia anglais [4] et d'un corpus de géotextes incluant des tweets, des POIs et des critiques d'utilisateurs tel que décrit dans la Section 4.1. La qualité des représentations a été évaluée sur le test d'analogie de Google [5] (Mikolov *et al.*, 2013b) et a obtenu une précision de 70% (contre 77% pour le modèle pré-entraîné Google News [6]).

3. https://nlp.stanford.edu/projects/snli/
4. https://dumps.wikimedia.org/
5. https://aclweb.org/aclwiki/Google_analogy_test_set_(State_of_the_art)
6. https://code.google.com/archive/p/word2vec/

4.4.2 *Configuration des modèles proposés*

Notre modèle est implémenté en utilisant l'API *Keras* (Chollet *et al.*). Sauf indication contraire, les hyperparamètres par défaut utilisés pour les expérimentations sont les suivants.

Pour le modèle **SGM**, concernant le *calcul des interactions locales*, nous utilisons une architecture composée de quatre couches : une couche d'entrée LCH (200 neurones ou *classes*), deux couches cachées pour chaque terme du tweet (10 neurones puis 1 neurone) et enfin une couche de sortie (1 neurone) pour calculer le score final $Score(t, p)$. Après la couche d'entrée, nous appliquons une technique de régularisation, l'abandon (ou *dropout*) avec un taux fixé à $0,2$, pour réduire le surajustement des réseaux de neurones (Hinton *et al.*, 2012).

Pour le scénario $\mathbf{SGM}_{Conv}$, concernant le *calcul des interactions locales*, nous avons fixé la taille maximale des tweets et des POIs à respectivement 16 et 1 000 mots. Dans la couche de convolution, nous avons utilisé des fenêtres de mots de taille 2×2, 3×3 et 4×4, chacune contenant 100 filtres de convolution. Après le *max pooling*, dans le perceptron multicouche, nous avons utilisé une architecture composée de trois couches, constituée de deux couches cachées (128 puis 32 neurones) et enfin une couche de sortie (1 neurone) pour calculer le score final $Score(t, p)$.

Concernant le *calcul des interactions globales*, pour les deux modèles, nous avons fixé le rayon r à 100 mètres et le seuil α à $0,01$ pour les deux jeux de données. Nous avons déterminé empiriquement les valeurs de μ et σ, et choisissons $\mu = 3,8$ km et $\sigma = 3,9$ km pour la collection de NY et $\mu = 4,7$ km et $\sigma = 5,9$ km pour la collection de SG. L'abandon, fixé à un taux de $0,2$, a été utilisé dans les couches cachées pour limiter le surajustement des réseaux de neurones.

Pour l'apprentissage des modèles (Section 3.5), nous fixons $R = 10$ POIs non-pertinents dans le calcul de la différence Δ (Équation 5.18). Enfin, pour la tâche de prédiction sémantique de l'emplacement, nous considérons, par tweet, un ensemble de 200 POIs candidats, sélectionnés parmi les POIs géographiquement (Haversine) et sémantiquement (BM25) les plus proches du tweet.

4.4.3 *Configuration des modèles de référence*

Sauf indication contraire, les modèles de référence sont paramétrés selon les détails fournis par les auteurs. Pour les modèles ARC-I et ARC-II, nous avons entraîné plusieurs versions, en variant la longueur des documents et les convolutions. Les paramètres retenus sont les suivants : la longueur des documents est fixée à 16 et 1 000 mots pour les tweets et les POIs. Pour le modèle ARC-I, nous utilisons une

couche de convolution composée de 32 filtres de taille 3. Pour le modèle ARC-II, nos effectuons deux convolutions successives composées de 16 et 32 filtres de taille 3. Concernant le modèle sBM, nous réutilisons le code fourni par Zhao *et al.* (2016), ainsi que leur configuration. Pour le modèle DRMM, nous fixons le nombre de classes à 35 et appliquons la méthode LCH pour le calcul des histogrammes d'appariement. L'architecture du perceptron multicouche est composée d'une couche cachée (5 neurones) et d'une couche de sortie (1 neurone). Pour le modèle aNMM, nous fixons le nombre de classes à 200. Le perceptron multicouche est quant à lui composé d'une couche cachée (10 neurones) et d'une couche de sortie (1 neurone).

Pour les modèles DISTILBERT et SBERT_{STS}, nous utilisons les modèles pré-entraînés fournis par Reimers et Gurevych (2019). Pour le modèle SBERT_X, nous avons utilisé le même protocole d'évaluation que celui décrit dans la Section 4.4.4, permettant de générer des sous-ensembles d'évaluation indépendants. Ainsi, pour chaque sous-ensemble, l'étape d'ajustement des modèles SBERT_X a été effectuée sur les quatre cinquièmes de la collection X et les similarités ont été calculées sur le cinquième restant. En ce qui concerne l'étape d'ajustement, nous considérons des triplets de la forme *(tweet, POI pertinent, POI non pertinent)* comme données d'entrée. Le modèle est entraîné pour minimiser l'erreur calculée à partir des triplets. Pour chaque tweet, nous considérons comme POI non pertinents, les 10 POIs les plus similaires selon la mesure du BM25. Autrement dit, pour ajuster le modèle SBERT_X, nous avons utilisé 10 triplets par tweet.

4.4.4 *Détails du protocole d'évaluation*

Nous adoptons le protocole de la validation croisée de type *k-fold* pour estimer la fiabilité de notre modèle. Il convient de noter que, dans nos jeux de données, et plus généralement avec les données Twitter, les tweets ne sont pas nécessairement indépendants et identiquement distribués (Mozetic *et al.*, 2018). En effet, dans notre cas, un utilisateur est susceptible de visiter plusieurs fois le même POI, et d'y poster plusieurs messages plus ou moins identiques. De fait, en appliquant une stratégie de validation croisée, chaque sous-ensemble (ou *fold*) peut ne pas être indépendant. Autrement dit, il est possible de trouver des paires (*utilisateur*, *POI*) similaires dans les données d'apprentissage et de test. Pour palier ce problème, lors du sous échantillonage des jeux de données, il est courant de s'assurer qu'aucun des utilisateurs ne se retrouve dans plusieurs sous-ensembles, garantissant ainsi l'indépendance de chaque échantillon, et donc une estimation non biaisée des résultats.

En gardant ceci à l'esprit, nous décidons d'effectuer une validation croisée de type *group k-fold*. Le *group k-fold* est une variante du *k-fold* qui garantit qu'un même groupe, ici un utilisateur de Twitter n'est pas représenté à la fois dans les données

de test et d'apprentissage. Plus spécifiquement, nous choisissons d'effectuer une validation à l'aide de 5 sous-échantillons par collection. Pour rappel, le principe de la validation croisée est le suivant :

1. découper le jeu de données en k sous-échantillons ;
2. sélectionner un des k échantillons comme ensemble de test et les $k-1$ échantillons restants comme ensemble d'apprentissage ;
3. entraîner le modèle à partir de l'ensemble d'apprentissage et calculer le score de performance sur l'ensemble de test grâce à des mesures d'évaluation ($Acc@k$ et MRR dans notre cas) ;
4. répéter l'opération 3 pour les $k-1$ échantillons restants ;
5. les mesures d'évaluation de chaque sous-échantillon sont ensuite résumées avec la moyenne arithmétique.

5 Résultats de l'évaluation

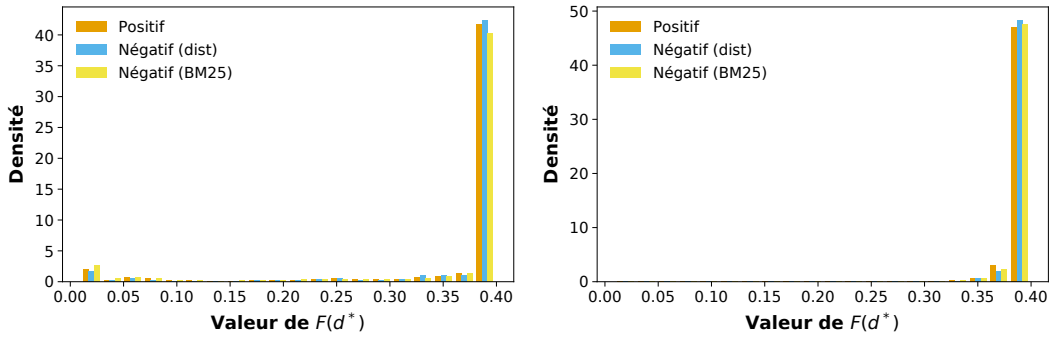
Nous présentons dans cette section les résultats de l'évaluation empirique et l'analyse qualitative réalisée pour évaluer l'efficacité du modèle **SGM**. Pour ce faire, nous répondons à chaque question de recherche, exprimées dans la Section 4, dans une sous-section dédiée. Pour commencer, dans la Section 5.1 nous répondons à **QR1** en évaluant l'impact intrinsèque du facteur d'amortissement spatial. Nous évaluons ensuite, dans la Section 5.2, la contribution de chacun des composants du modèle neuronal pour répondre à **QR2**. Puis, dans la Section 5.3, pour répondre à **QR3**, nous comparons les performances de notre modèle **SGM** avec les modèles d'appariement de référence détaillés dans la Section 4.3. Enfin, nous répondons à **QR4** dans la Section 5.4 en étudiant les hyperparamètres du modèle **SGM**.

5.1 *Évaluation intrinsèque du facteur d'amortissement spatial (QR1)*

Notre premier objectif est d'évaluer l'impact intrinsèque du facteur d'amortissement spatial $F(d^*)$ (Section 5.2.2). Pour cela, nous procédons en deux étapes. Premièrement, nous explorons les distributions du facteur d'amortissement spatial $F(d^*)$ sur les collections de NY et de SG en étudiant ses valeurs pour des paires de tweet-POI pertinentes et non pertinentes. Ensuite, nous tentons de vérifier empiriquement l'Hypothèse 2 portant sur la distribution spatiale des mots. Ainsi, nous étudions quantitativement et qualitativement si le facteur d'amortis-

sement spatial $F(d^*)$ a un impact intrinsèque sur les similarités mot à mot. Cette étape est essentielle puisque la composante des interactions locales du modèle **SGM** repose fortement sur ces similarités.

5.1.1 *Comparaison des distributions du facteur d'amortissement spatial.*



(a) Jeu de données de NY

(b) Jeu de données de SG

Figure 5.6 – Comparaison des distributions de $F(d^*)$ pour les paires tweet-POI pertinentes (Positif) et non pertinentes (Négatif).

Nous commençons cette analyse en étudiant les distributions du facteur d'amortissement spatial $F(d^*)$. Cela nous permet de voir, d'une part, si les distributions sont comparables dans les deux collections de NY et de SG, et d'autre part, si les distributions de $F(d^*)$ issues des paires positives sont les mêmes que celles issues des paires négatives. Par conséquent, pour chaque ensemble de données, nous calculons $F(d^*)$ pour toutes les paires tweet-POI pertinentes. Ensuite, pour chaque tweet, nous récupérons un ensemble de 2 POIs non pertinents, à savoir le POI le plus proche spatialement et le POI les plus similaires selon la mesure BM25. Nous calculons ensuite les distributions des valeurs de $F(d^*)$ associées et traçons dans la Figure 5.6 les histogrammes de densité pour les collections de NY (Figure 5.6a) et de SG (Figure 5.6b).

Tout d'abord, nous pouvons voir qu'une majorité des valeurs de $F(d^*)$ sont proches de $0,38$, quel que soit le type de paires (c.-à-d. positive ou négative) et la collection considérée. Deuxièmement, nous remarquons que la distribution des paires négatives est sensiblement identique à celle des paires positives, pour les deux collections. En effet, nous retrouvons des pics de densité pour les mêmes valeurs de $F(d^*)$ pour les paires positives et négatives, malgré des valeurs de densité légèrement différentes. En ce qui concerne la collection de NY (Figure 5.6a), nous constatons un pic de densité pour des valeurs de $F(d^*)$ comprises entre $0,01$ (ce qui correspond au seuil α) et $0,08$, ce qui suggère une forte utilisation de mots généraux dans les géotextes, en particulier dans l'ensemble négatif BM25. Pour la

collection de SG, les paires tweet-POI tirées de l'ensemble positif semblent plus impliquées dans l'utilisation de mots locaux. La distribution étant particulièrement concentrée sur des valeurs de $F(d^*)$ supérieures à $0,35$.

5.1.2 *Impact du facteur d'amortissement sur les similarités mot à mot*

Nous réalisons maintenant une seconde étude visant à vérifier si les similarités mot à mot suivent des tendances différentes avec et sans utilisation du facteur d'amortissement $F(d^*)$ (Équation 5.5). Pour atteindre cet objectif, nous construisons, pour chaque paire pertinente tweet-POI issues des jeux de données de NY et de SG, deux matrices d'interaction : l'une avec l'utilisation du facteur $F(d^*)$ telle que décrite par l'Équation 5.3 et donc notée I_l, et l'autre, sans l'utilisation du facteur $F(d^*)$, que nous notons I'_l. Après avoir calculé les deux matrices d'interaction I_l et I'_l, nous convertissons les valeurs de leurs éléments en variables de rang, respectivement notées rgI_l et rgI'_l. Enfin, pour déterminer si les deux variables suivent les mêmes tendances ou non, autrement dit, pour identifier si les rangs rgI_l et rgI'_l sont corrélés, nous calculons le coefficient de corrélation du rang de Spearman, noté ρ. La corrélation entre les deux variables sera élevée (ρ proche de 1) lorsque les observations auront un rang similaire, et faible (ρ proche de -1) lorsqu'il sera différent. Nous répétons ce processus, et calculons le coefficient de corrélation pour tous les tweets de nos collections de NY et de SG et affichons la distribution des résultats obtenus dans la Figure 5.7.

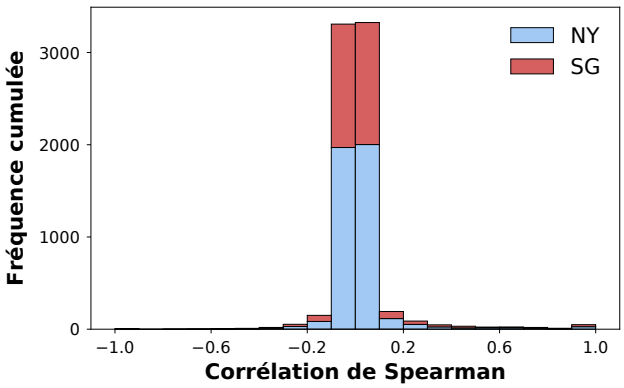


Figure 5.7 – Distribution des coefficients de corrélation de Spearman calculés entre les variables de rang rgI_l et rgI'_l issues des matrices d'interaction I_l et I'_l. La matrice d'interaction I_l est calculée selon l'Équation 5.3 et fait donc intervenir le facteur d'amortissement $F(d^*)$, contrairement à I'_l.

Comme nous pouvons le constater sur la Figure 5.7, presque tous les coefficients calculés ($90,2\%$) ont une valeur de corrélation de Spearman proche de 0 ($-0,1 \leq \rho \leq 0,1$), quel que soit le jeu de données considéré. En d'autres termes,

en corrigeant les similarités des paires de mots par un facteur spatial, nous avons complètement bouleversé l'importance de leur similarité sémantique. Cela fournit donc un indice sur l'influence de l'utilisation conjointe de caractéristiques syntaxiques et spatiales pour déterminer la similarité de paires de mots. Par la suite, dans la Section 5.2.2, nous verrons si l'influence est positive ou négative.

Dans le but d'obtenir un aperçu qualitatif du rôle du facteur d'amortissement spatial pour le calcul de la matrice d'interaction, nous analysons quelques ordonnancements de paires de mots issues d'un couple tweet-POI pertinent. Dans l'exemple présenté dans le Tableau 5.4, nous utilisons le tweet « *First Yankee vs. RedSox game ! #bronx #summer* », publié par un utilisateur depuis le *Yankee Stadium* dans le Bronx, un quartier de New York, au cours d'une rencontre de baseball entre les Yankees et les Red Sox. Nous affichons les rangs de quelques paires de mots issues des matrices I_l (calculée avec $F(d^*)$) et I'_l (calculée sans $F(d^*)$).

Paires de mots $(w_i^{(t)}, w_j^{(p)})$	**Rangs** $rgI'_l(w_i^{(t)}, w_j^{(p)})$	$rgI_l(w_i^{(t)}, w_j^{(p)})$	**Valeurs du facteur** $F(d^*)$
redsox - homeruns	86	11 △	0,398
redsox - berra	69	15 △	0,381
yankee - yanks	184	27 △	0,398
bronx - yankee	211	34 △	0,399
redsox - good	15	29 ▽	0,319
redsox - time	39	52 ▽	0,322
summer - original	21	87 ▽	0,316

Tableau 5.4 – Exemple d'ordonnancement de paires de mots $(w_i^{(t)}, w_j^{(p)})$ issues des matrices d'interaction I'_l (sans le facteur $F(d^*)$) et I_l (avec le facteur $F(d^*)$). Le symbole « △ » (resp. « ▽ ») indique une augmentation (resp. une diminution) du rang suite à l'utilisation du facteur dans le calcul de la similarité.

Dans le Tableau 5.4, nous remarquons que quatre paires de mots, situées dans la partie supérieure, ont une similarité qui a augmentée (c.-à-d. un meilleur rang) lorsqu'elle a été corrigée par le facteur $F(d^*)$. Au contraire, dans la partie basse, nous retrouvons trois paires de mots avec une similarité plus faible (c.-à-d. un rang moindre) lorsqu'elle a été corrigée par le facteur $F(d^*)$.
Considérons d'abord les paires de mots les plus populaires comme *(bronx, yankee)* ou *(reddsox, homeruns)*, qui passent respectivement du rang 211 à 34 et du rang 86 à 11. Intuitivement, il y a plus de chance de parler de *baseball* près d'un *stade de baseball*, ce qui se refléterai par de fortes similarités entre les mots considérés. De fait, cette stimulation de similarité que nous pouvons observer est tout a fait naturelle. En effet, « *Yankees* » (ou « *Yanks* ») est une équipe de baseball du Bronx,

dont le « *Yankee Stadium* » est le principal stade. Cette similarité spatiale est corroborée par le facteur d'amortissement qui atteint une valeur approximative de $0,39$. En revanche, les paires restantes, incluant par exemple *(reddsox, good)* et *(summer, original)*, sont respectivement passées du rang 15 à 29 et du rang 21 à 87 avec une valeur de $F(d^*)$ d'environ $0,31$ seulement. Cela pourrait être dû au caractère générique de ces mots puisqu'ils peuvent être utilisés dans des contextes spatiaux complètement différents. L'introduction du facteur d'amortissement permet de les écarter de manière significative.

5.2 *Analyse des scénarios (QR2)*

Nous poursuivons notre évaluation expérimentale en mesurant la pertinence de chacun des composants des modèles. Les résultats empiriques de l'évaluation en termes de *MRR* et d'*Acc@k* ($k = 1, 5, 10$), discutés dans toute section, sont résumés dans le Tableau 5.5 pour les jeux de données de NY (Tableau 5.5a) et de SG (Tableau 5.5b). Par ailleurs, nous calculons les taux d'améliorations relatifs (%Chg) des modèles **SGM** et $\mathbf{SGM}_{LCH}$ par rapport aux différents scénarios (Scénarios) et variantes de l'histogramme (Hist.).

La suite de cette section est organisée comme suit. Dans la Section 5.2.1 nous étudions la qualité de l'histogramme d'appariement calculé selon plusieurs méthodes d'agrégation. Nous poursuivons notre analyse en mesurant l'effet des interactions locales dans la Section 5.2.2. Enfin, dans la Section 5.2.3 nous évaluons l'apport des interactions globales textuelles et spatiales sur les performances des modèles.

5.2.1 *Analyse de l'histogramme d'appariement*

Nous l'avons vu dans la section précédente, l'histogramme d'appariement est un composant essentiel du modèle **SGM**. En effet, celui-ci permet d'apprendre des motifs d'appariement nécessaires au calcul du score de pertinence. La qualité des motifs appris dépend donc de la qualité des données transmises au réseau de neurones, et donc de la manière dont la matrice d'interaction est agrégée. Dans ce qui suit, nous allons d'abord étudier l'impact de chacune des méthodes d'agrégation proposées pour générer les histogrammes d'appariement. Ensuite, nous analyserons un histogramme d'appariement de plus près afin de comprendre comment le modèle l'utilise pour apprendre les représentations latentes.

		MRR		Acc@1		Acc@5		Acc@10	
	Modèle	**Valeur**	**%Chg**	**Valeur**	**%Chg**	**Valeur**	**%Chg**	**Valeur**	**%Chg**
	SGM	0,701	-	0,597	-	0,827	-	0,891	-
Scénarios	$\mathbf{SGM}_{I_l-F}$	0,662	+5,89% **	0,549	+8,74% **	0,801	+3,25% **	0,878	+1,48% **
	$\mathbf{SGM}_{I_l}$	0,603	+16,25% **	0,479	+24,63% **	0,763	+8,39% **	0,856	+4,09% **
	$\mathbf{SGM}_{\varnothing}$	0,575	+21,91% **	0,435	+37,24% **	0,751	+10,12% **	0,852	+4,58% **
	$\mathbf{SGM}_{I_g^S}$	0,687	+2,04% **	0,583	+2,40% **	0,809	+2,22% **	0,882	+1,02% *
	$\mathbf{SGM}_{I_g^T}$	0,589	+19,02% **	0,463	+28,94% **	0,742	+11,46% **	0,837	+6,45% **
Hist.	$\mathbf{SGM}_{LCH}$	0,592	-	0,463	-	0,749	-	0,849	-
	$\mathbf{SGM}_{CH}$	0,441	+34,24% **	0,283	+63,60% **	0,639	+17,21% **	0,792	+7,20% **
	$\mathbf{SGM}_{NCH}$	0,162	+265,43% **	0,043	+976,74% **	0,238	+214,71% **	0,464	+82,97% **
	$\mathbf{SGM}_{SH}$	0,513	+15,40% **	0,358	+29,33% **	0,707	+5,94% **	0,838	+1,31% *
	$\mathbf{SGM}_{Conv}$	0,574	+3,14% **	0,442	+4,75% **	0,747	+0,27%	0,851	-0,24%

(a) Jeu de données de NY

		MRR		Acc@1		Acc@5		Acc@10	
	Modèle	**Valeur**	**%Chg**	**Valeur**	**%Chg**	**Valeur**	**%Chg**	**Valeur**	**%Chg**
	SGM	0,757	-	0,676	-	0,854	-	0,903	-
Scénarios	$\mathbf{SGM}_{I_l-F}$	0,728	+3,98% **	0,639	+5,79% **	0,834	+2,40% **	0,888	+1,69% **
	$\mathbf{SGM}_{I_l}$	0,641	+18,10% **	0,527	+28,27% **	0,777	+9,91% **	0,854	+5,74% **
	$\mathbf{SGM}_{\varnothing}$	0,622	+21,70% **	0,501	+34,93% **	0,766	+11,49% **	0,849	+6,36% **
	$\mathbf{SGM}_{I_g^S}$	0,746	+1,47% *	0,659	+2,58% **	0,852	+0,23%	0,902	+0,11%
	$\mathbf{SGM}_{I_g^T}$	0,645	+17,36% **	0,533	+26,83% **	0,780	+9,49% **	0,854	+5,74% **
<	$\mathbf{SGM}_{LCH}$	0,641	-	0,527	-	0,777	-	0,854	-
	$\mathbf{SGM}_{CH}$	0,438	+46,35% **	0,286	+84,27% **	0,617	+25,93% **	0,617	+38,41% **
Hist.	$\mathbf{SGM}_{NCH}$	0,257	+149,42% **	0,104	+406,73% **	0,416	+86,78% **	0,624	+36,86% **
	$\mathbf{SGM}_{SH}$	0,575	+11,48% **	0,443	+18,96% **	0,735	+5,71% **	0,835	+2,28% **
	$\mathbf{SGM}_{Conv}$	0,620	+3,28% **	0,489	+7,21% **	0,778	-0,13%	0,858	+-0,47%

(b) Jeu de données de SG

Tableau 5.5 – Comparaison des performances du modèle **SGM** selon différentes configurations. La différence significative par rapport aux modèles **SGM** (Scénarios) ou $\mathbf{SGM}_{LCH}$ (Hist) est déterminée par le test t de Welch ($* : p < 0,05$, $** : p < 0,01$).

5.2.1.1 *Effet des différentes méthodes d'agrégation*

Dans cette section, nous menons un ensemble d'expériences pour déterminer quelle fonction d'agrégation utiliser, parmi celles listées dans la Section 3.2.3, pour construire les histogrammes d'appariement. Les principales observations qui ressortent du Tableau 5.5 sont les suivantes :

1. le modèle $\mathbf{SGM}_{LCH}$ surpasse largement les autres modèles pour les deux jeux de données. Avec une valeur de *MRR* de 0,641 (resp. 0,592) pour la collection de SG (resp. NY), il améliore les résultats de la méthode $\mathbf{SGM}_{CH}$ de 46,35% (resp. 34,24%). En effet, près de 52,7% (resp. 46,3%) des tweets sont correctement associés à leur POI lorsque nous considérons les résultats du top-1 (c.-à-d. *Acc*@1). Les bonnes performances du modèle $\mathbf{SGM}_{LCH}$ confirment donc que le modèle neuronal a tiré profit d'un signal d'entrée lissé et non linéaire, contrairement à celui du modèle $\mathbf{SGM}_{CH}$, issu d'une simple agrégation par comptage, qui a tendance à favoriser les documents les plus longs ;
2. le modèle $\mathbf{SGM}_{NCH}$ est nettement moins performant que les autres modèles. En effet, seulement 4,3% (resp. 10,4%) des tweets sont correctement associés à leur POI pour la collection de NY (resp. SG) en considérant les résultats du top-1 (c.-à-d. *Acc*@1). En envisageant une précision moins restrictive (c.-à-d. *Acc*@5 and *Acc*@10), les résultats ne sont pas plus prometteurs. Nous obtenons ainsi des valeurs de *MRR* très faibles, avec seulement 0,162 pour le jeu de données de NY et 0,257 pour celui de SG. Avec la méthode d'agrégation NCH, axée sur le comptage relatif des valeurs plutôt que absolu, le modèle $\mathbf{SGM}_{NCH}$ a perdu toute l'information sur la longueur des documents. Hors, dans une tâche de RI, cette donnée reste importante ;
3. la méthode par sommation, utilisée pour le modèle $\mathbf{SGM}_{SH}$ donne également de moins bons résultats que la méthode par comptage logarithmique. La valeur de l'*Acc*@1 atteint seulement 0,358 pour la collection de NY et 0,443 pour la collection de SG. En additionnant les valeurs de chaque classe, le modèle n'est pas en mesure de comprendre l'importance des différents niveaux des signaux d'appariement.
4. lorsque l'histogramme d'appariement est remplacé par un réseau de neurones à convolution ($\mathbf{SGM}_{Conv}$), nous remarquons que la performance du modèle est dégradée. Plus spécifiquement, la valeur *MRR* du modèle $\mathbf{SGM}_{Conv}$ diminue de 0,592 à 0,574 (resp. de 0,641 à 0,620) par rapport au modèle $\mathbf{SGM}_{LCH}$ pour le jeu de données de NY (resp. SG). Ainsi, seuls 44% (resp. 48%) des tweets sont correctement associés à leur POI correspondant pour la collection de de NY (resp. SG) lorsque nous considérons le résultats dans la top-1 (c.-à-d. *Acc*@1). Cet écart de performance peut s'expliquer par la spécificité de la convolution. En effet, de par sa définition, la convolution permet de capturer les positions et l'ordre d'apparition des motifs d'appariement. Les modèles de RI qui utilisent la convolution sont généralement conçus pour résoudre des tâches de TALN, telles que l'élaboration de systèmes de questions-réponses, la traduction automatique ou la désambiguïsation lexicale. Hors, comme l'ont déjà démontré Huang *et al.* (2013), ce type de tâche concerne plutôt l'identification de la signification sémantique et l'inférence des relations sémantiques entre deux éléments de texte. C'est

ainsi que la convolution prend sens, en repérant des portions de textes pertinentes. Dans notre cas, nous avons plutôt affaire à une tâche de RI *ad-hoc*, qui consiste principalement à déterminer si un document (ici un POI) est pertinent pour une requête donnée (ici un tweet). Ce n'est donc ni la position ni l'ordre d'apparition des motifs d'appariement qui nous intéresse, mais plutôt les niveaux de similarité, que nous récupérons à l'aide des histogrammes d'appariement.

5.2.1.2 *Analyse des poids de l'histogramme d'appariement*

Nous allons maintenant étudier l'histogramme d'appariement de plus près, et tenter de visualiser comment le modèle l'utilise pour déterminer les représentations latentes des interactions locales. Dans cette optique, nous avons récupéré la matrice de poids $\mathbf{W}^{(2)}$ du modèle **SGM** qui, combinée à l'histogramme d'appariement, permet de calculer les représentations latentes $\mathbf{z}_i^{(2)}$ (Équation 5.9). Puisque la matrice de poids est bidimensionnelle (200 classes $\times$ 10 noeuds) , nous avons moyenné les poids de chaque classe et affiché dans la Figure 5.8 sa courbe moyenne d'évolution (ligne bleu foncé). La zone bleu clair correspond à des valeurs de poids situées à plus ou moins l'écart-type autour de la moyenne. L'axe des abscisses représente l'index des classes de l'histogramme, dont les valeurs de signaux s'étendent de $[-K(0), K(0)]$, et l'axe des ordonnées correspond au poids moyen de chaque classe.

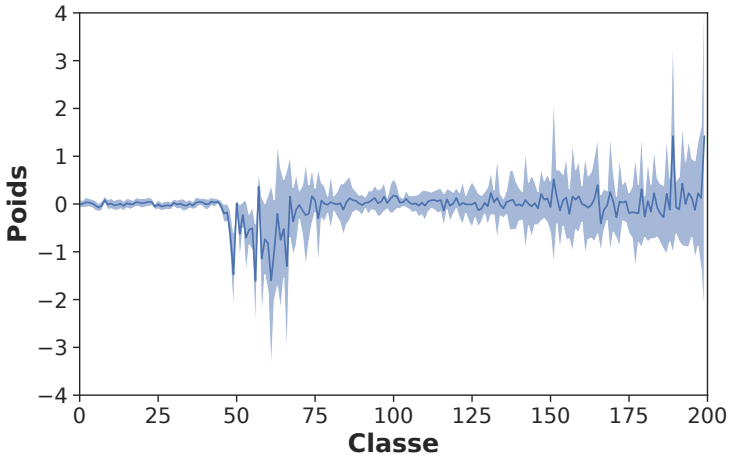


Figure 5.8 – Visualisation des poids $\mathbf{W}^{(2)}$ associés aux histogrammes d'appariement pour le modèle **SGM**. L'axe des abscisses est l'index des classes de l'histogramme, et l'axe des ordonnées est la valeur du poids appris pour chaque classe. La courbe bleue représente la moyenne des poids de chaque classe. La zone bleue est la distribution à +/- l'écart-type autour du poids moyen.

Nous pouvons remarquer, à partir de la Figure 5.8, que le signal peut être divisé en trois parties, où chacune d'entre elle présente un comportement spécifique :

1. dans les classes 0 à 45, correspondant à une plage de similarité négative sur l'intervalle $[-K(0); -0,25]$, le signal est relativement plat et le poids proche de 0. La raison principale est que les scores de similarité, calculés à partir des plongements lexicaux, sont traditionnellement positifs. Comme les valeurs de ces classes sont généralement égales à 0, elles ne sont pas considérées lors de l'entraînement du modèle, engendrant ainsi des poids nuls ;
2. dans les classes 45 à 100, correspondant à une plage de similarité négative sur l'intervalle $[-0,25; 0]$, le signal est plus chaotique. Le modèle semble allouer une grande importance aux valeurs de ces classes et les considère utiles pour déterminer les représentations latentes des interactions. Nous pouvons raisonnablement soutenir que le modèle utilise ces signaux pour diminuer le score de pertinence des documents ;
3. dans les classes 100 à 200, correspondant à une plage de similarité positive sur l'intervalle $[0, K(0)]$, l'intensité du signal augmente progressivement. Plus les signaux de similarités sont forts (c.-à-d. plus l'index des classes est élevé), plus ils auront une influence pour le calcul des représentations latentes des interactions. Ici, nous pouvons penser que le modèle utilise ces signaux pour augmenter le score de pertinence des documents.

5.2.2 *Effet des interactions locales*

Nous continuons notre analyse en nous focalisant maintenant sur les interactions locales. Pour cela, nous évaluons l'impact du facteur d'amortissement spatial $F(d^*)$ sur les performances du modèle **SGM**.

Dans un premier temps, nous proposons d'évaluer l'impact du facteur d'amortissement en comparant le modèle complet **SGM** avec le scénario $\mathbf{SGM}_{I_l - F}$. Comme résumé dans le Tableau 5.5, enrichir les matrices d'interaction avec un facteur spatial permet un gain de performance significatif, quels que soient les modèles et les collections considérés. En effet, regardons en détail le modèle **SGM**. Son *Acc*@1 progresse de 0,639 à 0,676 (resp. 0,549 à 0,597) pour le jeu de donnés de SG (resp. NY). Cela correspond à un taux d'amélioration significatif de 5,79% (resp. 8,74%). Lorsque nous considérons des valeurs de précision plus élevées (c.-à-d. *Acc*@10), près de 9 tweets sur 10 sont correctement associés à leur POI lorsque nous intégrons le facteur d'amortissement.

Dans un deuxième temps, nous comparons le modèle dégradé $\mathbf{SGM}_{I_l}$ avec le modèle $\mathbf{SGM}_{\varnothing}$ pour étudier la contribution du facteur d'amortissement spatial $F(d^*)$ indépendamment des interactions globales. Comme nous pouvons le voir

dans le Tableau 5.5, intégrer le facteur $F(d^*)$ permet d'améliorer significativement les performances du modèle, quelle que soit la collection. Notons que l'amélioration est encore plus marquée dans cette configuration, avec une valeur de l'*Acc*@1 qui passe de $0,435$ à $0,463$ (resp. de $0,501$ à $0,527$) pour la collection de NY (resp. SG), soit un taux d'amélioration de $6,44\%$ (resp. $5,19\%$).

En résumé, ces résultats indiquent clairement que la prise en compte des interactions locales spatiales mot à mot au sein des modèles neuronaux est utile pour notre tâche de prédiction, et ce, quelle que soit l'architecture adoptée avec l'utilisation ou non des interactions globales.

5.2.3 *Effet des interactions globales*

Pour compléter l'analyse des différents scénarios, et évaluer l'impact de chacun des composants des modèles, nous nous intéressons maintenant aux interactions globales. Dans cette section, nous allons voir comment l'appariement exact de pertinence (via les interactions locales) et l'appariement sémantique (via les interactions globales) se complètent mutuellement, et dans quelle mesure les caractéristiques globales textuelle et spatiale sont utiles pour l'appariement tweet-POI.

En comparant les modèles **SGM** et $\mathbf{SGM}_{I_l}$ dans le Tableau 5.5, nous voyons clairement que ne pas modéliser les interactions globales dégrade considérablement les performances des modèles. Par exemple, l'*Acc*@1 atteint seulement $0,463$ (resp. $0,527$) pour le jeux de données de NY (resp. SG) contre $0,597$ (resp. $0,676$) avec le modèle **SGM**. En ce qui concerne la *MRR*, nous obtenons une valeur de $0,592$ (resp. $0,641$) pour la collection de NY (resp. SG) contre $0,701$ (resp. $0,757$) avec le modèle **SGM**.

En examinant la contribution intrinsèque de chaque interaction globale I_g^S et I_g^T, nous constatons que l'interaction spatiale I_g^S contribue généralement mieux à la performance des modèles que l'interaction textuelle. Par exemple, dans le cas où les caractéristiques spatiales ne sont pas prises en compte dans le modèle **SGM** (c.-à-d. $\mathbf{SGM}_{I_g^T}$), la *MRR* diminue considérablement de $0,701$ à $0,589$ (resp. de $0,757$ à $0,641$) pour la collection de NY (resp. SG). Sans tenir compte des caractéristiques textuelles I_g^T dans le modèle **SGM** (c.-à-d. $\mathbf{SGM}_{I_g^S}$), la *MRR* ne diminue que très légèrement de $0,701$ à $0,687$ (resp. $0,757$ à $0,746$) pour les données de NY (resp. SG). Néanmoins, en comparant le modèle $\mathbf{SGM}_{I_l}$ avec le modèle $\mathbf{SGM}_{I_g^T}$, nous remarquons que, même si les contributions intrinsèques de l'interaction textuelle I_g^T ne sont pas nécessairement significatives selon les collections et les mesures considérés, sa combinaison avec l'interaction spatiale I_g^S permet d'améliorer significativement la performance globale des modèles.

5.3 *Comparaison des performances (QR3)*

Le troisième objectif de cette évaluation expérimentale est de mesurer l'efficacité du modèle **SGM** par rapport aux modèles d'appariement de référence détaillés dans la Section 4.3. Le Tableau 5.6 résume les résultats empiriques obtenus en termes de *MRR* et d'*Acc@k* ($k = 1, 5, 10$) pour les collections de NY (Tableau 5.6a) et de SG (Tableau 5.6b). L'organisation de la section est la suivante. Nous commençons par étudier les performances globales du modèle **SGM** à travers une analyse quantitative (Section 5.3.1), puis nous menons une analyse qualitative sur un échantillon de tweets.

	MRR		Acc@1		Acc@5		Acc@10	
Modèle	**Valeur**	**%Chg**	**Valeur**	**%Chg**	**Valeur**	**%Chg**	**Valeur**	**%Chg**
Notre contribution								
SGM	0,701	-	0,597	-	0,827	-	0,891	-
Appariement spatial								
DIST	0,501	+39,92% *	0,410	+45,61% *	0,603	+37,15% *	0,691	+28,94% *
Appariement textuel								
BM25	0,452	+55,09% *	0,326	+83,13% *	0,706	+ 17,14% *	0,872	+2,18% *
TFIDF$^+$-D	0,416	+68,51% *	0,311	+91,96% *	0,548	+50,91% *	0,633	+40,76% *
Appariement spatial et textuel								
CLASS	0,584	+20,03% *	0,498	+19,88% *	0,684	+20,91% *	0,724	+23,07% *
SBM	0,390	+79,74% *	0,343	+74,05% *	0,451	+83,37% *	0,473	+88,37% *
Appariement neuronal								
ARC-I	0,546	+28,39% *	0,380	+57,11% *	0,772	7,12% *	0,881	+1,14% *
ARC-II	0,515	+36,12% *	0,341	+75,07% *	0,755	+9,54% *	0,870	+2,41% *
ANMM	0,596	+17,62% *	0,461	+29,50% *	0,771	+7,26% *	0,871	+2,30% *
ANMM+$F(d^*)$	0,599	+17,03% *	0,466	+28,11% *	0,765	8,10% *	0,868	+2,65% *
DRMM	0,644	+8,85% *	0,514	+16,15% *	0,810	+2,10% *	0,894	+0,34%
DRMM+$F(d^*)$	0,622	+12,70% *	0,492	+21,34% *	0,786	+5,22% *	0,873	+2,06% *
Appariement de représentations distribuées de géotextes								
WORD2VEC	0,356	+96,91% *	0,218	+173,85% *	0,499	+65,73% *	0,650	+37,08% *
DISTILBERT	0,187	+274,48% *	0,095	+528,42% *	0,258	+220,54% *	0,374	+138,23% *
SBERT$_{STS}$	0,300	+133,67% *	0,177	+237,29% *	0,418	+97,84% *	0,565	+57,70% *
SBERT$_{NY}$	0,712	-1,54% *	0,612	-2,45% *	0,821	+0,73%	0,900	-1,00%

(a) Jeu de données de NY

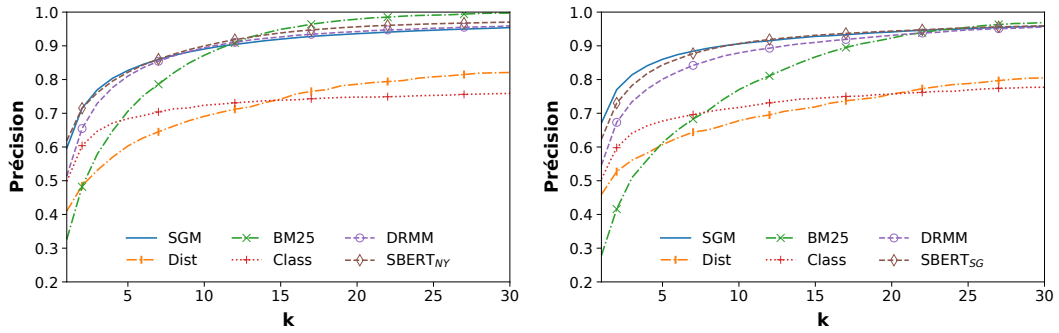
	MRR		Acc@1		Acc@5		Acc@10	
Modèle	**Valeur**	**%Chg**	**Valeur**	**%Chg**	**Valeur**	**%Chg**	**Valeur**	**%Chg**
Notre contribution								
SGM	0,757	-	0,676	-	0,854	-	0,903	-
Appariement spatial								
DIST	0,534	+41,76% *	0,459	+47,28% *	0,607	+40,69% *	0,678	+33,19% *
Appariement textuel								
BM25	0,381	+98,69% *	0,278	+143,17% *	0,613	+39,31% *	0,770	+17,27% *
TFIDF$^+$-D	0,411	+84,18% *	0,314	+115,29% *	0,545	+56,70% *	0,622	+45,18% *
Appariement spatial et textuel								
CLASS	0,586	+29,18% *	0,505	+33,86% *	0,677	+26,14% *	0,717	+25,94% *
SBM	0,200	+278,50% *	0,170	+297,65% *	0,241	+254,36% *	0,256	+252,73% *
Appariement neuronal								
ARC-I	0,511	+48,14% *	0,367	+84,20% *	0,694	+23,05% *	0,809	+11,62% *
ARC-II	0,500	+51,40% *	0,355	+90,42% *	0,680	+25,59% *	0,808	+11,76% *
ANMM	0,623	+21,51% *	0,502	+34,66% *	0,776	+10,05% *	0,858	+5,24% *
ANMM+$F(d^*)$	0,620	+22,10% *	0,501	+34,93% *	0,766	+11,49% *	0,854	+5,74% *
DRMM	0,661	+14,52% *	0,546	+23,81% *	0,801	+6,62% *	0,879	+2,73% *
DRMM+$F(d^*)$	0,653	+15,93% *	0,538	+25,65% *	0,795	+7,42% *	0,870	+3,79% *
Appariement de représentations distribuées de géotextes								
WORD2VEC	0,320	+136,56% *	0,187	+261,50% *	0,449	+90,20% *	0,608	+48,52% *
DISTILBERT	0,171	+342,69% *	0,085	+695,30%	0,239	+257,32%	0,344	+162,50% *
SBERT$_{STS}$	0,282	+168,44% *	0,165	+309,70% *	0,391	+118,41% *	0,527	+71,35% *
SBERT$_{SG}$	0,721	+5,00% *	0,623	+8,51% *	0,844	+1,18% *	0,906	-0,33%

(b) Jeu de données de SG

Tableau 5.6 – Comparaison des performances du modèle **SGM** par rapport aux modèles de référence, La différence significative par rapport à **SGM** est déterminée par le test t de Welch ($* : p < 0,01$)

5.3.1 *Analyse des performances globales*

Dans l'ensemble, nous remarquons à partir du Tableau 5.6, que le modèle **SGM** surpasse largement les performances de presque tous les modèles de l'état-de-l'art (15/16), quels que soient les mesures et les jeux de données considérés. La seule exception est la variante SBERT$_X$ qui a été spécialement entraînée sur la tâche d'appariement sémantique. Les résultats indiquent donc que le modèle **SGM** permet un appariement tweet-POI plus efficace. Plus précisément, près de 6 tweets sur 10 sont correctement associés à leur POI pour les deux collections, selon les résultats du top-1 (c.-à-d. *Acc*@1). Cela correspond à un taux d'amélioration signi-



(a) Jeu de données de NY (b) Jeu de données de SG

Figure 5.9 – Évolution de l'*Acc@k* pour le modèle **SGM** et une sélection de modèles de référence.

ficatif compris entre 16, 15% et 91, 96% (resp. 23, 81% et 297, 65%) par rapport aux performances des modèles de référence pour la collection de NY (resp. SG). En regardant l'*Acc*@10 du modèle **SGM**, près de 90% des tweets sont correctement associés à leurs POIs pour les deux jeux de données. Sa *MRR* atteint des valeurs de 0, 701 et 0, 757 pour les collections de NY et de SG, augmentant respectivement les performances de l'appariement de 34, 91% et 64, 17% en moyenne.

Pour approfondir l'analyse des résultats, nous traçons, dans la Figure 5.9, l'évolution de l'*Acc@k*, pour $k \in [\![1, 30]\!]$, du modèle **SGM** et des modèles de référence, pour les jeux de données de NY (Figure 5.9a) et de SG (Figure 5.9b). Pour une meilleure lisibilité, nous n'affichons que les modèles représentatifs : Dist, BM25, Class, Arc-I et SBERT$_X$. En recoupant les résultats mis en évidence dans le Tableau 5.6 et la Figure 5.9, nous faisons les observations suivantes.

Appariement spatial. Utiliser les caractéristiques spatiales pour l'appariement d'objets géotextuels est incontestablement bénéfique. En effet, le modèle Dist (ligne tiretée-pointillée verte) est l'un des modèles de référence les plus performants pour de faibles valeurs de k. Nous pouvons voir dans le Tableau 5.6 que plus de 40% des tweets sont correctement associés à leur POI lorsque nous considérons le résultat du top-1 (c.-à-d. *Acc*@1). Toutefois, les résultats restent inférieurs à ceux obtenus par le modèle **SGM**, indiquant donc l'intérêt de l'utilisation conjointe des caractéristiques textuelles et spatiales.
Par ailleurs, en examinant les performances des modèles sBM et Class (ligne pointillée violette), et en comparant leurs performances avec les modèles Dist et **SGM**, nous pouvons conclure que la méthode utilisée pour combiner les caractéristiques spatiales et textuelles est cruciale. La combinaison de la distance avec un modèle bayésien (sBM) dégrade les performances. L'*Acc*@1 diminue de 0, 410 à 0, 343 (resp. de 0, 459 à 0, 170) pour la collection de NY (resp. SG). En revanche, l'utilisation

conjointe des caractéristiques spatiales et d'un modèle de langue (CLASS) améliore l'efficacité de l'appariement de 21% (resp. 10%) pour la collection de NY (resp. SG).

Appariement textuel. L'appariement textuel exact est certes utile, mais insuffisant pour capturer efficacement les relations sémantiques entre les textes. Ces relations ne peuvent être déduites qu'à un niveau spatial ou à un niveau profond de représentations des sens des mots (p. ex. via des plongements lexicaux). En effet, parmi les modèles de référence qui reposent sur l'appariement textuel (c.-à-d. BM25, TFIDF$^+$-D), le modèle BM25 obtient une *Acc*@1 très faible, de seulement 0,326 et 0,278 pour les collections de NY et de SG, mais atteint rapidement de bonnes valeurs de précision, et dépasse tous les autres modèles lorsque nous considérons des valeurs de k plus élevées, comme nous pouvons le voir dans la Figure 5.9 (ligne tiretée-pointillée rouge).

Appariement sémantique. Utiliser seulement l'appariement sémantique, comme c'est le cas pour les modèles neuronaux d'appariement (ARC-I, ARC-II, aNMM, DRMM), ne suffit pas pour pallier l'inadéquation du vocabulaire. En revanche, combiner les approches d'appariement exact et sémantique, comme nous l'avons proposé avec le modèle **SGM**, est bénéfique pour l'appariement tweet-POI. En effet, en regardant les résultats des modèles ARC-I et ARC-II qui n'utilisent que l'appariement sémantique, nous pouvons remarquer qu'ils se classent parmi les pires modèles pour la collection de NY, avec une *Acc*@1 qui ne dépasse pas 38%. Il en est de même pour le jeu de données de SG, qui présente une *Acc*@1 de seulement 36,7%.
En ce qui concerne les modèles aNMM et DRMM (ligne tiretée violette), nous remarquons qu'ils surpassent tous les autres modèles de référence lorsque $k = 1$ pour la collection de SG, et $k = 2$ pour la collection de NY, mais n'atteignent pas les performances du modèle **SGM**. En considérant des valeurs plus élevées de k ($k > 8$ pour la collection de NY et $k > 22$ pour la collection de SG), nous voyons sur la Figure 5.9 que les modèles neuronaux parviennent à atteindre les performances du modèle **SGM**.
Enfin, en examinant les variantes DRMM+$F(d^*)$ et aNMM+$F(d^*)$, augmentées par le facteur d'amortissement $F(d^*)$, nous pouvons conclure que celui-ci ne permet pas d'obtenir un gain de performance significatif. Au contraire, combiner le facteur d'amortissement avec le réseau de pondération des mots (*term gating network*) dégrade les performances de l'appariement. La *MRR* du modèle DRMM diminue de 0,644 à 0,622 (resp. de 0,661 à 0,653) pour la collection de NY (resp. SG) lorsque nous ajoutons le facteur d'amortissement $F(d^*)$.

Appariement de représentations distribuées de géotextes. L'analyse des résultats des modèles d'appariement s'appuyant sur les représentations distribuées des géotextes nous permet de constater que le simple appariement de représentations

de tweets et de POIs est généralement beaucoup moins efficace que l'apprentissage d'une fonction d'appariement sémantique spatial comme le fait le modèle **SGM**. Plus spécifiquement, nous pouvons remarquer, les modèles **SBERT** mis à part, que l'amélioration de la *MRR* varie entre 96, 91% et 274, 87% (resp. entre 136, 56% et 342, 69%) pour la collection de NY (resp. SG) pour le modèle **SGM**.
En ce qui concerne les modèles s'appuyant sur l'architecture de BERT, nous constatons ce qui suit : *(1)* l'utilisation des plongements lexicaux DistillBert sans ajustements réalise de moins bonnes performances que les plongements lexicaux traditionnels Word2Vec. Plus précisément, pour le jeu de données de NY (resp. SG), environ 22% (resp. 19%) des tweets sont correctement associés à leur POI correspondant avec le modèle Word2Vec, contre environ 10% (resp. 9%) avec le modèle DistillBert. Nos résultats corroborent donc les conclusions de Reimers et Gurevych (2019) ; *(2)* : l'ajustement des représentations distribuées du modèle BERT à l'aide d'une tâche de similarité sémantique (c.-à-d. les modèles SBERT_{STS}, SBERT_{NY} et SBERT_{SG}) permet d'améliorer les performances de prédiction. Cette augmentation est faible lorsque la collection utilisée pour l'étape d'ajustement est différente de la collection de test (c.-à-d. SBERT_{STS}), mais elle est significative et permet d'obtenir des résultats comparables au modèle **SGM** lorsque la collection utilisée pour l'étape d'ajustement est identique à la collection de test (c.-à-d. SBERT_{X}). En effet, la *MRR* augmente de 0, 300 à 0, 712 (resp. de 0, 282 à 0, 721) pour le jeu de données de NY (resp. SG). Par ailleurs, nous notons que le modèle SBERT_{NY} réalise une performance légèrement plus élevée que le modèle **SGM** sur le jeu de données NY, avec une *Acc@1* de 0, 612 mais que l'*Acc@1* du modèle SBERT_{SG} est légèrement plus faible (0, 623) que le modèle **SGM** sur la collection de SG. Ainsi, ces représentations peuvent être utilisées pour un appariement sémantique efficace au prix d'un ajustement très spécifique et coûteux en temps et en ressources.

5.3.2 *Analyse qualitative d'un échantillon de tweets*

Nous effectuons maintenant une analyse qualitative au niveau des tweets, pour déterminer les raisons du succès ou de l'échec du modèle **SGM** par rapport aux modèles de référence. Nous choisissons, dans chaque catégorie définie dans la Section 4.3, les modèles qui ont donné les meilleurs résultats en terme d'*Acc*@1, à savoir Dist, BM25 et SBERT_{X}. Nous commençons par identifier les ensembles de tweets pour lesquels notre modèle **SGM** a fait moins bien (T^{-}), à obtenu le même résultat ($\text{T}^{=}$), ou a fait mieux (T^{+}) que les modèles sélectionnés. Les résultats sont présentés dans le Tableau 5.7.

D'après le Tableau 5.7, nous pouvons remarquer que, par rapport au modèle Dist, le modèle **SGM** améliore la qualité d'appariement de près de la moitié des tweets pour les deux jeux de données. Plus précisément, 49, 60% (resp. 44, 40%) des tweets sont mieux associés à leur POI avec le modèle **SGM** qu'avec le modèle

	NY			SG		
Modèle	$\mathbf{T^+}$	$\mathbf{T^=}$	$\mathbf{T^-}$	$\mathbf{T^+}$	$\mathbf{T^=}$	$\mathbf{T^-}$
DIST	49,60%	34,58%	15,82%	44,40%	39,25 %	16,35%
BM25	61,21%	23,03%	15,76%	67,65%	18,01%	14,34%
DRMM	52,09%	32,55%	15,36%	48,81%	33,61%	17,58%
SBERT$_X$	28,20%	42,82%	28,98%	27,23%	48,26%	24,51%

*T^+ : pourcentage de tweets pour lesquels **SGM** dépasse le modèle considéré.*
*$T^=$: pourcentage de tweets pour lesquels **SGM** a fait autant que le modèle considéré.*
*T^- : pourcentage de tweets pour lesquels **SGM** ne dépasse pas le modèle considéré.*

Tableau 5.7 – Analyse qualitative des performances du modèle **SGM**.

DIST pour le jeu de données de NY (resp. SG). Cela confirme l'importance d'associer les caractéristiques spatiales et textuelles, comme le fait le modèle **SGM**, pour améliorer l'appariement tweet-POI.
En examinant maintenant le modèle BM25, nous constatons que le modèle **SGM** l'égale ou le dépasse pour près de 95% des tweets. Ces résultats démontrent qu'un appariement exact n'est pas en mesure de résoudre le problème de discordance du vocabulaire pourtant très présent dans la tâche de prédiction sémantique de l'emplacement, comme nous l'avons évoqué dans la Section 2.
Lorsque nous comparons le modèle **SGM** avec le modèle DRMM, nous observons qu'il améliore les performances de 52, 09% (resp. 48, 81%) des tweets pour la collection de NY (resp. SG), tandis qu'il pèche pour seulement 15, 36% (resp. 17, 58%) des tweets. Ces résultats nous confortent dans l'utilisation des interactions globales et locales, mais aussi dans l'intégration du facteur d'amortissement $F(d^*)$ pour effectuer un appariement de qualité.
Enfin, comme l'expliquent les performances proches, la comparaison entre le modèle **SGM** et le modèle SBERT$_X$ révèle une tendance similaire entre les deux collections de données : ces deux modèles atteignent des performances équivalentes pour près de la moitié des tweets (42, 82% dans la collection de NY, 48, 26% dans la collection de SG) et chacun surpasse son homologue pour environs un quart des autres tweets (p. ex. 28, 20% pour l'ensemble de données de NY). Cela corrobore la force de l'appariement sémantique établie par les deux modèles. Alors que le modèle **SGM** repose sur des interactions spatiales explicites lors de l'appariement du tweet avec un POI, le modèle SBERT$_X$ semble les inférer grâce à l'ajustement du modèle SBERT avec la collection qui implique le même espace géographique. Cela pourrait d'ailleurs expliquer les performances inférieures du modèle SBERT$_{STS}$ comme nous l'avons montré précédemment.

	SGM	DIST	BM25	DRMM	SBERT$_X$
Tweet #1 (Négatif) best lamb flushing lamb skewers lamb burgers lamb noodles #asianbrunch	1. Starbucks	**1. Biang!**	1. Xi'an Famous Foods	1. Starbuck	**1. Biang!**
	2. Pho Bang	2. Xi'an Famous Foods	**2. Biang!**	2. Pho Bang	2. Hing Long Supermarket
	3. Queens Library	3. Prudential Insurance Co.	3. Lamb Noodle Soup	**3. Biang!**	3. Hai Zhen Zhu Live Seafood Restaurant
	4. Biang!	4. Pho Hoang	4. Lamb BBQ	4. Xi'an Famous Foods	4. Q48 moving target
	5. Xi'an Famous Foods	5. Mi Ni Sweet Food	5. Lan Zhou Hand Made Noodle	5. Lan Zhou Hand Made Noodle	5. The Ho-ho
Tweet #2 (Positif) best burgers bayridge brooklyn new york ny #bestburger #bayridgebrooklyn	**1. Brooklyn Beet Compagny**	1. McGovern's Wine Cellar	1. Le Nar Cafe	1. Vesuvio Pizzeria Restaurant	1. Cream Doughnuts
	2. Vesuvio Pizzeria Restaurant	2. Gaspar Veliz	2. Vicky Simegiatos Dance School	2. The Burger Bistro	2. Mohims Indian Cuisine
	3. Longbow Pub Pantry	3. Yours Car Service	3. Chubby's Chicken Burgers	**3. Brooklyn Beet Compagny**	3. Catch 22
	4. The Burger Bistro	4. Dmitry Maryanovsky	4. The Burger Bistro	4. Longbow Pub Pantry	**4. Brooklyn Beet Compagny**
	5. Polonica Restaurant	5. Ridge Computer Service	**5. Brooklyn Beet Compagny**	5. Taj Mahal	5. Bay Kebab

Tableau 5.8 – Exemples d'échec et de succès d'appariements effectués par le modèle **SGM**.

Afin de mieux interpréter ces résultats, nous présentons dans le Tableau 5.8 un exemple d'échec et de succès d'appariements réalisés par le modèle **SGM**. Plus précisément, nous rapportons pour un appariement échoué et un appariement réussi, le top-5 des POIs retournés par le modèle **SGM** et par les modèles de référence DIST, BM25 et DRMM, incluant (s'il est présent), le POI pertinent (en gras).

Nous pouvons observer à partir du Tableau 5.8 que pour l'instance négative, le modèle **SGM** obtient des résultats comparables avec le modèle DRMM puisqu'ils classent respectivement le POI pertinent (**Biang!**) aux rangs 3 et 4 et qu'ils ren-

voient les mêmes résultats aux rangs 1 (*Starbuck*) et 2 (*Pho Bang*). Cela pourrait s'expliquer par le fait que les deux modèles s'appuient sur les associations sémantiques qui se produisent entre les mots *asian* et *noodle* qui poussent plutôt le POI *Pho Bang*, un autre restaurant asiatique, au rang 2. Le modèle **SGM** n'a pas réussi à obtenir de meilleurs résultats que le modèle DRMM, même en utilisant le facteur spatial, probablement parce que l'association sémantique entre ces mots est pertinente quelle que soit la dimension spatiale. Les modèles DIST et SBERT$_X$ sont quant à eux les meilleures modèles. La bonne performance obtenue par le modèle DIST s'explique simplement par la métrique de la distance : le POI pertinent étant celui qui est le plus proche du tweet. Les meilleurs classement du POI pertinent dans les modèles SBERT$_X$ (rang 1) et BM25 (rang 2) sont principalement liés à la présence des mots *asian*, *lamb* et *noodles*. L'association sémantique contextuelle de ces mots pousse **Biang !**, un restaurant asiatique réputé pour ses nouilles maison, au rang 1 avec le modèle SBERT$_X$, tandis que les statistiques de distribution de ces mots sur les jeux de données le poussent au rang 2 avec le modèle BM25. Dans l'ensemble, il semble que la prise en compte des similarités mot à mot dans l'espace biaise les résultats lorsque ces similarités sont par essence indépendantes de l'espace.

En regardant le tweet positif et les résultats obtenus avec le modèle **SGM** et les modèles de l'état-de-l'art, il semble que les interactions spatiales entre les mots *bay ridge*, *brooklyn* et *burger* ont contribué à remonter **Brooklyn Beet Compagny**, connu pour ses hamburgers à la betterave dans le district de Bay Ridge, et *Vesuvio Pizzeria & restaurant*, aux rangs 1 et 2. Plus précisément, le rang de **Brooklyn Beet Compagny** pourrait s'expliquer par la caractéristique spatiale globale utilisée dans le modèle **SGM** puisque celui-ci est spatialement plus proche du tweet que le *Vesuvio Pizzeria*. Toutes ces remarques permettent également d'expliquer le classement obtenu à l'aide du modèle *DRMM*, où le classement de ces POIs candidats est inversé puisque la distance géographique n'est pas prise en compte. Les raisons de l'échec des modèles *BM25* et SBERT$_X$ sont peut-être différentes. Le modèle *BM25* s'appuie principalement sur la présence des mots *bayridge*, *brooklyn* et *new York* dans le tweet et les POIs candidats, ce qui a conduit à dégrader le classement de **Brooklyn Beet Company** par rapport à d'autres POIs dont les descriptions comprennent d'autres mots ayant de meilleurs caractéristiques d'appariement. La mauvaise performance du modèle SBERT$_X$ pourrait s'expliquer par les spécificités géographiques de la plupart des mots du tweet (*bayridge*, *brooklyn*, *new york*). Le modèle SBERT$_X$ n'a en effet pas permis d'obtenir une représentation contextuelle du tweet qui soit proche de celle construite pour le POI **Brooklyn Beet Company** à partir de sa description qui est plutôt liée sémantiquement aux hamburgers et à la nourriture en général. En examinant le premier POI (*Cream Doughnuts*) retourné parle modèle SBERT$_X$, nous supposons que le modèle a pu mettre en évidence, à partir du jeu de données utilisé pour l'ajustement, la similarité sémantique contex-

tuelle entre les mots *bay ridge* et *donuts* puisque *Cream Doughnuts* est le premier établissement qui propose des donuts dans le quartier de Bay Ridge.

5.4 *Analyse de sensibilité des paramètres de SGM (QR4)*

Nous l'avons vu tout au long de cette évaluation expérimentale, le modèle **SGM** se démarque très nettement des modèles de référence. Ainsi, pour conclure notre évaluation, nous menons une dernière analyse visant à étudier la sensibilité des paramètres du modèle **SGM**. Dans ce qui suit, nous analysons l'impact du nombre de classes utilisées pour construire les histogrammes d'appariement, puis nous étudions l'effet du rayon r, utilisé pour normaliser la distance tweet-POI.

5.4.1 *Impact du nombre de classes*

Comme évoqué dans la Section 5.2.1, les bonnes performances du modèle **SGM** dépendent en partie de la qualité des représentations latentes des interactions locales, et donc de la qualité de l'histogramme d'appariement. Dans cette section, nous avions déterminé que la méthode *LCH* était la plus performante pour construire les représentations latentes. Nous nous attachons maintenant à analyser l'influence du nombre de classes utilisées pour agréger la matrice d'interaction. Pour cela, nous nous focalisons sur le modèle $\mathbf{SGM}_{I_l}$ qui ne considère que les interactions locales, et faisons varier le nombre de classes entre 10 et 600.

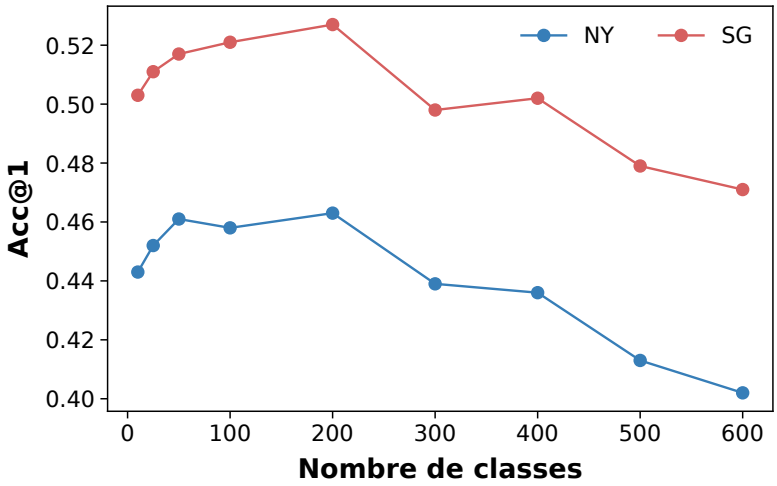


Figure 5.10 – Évolution de l'*Acc*@1 du modèle $\mathbf{SGM}_{I_l}$ en fonction du nombre de classes utilisées pour construire l'histogramme d'appariement.

La Figure 5.10 présente l'évolution de l'*Acc*@1 du modèle $\mathbf{SGM}_{I_l}$ en fonction du nombre de classes utilisées pour agréger l'histogramme d'appariement. Comme nous pouvons le voir, l'*Acc*@1 augmente progressivement au fur et à mesure que le nombre de classes augmente. Toutefois, les meilleures performances sont obtenues lorsque nous considérons 200 classes. Passé ce stade, la qualité du modèle se dégrade sévèrement. Il est clair que considérer un trop grand nombre de classes conduit le modèle à surapprendre les données d'apprentissage, baissant ainsi la qualité de la prédiction. A contrario, si le nombre de classe n'est pas assez élevé, le modèle ne sera pas en mesure de distinguer les différents signaux d'appariement.

5.4.2 *Impact du rayon r*

	NY		SG	
Rayon r	**Valeur**	**%Chg**	**Valeur**	**%Chg**
$r = 100$ *(défaut)*	0,597	-	0,671	-
$r = 50$	0,585	+2,05% *	0,655	+2,44% *
$r = 150$	0,570	+4,74% *	0,649	+3,39% *

Tableau 5.9 – Comparaison des performances du modèle **SGM** pour différentes valeurs de rayon r. La différence significative par rapport à la configuration par défaut ($r = 100$) est déterminée par le test t de Welch ($* : p < 0,01$).

Le deuxième paramètre que nous évaluons est le rayon r, utilisé pour normaliser la distance tweet-POI lors du calcul l'interaction globale spatiale (Section 3.3.1). Nous nous focalisons ici sur le modèle complet **SGM** pour établir la valeur seuil du rayon r. Comme nous pouvons le voir dans le Tableau 5.9, le modèle est très sensible à la valeur r choisie. Une valeur trop faible conduira à sous-estimer la densité des POIs autour du POI candidat p. Au contraire, une valeur trop élevée de r considérera trop d'informations, le modèle ne sera plus en mesure d'utiliser la distance comme facteur discriminant pour faire correspondre un tweet à un POI.

6 Bilan

Nous avons présenté dans ce chapitre notre contribution portant sur l'apprentissage d'un modèle neuronale d'interaction pour résoudre la tâche de prédiction sémantique de l'emplacement. Dans un premier temps, nous avons détaillé l'intérêt de la tâche pour le monde réel, mais aussi les contraintes inhérentes qui complexifient l'appariement d'objets géotextuels. Nous avons ensuite formalisé la tâche d'annotation et mené une analyse préliminaire permettant non seulement

d'illustrer la difficulté de la tâche, mais aussi de montrer l'importance et les limites du facteur spatial pour réaliser un appariement efficace. De là, nous en avons déduit une hypothèse qui a guidé nos travaux de recherche.

Avec tout cela en tête, nous avons proposé un modèle neuronal d'appariement axé sur l'interaction. Contrairement à des architectures neuronales traditionnelles utilisées pour l'appariement de textes, notre modèle capture conjointement des motifs d'appariement exact via des interactions locales, et des motifs d'appariement sémantique via des interactions globales textuelles et spatiales entre un tweet et un POI. Les principales caractéristiques de notre modèle sont les suivantes. Pour calculer les interactions locales mot à mot, nous avons tiré profit des distributions spatiales des mots pour révéler des similarités sémantiques plus ou moins prononcées. Nous avons proposé d'utiliser des histogrammes d'appariement pour agréger les signaux selon leur force. Nous avons complété cet appariement de pertinence avec un appariement sémantique global afin de détecter conjointement des régularités textuelles et spatiales et de les regrouper dans un espace de représentation commun.

Nous avons mené une évaluation expérimentale pour mesurer la qualité de notre modèle, selon plusieurs aspects. Plus précisément, nous avons commencé par évaluer l'impact des différents histogrammes et de chaque interaction (globale et locale) sur les performances des modèles. Les résultats de l'analyse ont montré l'importance du facteur spatial pour corriger les similarités mot à mot lors de la construction des interactions locales. De plus, nous avons vu que la méthode la plus saillante pour les construire est l'histogramme d'appariement. En effet, ce n'est ni la position ni l'ordre d'apparition des motifs d'appariement qui nous intéresse, mais les niveaux de similarités. Enfin, nous avons montré l'importance des interactions globales, et leur utilisation conjointe, pour améliorer les performances des modèles, même si la caractéristique spatiale semble la plus importante. Nous avons ensuite comparé nos deux réseaux de neurones avec des modèles d'appariement de l'état-de-l'art. Les résultats ont montré que nos propositions permettent d'améliorer de manière significative l'efficacité de l'appariement. Nous avons ainsi vu la nécessité de combiner les caractéristiques spatiales et textuelles à différents niveaux, ce que ne font pas les modèles traditionnels. Enfin, nous avons testé la sensibilité des paramètres du modèle neuronal utilisant des histogrammes d'appariement et vu dans quelle mesure le choix du nombre de classes pour le calcul des représentations latentes influence la qualité de la prédiction. Nous avons aussi déterminé l'impact du rayon utilisé pour calculer l'interaction globale spatiale.

Partie III

CONCLUSION

CONCLUSION

Synthèse des contributions

Les travaux présentés dans ce manuscrit s'inscrivent dans le contexte général de la RI, et plus spécifiquement sur l'utilisation d'approches neuronales dans un contexte de RIG.
Nous nous sommes particulièrement intéressés aux approches qui exploitent les représentations distribuées des mots ainsi que les modèles neuronaux pour améliorer les représentations de géotextes et apprendre des caractéristiques d'appariement. Nous avons axé notre état-de-l'art sur ces lignes de recherche et avons consacré deux chapitres à la description des différents modèles proposés dans ce cadre. Plus spécifiquement, le Chapitre 2 s'est focalisé sur les approches traditionnelles pour la prédiction sémantique de l'emplacement à partir de contenus générés dans les réseaux sociaux. Le Chapitre 3 s'est quant à lui consacré à détailler l'utilisation des approches neuronales pour l'apprentissage de représentations distribuées de textes ou d'objets et l'appariement de textes dans des tâches de RI.

Dans ce contexte, nous avons abordé deux questions principales :

1. Dans un premier temps, nous nous sommes intéressés à la représentation d'objets géotextuels par le biais de plongements lexicaux. De précédents travaux ont introduit l'existence de langages sensibles à la localisation, qui sont suivis par une variation des mots et des sujets en fonction des contextes géospatiaux (Backstrom *et al.*, 2008; Han *et al.*, 2012; Laere *et al.*, 2014). Par ailleurs, plusieurs travaux ont montré l'apport des plongements lexicaux pour représenter des mots (Mikolov *et al.*, 2013a,b) ou des phrases (Le et Mikolov, 2014), mais aussi leurs limites lorsqu'il s'agit de représenter les contraintes relationnelles telles que la polysémie (Faruqui *et al.*, 2015; Vulic *et al.*, 2018). Ainsi, nous nous sommes posés la question suivante : *Comment intégrer la sémantique distributionnelle et l'information spatiale pour améliorer la représentation des mots et des objets géotextuels pour des tâches de RIG portées sur l'appariement de géotextes ?*

   Pour répondre à cette première question de recherche, nous avons proposé deux méthodes de régularisation a posteriori de plongements lexicaux qui exploitent les répartitions spatiales des mots pour identifier des relations

sémantiques locales entre les mots ainsi que des sens locaux. L'objectif de ces méthodes est d'améliorer la représentation d'objets géotextuels pour résoudre des tâches de similarités et d'appariement de géotextes. Tout d'abord, nous avons introduit deux méthodes pour détecter les spécificités locales des mots, autrement dit, les différents sens locaux. Notre première méthode s'est appuyée sur une méthode de *clustering*, à l'aide de l'algorithme des k-moyennes. La seconde méthode s'est quant à elle appuyée sur un partitionnement probabiliste à l'aide d'estimations de densités. Une fois les sens locaux déterminés, nous avons corrigé des plongements lexicaux préalablement entraînés, à l'aide d'une fonction de régularisation a posteriori qui intègre les répartitions spatiales des mots. Cette dernière a pour objectif de rapprocher dans l'espace vectoriel les représentations des mots proches spatialement, et d'éloigner les représentations des mots distants spatialement. Nous avons mené une évaluation empirique et une étude expérimentale pour montrer l'efficacité de nos plongements lexicaux spatiaux dans deux tâches de RIG, à savoir la similarité de POIs et la prédiction sémantique de l'emplacement afin d'évaluer respectivement leur effet intrinsèque et extrinsèque. Les résultats de l'analyse ont montré l'intérêt des plongements lexicaux spatiaux pour construire les représentations latentes des géotextes, permettant ainsi de résoudre efficacement les différentes tâches proposées.

2. Dans un deuxième temps, nous avons axé nos travaux de recherche sur l'appariement d'objets géotextuels issus des réseaux sociaux pour résoudre la tâche de prédiction sémantique de l'emplacement. La plupart des approches existantes (Dalvi *et al.*, 2009a,b; Zhao *et al.*, 2016) se sont appuyées sur des modèles de langue ou de traduction ainsi que des réseaux bayésiens pour modéliser les distributions des mots. Cependant, ces approches se sont révélées peu performantes face à la nature bruités des publications issues des RSNs. Par ailleurs, les nouvelles approches d'apprentissage automatique, s'appuyant sur la sémantique distributionnelle (Mikolov *et al.*, 2013a,b), ont montré leur efficacité pour résoudre des tâches de RI (Guo *et al.*, 2016; Mitra *et al.*, 2016). Dès lors, nous nous sommes posés la question suivante : *Comment exploiter la sémantique distributionnelle et l'information spatiale pour améliorer la représentation des mots et des objets géotextuels pour des tâches de RIG portées sur l'appariement de géotextes ?*

Pour répondre à cette deuxième question, nous avons d'abord montré les limites d'un appariement s'appuyant sur une mise en correspondance exacte des termes des géotextes et l'utilisation de l'information spatiale pour résoudre la tâche de prédiction sémantique de l'emplacement. Nous avons ensuite motivé l'utilisation des plongements lexicaux et des architectures neuronales axées sur l'interaction pour capturer et combiner les structures de similarités non linéaires en tenant compte des interactions spatiales et

textuelles. Dans cette contribution, nous avons ainsi proposé un réseau de neurones, permettant de considérer conjointement les interactions d'appariement local, c.-à-d. les signaux de pertinence, et les interactions d'appariement global, c.-à-d. les signaux sémantiques. Tout d'abord, nous avons proposé d'intégrer un facteur d'amortissement spatial sur les interactions entre les mots des tweets et des POIs afin de discriminer les similarités sémantiques des paires de mots en fonction de leur répartition géographique. Ces interactions ont été traitées dans un réseau de neurones à l'aide d'histogrammes d'appariement. Ainsi, nous avons proposé une architecture complète permettant d'apprendre la fonction d'appariement des tweets avec des POIs. Cette dernière permet de considérer conjointement les interactions locales lissées par un facteur spatial et les interactions globales spatiales et textuelles. Nous avons mené une évaluation expérimentale complète et des analyses qualitatives approfondies pour évaluer : *(1)* l'influence des distributions spatiales des mots utilisées comme critère d'amortissement des interactions locale ; *(2)* l'impact individuel et conjoint des interactions locales et globales pour apprendre les signaux d'appariement tweet-POI ; *(3)* l'efficacité de notre modèle par rapport à des modèles de référence ; et *(4)* une analyse de l'impact des paramètres clés des modèles sur la performance globale. Par ces expérimentations, nous avons montré que les interactions locales contribuent à révéler des similarités mot à mot limitées dans l'espace. Nous avons également montré que combiner les interactions locales et globales permet d'obtenir des améliorations significatives par rapport à des modèles de référence.

Perspectives

Nos contributions et expérimentations présentées dans cette thèse peuvent être étendues dans plusieurs directions :

À court terme, nos perspectives portent sur les aspects suivants :

— Nous avons seulement considéré les plongements lexicaux traditionnels proposés par Mikolov *et al.* (2013a,b). Cependant, des travaux de recherche d'information plus récents (MacAvaney *et al.*, 2019; Dai et Callan, 2019) considèrent des plongements lexicaux contextuels, tels que BERT (Devlin *et al.*, 2019) ou ELMo (Peters *et al.*, 2018), où chaque occurrence d'un mot dans le corpus peut être représenté par un vecteur différent. L'avantage des représentations contextuelles est que les différents sens, ou dans notre cas, les spécificités locales d'un mot, ne sont pas représentées par un seul et même

vecteur. Ainsi, les résultats des différents modèles d'appariement proposés dans le Chapitre 5 pourraient être améliorés en utilisant des plongements lexicaux contextuels. Un élément majeur de réflexion dans cette perspective est d'intégrer le contexte de localisation au delà du contexte en lien avec une proximité lexicale. Une piste à envisager est d'étendre le mécanisme d'attention à la prise en compte jointe de la séquence « géographique » des mots et de leur séquence positionnelle dans le texte.

— Dans nos travaux, nous nous sommes concentrés sur les caractéristiques spatiales et textuelles des géotextes comme critères d'appariement. De ce fait, pour les interactions globales, nous avons uniquement considéré une mesure de distance spatiale entre les géotextes et de distance sémantique entre leurs représentations distribuées. De précédents travaux ont montré l'importance du contexte des publications pour l'inférence de l'emplacement, et plus particulièrement du contexte temporel (Mahmud *et al.*, 2014; Li *et al.*, 2011a). En effet, il ressort que l'heure de publication des contenus est un indicateur pertinent de l'emplacement. À titre d'exemple, un club aura tendance à être plus actif la nuit, alors qu'un parc le sera le weekend. Ainsi, nous pensons que tirer profit de l'horodatage des tweets en les intégrant dans le modèle en tant que caractéristique d'appariement global supplémentaire permettrait d'améliorer les performances de notre modèle. Dans ce sens, le calcul des interactions globales peut être étendu à la combinaison des dimensions spatiales et temporelles en comparant par exemple l'heure de publication des tweets et les périodes d'affluence des POIs.

À plus long terme, nos perspectives portent sur les aspects suivants :

— Pour la prédiction sémantique de l'emplacement (Chapitre 5), nous avons supposé que les tweets étaient géolocalisés. Cependant, la littérature reconnaît que les utilisateurs sont réticents à fournir des informations de localisation, ce qui conduit à n'avoir qu'entre 1 et 4% de tweets géolocalisés (Hecht *et al.*, 2011; Ryoo et Moon, 2014). Même si cela représente plus de quatre millions de tweets par jours, une perspective pertinente serait de considérer des tweets géolocalisés ou non. Une amélioration possible consisterait à prédire automatiquement l'emplacement physique des tweets à l'aide des méthodes de l'état-de-l'art (Ajao *et al.*, 2015; Zheng *et al.*, 2018; Chong et Lim, 2018) et à évaluer ensuite l'impact de cette prédiction sur la robustesse et la performance de la prédiction sémantique de l'emplacement. Une autre piste de recherche intéressante consisterait à ordonner conjointement une liste de lieux candidats à associer à l'emplacement physique du tweet et une liste de POIs pertinents pour la prédiction sémantique de l'emplacement. Notre intuition étant que la région de publication d'un tweet délimite spatialement les POIs qu'ils commentent, une approche d'apprentissage multitâche (Zhang et Yang) permettrait de saisir les associations sémantiques tweet-POI.

Plus précisément, en partageant les représentations entre les deux tâches connexes que sont la prédiction de l'emplacement du géotexte et la prédiction sémantique de l'emplacement, nous pouvons permettre à notre modèle de mieux généraliser sur les tweets non géolocalisés.

— Dans le contexte de nos travaux actuels, nous n'avons considéré que le contenu textuel des géotextes. Hors, ces derniers englobent généralement de riches informations visuelles telles que des photos et des vidéos. Intuitivement, la prise en compte d'un contexte supplémentaire aussi riche permettrait d'améliorer l'estimation de la pertinence des POIs. Ainsi, un travail futur intéressant serait d'étendre le travail aux géotextes multimodaux. Cela donnerait lieu à des défis spécifiques dans la phase d'apprentissage. Plus précisément, le simple fait de combiner des distributions spatiales de mots s'appuyant sur la matrice d'interaction et des représentations visuelles apprises séparément affaiblirait l'association entre les modalités. Des perspectives telles que l'apprentissage de bout-en-bout d'un espace intégrant plusieurs modalités en exploitant conjointement les informations visuelles et la distribution spatiale des mots seraient intéressantes.

BIBLIOGRAPHIE

Benjamin Adams, Grant McKenzie et Mark Gahegan : Frankenplace : Interactive thematic mapping for ad hoc exploratory search. *In Proceedings of the 24th International Conference on World Wide Web, WWW 2015*, pages 12–22, 2015.

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu et Quoc V. Le : Towards a human-like open-domain chatbot. *CoRR*, 2020.

Shane Ahern, Mor Naaman, Rahul Nair et Jeannie Hui-I Yang : World explorer : visualizing aggregate data from unstructured text in geo-referenced collections. *In Proceedings of the 7th Joint Conference on Digital Libraries, JCDL 2007*, pages 1–10, 2007.

Amr Ahmed, Liangjie Hong et Alexander J. Smola : Hierarchical geographical modeling of user locations from social media posts. *In Proceedings of the 22nd International World Wide Web Conference, WWW 2013*, pages 25–36, 2013.

Qingyao Ai, Liu Yang, Jiafeng Guo et W. Bruce Croft : Improving language estimation with the paragraph vector model for ad-hoc retrieval. *In Proceedings of the 39th International Conference on Research and Development in Information Retrieval, SIGIR 2016*, pages 869–872, 2016.

Oluwaseun Ajao, Jun Hong et Weiru Liu : A survey of location inference techniques on twitter. *Journal of Information Science*, 41(6):855–864, 2015.

Panos Alexopoulos, Carlos Ruiz et JM Gomez-Perez : Optimizing geographical entity and scope resolution in texts using non-geographical semantic information. *In Proceedings of the 6th International Conference on Advances in Semantic Processing, SEMAPRO 2012*, pages 65–70, 2012.

Saad Aloteibi et Mark Sanderson : Analyzing geographic query reformulation : An exploratory study. *Journal of the Association for Information Science and Technology*, 65(1):13–24, 2014.

Einat Amitay, Nadav Har'El, Ron Sivan et Aya Soffer : Web-a-where : geotagging web content. *In Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval, SIGIR 2004*, pages 273–280, 2004.

Geoffrey Andogah, Gosse Bouma et John Nerbonne : Every document has a geographical scope. volume 81-82, pages 1–20, 2012.

Leonardo Andrade et Mário J. Silva : Relevance ranking for geographic IR. *In Proceedings of the 3rd Workshop On Geographic Information Retrieval, GIR 2006*, 2006.

Gopala K Anumanchipalli, Josh Chartier et Edward F Chang : Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498, 2019.

Sanjeev Arora, Yingyu Liang et Tengyu Ma : A simple but tough-to-beat baseline for sentence embeddings. *In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*, pages 1–16, 2017.

Lars Backstrom, Jon M. Kleinberg, Ravi Kumar et Jasmine Novak : Spatial variation in search engine queries. *In Proceedings of the 17th International Conference on World Wide Web, WWW 2008*, pages 357–366, 2008.

Ricardo Baeza-Yates et Berthier Ribeiro-Neto : Modern information retrieval. 1999.

Dzmitry Bahdanau, Kyunghyun Cho et Yoshua Bengio : Neural machine translation by jointly learning to align and translate. *In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, pages 1–15, 2015.

Jordan Bakerman, Karl Pazdernik, Alyson G. Wilson, Geoffrey Fairchild et Rian Bahran : Twitter geolocation : A hybrid approach. *ACM Transactions on Knowledge Discovery from Data*, 12(3):1–17, 2018.

Marco Baroni, Georgiana Dinu et Germán Kruszewski : Don't count, predict ! A systematic comparison of context-counting vs. context-predicting semantic vectors. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pages 238–247, 2014.

Petr Baudis et Jan Sedivý : Sentence pair scoring : Towards unified framework for text comprehension. *CoRR*, 2016.

Rudolf Bayer : The universal b-tree for multidimensional indexing : general concepts. *In Proceedings of the Worldwide Computing and Its Applications, International Conference, WWCA 1997*, pages 198–209, 1997.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent et Christian Janvin : A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

Tanusri Bhattacharya, Lars Kulik et James Bailey : Automatically recognizing places of interest from unreliable GPS data using spatio-temporal density estimation and line intersections. *Pervasive and Mobile Computing*, 19:86–107, 2015.

Christopher M Bishop *et al.* : *Neural networks for pattern recognition*. Oxford university press, 1995.

Piotr Bojanowski, Edouard Grave, Armand Joulin et Tomas Mikolov : Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Cécile Bothorel, Neal Lathia, Romain Picot-Clémente et Anastasios Noulas : Location recommendation with social media data. *In* Peter Brusilovsky et Daqing He, éditeurs : *Social Information Access - Systems and Technologies*, volume 10100, pages 624–653. 2018.

Nieves R. Brisaboa, Miguel Rodríguez Luaces, Ángeles Saavedra Places et Diego Seco : Exploiting geographic references of documents in a geographical information retrieval system using an ontology-based index. *GeoInformatica*, 14 (3):307–331, 2010.

Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souza, Jennifer C. Lai et Robert L. Mercer : Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

Travis Brown, Jason Baldridge, Maria Esteva et Weijia Xu : The substantial words are in the ground and sea : computationally linking text and geography. *Texas Studies in Literature and Language*, 54(3):324–339, 2012.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton et Gregory N. Hullender : Learning to rank using gradient descent. *In Proceedings of the 22nd International Conference on Machine Learning, ICML 2005*, pages 89–96, 2005.

Guoray Cai : Geovsm : An integrated retrieval model for geographic information. *In Proceedings of the 2nd Geographic Information Science, GIS 2002*, pages 65–79, 2002.

Guoray Cai : Relevance ranking in geographical information retrieval. *SIGSPATIAL Special*, 3(2):33–36, 2011.

M. Emre Celebi, Hassan A. Kingravi et Patricio A. Vela : A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1):200–210, 2013.

Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim et Jaewoo Kang : Content-aware hierarchical point-of-interest embedding model for successive POI recommendation. *In Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 3301–3307, 2018.

Lisi Chen, Gao Cong, Christian S. Jensen et Dingming Wu : Spatial keyword query processing : An experimental evaluation. *Proceedings of the VLDB Endowment*, 6(3):217–228, 2013.

Peixin Chen, Wu Guo, Zhi Chen, Jian Sun et Lanhua You : Gated convolutional neural network for sentence matching. *In Proceedings of the 19th Annual Conference of the International Speech Communication Association, ISCA 2018*, pages 2853–2857, 2018.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang et Diana Inkpen : Enhanced LSTM for natural language inference. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 1657–1668, 2017.

Yen-Yu Chen, Torsten Suel et Alexander Markowetz : Efficient query processing in geographic web search engines. *In Proceedings of the 32th International Conference on Management of Data, SIGMOD 2006*, pages 277–288, 2006.

Jianpeng Cheng, Zhongyuan Wang, Ji-Rong Wen, Jun Yan et Zheng Chen : Contextual text understanding in distributional semantic space. *In Proceedings of the 24th International Conference on Informationand Knowledge Management, CIKM 2015*, pages 133–142, 2015.

Zhiyuan Cheng, James Caverlee et Kyumin Lee : You are where you tweet : a content-based approach to geo-locating twitter users. *In Proceedings of the 19th Conference on Information and Knowledge Management, CIKM 2010*, pages 759–768, 2010.

France Cheong et Christopher Cheong : Social media data mining : A social network analysis of tweets during the 2010-2011 australian floods. *In Proceedings of the 15th Pacific Asia Conference on Information Systems, PACIS 2011*, page 46, 2011.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk et Yoshua Bengio : Learning phrase representations using RNN encoder-decoder for statistical machine translation. *In Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1724–1734, 2014.

François Chollet *et al.* : Keras. https://keras.io.

Wen-Haw Chong et Ee-Peng Lim : Exploiting contextual information for fine-grained tweet geolocation. *In Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017*, pages 488–491, 2017a.

Wen-Haw Chong et Ee-Peng Lim : Tweet geolocation : Leveraging location, user and peer signals. *In Proceedings of the Conference on Information and Knowledge Management, CIKM 2017*, pages 1279–1288, 2017b.

Wen-Haw Chong et Ee-Peng Lim : Exploiting user and venue characteristics for fine-grained tweet geolocation. *Transactions on Information Systems*, 36(3):26 :1–26 :34, 2018.

Maria Christoforaki, Jinru He, Constantinos Dimopoulos, Alexander Markowetz et Torsten Suel : Text vs. space : efficient geo-search query processing. *In Proceedings of the 20th Conference on Information and Knowledge Management, CIKM 2011*, pages 423–432, 2011.

Dan C. Ciresan, Ueli Meier et Jürgen Schmidhuber : Multi-column deep neural networks for image classification. *In Proceedings of the 22th Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pages 3642–3649, 2012.

Paul Clough, Mark Sanderson et Hideo Joho : Extraction of semantic annotations from textual web pages. *Technical Report D15*, 6201, 2004.

Anne Cocos et Chris Callison-Burch : The language of place : Semantic value from geospatial context. *In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017*, pages 99–104, 2017.

Daniel Cohen, Qingyao Ai et W. Bruce Croft : Adaptability of neural networks on varying granularity IR tasks. *CoRR*, 2016.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu et Pavel P. Kuksa : Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

Gao Cong, Christian S. Jensen et Dingming Wu : Efficient retrieval of the top-k most relevant spatial web objects. *The VLDB Journal*, 2(1):337–348, 2009.

Alexis Conneau et Douwe Kiela : Senteval : An evaluation toolkit for universal sentence representations. *In Proceedings of the 11th International Conference on Language Resources and Evaluation, LREC 2018*, pages 1–6, 2018.

Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek et Amine Benhalloum : Trans-gram, fast cross-lingual word-embeddings. *In Proceedings of the 20th Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1109–1113, 2015.

Nick Craswell : Mean reciprocal rank. *In Encyclopedia of Database Systems*, page 1703. 2009.

W. Bruce Croft, Donald Metzler et Trevor Strohman : *Search Engines - Information Retrieval in Practice*. Pearson Education, 2009.

Zhuyun Dai et Jamie Callan : Deeper text understanding for IR with contextual neural language modeling. *In Proceedings of the 42nd International Conference on Research and Development in Information Retrieval, SIGIR 2019*, pages 985–988, 2019.

Zhuyun Dai, Chenyan Xiong, Jamie Callan et Zhiyuan Liu : Convolutional neural networks for soft-matching n-grams in ad-hoc search. *In Proceedings of the 11th International Conference on Web Search and Data Mining, WSDM 2018*, pages 126–134, 2018.

Nilesh Dalvi, Ravi Kumar et Bo Pang : Object matching in tweets with spatial models. *In Proceedings of the 5th International Conference on Web Search and Web Data Mining, WSDM 2012*, pages 43–52, 2012.

Nilesh Dalvi, Ravi Kumar, Bo Pang et Andrew Tomkins : Matching reviews to objects using a language model. *In Proceedings of the 14th Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*, pages 609–618, 2009a.

Nilesh Dalvi, Ravi Kumar, Bo Pang et Andrew Tomkins : A translation model for matching reviews to objects. *In Proceedings of the 18th Conference on Information and Knowledge Management, CIKM 2009*, pages 167–176, 2009b.

Yann N. Dauphin, Razvan Pascanu, Çaglar Gülçehre, KyungHyun Cho, Surya Ganguli et Yoshua Bengio : Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *In Proceedings of the 27th Conference on Neural Information Processing Systems, NIPS 2014*, pages 2933–2941, 2014.

Michael De Smith et Michael F Goodchild : *Geospatial analysis : a comprehensive guide to principles, techniques and software tools*. Troubador Publishing Ltd., 2007.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas et Richard A. Harshman : Indexing by latent semantic analysis. *Journal of the Association for Information Science and Technology*, 41(6):391–407, 1990.

Tiago M. Delboni, Karla A. V. Borges, Alberto H. F. Laender et Clodoveu A. Davis Jr. : Semantic expansion of geographic web queries based on natural language positioning expressions. *Transactions in GIS*, 11(3):377–397, 2007.

Li Deng et Dong Yu : Deep learning : Methods and applications. *Foundations and Trends in Signal Processing*, 7(3-4):197–387, 2014.

John S. Denker, W. R. Gardner, Hans Peter Graf, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, Lawrence D. Jackel, Henry S. Baird et Isabelle Guyon : Neural network recognizer for hand-written zip code digits. *In Proceedings of the 1st Conference on Neural Information Processing Systems, NIPS 1988*, pages 323–331, 1988.

Romain Deveaud, M-Dyaa Albakour, Craig Macdonald et Iadh Ounis : Experiments with a venue-centric model for personalisedand time-aware venue suggestion. *In Proceedings of the 24th International Conference on Information and Knowledge Management, CIKM 2015*, pages 53–62, 2015.

Jacob Devlin, Ming-Wei Chang, Kenton Lee et Kristina Toutanova : 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, NAACL-HLT 2019*, pages 4171–4186, 2019.

Sander Dieleman et Benjamin Schrauwen : End-to-end learning for music audio. *In Proceedings of the 39th International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014*, pages 6964–6968, 2014.

Junyan Ding, Luis Gravano et Narayanan Shivakumar : Computing geographical scopes of web resources. *In Proceedings of 26th International Conference on Very Large Data Bases, VLDB 2000*, pages 545–556, 2000.

Mark Dredze, Miles Osborne et Prabhanjan Kambadur : Geolocation for twitter : Timing matters. *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2016*, pages 1064–1069, 2016.

John C. Duchi, Elad Hazan et Yoram Singer : Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learnaring Research*, 12:2121–2159, 2011.

Jacob Eisenstein, Brendan O'Connor, Noah A. Smith et Eric P. Xing : A latent variable model for geographic lexical variation. *In Proceedings of the 15th Conference on Empirical Methods in Natural Language Processing, EMNLP 2010*, pages 1277–1287, 2010.

Yixing Fan, Jiafeng Guo, Yanyan Lan, Jun Xu, Chengxiang Zhai et Xueqi Cheng : Modeling diverse relevance patterns in ad-hoc retrieval. *In The 41st International Conference on Research & Development in Information Retrieval, SIGIR 2018*, pages 375–384, 2018.

Yuan Fang et Ming-Wei Chang : Entity linking on microblogs with spatial and temporal signals. *Transactions of the Association for Computational Linguistics*, 2: 259–272, 2014.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy et Noah A. Smith : Retrofitting word vectors to semantic lexicons. *In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2015*, pages 1606–1615, 2015.

Shanshan Feng, Gao Cong, Bo An et Yeow Meng Chee : Poi2vec : Geographical latent representation for predicting future visitors. *In Proceedings of the 31st Conference on Artificial Intelligence, AAAI 2017*, pages 102–108, 2017.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman et Eytan Ruppin : Placing search in context : the concept revisited. *In Proceedings of the 10th International World Wide Web Conference, WWW 2001*, pages 406–414, 2001.

David Flatow, Mor Naaman, Ke Eddie Xie, Yana Volkovich et Yaron Kanza : On the accuracy of hyper-local geotagging of social media content. *In Proceedings of the 8th International Conference on Web Search and Data Mining, WSDM 2015*, pages 127–136, 2015.

Patricia Frontiera, Ray R. Larson et John Radke : A comparison of geometric approaches to assessing spatial similarity for GIR. *International Journal of Geographical Information Science*, 22(3):337–360, 2008.

Qingqing Gan, Josh Attenberg, Alexander Markowetz et Torsten Suel : Analysis of geographic queries in a search engine log. *In Proceedings of the 1st International Workshop on Location and the Web, LocWeb 2008*, volume 300, pages 49–56, 2008.

Judith Gelernter et Nikolai Mushegian : Geo-parsing messages from microtext. *Transactions in GIS*, 15(6):753–773, 2011.

Michael F. Goodchild : Twenty years of progress : Giscience in 2010. *Journal of Spatial Information Science*, 1(1):3–20, 2010.

Mark Graham, Scott A Hale et Devin Gaffney : Where in the world are you ? geolocation and language identification in twitter. *The Professional Geographer*, 66 (4):568–578, 2014.

Alex Graves, Abdel-rahman Mohamed et Geoffrey E. Hinton : Speech recognition with deep recurrent neural networks. *In Proceedings of the 38th International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013*, pages 6645–6649, 2013.

Edward Grefenstette, Phil Blunsom, Nando de Freitas et Karl Moritz Hermann : A deep architecture for semantic parsing. *In Proceedings of the 52th Workshop on Semantic Parsing, ACL 2014*, pages 1–6, 2014.

Jiafeng Guo, Yixing Fan, Qingyao Ai et W. Bruce Croft : A deep relevance matching model for ad-hoc retrieval. *In Proceedings of the 25th International Conference on Information and Knowledge Management, CIKM 2016*, pages 55–64, 2016.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft et Xueqi Cheng : A deep look into neural ranking models for information retrieval. *CoRR*, 2019.

Stephen Guo, Ming-Wei Chang et Emre Kiciman : To link or not to link ? A study on end-to-end tweet entity linking. *In Proceedings of the 2013 Human Language Technologies : Conference of the North American Chapter of the Association of Computational Linguistics, HLT-NAACL 2013*, pages 1020–1030, 2013.

Antonin Guttman : R-trees : A dynamic index structure for spatial searching. *In* Beatrice Yormark, éditeur : *Proceedings of the 1984 International Conference on Management of Data, SIGMOD 1984*, pages 47–57, 1984.

Nur Al Hasan Haldar, Jianxin Li, Mark Reynolds, Timos Sellis et Jeffrey Xu Yu : Location prediction in large-scale social networks : an in-depth benchmarking study. *The International Journal on Very Large Data Bases*, 28(5):623–648, 2019.

Bo Han, Paul Cook et Timothy Baldwin : Geolocation prediction in social media data by finding location indicative words. *In Proceedings of the 24th International Conference on Computational Linguistics, COLING 2012*, pages 1045–1062, 2012.

J. Han, A. Sun, G. Cong, W. X. Zhao, Z. Ji et M. C. Phan : Linking fine-grained locations in user comments. *IEEE Transactions on Knowledge and Data Engineering*, 30(1):59–72, 2018.

Mengyue Hang, Ian Pytlarz et Jennifer Neville : Exploring student check-in behavior for improved point-of-interest prediction. *In Proceedings of the 24th International Conference on Knowledge Discovery & Data Mining, KDD 2018*, pages 321–330, 2018.

Pei-Yi Hao, Weng-Hang Cheang et Jung-Hsien Chiang : Real-time event embedding for POI recommendation. *Neurocomputing*, 349:1–11, 2019.

Brent J. Hecht, Lichan Hong, Bongwon Suh et Ed H. Chi : Tweets from justin bieber's heart : the dynamics of the location field in user profiles. *In Proceedings of the 29th International Conference on Human Factors in Computing Systems, CHI 2011*, pages 237–246, 2011.

Djoerd Hiemstra et Wessel Kraaij : Twenty-one at trec-7 : Ad-hoc and cross-language track. *Nist Special Publication*, pages 227–238, 1999.

Felix Hill, Kyunghyun Cho et Anna Korhonen : Learning distributed representations of sentences from unlabelled data. *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT 2016*, pages 1367–1377, 2016.

Linda L. Hill : *Georeferencing - The Geographic Associations of Information*. Digital libraries and electronic publishing. MIT Press, 2009.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever et Ruslan Salakhutdinov : Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, 2012.

Thi Bich Ngoc Hoang et Josiane Mothe : Location extraction from tweets. *Information Processing and Management*, 54(2):129–144, 2018.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber *et al.* : Gradient flow in recurrent nets : the difficulty of learning long-term dependencies. 2001.

Sepp Hochreiter et Jürgen Schmidhuber : Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater et Gerhard Weikum : Robust disambiguation of named entities in text. *In Proceedings of the 16th Conference on Empirical Methods in Natural Language Processing, EMNLP 2011*, pages 782–792, 2011.

Shenda Hong, Meng Wu, Hongyan Li et Zhengwu Wu : Event2vec : Learning representations of events on temporal sequences. *In Proceedings of the 1st International Joint Conference, APWeb-WAIM 2017*, pages 33–47, 2017.

Baotian Hu, Zhengdong Lu, Hang Li et Qingcai Chen : Convolutional neural network architectures for matching natural language sentences. *In Proceedings of the 28th Conference on Neural Information Processing Systems, NIPS 2014*, pages 2042–2050, 2014.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero et Larry Heck : Learning deep structured semantic models for web search using clickthrough data. *In Proceedings of the 22nd International Conference on Information and Knowledge Management, CIKM 2013*, pages 2333–2338, 2013.

Ignacio Iacobacci, Mohammad Taher Pilehvar et Roberto Navigli : Sensembed : Learning sense embeddings for word and relational similarity. *In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, pages 95–105, 2015.

Ignacio IACOBACCI, Mohammad Taher PILEHVAR et Roberto NAVIGLI : Embeddings for word sense disambiguation : An evaluation study. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, pages 897—-907, 2016.

Muhammad IMRAN, Carlos CASTILLO, Fernando DIAZ et Sarah VIEWEG : Processing social media messages in mass emergency : A survey. *ACM Computing Surveys*, 47(4):1–67, 2015.

Sergey IOFFE et Christian SZEGEDY : Batch normalization : Accelerating deep network training by reducing internal covariate shift. *In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pages 448–456, 2015.

Shuiwang JI, Wei XU, Ming YANG et Kai YU : 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

Zongcheng JI, Aixin SUN, Gao CONG et Jialong HAN : Joint recognition and linking of fine-grained locations from tweets. *In Proceedings of the 25th International Conference on World Wide Web, WWW 2016*, pages 1271–1281, 2016.

Christopher B. JONES et Ross S. PURVES : Geographical information retrieval. *In Encyclopedia of Database Systems*, pages 1227–1231. 2009.

Armand JOULIN, Edouard GRAVE, Piotr BOJANOWSKI et Tomas MIKOLOV : Bag of tricks for efficient text classification. *In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017*, pages 427—-431, 2017.

Nal KALCHBRENNER et Phil BLUNSOM : Recurrent continuous translation models. *In Proceedings of the 18th Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pages 1700–1709, 2013.

Nal KALCHBRENNER, Edward GREFENSTETTE et Phil BLUNSOM : A convolutional neural network for modelling sentences. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pages 655–665, 2014.

Sanjay KAMATH, Brigitte GRAU et Yue MA : Predicting and integrating expected answer types into a simple recurrent neural network model for answer sentence selection. *Computación y Sistemas*, 23(3), 2019.

Anne KAO et Steve R POTEET : *Natural language processing and text mining*. Springer Science & Business Media, 2007.

Andrej KARPATHY, George TODERICI, Sanketh SHETTY, Thomas LEUNG, Rahul SUKTHANKAR et Fei-Fei LI : Large-scale video classification with convolutional neural networks. *In Proceedings of the 24th Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pages 1725–1732, 2014.

Tom Kenter, Alexey Borisov et Maarten de Rijke : Siamese CBOW : optimizing word embeddings for sentence representations. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, 2016.

Seonhoon Kim, Inho Kang et Nojun Kwak : Semantic sentence matching with densely-connected recurrent and co-attentive information. *In Proceddings of the 33rd Conference on Artificial Intelligence, AAAI 2019*, pages 6586–6593, 2019.

Yoon Kim : Convolutional neural networks for sentence classification. *In Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1746–1751, 2014.

Diederik P. Kingma et Jimmy Ba : Adam : A method for stochastic optimization. *In Proceedins of the 3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Sheila Kinsella, Vanessa Murdock et Neil O'Hare : "i'm eating a sandwich in glasgow" : modeling locations with tweets. *In Proceedings of the 3rd International CIKM Workshop on Search and Mining User-Generated Contents, SMUC 2011*, pages 61–68, 2011.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba et Sanja Fidler : Skip-thought vectors. *In Proceedings of the 28th Annual Conference on Neural Information Processing Systems, NIPS 2015*, pages 3294–3302, 2015.

Alex Krizhevsky, Ilya Sutskever et Geoffrey E. Hinton : Imagenet classification with deep convolutional neural networks. *In Proceedings of the 26th Annual Conference on Neural Information Processing Systems, NIPS 2012*, pages 1106–1114, 2012.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan et Soumen Chakrabarti : Collective annotation of wikipedia entities in web text. *In Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining, SIGKDD 2009*, pages 457–466, 2009.

Abhinav Kumar et Jyoti Prakash Singh : Location reference identification from tweets during emergencies : A deep learning approach. *International Journal of Disaster Risk Reduction*, 33:365 – 375, 2019.

Shamanth Kumar, Geoffrey Barbier, Mohammad Ali Abbasi et Huan Liu : Tweettracker : An analysis tool for humanitarian and disaster relief. *In Proceedings of the 5th International Conference on Weblogs and Social Media, ICWSM 2011*, pages 1–2, 2011.

Haewoon Kwak, Changhyun Lee, Hosung Park et Sue B. Moon : What is twitter, a social network or a news media ? *In Proceedings of the 19th International Conference on World Wide Web, WWW 2010*, pages 591–600, 2010.

Olivier Van Laere, Jonathan A. Quinn, Steven Schockaert et Bart Dhoedt : Spatially aware term selection for geotagging. *IEEE Transactions on Knowledge and Data Engineering*, 26(1):221–234, 2014.

John D. Lafferty, Andrew McCallum et Fernando C. N. Pereira : Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *In Proceedings of the 18th International Conference on Machine Learning ICML 2001*, pages 282–289, 2001.

Siwei Lai, Liheng Xu, Kang Liu et Jun Zhao : Recurrent convolutional neural networks for text classification. *In Proceedings of the 29th Conference on Artificial Intelligence, AAAI 2015*, pages 2267–2273, 2015.

Ray R Larson : Geographic information retrieval and spatial browsing. *Geographic information systems and libraries : patrons, maps, and spatial information*, pages 81–124, 1996.

Ray R. Larson : Ranking approaches for GIR. *SIGSPATIAL Special*, 3(2):37–41, 2011.

Ray R. Larson et Patricia Frontiera : Spatial ranking methods for geographic information retrieval (GIR) in digital libraries. *In Proceedings og the 8th European Conference on Research and Advanced Technology for Digital Librarie, ECDL 2004*, pages 45–56, 2004.

Steve Lawrence, C. Lee Giles, Ah Chung Tsoi et Andrew D. Back : Face recognition : a convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

Quoc V. Le et Tomas Mikolov : Distributed representations of sentences and documents. *In Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, pages 1188–1196, 2014.

Claudia Leacock et Martin Chodorow : *Combining Local Context and WordNet Similarity for Word Sense Identification*, volume 49, pages 265–283. 01 1998.

Rémi Lebret et Ronan Collobert : Word embeddings through hellinger PCA. *In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014*, pages 482–490, 2014.

Yann LeCun, Yoshua Bengio *et al.* : Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10):1995, 1995.

Yann LeCun, Yoshua Bengio et Geoffrey Hinton : Deep learning. *Nature*, 521 (7553):436–444, 2015.

Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard et Lawrence D. Jackel : Handwritten digit recognition with a back-propagation network. *In Proceedings of the 2nd Conference on Neural Information Processing Systems, NIPS 1989*, pages 396–404, 1989.

Yann LeCun, Léon Bottou, Yoshua Bengio et Patrick Haffner : Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Kisung Lee, Raghu K. Ganti, Mudhakar Srivatsa et Ling Liu : When twitter meets foursquare : tweet location prediction using foursquare. *In Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems : Computing, Networking and Services, MOBIQUITOUS 2014*, pages 198–207, 2014.

Jochen L. Leidner : An evaluation dataset for the toponym resolution task. *Computers, Environment and Urban Systems*, 30(4):400–417, 2006.

Jochen L. Leidner et Michael D. Lieberman : Detecting geographical references in the form of place names and associated spatial natural language. *SIGSPATIAL Special*, 3(2):5–11, 2011.

Omer Levy et Yoav Goldberg : Dependency-based word embeddings. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pages 302–308, 2014.

Chenliang Li et Aixin Sun : Fine-grained location extraction from tweets with temporal awareness. *In The 37th International Conference on Research and Development in Information Retrieval, SIGIR 2014*, pages 43–52, 2014.

Chenliang Li et Aixin Sun : Extracting fine-grained location with temporal awareness in tweets : A two-stage approach. *Journal of the Association for Information Science and Technology*, 68(7):1652–1670, 2017.

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun et Bu-Sung Lee : Twiner : named entity recognition in targeted twitter stream. *In Proceedings of the 35th International Conference on Research and Development in Information Retrieval, SIGIR 2012*, pages 721–730, 2012.

Hang Li et Zhengdong Lu : Deep learning for information retrieval. *In Proceedings of the 39th International Conference on Research and Development in Information Retrieval, SIGIR 2016*, pages 1203–1206, 2016.

Wen Li, Pavel Serdyukov, Arjen P. de Vries, Carsten Eickhoff et Martha A. Larson : The where in the tweet. *In Proceedings of the 20th Conference on Information and Knowledge Management, CIKM 2011*, pages 2473–2476, 2011a.

Xuefei Li, HongYun Cai, Zi Huang, Yang Yang et Xiaofang Zhou : Social event identification and ranking on flickr. *World Wide Web*, 18(5):1219–1245, 2015.

Zhisheng Li, Ken C. K. Lee, Baihua Zheng, Wang-Chien Lee, Dik Lun Lee et Xufa Wang : Ir-tree : An efficient index for geographic document search. *Transactions on Knowledge and Data Engineering*, 23(4):585–599, 2011b.

Michael D. Lieberman, Hanan Samet, Jagan Sankaranarayanan et Jon Sperling : STEWARD : architecture of a spatio-textual search engine. *In Proceedings of the 15th International Symposium on Geographic Information Systems, GIS 2007*, page 25, 2007.

John Lingad, Sarvnaz Karimi et Jie Yin : Location extraction from disaster-related microblogs. *In Proceedings of the 22nd International World Wide Web Conference, WWW 2013*, pages 1017–1020, 2013.

Kun-Lin Liu, Wu-Jun Li et Minyi Guo : Emoticon smoothed language models for twitter sentiment analysis. *In Proceedings of the 26th Conference on Artificial Intelligence, AAAI 2012*, pages 1678–1684, 2012.

Tie-Yan Liu : Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

Xiaohua Liu, Furu Wei, Shaodian Zhang et Ming Zhou : Named entity recognition for tweets. *ACM Transactions on Intelligent Systems and Technology*, 4(1):3 :1–3 :15, 2013.

Xiaohua Liu, Shaodian Zhang, Furu Wei et Ming Zhou : Recognizing named entities in tweets. *In Proceedinfs of the 49th Annual Meeting of the Association for Computational Linguistics, ACL 2011*, pages 359–367, 2011.

Zhi Liu, Yan Huang et Joshua R. Trampier : LEDS : local event discovery and summarization from tweets. *In Proceedings of the 24th International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2016*, pages 53 :1–53 :4, 2016.

Lajanugen Logeswaran et Honglak Lee : An efficient framework for learning sentence representations. *In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, pages 1–16, 2018.

Paul A Longley, Michael F Goodchild, David J Maguire et David W Rhind : *Geographic information science and systems*. John Wiley & Sons, 2015.

Zhengdong Lu et Hang Li : A deep architecture for matching short texts. *In Proceedings of the 27th Conference on Neural Information Processing Systems, NIPS 2013*, pages 1367–1375, 2013.

Kevin Lund et Curt Burgess : Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*, 28(2): 203–208, 1996.

Sean MacAvaney, Andrew Yates, Arman Cohan et Nazli Goharian : CEDR : contextualized embeddings for document ranking. *In Proceedings of the 42nd International Conference on Research and Development in Information Retrieval, SIGIR 2019*, pages 1101–1104, 2019.

James MacQueen *et al.* : Some methods for classification and analysis of multivariate observations. *In Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, BSMSP 1967*, pages 281–297, 1967.

Amr Magdy, Mohamed F. Mokbel, Sameh Elnikety, Suman Nath et Yuxiong He : Mercury : A memory-constrained spatio-temporal real-time search on microblogs. *In proceedings of the 30th International Conference on Data Engineering, ICDE 2014*, pages 172–183, 2014.

Jalal Mahmud, Jeffrey Nichols et Clemens Drews : Home location identification of twitter users. *ACM Transactions on Intelligent Systems and Technology*, 5(3):47 :1–47 :21, 2014.

Deepa Mallela, Dirk Ahlers et Maria Soledad Pera : Mining twitter features for event summarization and rating. *In Proceedings of the 15th International Conference on Web Intelligence, WI 2017*, pages 615–622, 2017.

Shervin Malmasi et Mark Dras : Location mention detection in tweets and microblogs. *In Proceedings of the 14th International Conference of the Pacific Association for Computaitonal Linguistics, PACLING 2015*, volume 593, pages 123–134, 2015.

Massimiliano Mancini, José Camacho-Collados, Ignacio Iacobacci et Roberto Navigli : Embedding words and senses together via joint knowledge-enhanced training. *In Proceedings of the 21st Conference on Computational Natural Language Learning, CoNLL 2017*, pages 100–111, 2017.

Thomas Mandl : Evaluating GIR : geography-oriented or user-oriented ? *SIGSPATIAL Special*, 3(2):42–45, 2011.

Christopher Manning, Prabhakar Raghavan et Hinrich Schütze : Introduction to information retrieval. *Computational Linguistics*, 35(2):307–309, 2009.

Bruno Martins, Mário J. Silva et Marcirio Silveira Chaves : Challenges and resources for evaluating geographical IR. *In Proceedings of the 2nd Workshop On Geographic Information Retrieval, GIR 2005*, pages 65–69, 2005.

Warren S McCulloch et Walter Pitts : A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

Kevin S. McCurley : Geospatial mapping and navigation of the web. *In Proceedings of the 10th International World Wide Web Conference, WWW 2001*, pages 221–229, 2001.

Tomas Mikolov, Kai Chen, Greg Corrado et Jeffrey Dean : Efficient estimation of word representations in vector space. *In Proceedings of the 1st International Conference on Learning Representations, ICLR 2013*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado et Jeff Dean : Distributed representations of words and phrases and their compositionality. *In Proceedings of the 27th Annual Conference on Neural Information Processing Systems, NIPS 2013*, pages 3111–3119, 2013b.

David N. Milne et Ian H. Witten : Learning to link with wikipedia. *In Proceedings of the 17th Conference on Information and Knowledge Management, CIKM 2008*, pages 509–518, 2008.

Weiqing Min, Bing-Kun Bao, Changsheng Xu et M. Shamim Hossain : Cross-platform multi-modal topic modeling for personalized inter-platform recommendation. *IEEE Transactions on Multimedia*, 17(10):1787–1801, 2015.

Bhaskar Mitra et Nick Craswell : An introduction to neural information retrieval. *Foundations and Trends in Information Retrieval*, 13(1):1–126, 2018.

Bhaskar Mitra, Fernando Diaz et Nick Craswell : Learning to match using local and distributed representations of text for web search. *In Proceedings of the 26th International Conference on World Wide Web, WWW 2017*, pages 1291–1299, 2017.

Bhaskar Mitra, Eric T. Nalisnick, Nick Craswell et Rich Caruana : A dual embedding space model for document ranking. *CoRR*, 2016.

Jose G. Moreno, Romaric Besançon, Romain Beaumont, Eva D'hondt, Anne-Laure Ligozat, Sophie Rosset, Xavier Tannier et Brigitte Grau : Combining word and entity embeddings for entity linking. *In Proceedings of the 14th Extended Semantic Web Conference, ESWC 2017*, volume 10249, pages 337–352, 2017.

Andrea Moro, Alessandro Raganato et Roberto Navigli : Entity linking meets word sense disambiguation : a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.

Igor Mozetic, Luís Torgo, Vítor Cerqueira et Jasmina Smailovic : How to evaluate sentiment classifiers for twitter time-ordered data ? *Plos One*, 13(3):1–20, 03 2018.

Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen et Steve J. Young : Counter-fitting word vectors to linguistic constraints. *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2016*, pages 142–148, 2016.

Nikola Mrksic, Ivan Vulic, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gasic, Anna Korhonen et Steve J. Young : Semantic specialisation of distributional word vector spaces using monolingual and cross-lingual constraints. *CoRR*, 2017.

Eric T. Nalisnick, Bhaskar Mitra, Nick Craswell et Rich Caruana : Improving document ranking with dual word embeddings. *In Proceedings of the 25th International Conference on World Wide Web, WWW 2016*, pages 83–84, 2016.

Ramesh Nallapati, Feifei Zhai et Bowen Zhou : Summarunner : A recurrent neural network based sequence model for extractive summarization of documents. *In Proceedings of the 31st Conference on Artificial Intelligence, AAAI 2017*, pages 3075–3081, 2017.

Navy : *Admiralty manual of navigation*, volume 1. 1987.

Andrew Y Ng : Feature selection, l 1 vs. l 2 regularization, and rotational invariance. *In Proceedings of the 21st International Conference on Machine Learning, ICML 2004*, page 78, 2004.

Gia-Hung Nguyen : *Modèles neuronaux pour la recherche d'information : approches dirigées par les ressources sémantiques*. Thèse de doctorat, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2018.

Gia-Hung Nguyen, Lynda Tamine, Laure Soulier et Nathalie Souf : A tri-partite neural document language model for semantic information retrieval. *In Proceedings of the 15th Extended Semantic Web Conference, ESWC 2018*, pages 445–461, 2018.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde et Ngoc Thang Vu : Hierarchical embeddings for hypernymy detection and directionality. *In Proceedings of the 22th Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 233–243, 2017.

Thanh-Son Nguyen, Hady Wirawan Lauw et Panayiotis Tsaparas : Review synthesis for micro-review summarization. *In Proceedings of the Eighth International Conference on Web Search and Data Mining, WSDM 2015*, pages 169–178, 2015.

John Nickolls, Ian Buck, Michael Garland et Kevin Skadron : Scalable parallel programming with CUDA. *ACM Queue*, 6(2):40–53, 2008.

Rodrigo Nogueira et Kyunghyun Cho : Passage re-ranking with BERT. *CoRR*, 2019.

Neil O'Hare et Vanessa Murdock : Modeling locations with social media. *Information Retrieval*, 16(1):30–62, 2013.

Kezban Dilek Onal, Ye Zhang, Ismail Sengor Altingovde, Md Mustafizur Rahman, Pinar Karagoz, Alex Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten McNamara *et al.* : Neural information retrieval : At the end of the early years. *Information Retrieval Journal*, pages 1–72, 2017.

Ozer Ozdikis, Heri Ramampiaro et Kjetil Nørvåg : Locality-adapted kernel densities of term co-occurrences for location prediction of tweets. *Information Processing & Management*, 56(4):1280 – 1299, 2019.

Damien Palacio, Guillaume Cabanac, Christian Sallaberry et Gilles Hubert : On the evaluation of geographic information retrieval systems - evaluation framework and case study. *International Journal on Digital Libraries*, 11(2):91–109, 2010.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan et Xueqi Cheng : Text matching as image recognition. *In Proceedings of the 30th Conference on Artificial Intelligence, AAAI 2016*, pages 2793–2799, 2016.

Emanuel Parzen : On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

Francis Jeffry Pelletier : The principle of semantic compositionality. *Topoi*, 13 (1):11–24, 1994.

Jeffrey Pennington, Richard Socher et Christopher D. Manning : Glove : Global vectors for word representation. *In Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1532–1543, 2014.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee et Luke Zettlemoyer : Deep contextualized word representations. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2018*, pages 2227–2237, 2018.

David Martin Powers : Evaluation : from precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, pages 37–63, 2011.

Reid Priedhorsky, Aron Culotta et Sara Y. Del Valle : Inferring the origin locations of tweets with quantitative confidence. *In Proceedings of the 17th Conference on Computer Supported Cooperative Work, CSCW 2014*, pages 1523–1536, 2014.

Ross S. Purves, Paul D. Clough, Christopher B. Jones, Avi Arampatzis, Bénédicte Bucher, David Finch, Gaihua Fu, Hideo Joho, Awase Khirni Syed, Subodh Vaid et Bisheng Yang : The design and implementation of SPIRIT : a spatially aware search engine for information retrieval on the internet. *International Journal of Geographical Information Science*, 21(7):717–745, 2007.

Ross S. Purves, Paul D. Clough, Christopher B. Jones, Mark M. Hall et Vanessa Murdock : Geographic information retrieval : Progress and challenges in spatial search of text. *Foundations and Trends in Information Retrieval*, 12(2-3):164–318, 2018.

Ross S Purves, Alistair Edwardes et Mark Sanderson : Describing the where–improving image annotation and search through geography. *In Proceedings of the 2008 workshop on Metadata Mining for Image Understanding (MMIU 2008)*, pages 1–11, 2008.

Yifan Qiao, Chenyan Xiong, Zheng-Hao Liu et Zhiyuan Liu : Understanding the behaviors of BERT in ranking. *CoRR*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei et Ilya Sutskever : Language models are unsupervised multitask learners. *OpenAI Blog*, 1 (8):9, 2019.

Vineeth Rakesh, Chandan K. Reddy, Dilpreet Singh et Ramachandran M. S. : Location-specific tweet detection and topic summarization in twitter. *In Advances in Social Networks Analysis and Mining, ASONAM 2013*, pages 1441–1444, 2013.

Jinfeng Rao, Hua He et Jimmy J. Lin : Noise-contrastive estimation for answer selection with deep neural networks. *In Proceedings of the 25th International Conference on Information and Knowledge Management, CIKM 2016*, pages 1913–1916, 2016.

Lev-Arie Ratinov et Dan Roth : Design challenges and misconceptions in named entity recognition. *In Proceedings of the 13th Conference on Computational Natural Language Learning, CoNLL 2009*, pages 147–155, 2009.

Mirco Ravanelli et Yoshua Bengio : Speaker recognition from raw waveform with sincnet. *In Proceedings of the 7th Spoken Language Technology Workshop, SLT 2018*, pages 1021–1028, 2018.

Russell D. Reed et Robert J. Marks : *Neural Smithing : Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press, 1998.

Radim Rehurek et Petr Sojka : Software framework for topic modelling with large corpora. *In Proceedings of the 7th International Conference on Language Resources and Evaluation, Workshop on New Challenges for NLP Frameworks, LREC 2010*, pages 45–50, 2010.

Tumasch Reichenbacher, Stefano De Sabbata, Ross S. Purves et Sara Irina Fabrikant : Assessing geographic relevance for mobile search : A computational model and its validation via crowdsourcing. *Journal of the Association for Information Science and Technology*, 67(11):2620–2634, 2016.

Nils Reimers et Iryna Gurevych : Sentence-bert : Sentence embeddings using siamese bert-networks. *In Proceedings of the 24th Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 3980–3990, 2019.

Burghard B Rieger : *On distributed representation in word semantics*. International Computer Science Institute Berkeley, CA, 1991.

Alan Ritter, Sam Clark, Mausam et Oren Etzioni : Named entity recognition in tweets : An experimental study. *In Proceedings of the 16th Conference on Empirical Methods in Natural Language Processing, EMNLP 2011*, pages 1524–1534, 2011.

Stephen E. Robertson et Karen Spärck Jones : Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

João B. Rocha-Junior, Orestis Gkorgkas, Simon Jonassen et Kjetil Nørvåg : Efficient processing of top-k spatial keyword queries. *In Proceedings of the 12th International Symposium in Spatial and Temporal Databases, SSTD 2011*, pages 205–222, 2011.

Douglas LT Rohde, Laura M Gonnerman et David C Plaut : An improved model of semantic similarity based on lexical co-occurrence. *Communications of the ACM*, 8(627-633):116, 2006.

Stephen Roller, Michael Speriosu, Sarat Rallapalli, Benjamin Wing et Jason Baldridge : Supervised text-based geolocation using language models on an adaptive grid. *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012*, pages 1500–1510, 2012.

Frank Rosenblatt : The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Peter J Rousseeuw : Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Sebastian Ruder : An overview of gradient descent optimization algorithms. *CoRR*, 2016.

David E Rumelhart, Geoffrey E Hinton et Ronald J Williams : Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

KyoungMin Ryoo et Sue Moon : Inferring twitter user locations with 10 km accuracy. *In Proceedings of the 23rd International World Wide Web Conference, WWW 2014*, pages 643–648, 2014.

Stefano De Sabbata, Omar Alonso et Stefano Mizzaro : Classical vs. crowdsourcing surveys for eliciting geographic relevance criteria. *In Proceedings of the 3rd Italian Information Retrieval Workshop, CEUR Workshop 2012*, volume 835, pages 65–72, 2012.

Adam Sadilek, Henry A. Kautz et Jeffrey P. Bigham : Finding your friends and following them to where you are. *In Proceedings of the 5th International Conference on Web Search and Web Data Mining, WSDM 2012*, pages 723–732, 2012.

Gerard Salton : Information storage and retrieval. *Reports on Analysis, Search and iterative Retrieval*, 1968.

Gerard Salton et Michael McGill : *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1984.

Gerard Salton, A. Wong et Chung-Shu Yang : A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

Mark Sanderson et Janet Kohler : Analyzing geographic queries. *In Proceedings of the 5th Workshop On Geographic Information Retrieval, GIR 2008*, pages 8–10, 2004.

Victor Sanh, Lysandre Debut, Julien Chaumond et Thomas Wolf : Distilbert, a distilled version of BERT : smaller, faster, cheaper and lighter. *CoRR*, 2019.

Rudolf Schneider, Sebastian Arnold, Tom Oberhauser, Tobias Klatt, Thomas Steffek et Alexander Löser : Smart-md : Neural paragraph retrieval of medical topics. *In Proceedings of the 27th International Conference on World Wide Web, WWW 2018*, pages 203–206, 2018.

Axel Schulz, Aristotelis Hadjakos, Heiko Paulheim, Johannes Nachtwey et Max Mühlhäuser : A multi-indicator approach for geolocalization of tweets. *In Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013*, pages 573–582, 2013.

Aliaksei Severyn et Alessandro Moschitti : Learning to rank short text pairs with convolutional deep neural networks. *In Proceedings of the 38th International Conference on Research and Development in Information Retrieval, SIGIR 2015*, pages 373–382, 2015.

Khaled Shaalan : A survey of arabic named entity recognition and classification. *Computational Linguistics*, 40(2):469–510, 2014.

Blake Shaw, Jon Shea, Siddhartha Sinha et Andrew Hogue : Learning to rank for spatiotemporal search. *In Proceedings of the 6th International Conference on Web Search and Data Mining, WSDM 2013*, pages 717–726, 2013.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng et Grégoire Mesnil : A latent semantic model with convolutional-pooling structure for information retrieval. *In Proceedings of the 23rd International Conference on Conference on Information and Knowledge Management, CIKM 2014*, pages 101–110, 2014a.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng et Grégoire Mesnil : Learning semantic representations using convolutional neural networks for web search. *In 23rd International World Wide Web Conference, WWW 2014*, pages 373–374, 2014b.

Mário J. Silva, Bruno Martins, Marcirio Silveira Chaves, Ana Paula Afonso et Nuno Cardoso : Adding geographic scopes to web resources. *Computers, Environment and Urban Systems*, 30(4):378–399, 2006.

Ilya Sutskever, Oriol Vinyals et Quoc V. Le : Sequence to sequence learning with neural networks. *In Proceedings of the 24th Conference on Neural Information Processing Systems, NIPS 2014*, pages 3104–3112, 2014.

Lynda Tamine, Laure Soulier, Gia-Hung Nguyen et Nathalie Souf : Offline versus online representation learning of documents using external knowledge. *Transactions on Information Systems*, 37(4):42 :1–42 :34, 2019.

Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv et Ming Zhou : Multiway attention networks for modeling sentence pairs. *In Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 4411–4417, 2018.

Jaime Teevan, Daniel Ramage et Meredith Ringel Morris : #twittersearch : a comparison of microblog search and web search. *In Proceedings of the 4th International Conference on Web Search and Web Data Mining, WSDM 2011*, pages 35–44, 2011.

Benjamin E. Teitler, Michael D. Lieberman, Daniele Panozzo, Jagan Sankaranarayanan, Hanan Samet et Jon Sperling : Newsstand : a new view on news. *In proceedings of the 16th International Symposium on Advances in Geographic Information Systems, GIS 2008*, page 18, 2008.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das et Ellie Pavlick : What do you learn from context ? probing for sentence structure in contextualized word representations. *In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019*, pages 1–17, 2019.

Waldo R Tobler : A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240, 1970.

Subodh Vaid, Christopher B. Jones, Hideo Joho et Mark Sanderson : Spatio-textual indexing for geographical search on the web. *In proceedings of the 9th International Symposium in Spatial and Temporal Databases, SSTD 2005*, volume 3633, pages 218–235, 2005.

Maria Vasardani, Stephan Winter et Kai-Florian Richter : Locating place names from place descriptions. *International Journal of Geographical Information Science*, 27(12):2509–2532, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser et Illia Polosukhin : Attention is all you need. *In Proceedings of the 31th Conference on Neural Information Processing Systems, NIPS 2017*, pages 5998–6008, 2017.

Sarah Vieweg, Amanda Lee Hughes, Kate Starbird et Leysia Palen : Microblogging during two natural hazards events : what twitter may contribute to situational awareness. *In Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010*, pages 1079–1088, 2010.

Thaddeus Vincenty : Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, 23(176):88–93, 1975.

Ivan Vulic, Goran Glavas, Nikola Mrksic et Anna Korhonen : Post-specialisation : Retrofitting vectors of words unseen in lexical resources. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 516–527, 2018.

Ivan Vulic et Marie-Francine Moens : Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. *In Proceedings of the 38th International Conference on Research and Development in Information Retrieval, SIGIR 2015*, pages 363–372, 2015.

Ivan VULIC et Nikola MRKSIC : Specialising word vectors for lexical entailment. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2018*, pages 1134–1145, 2018.

Ivan VULIC, Nikola MRKSIC, Roi REICHART, Diarmuid Ó SÉAGHDHA, Steve J. YOUNG et Anna KORHONEN : Morph-fitting : Fine-tuning word vector spaces with simple language-specific rules. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 56–68, 2017.

Hanna M. WALLACH : Topic modeling : beyond bag-of-words. *In Proceedings of the 23rd International Conference on Machine Learning, ICML 2006*, pages 977–984, 2006.

Shengxian WAN, Yanyan LAN, Jiafeng GUO, Jun XU, Liang PANG et Xueqi CHENG : A deep architecture for semantic matching with multiple positional sentence representations. *In Proceedings of the 30th Conference on Artificial Intelligence, AAAI 2016*, pages 2835–2841, 2016a.

Shengxian WAN, Yanyan LAN, Jun XU, Jiafeng GUO, Liang PANG et Xueqi CHENG : Match-srnn : Modeling the recursive matching structure with spatial RNN. *In Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016*, pages 2922–2928, 2016b.

Bingning WANG, Kang LIU et Jun ZHAO : Inner attention based recurrent neural networks for answer selection. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, pages 1288—-1297, 2016.

Ruishuang WANG, Zhao LI, Jian CAO, Tong CHEN et Lei WANG : Convolutional recurrent neural networks for text classification. *In Proceedings of the 18th International Joint Conference on Neural Networks, IJCNN 2019*, pages 1–6, 2019.

Jason WESTON, Sumit CHOPRA et Keith ADAMS : #tagspace : Semantic embeddings from hashtags. *In Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1822–1827, 2014.

Benjamin WING et Jason BALDRIDGE : Hierarchical discriminative classification for text-based geolocation. *In Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 336–348, 2014.

Ian H. WITTEN, Alistair MOFFAT et Timothy C. BELL : *Managing Gigabytes : Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, 1994.

Allison WOODRUFF et Christian PLAUNT : GIPSY : automated geographic indexing of text documents. *Journal of the American Society for Information Science*, 45(9):645–655, 1994.

Zhibiao Wu et Martha Stone Palmer : Verb semantics and lexical selection. *In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, ACL 1994*, pages 133–138, 1994.

Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen et Sen Wang : Learning graph-based POI embedding for location-based recommendation. *In Proceedings of the 25th International Conference on Information and Knowledge Management, CIKM 2016*, pages 15–24, 2016.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu et Russell Power : End-to-end neural ad-hoc ranking with kernel pooling. *In Proceedings of the 40th International Conference on Research and Development in Information Retrieval, SIGIR 2017*, pages 55–64, 2017.

Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu et Tie-Yan Liu : RC-NET : A general framework for incorporating knowledge into word representations. *In Proceedings of the 23rd International Conference on Conference on Information and Knowledge Management, CIKM 2014*, pages 1219–1228, 2014.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda et Yoshiyasu Takefuji : Joint learning of the embedding of words and entities for named entity disambiguation. *In Proceedings of the 20th Conference on Computational Natural Language Learning, CoNLL 2016*, pages 250–259, 2016.

Bo Yan, Krzysztof Janowicz, Gengchen Mai et Song Gao : From ITDL to place2vec : Reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. *In Proceedings of the 25th International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2017*, pages 35 :1–35 :10, 2017.

Liu Yang, Qingyao Ai, Jiafeng Guo et W. Bruce Croft : anmm : Ranking short answer texts with attention-based neural matching model. *In Proceedings of the 25th International Conference on Information and Knowledge Management, CIKM 2016*, pages 287–296, 2016.

Yuan Yao, Lorenzo Rosasco et Andrea Caponnetto : On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Hongzhi Yin, Bin Cui, Ling Chen, Zhiting Hu et Chengqi Zhang : Modeling location-based user rating profiles for personalized recommendation. *ACM Transactions on Knowledge Discovery from Data*, 9(3):1–41, 2015.

Mo Yu et Mark Dredze : Improving lexical embeddings with semantic knowledge. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pages 545–550, 2014.

Quan YUAN, Gao CONG, Zongyang MA, Aixin SUN et Nadia MAGNENAT-THALMANN : Who, where, when and what : discover spatio-temporal topics for twitter users. *In Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining, KDD 2013*, pages 605–613, 2013.

Yisleidy Linares ZAILA et Danilo MONTESI : Geographic information extraction, disambiguation and ranking techniques. *In Proceedings of the 9th Workshop on Geographic Information Retrieval, GIR 2015*, pages 11 :1–11 :7, 2015.

Hamed ZAMANI et W. Bruce CROFT : Estimating embedding vectors for queries. *In Proceedings of the 6th International Conference on the Theory of Information Retrieval, ICTIR 2016*, pages 123–132, 2016.

Hamed ZAMANI et W. Bruce CROFT : Relevance-based word embedding. *In Proceedings of the 40th International Conference on Research and Development in Information Retrieval, SIGIR 2017*, pages 505–514, 2017.

Matthew D. ZEILER : ADADELTA : an adaptive learning rate method. *CoRR*, 2012.

Chao ZHANG, Liyuan LIU, Dongming LEI, Quan YUAN, Honglei ZHUANG, Tim HANRATTY et Jiawei HAN : Triovecevent : Embedding-based online local event detection in geo-tagged tweet streams. *In Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining, KDD 2017*, pages 595–604, 2017a.

Chao ZHANG, Guangyu ZHOU, Quan YUAN, Honglei ZHUANG, Yu ZHENG, Lance M. KAPLAN, Shaowen WANG et Jiawei HAN : Geoburst : Real-time local event detection in geo-tagged tweet streams. *In Proceedings of the 39th International Conference on Research and Development in Information Retrieval, SIGIR 2016*, pages 513–522, 2016.

Dongxiang ZHANG, Chee-Yong CHAN et Kian-Lee TAN : Processing spatial keyword query as a top-k aggregation query. *In Proceedings of the 37th International Conference on Research and Development in Information Retrieval, SIGIR 2014*, pages 355–364, 2014.

Haoyu ZHANG, Jingjing CAI, Jianjun XU et Ji WANG : Pretraining-based natural language generation for text summarization. *In Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019*, pages 789–797, 2019.

Jia-Dong ZHANG et Chi-Yin CHOW : Geosoca : Exploiting geographical, social and categorical correlations for point-of-interest recommendations. *In Proceedings of the 38th International Conference on Research and Development in Information Retrieval, SIGIR 2015*, pages 443–452, 2015.

Wei ZHANG et Judith GELERNTER : Geocoding location expressions in twitter messages : A preference learning method. *Journal of Spatial Information Science*, 9 (1):37–70, 2014.

Yating Zhang, Adam Jatowt et Katsumi Tanaka : Is tofu the cheese of asia ? : Searching for corresponding objects across geographical areas. *In Proceedings of the 26th International Conference on World Wide Web, WWW 2017*, pages 1033–1042, 2017b.

Yu Zhang et Qiang Yang : A survey on multi-task learning. *CoRR*.

Yue Zhang et Stephen Clark : Joint word segmentation and POS tagging using a single perceptron. *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, ACL 2008*, pages 888–896, 2008.

Kaiqi Zhao, Gao Cong et Aixin Sun : Annotating points of interest with geo-tagged tweets. *In Proceedings of the 25th International Conference on Information and Knowledge Management, CIKM 2016*, pages 417–426, 2016.

Shenglin Zhao, Tong Zhao, Irwin King et Michael R. Lyu : Geo-teaser : Geo-temporal sequential embedding rank for point-of-interest recommendation. *In Proceedings of the 26th International Conference on World Wide Web Companion, WWW 2017*, pages 153–162, 2017.

Guoqing Zheng et Jamie Callan : Learning to reweight terms with distributed representations. *In Proceedings of the 38th International Conference on Research and Development in Information Retrieval, SIGIR 2015*, pages 575–584, 2015.

Xin Zheng, Jialong Han et Aixin Sun : A survey of location prediction on twitter. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1652–1671, 2018.

Xiaoqiang Zhou, Baotian Hu, Qingcai Chen et Xiaolong Wang : Recurrent convolutional neural network for answer selection in community question answering. *Neurocomputing*, 274:8–18, 2018.

Yinghua Zhou, Xing Xie, Chuang Wang, Yuchang Gong et Wei-Ying Ma : Hybrid index structures for location-based web search. *In Proceedings of the 14th International Conference on Information and Knowledge Management, CIKM 2005*, pages 155–162, 2005.

Ming Zhu, Aman Ahuja, Wei Wei et Chandan K. Reddy : A hierarchical attention retrieval model for healthcare question answering. *In Proceedings of the 28th World Wide Web Conference, WWW 2019*, pages 2472–2482, 2019.

Wenbo Zong, Dan Wu, Aixin Sun, Ee-Peng Lim et Dion Hoe-Lian Goh : On assigning place names to geography related web pages. *In Proceedings of the 2005 Joint Conference on Digital Libraries, JCDL 2005*, pages 354–362, 2005.

Guido Zuccon, Bevan Koopman, Peter Bruza et Leif Azzopardi : Integrating and evaluating neural word embeddings in information retrieval. *In Proceedings of the 20th Australasian Document Computing Symposium, ADCS 2015*, pages 1–8, 2015.