Réseaux de Neurones pour la Représentation de Contextes Continus des Mots

Killian Janod[*,**] **— Mohamed Morchid**[*] **— Richard Dufour**[*] **— Georges Lianrès**[*]

[*] *Université d'Avignon, LIA (France),* [**] *ORKIS - Aix en Provence (France)*

RÉSUMÉ. *Les méthodes d'apprentissage profond s'appuient de plus en plus sur des représentations vectorielles continues des mots. Ces méthodes, déjà appliquées avec succès dans de nombreuses tâches de traitement automatique du langage naturel écrit et oral, sont capables de représenter des mots ainsi que les relations les liant. De manière générale, ces méthodes utilisent des représentations par "sac-de-mots" et traitent donc tous les mots d'un contexte de façon égale. Cet article propose une méthode originale qui s'appuie sur les modèles de contextes continus en intégrant la position relative des mots dans un contexte. Les résultats montrent que l'information portée par les contextes continus permet un gain jusqu'à 7 % sur le test qualitatif "de relation sémantique" et permet d'obtenir des résultats pertinents pour une application concrète (identification de thèmes de dialogues dans le cadre du projet DECODA), alors que des performances faibles sont obtenues sans l'information contextuelle.*

ABSTRACT. *Deep learning methods use more and more word embedding representations. Those methods, which have been already applied with success on various tasks of written and spoken natural language processing, are able to represent words and the relations between them. Usually in those methods context windows are represented as bag-of-words, i.e. every word in the context is treated equally. This paper proposes an original method inspired from the Continuous Context Models by integrating words relative positions. The results observed confirm that the information given by continuous context models allow us to gain more than 7 % on the Word Relationship test and achieve relevant results on a real application (theme identification with the DECODA corpus) that couldn't be done without this original information .*

MOTS-CLÉS : *sac-de-mots, Contextes Continus, réseau de neurones artificiel, Word2vec*

KEYWORDS: *Word embeddings, Continuous context, Neural network, Word2vec*

1. Introduction

Le choix d'une bonne représentation des mots est souvent indispensable pour mener à bien des tâches de traitement automatique du langage naturel écrit ou oral. La représentation par "sac-de-mots" est la plus couramment utilisée, les documents étant représentés par les vecteurs de fréquence des mots qui les composent. Cependant, cette représentation a pour défaut de ne capturer que très peu d'informations sur la structure des documents et leur relation entre eux. Pour cette raison, les approches à base de n-grammes ont été introduites. Cette représentation associe à chaque mot son historique proche pour modéliser des occurrences plus complexes. Dans (Sahlgren, 2008), l'auteur propose de capturer les relations sémantiques entre les mots en utilisant "l'Hypothèse de Distribution" qui dicte que "*les mots apparaissant dans le même contexte ont la même signification*". Les représentations découlant de cette hypothèse projettent les mots dans l'espace de tous les contextes présents dans les données d'apprentissage. Les projections dans ces espaces sont le plus souvent creuses et de très grande dimension. Ces projections souffrent de ce qui est appelé "Fléau de la dimension". D'autres représentations avec un plus haut niveau d'abstraction, comme l'Allocation Latente de Dirichlet (*Latent Dirichlet Allocation* (LDA) (Blei *et al.*, 2003)) cherchent à modéliser les thèmes latents d'un corpus de documents.

Plus récemment, des méthodes s'appuyant sur des réseaux de neurones artificiels (Bengio *et al.*, 2003) se sont imposées. Ces méthodes sont capables de représenter un mot par un vecteur plein de taille contrôlée. Ce vecteur correspond à la position du mot dans un espace multi-dimensionnel. Ce type de technique est appelé *Word embedding*. Ces méthodes étaient d'abord employées pour des Modèles de Langue Neuronaux (Bengio et Heigold, 2014 ; Collobert *et al.*, 2011) et ensuite appliquées dans de nombreuses tâches de traitement du langage. (Do *et al.*, 2014 ; Vaswani *et al.*, 2013 ; Mesnil *et al.*, 2015). Parmi ces approches, la méthode *Word2vec* (Mikolov *et al.*, 2013a) apparaît aujourd'hui à l'état-de-l'art en matière de représentation distribuée des mots. *Word2vec* propose deux architectures s'appuyant sur les réseaux de neurones artificiels et conçues pour traiter une grande quantité de textes ainsi qu'inférer une structure linéaire qui modélise des relations sémantiques et syntaxiques liant les mots. Cette modélisation a prouvé son efficacité dans plusieurs tâches en traitement du langage naturel (TALN) oral et écrit. Les réseaux de neurones artificiels *Word2vec* construisent, pour chaque mot, une fenêtre de contexte. À l'intérieur de cette fenêtre, tous les mots sont traités de façon égale.

Ce papier introduit une pondération log-linéaire des mots s'appuyant sur les contextes continus de ces mots (Bigot *et al.*, 2013) intégrant une information structurelle des mots. Cette méthode permet de réduire l'importance de la taille maximum de la fenêtre du contexte. Cette approche originale a été appliquée sur deux tâches du TALN : le test "*Semantic-Syntactic Word Relationship*" (Mikolov *et al.*, 2013a), permettant de vérifier que le modèle a correctement capturé les relations sémantiques et syntaxiques liant les mots, et une tâche plus appliquée dans un contexte réaliste, à savoir l'identification de thèmes dans des dialogues bruités dans le cadre du projet DECODA (Bechet *et al.*, 2012).

L'article est organisé comme suit : la section 2 détaille les deux architectures de réseaux de neurones artificiels *Word2vec* ainsi que la fonction de pondération intégrée dans ces architectures. Les expériences et leurs résultats sont décrits dans la section 3 avant de conclure dans la section 4.

2. Approche proposée

La section 2.1 présente le framework *Word2vec* en détaillant l'architecture réseaux de neurones artificiels "Sac-de-mots continus"(*Continuous Bag-Of-Words*, *CBOW*), et l'architecture *Skip-gram*. Les fenêtres de contextes dynamiques et l'approche s'appuyant sur les modèles de contextes continus sont définies et expliquées dans la section 2.2.

2.1. *Les réseaux de neurones artificiels Word2vec*

La méthode *Word2vec* est définie dans (Mikolov *et al.*, 2013a). Cette méthode propose deux réseaux de neurones artificiels simples (*i.e.* peu profonds) : l'architecture *CBOW* et l'architecture *Skip-gram*. Ces architectures ont besoin, pour s'entrainer, de mots centraux (ou mots d'attention) et leurs fenêtres de contexte respectives. Une fenêtre de contexte correspond aux n mots précédents et n mots suivants un mot central. La valeur de n est à adapter selon la tâche et les données.

Chacune de ces architectures est composée de trois couches. Une couche d'entrée, une couche cachée et une couche de sortie.

Le schema 1 présente l'architecture *CBOW* et le schéma 2 l'architecture *Skip-gram* pour le segment de texte : "$w_{-2}\ w_{-1}\ w_{central}\ w_{+1}\ w_{+2}$", et dont le mot d'attention est $w_{central}$. La couche d'entrée contient un sac-de-mots contenant la fenêtre de contexte pour le *CBOW* ou le mot central pour le *Skip-gram*. La couche cachée contient la projection de l'entrée dans la matrice globale des poids. La couche de sortie est la prédiction du modèle. Soit un mot pour l'architecture *CBOW*, et un contexte pour l'architecture *Skip-gram*. Cette prédiction est uniquement utilisée pour calculer l'erreur des réseaux et la rétro-propagation du gradient. Cette rétro-propagation permet de corriger la matrice globale en rapprochant dans l'espace multi-dimensionnel les mots de leurs contextes respectifs.

La couche de sortie des réseaux est composée de neurones artificiels avec une fonction d'activation *Softmax*. Dans le but de réduire la complexité algorithmique de cette fonction, les auteurs dans (Mikolov *et al.*, 2013a) ont proposé deux alternatives : le *Softmax hiérarchique*(*Hierarchical softmax*) et *l'échantillonnage négatif*(*negative sampling*). Elles ont permis un accroissement important de la vitesse de traitement du réseau. D'après (Mikolov *et al.*, 2013a), cette augmentation de la vitesse de traitement permet aux modèles d'apprendre sur de plus importantes quantités de textes dans un temps raisonnable, un plus grand nombre de données d'apprentissage impliquant de meilleures représentations (cf. schéma 3).

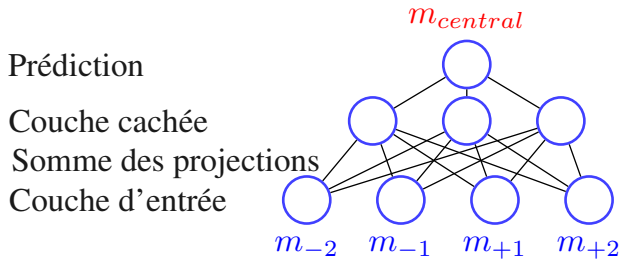

Figure 1. *Réseau de neurones CBOW sans pondération qui prédit un mot sachant les mots dans la fenêtre de contexte (deux mots avant et deux mots après dans ce schéma).*

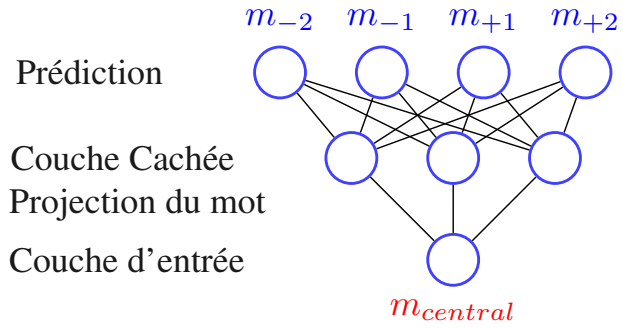

Figure 2. *Réseau de neurones Skip-gram sans pondération qui tente de prédire sachant un mot son contexte(deux mots avant et deux mots après dans ce schéma).*

2.1.1. *Sacs-de-mots continus (CBOW)*

L'architecture du *CBOW* est un réseau de neurones artificiels simple et log-linéaire. La couche d'entrée de ce réseau de neurones utilise des "sacs-de-mots" binaires représentant une fenêtre de contexte. Dans cette configuration, les vecteurs d'entrée sont des vecteurs de la taille du vocabulaire avec un 1 dans la colonne i si le mot i est présent dans le document, 0 sinon. Chaque mot est alors projeté dans la matrice globale, l'ensemble des représentations est ensuite additionné pour former une unique couche cachée. Cette couche cachée passe par la couche de sortie et les fonctions d'activation type "Softmax" tentent de prédire le mot au coeur de la fenêtre. L'erreur de prédiction

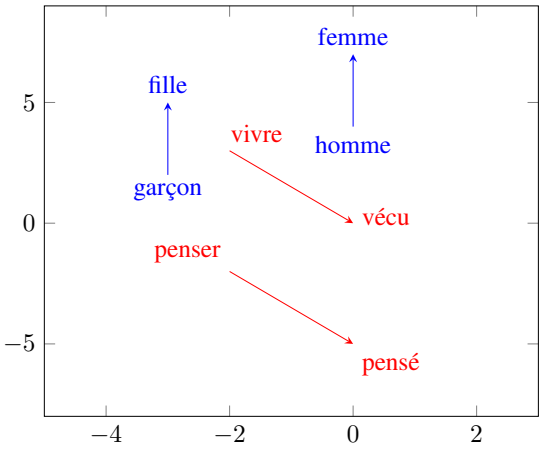

Figure 3. *Exemple de relations inter-mots*

est ensuite utilisée pour corriger les matrices de poids via une rétro-propagation de gradient. Cette architecture essaie de maximiser la vraisemblance ci-dessous :

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_t|w_{t-\frac{c}{2}}...w_{t+\frac{c}{2}}) \quad [1]$$

dans laquelle T est la taille des données d'apprentissage et c est la taille maximum de la fenêtre de contexte. L'architecture *CBOW* est plus efficiente que son homologue *Skip-gram* et capture une meilleure représentation des mots fréquents (Mikolov *et al.*, 2013a).

2.1.2. *Architecture Skip-gram*

L'architecture *Skip-gram* est également un réseau de neurones artificiels simple et log-linéaire. Ce réseau apprend, contrairement au *CBOW*, à prédire une fenêtre de contexte sachant le mot au coeur de celle-ci (voir figure 2). La couche d'entrée du réseau ne contient donc que la représentation en sac-de-mots binaire du mot au cœur du contexte. Ce mot est projeté dans la matrice de poids globale, puis transmis à la couche de sortie qui va prédire un mot. Cette prédiction est ensuite corrigée par rétro-propagation pour chacun des mots de la fenêtre de contexte. Un réseau *Skip-gram* maximise la vraisemblance suivante :

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j=t-c,j\neq t}^{t+c}\log p(w_j|w_t) \quad [2]$$

Comparativement au *CBOW*, les réseaux *Skip-gram* apprennent de meilleures représentations sémantiques et sont plus adaptés aux mots peu fréquents (Mikolov *et al.*, 2013b).

2.2. *Fenêtre de contexte dynamique (DW) et Fonction de pondération par contextes continus (CCM)*

Dans (Mikolov *et al.*, 2013a), la fenêtre de contexte dynamique est définie et attribuée au modèle *Skip-gram*. Mais le "Word2vec toolkit"[1] applique cette méthode à la fois pour le *CBOW* et le *Skip-gram*. La fenêtre de contexte dynamique permet au modèle *Skip-gram* d'ignorer aléatoirement des mots aux extrémités de la fenêtre de contexte. Une réduction de la fenêtre implique une réduction du nombre de rétropropagations et donc une accélération du traitement. En contrepartie, le réseau ignore parfois des relations de plus longue distance. En ignorant de façon régulière les mots éloignés du centre de la fenêtre de contexte, le réseau applique une forme de pondération linéaire aux relations éloignées. En plus d'ignorer des relations, cette méthode attribue à tous les mots dans la fenêtre de contexte la même importance (binaire). Une fonction de pondération log-linéaire s'appuyant sur les modèles de contextes continus (Bigot *et al.*, 2013) est proposée comme alternative à la fenêtre de contexte dynamique. Cette nouvelle fonction de pondération donne alors un poids à chaque mot qui ne dépend que de la distance séparant les mots du contexte à pondérer du mot au cœur du contexte. La fonction de pondération de distance est définie par :

$$\frac{\alpha}{b + \beta log(d)} \qquad [3]$$

où d est la distance, en nombre de mots, séparant le mot à pondérer et le mot de référence ; α, b et β sont les paramètres de la fonction de distance qui définissent l'impact des mots les plus éloignés. Le graphique 4 montre la différence d'importance accordée aux mots d'un contexte selon son éloignement au mot central, pour l'approche classique avec fenêtre de contexte dynamique (dynamic window ou DW) et avec la pondération par contextes continus *CCM*. Notons que l'importance attribuée par la fenêtre dynamique n'est vraie que pour un nombre de tirage infini d'un couple *mot-contexte* en particulier.

La fonction de contextes continus est capable de mettre en avant les mots les plus proches tout en conservant les relations plus distantes (cf. schéma 4). En effet, une approche par pondération permet de conserver tous les mots du contexte (cf. figure 4-courbe rouge), alors qu'avec un re-échantillonnage, les relations ont seulement une probabilité d'être capturée (cf. figure 4-histogramme bleu). Une méthode par pondération a deux effets sur l'apprentissage : d'un côté, en pondérant les mots lors de l'apprentissage au lieu de les négliger, elle permet au réseaux de neurones artificiels d'utiliser cette information de distance ; d'un autre côté, ignorer des mots est aussi utile pour rendre le modèle plus rapide. La méthode par pondération ralentit les réseaux *Skip-gram* de 20%. Par contre, un impact négligeable sur le réseau *CBOW* a été observé dans le cadre de nos expériences.

1. https ://code.google.com/p/word2vec

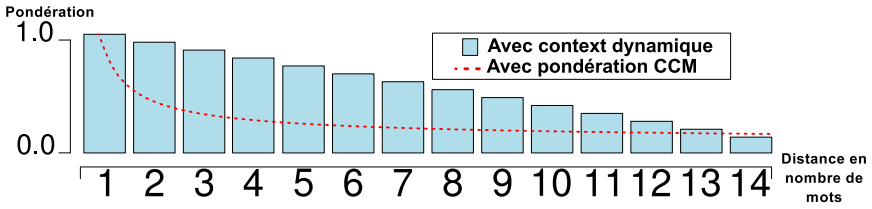

Figure 4. *Importance des mots dans un contexte avec et sans modèle de contextes continus. L'histogramme présente l'espérance de la pondération attribué globalement pour un contexte donnée par ré-échantillonnage du contexte. La courbe rouge présente la pondération effectivement attribuée dans une fenêtre de contexte par la fonction de contextes continus.*

3. Expériences et Résultats

Dans cette section, nous allons d'abord analyser les différences des représentations produites par des modèles avec et sans fonction de pondération *CCM* (cf. section 3.1). Deux expériences sont expliquées ensuite pour évaluer l'efficacité de l'approche avec une pondération par contextes continus (cf. sections 3.2 et 3.3).

3.1. *Impact sémantique des modèles de contextes continus (CCM)*

En comparant les voisins de plusieurs exemples de mots (cf. tableau 1), nous pouvons remarquer que, avec la pondération *CCM* que nous proposons, les mots ont tendance à être regroupés par thématique, alors que les mots avec la fenêtre classique "DW" ne le sont pas. Par exemple, pour le mot "Holidays", nous trouvons d'abord la thématique des *loisirs* avec ("holiday", "vacation" and "festivities") puis en second plan la thématique *religion* ("thanksgiving", "easter" et "christmas"). Comme cette séparation thématique est une observation de l'effet de la fonction de pondération *CCM* plus qu'une évaluation, l'impact des modèles avec pondération *CCM* est mesuré plus en détail par deux expériences différentes. La première expérience, détaillée dans la section 3.2, évalue la capacité du modèle à apprendre les relations sémantiques et syntaxiques des données d'apprentissage. Bien que cette évaluation permette de mesurer la qualité intrinsèque des modèles, elle en dit peu sur leur efficacité dans des cas d'utilisation plus pratiques. La seconde expérience, présentée dans la section 3.3, a pour but de mesurer l'efficacité du modèle dans une tâche d'identification de thématique conversationnelle.

3.2. *Evaluation des relations sémantiques et syntaxiques*

Le "*Semantic-Syntactic Word relationship test set*" est défini dans (Mikolov *et al.*, 2013a). L'objectif de cette suite de tests est de vérifier que la représentation

Tableau 1. *Exemple de contexte des mots sans (DW) et avec pondération CCM.*

Holidays		Meat	
Avec Distance	Avec DW	Avec Distance	Avec DW
holiday	vacations	chicken	pork
vacation	thanksgiving	beef	not-pasterised
festivities	vacation	pork	mutton
thanksgiving	christmas	milk	eggs
easter	celebration	eggs	cattle
christmas	easter	seafood	chicken

des mots a bien capturé les relations sémantiques et syntaxiques liant les mots. Ce test est composé d'environ 19 000 questions divisées en deux catégories (Sémantique et Syntaxique). Chaque question est composée de deux paires de mots partageant une même relation. Par exemple les mots : *USA - Dollars* et *Europe - Euros* partagent la relation **est la monnaie de**. Chaque question est considérée comme correcte si l'équation [4] est vraie dans l'espace multidimensionnel.

$$w_{1c1} + w_{2c1} = w_{1c2} + w_{2c2}. \qquad [4]$$

La nature polysémique des mots, ainsi que l'absence de contrainte sur l'espace de projection, impliquent que l'équation [4] est pratiquement impossible. L'utilisation des propriétés, notamment algébriques, de l'espace de projection permettent de relâcher quelques contraintes sur la formulation de la question qui devient w_{1c2} **est-il le mot le plus proche de** $w_{1c1} - w_{2c1} + w_{2c2}$. La métrique utilisée, appelée couramment "précision", évalue un modèle par la proportion de bonnes réponses par rapport au nombre de questions. Tous les modèles utilisés pour cette expérience sont entraînés avec le même corpus de données en anglais composé de :

– l'ensemble de données "One Billion Word Language Modeling Benchmark" (31 millions de documents - 700 millions de mots),

– premier millard de caractères d'un corpus issu de Wikipedia (124 303 documents - 124 millions de mots),

– corpus GigaWord anglais de 1994 à 2011 (190 millions de documents - 3 771 millions de mots),

– Brown Corpus (57 341 documents - 1 million de mots).

Tous les documents du corpus sont passés en minuscule. Puis les mots sont segmentés par la présence d'espaces et de tirets. Enfin, la ponctuation a été retirée. Après prétraitement, l'ensemble du corpus contient 4 milliards de mots pour un vocabulaire d'environ 1 million de mots uniques. L'entraînement d'un modèle *Skip-gram* sur ces données a été réalisé pendant environ 12 heures sur une machine possédant 8 cœurs.

Pour cette expérience, plusieurs réseaux de neurones artificiels avec des paramètres différents ont été évalués. Le modèle faisant office de *baseline* (comme défini dans (Mikolov *et al.*, 2013a)) est composé d'une couche cachée de 300 neurones et d'une fenêtre de contexte de taille 10. Deux tailles de couches cachées (120 et 300) et trois tailles de contextes (10, 15 et 100) sont comparées. Comme moins de 1 % des documents du corpus dépassent les 100 mots, un contexte de taille 100 est ici considéré comme équivalent à l'utilisation du document entier comme contexte. Enfin, deux fonctions de pondération sont utilisées (Distance 1 et Distance 2) définies par l'équation [5] :

$$Distance1 = \frac{1 + log(2)}{1 + log(d)} \qquad \text{et} \qquad Distance2 = \frac{log(10)}{5 * log(d)} \qquad [5]$$

où d est la distance entre le mot auquel on attribue un score et le centre du context en nombre de mots.

Résultats

Les tableaux 2 et 3 montrent que l'utilisation de la fonction de pondération par *CCM* obtient globalement de meilleurs résultats. L'écart de performance le plus important est obtenu en considérant un document entier comme contexte, atteignant un gain absolu de précision de 7 % pour le *CBOW* et de 7,7 % pour le *Skip-gram*. Nous pouvons aussi noter, dans le tableau 2, que plus le contexte est large, plus le gain est important. De plus, le tableau 3 montre que la Distance 1 est plus favorable au modèle *Skip-gram*, avec des gains atteignant 2,1 % en absolu, alors que la Distance 2 est favorable au modèle *CBOW*, avec des gains absolus en précision jusqu'à 4,3 %. Enfin, le tableau 3 met en évidence que plus la couche cachée est large (i.e. plus le nombre de neurones est important) plus les modèles sont capables de retenir de l'information portée par les contextes continus.

Tableau 2. *Précision (%) en fonction de la taille des contextes (10, 15 and 100 mots).*

	Skip-gram			*CBOW*		
Taille de la couche cachée	300					
Taille du contexte	10	15	100	10	15	100
avec DW	50.0	50.9	43.7	39	38.9	36.9
avec Distance 1	55.0	53.7	51.4	39.9	39.6	43.9

Tableau 3. *Précision (%) sans (avec DW) et avec pondération CCM (Distance 1 et Distance 2) dans des espaces plus petits.*

	Skip-gram		*CBOW*	
Taille du contexte	10			
Taille de la couche cachée	120	300	120	300
Avec DW	43.9	50.0	29.0	39.0
Avec Distance 1	45.1	55.0	30.3	39.9
Avec Distance 2	40.0	52.1	31.5	43.5

Tableau 4. *Description des thèmes du corpus DECODA.*

	Nombre d'échantillons		
label	**Train**	**Dev**	**Test**
Problème d'itinéraire	145	44	67
Objets trouvés	143	33	63
Horaires	47	7	18
Carte de transport	106	24	47
État du trafic	202	45	90
Prix du ticket	19	9	11
Infractions	47	4	18
Offres spéciales	31	9	13
Total	**740**	**175**	**327**

3.3. *Classification de conversations transcrites automatiquement*

La seconde évaluation mesure les performances des deux types de représentation pour une tâche de classification automatique. Cette expérience est réalisée au moyen du corpus du projet DECODA (Bechet *et al.*, 2012 ; Morchid *et al.*, 2015 ; Morchid *et al.*, 2014a ; Morchid *et al.*, 2014b), ayant pour objectif d'identifier le thème abordé dans une conversation téléphonique. Le corpus DECODA est composé de 1 067 conversations téléphoniques découpées en trois morceaux, un corpus d'entraînement (Train) de 740 dialogues, un corpus de développement (Dev) de 175 dialogues, et 327 dialogues dans un corpus de Test. Les conversations ont été manuellement réparties parmi 8 thèmes.

Le système de reconnaissance de la parole utilisé est Speeral (Linares *et al.*, 2007). Le modèle acoustique est estimé sur 150 heures de documents parlés en condition téléphonique. Le vocabulaire du système est de 5 782 mots. Un modèle de langue tri-grammes est appris à partir d'un modèle de langue standard adapté avec les transcriptions du corpus de train. Une liste d'arrêt de 126 mots est utilisée[2] afin de supprimer les mots inutiles. Le système de transcription atteint un taux d'erreur-mot de 33,8 % sur le train, de 45,2 % sur le dev., et 49,5 % sur le test. Ces taux d'erreur-mot particulièrement élevés sont principalement dus à des disfluences verbales et à des mauvaises conditions acoustiques (par exemple, bruits de fond ou communication depuis un smartphone).

La projection des conversations dans l'espace multidimensionnel Word2vec est réalisée de la manière suivante : tout d'abord, les 1000 mots les plus discriminants sont sélectionnés en utilisant un combinaison de TF-IDF associé à un critère de pureté de Gini. Chaque dialogue est ensuite associé à un vecteur de scores représentant la distance entre la somme des mots du dialogue et chacun des mots discriminants. Enfin, les vecteurs de distance sont utilisés pour attribuer un thème à chaque conversation au moyen d'une approche de classification automatique. Pour mesurer l'efficacité de la

2. http ://code.google.com/p/stop-words/

pondération *CCM* proposée, 4 modèles sont utilisés : deux architectures *CBOW*, une avec la Distance 2 et une avec DW, et deux modèles *Skip-gram*, un avec la Distance 1 et un avec DW. Comme le projet DECODA est en français, ces modèles sont entraînés sur le corpus français suivant :

– le corpus français GigaWord (17 millions de documents - 500 millions de mots),

– une partie de Wikipedia (16 millions de documents - 400 millions de mots),

– des extraits de presse française issus de l'Agence France Presse (AFP), Le Monde et Le Soir (56 millions de documents - 737 millions de mots),

– un ensemble de documents extraits d'Internet (4 millions de documents - 108 millions de mots),

– des transcriptions manuelles issues de campagnes d'évaluation récentes (ESTER, EPAC, ETAPE et REPERE) (411 000 documents - 379 millions de mots).

Ce corpus subit la même phase de pré-traitement que celui présenté dans la section 3.2. Ce corpus contient approximativement 2 milliards de mots pour un vocabulaire d'environ 3 millions de mots uniques. Pour cette tâche, deux différents types de classifieurs sont utilisés. Le premier est un *Gradient Tree Boosting*(*GBT*) (Pedregosa *et al.*, 2011 ; Friedman *et al.*, 2001). Le classifieur *GBT* est une généralisation des algorithmes de *boosting* utilisant une fonction de coût. Ce classifieur est employé comme baseline pour ses performances ainsi que le peu de paramétrage nécessaire. Le second classifieur utilisé est un réseau de neurones artificiel, appelé *Multilayer Perceptron*(MLP) (Ruck *et al.*, 1990 ; Bastien *et al.*, 2012), composé de 3 couches dans cette expérience. Une couche d'entrée de 95 neurones artificiels, 32 neurones artificiels dans la couche cachée et 8 neurones dans la couche de sortie avec des sigmoïds puis softmax pour les fonctions d'activation. Enfin, ces réseaux utilisent la méthode de *dropout* pour la régularisation. Les deux classifieurs sont entraînés et évalués séparément avec chacun des 4 modèles Word2vec.

Résultats

Les précisions reportées dans le tableau 5 sont mesurées à la fois sur le corpus de dev. et le corpus de test avec respectivement les modèles *Skip-gram* et *CBOW*. Ce tableau montre que toutes les configurations testées voient leurs performances s'améliorer en utilisant l'information de distance contextuelle (*CCM*). Nous constatons que les résultats du modèle *Skip-gram* avec le *GBT* augmentent de 10 %, et que les résultats avec le *MLP* augmentent considérablement la précision avec un gain absolu de 20 % et un score maximum de 70 %. De même, les résultats utilisant les modèles *CBOW* voient leurs précisions doubler grâce à la pondération *CCM* que nous proposons. Le classifieur *GBT* obtient un score de 60 % et le *MLP* passe de 31 % à 71 % en termes de précision. Les résultats du *MLP* sont mesurés toutes les 10 époques d'apprentissage à la fois sur le corpus de dev. et le corpus de test. Ces résultats sont représentés dans le graphique 5. Nous remarquons que les modèles utilisant la pondération *CCM* obtiennent de meilleurs résultats et convergent plus rapidement que les autres modèles.

Ces expériences montrent qu'utiliser une pondération *CCM* dans un modèle Word2Vec produit des représentations avec une plus forte influence thématique. Elle

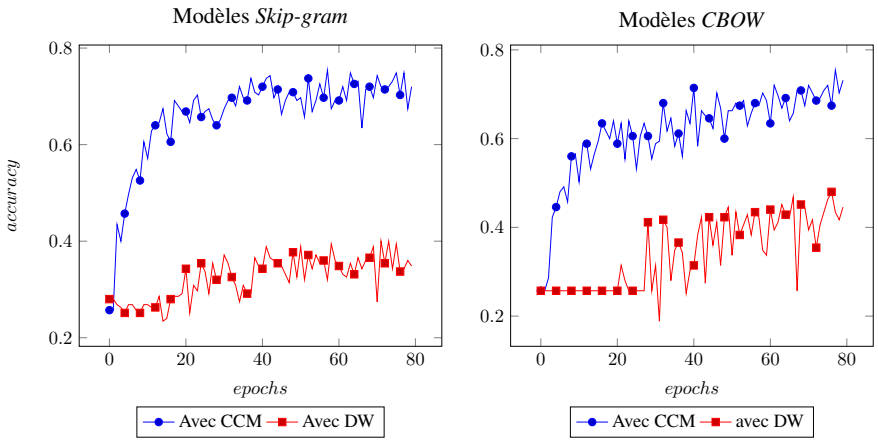

Figure 5. *Précision du classifieur MLP au fil des époques (0 à 80) pour les architectures Skip-gram et CBOW.*

Tableau 5. *Précision de la classification (%) sur la classification de thèmes en utilisant 2 classifiers différents (GBT et MLP).*

	Skip-gram		*CBOW*	
	Dev	Test	Dev	Test
GBT avec DW	39	42	28	27
GBT avec CCM	56	52	66	60
MLP avec DW	50	50	41	37
MLP avec CCM	75	**70**	74	**71**

montrent aussi que plus le contexte est important, plus l'information contenue dans la pondération *CCM* a un effet bénéfique. De plus, les réseaux de neurones apprenant à projeter les mots dans un espace multidimensionnel ont besoin d'un plus grand nombre de neurones artificiels pour capturer l'information supplémentaire contenue par la pondération *CCM*. Cette information apporte des modèles améliorés, comme le montre le "test sémantique et syntaxique" (cf. section 3.2), et fournit aux outils de classfication des caractéristiques plus adaptées pour la classification thématique de données textuelles.

4. Conclusion

La fenêtre de contexte dynamique proposée de manière classique utilise les mots, représentés par des sacs-de-mots binaires, tout en ignorant aléatoirement les mots en bordure de contexte. Dans ce contexte, tous les mots sont traités de façon égale. Cet article propose une alternative originale sous la forme d'une fonction de pondération s'appuyant sur les modèles de contextes continus capable de préserver les relations distantes. Cette méthode a été évaluée au moyen d'un test de similarité syntaxique

et sémantique, où un gain de 7 % a été observé, ainsi qu'une tâche de classification thématique de dialogues apportant un gain de plus de 20 %. Ces expériences ont aussi montré que les modèles intégrant notre proposition de pondération du contexte continu des mots est utile pour la classification thématique de documents textuels. Nous prévoyons d'étendre ce travail en étudiant l'impact de différents types de fonctions de pondération du contexte continu des mots et en ajoutant la même information pour d'autres formes de représentations distribuées.

5. Bibliographie

Bastien F., Lamblin P., Pascanu R., Bergstra J., Goodfellow I. J., Bergeron A., Bouchard N., Bengio Y., « Theano : new features and speed improvements », , Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

Bechet F., Maza B., Bigouroux N., Bazillon T., El-Beze M., De Mori R., Arbillot E., « DECODA : a call-centre human-human spoken conversation corpus. », *LREC*, p. 1343-1347, 2012.

Bengio S., Heigold G., « Word embeddings for speech recognition », *Proceedings of the 15th Conference of the International Speech Communication Association, Interspeech*, 2014.

Bengio Y., Ducharme R., Vincent P., « A Neural probabilistic language model », *Journal of Machine Learning Research*, vol. 3, p. 1137-1155, 2003.

Bigot B., Senay G., Linares G., Fredouille C., Dufour R., « Combining Acoustic Name Spotting and Continuous Context Models to improve Spoken Person Name Recognition in Speech », *Interspeech*p. 2539-2543, 2013.

Blei D. M., Ng A. Y., Jordan M. I., « Latent dirichlet allocation », *the Journal of machine Learning research*, vol. 3, p. 993-1022, 2003.

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., Kuksa P., « Natural language processing (almost) from scratch », *The Journal of Machine Learning Research*, vol. 12, p. 2493-2537, 2011.

Do Q.-K., Allauzen A., Yvon F., « Modèles de langue neuronaux : une comparaison de plusieurs stratégies d'apprentissage », *TALN 2014*, 2014.

Friedman J., Hastie T., Tibshirani R., *The elements of statistical learning*, vol. 1, Springer series in statistics Springer, Berlin, 2001.

Linares G., Nocéra P., Massonie D., Matrouf D., « The lia speech recognition system : from 10xrt to 1xrt », *Text, Speech and Dialogue*, Springer, p. 302-308, 2007.

Mesnil G., Mikolov T., Ranzato M., Bengio Y., « Ensemble of Generative and Discriminative Techniques for Sentiment Analysis of Movie Reviews », 2015.

Mikolov T., Corrado G., Chen K., Dean J., « Efficient Estimation of Word Representations in Vector Space », *Proceedings of the International Conference on Learning Representations (ICLR 2013)*p. 1-12, 2013a.

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., « Distributed representations of words and phrases and their compositionality », *Advances in Neural Information Processing Systems*, p. 3111-3119, 2013b.

Morchid M., Dufour R., Bouallegue M., Linarès G., « Author-Topic based Representation of Call-Center Conversations », *International Spoken Language Technology Workshop (SLT) 2014*, IEEE, 2014a.

Morchid M., Dufour R., Bousquet P.-M., Bouallegue M., Linarès G., De Mori R., « Improving Dialogue Classification using a Topic Space Representation and a Gaussian Classifier based on the Decision Rule », *International Conference on Acoustic, Speech and Signal Processing (ICASSP) 2014*, IEEE, 2014b.

Morchid M., Dufour R., Linarès G., Hamadi Y., « Latent Topic Model based Representations for a Robust Theme Identification of Highly Imperfect Automatic Transcriptions », *International Conference on Intelligent Text Processing and Computational Linguistics (CICLing) 2015*, 2015.

Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay E., « Scikit-learn : Machine Learning in Python », *Journal of Machine Learning Research*, vol. 12, p. 2825-2830, 2011.

Ruck D. W., Rogers S. K., Kabrisky M., Oxley M. E., Suter B. W., « The multilayer perceptron as an approximation to a Bayes optimal discriminant function », *Neural Networks, IEEE Transactions on*, vol. 1, n° 4, p. 296-298, 1990.

Sahlgren M., « The distributional hypothesis », *Italian Journal of Linguistics*, vol. 20, n° 1, p. 33-54, 2008.

Vaswani A., Zhao Y., Fossum V., Chiang D., « Decoding with Large-Scale Neural Language Models Improves Translation. », *EMNLP*, Citeseer, p. 1387-1392, 2013.