Exploitation de documents médicaux par les techniques d'embedding : application au typage automatique de documents

Mikaël Dusenne[1,2], Julien Grosjean[1,2], Lina Soualmia[2,3], Clément Massonnaud[1], Stéphane Canu[3], Stefan J. Darmoni[1,2]

[1] DÉPARTEMENT D'INFORMATION ET D'INFORMATIQUE BIOMÉDICALE, Centre Hospitalo-Universitaire de Rouen, Rouen, France
mikael.dusenne@chu-rouen.fr

[2] UMR_S 1142, F-75006 Sorbonne Univ., Inserm LIMICS, Paris, France

[3] LITIS EA 4108, F-76000, Normandie Univ., UNIROUEN, UNIHAVRE, INSA Rouen, Rouen, France

Résumé : **Introduction :** Les documents non structurés contiennent la majeure partie de l'information utile d'un dossier patient informatisé. Les techniques de traitement automatique de la langue permettant d'exploiter ces données sont en constante évolution. Les techniques d'embedding transforment des concepts non structurés en un espace vectoriel multidimensionnel. Il est ensuite possible d'exploiter les données sous forme numérique afin d'accomplir différentes tâches, supervisées ou non (classification, création de clusters sémantiques, quantification de similarité sémantique).
Méthodes : Nous étudions dans cet article une des possibles applications des techniques d'embeddings aux documents médicaux non structurés. En utilisant plus de 15 millions de documents médicaux disponibles dans l'entrepôt de données cliniques du Centre Hospitalo-Universitaire de Rouen, nous créons des *document embeddings* avec doc2vec après avoir identifié les hyper-paramètres optimaux sur un sous-ensemble de documents. Un réseau de neurones est ensuite entraîné pour prédire le type de document en fonction de l'embedding qui le représente. Nous analysons les performances de classification en utilisant le pourcentage de bonne classification.
Résultats : Le choix des hyper-paramètres fait grandement varier la qualité des embeddings générés, et ces paramètres semblent être très dépendants des données utilisées. La classification des types de documents présentait un pourcentage de classification correcte de 99,07% sur 110 481 documents de l'ensemble de test.
Conclusion : Les premiers résultats obtenus montrent que les techniques d'*embedding* semblent offrir des avantages supérieurs aux autres méthodes de traitement automatiques de la langue, et permettent de répondre à des problématiques nouvelles.
Mots-clés : Apprentissage automatique, Apprentissage profond, Réseaux de neurones, Traitement automatique de la langue

1 Introduction

L'exploitation de données produites par les hôpitaux peut être très utile dans un contexte de recherche de médicale et de construction d'outils d'aide à la prise en charge des patients. Les données pertinentes d'un entrepôt de données de santé (EDS) sont représentées à 80% sous forme non structurée. (Raghavan *et al.*, 2014) ont par ailleurs montré que les données non structurées étaient essentielles pour répondre aux critères d'inclusion des études cliniques dans 59% à 77% des cas. Afin de fournir des résultats intéressants face à diverses problématiques, il est nécessaire d'exploiter ces données de la façon la plus efficace possible.

Les techniques de traitement automatique de la langue (TAL) existent depuis de nombreuses années et abordent la tâche de différentes manières. Le principal inconvénient des techniques classiques est leur représentation peu efficace du texte, dont la très haute dimensionnalité résulte en une sparcité importante, et dont la nature ne permet pas de convoyer les liens sémantiques qui existent entre les mots d'un texte.

L'absence de technique permettant de représenter des mots sous forme numérique de façon efficace a été un frein majeur au développement du TAL.

Les techniques de *word embedding* (Mikolov *et al.*, 2013) consistent en un apprentissage semi supervisé de données non structurées, et transformant chaque mot en un vecteur de

nombres réels. Ces vecteurs ont une dimension ne dépendant pas de la taille du vocabulaire, et seront généralement bien plus denses que les données utilisées dans les algorithmes de TAL reposant sur le concept de sac de mots. De plus, les vecteurs générés conservent une valeur sémantique, et les mots sémantiquement semblables seront proches les uns des autres dans l'espace vectoriel. Cette dernière propriété est absente des algorithmes classiques, et offre la possibilité d'effectuer, par le biais d'opérations vectorielles, des transformations et rapprochements sémantiques, permettant de réaliser à la fois des tâches de classification, supervisées, et des tâches non supervisées telles que la création de clusters sémantiques.

Les implémentations des *word embeddings* sont nombreuses. Word2vec (Mikolov *et al.*, 2013), développée en 2013, est la plus connue. Elle était la première implémentation facilement utilisable et fournissant des résultats surpassant l'état de l'art sur les tâches évaluées.

Depuis, de nombreuses autres implémentations ont été créées, abordant la façon de générer les vecteurs de façons variées. De plus, la notion d'embedding peut être considérée de façon plus abstraite et par exemple doc2vec (Le & Mikolov, 2014), développé en 2014, permet de générer un vecteur par document (et non pas par mot). Cela permet donc d'appliquer les techniques d'embeddings et de regrouper des documents en fonction de leurs contenus sémantiques, afin de les catégoriser automatiquement, ou de retrouver les documents les plus proches d'un document donné.

(Dynomant *et al.*, 2019a) ont exploré doc2vec dans un contexte de littérature médicale, utilisant les résumés des articles issus de PubMed afin d'implémenter une fonctionnalité de recommandation d'articles similaires à un article donné. Bien qu'utilisant une ressource en langue anglaise et n'étant pas composée de documents médicaux, les résultats montrent que Doc2vec est un outil intéressant et mérite d'être exploité dans le contexte d'un EDS. Ces outils offrent des possiblités d'exploitation des données non structurées jusqu'ici impossibles à mettre en place, comme la catégorisation automatique de documents.

Les documents de santé peuvent être de différents types, parmi lesquels figurent : compte-rendu d'hospitalisation, compte-rendu d'acte, compte-rendu d'accouchement, ordonnance, compte-rendu de biologie. Ces types sont renseignés dans le système d'information manuellement lors de la saisie du document. Cependant, de nombreux documents du système d'information du CHU de Rouen ne sont pas typés. Le développement d'un outil capable de déterminer automatiquement le type d'un document permettrait d'améliorer la qualité des données du système d'information et de l'EDS, et d'éviter la saisie manuelle du type de futurs documents. Cette tâche supervisée est aussi un moyen d'évaluer objectivement la capacité des *embeddings* à exploiter des données non structurées pour accomplir une tâche concrète.

L'objectif de ce travail est d'explorer l'utilisation des *document embeddings* pour catégoriser automatiquement les différents types de documents médicaux non structurés issus d'un entrepôt de données de santé.

2 Méthodes

Le Centre Hospitalo-Universitaire (CHU) de Rouen dispose d'un entrepôt de données médicales contenant 15,7 millions de documents médicaux concernant deux millions de patients, issus de l'activité de l'hôpital entre 1992 et 2019.

Ces documents sont constitués entre autres de comptes-rendus hospitaliers, comptes-rendus de consultation, ordonnances, comptes-rendus d'actes, de chimiothérapie, de résultats de laboratoire, rédigés en langue française.

Notre travail, s'inscrivant dans le cadre d'une thèse de science débutée en Octobre 2019 et faisant suite à celle réalisée par Émeric Dynomant, consiste en l'étude de l'application des *embeddings* à plusieurs niveaux d'agrégation :

— *Word Embeddings* : création d'un annotateur sémantique hybride, utilisant l'annotateur sémantique déjà existant dans l'entrepôt (Sakji *et al.*, 2010), basé sur les techniques de sacs de mots. L'objectif est d'améliorer les performances de l'annotateur notamment au niveau des erreurs d'orthographes, nombreuses dans les documents, et des abréviations, nombreuses elles aussi et dont le sens dépend fortement du contexte (une même abréviation peut avoir un sens différent selon la spécialité).

— *Document Embeddings* : création de embeddings centrés sur les documents, permettant de classer automatiquement les différents types de documents (par exemple compte-rendu d'hospitalisation, compte-rendu d'acte, ordonnance, compte-rendu de biologie). La détermination du type de document permettra de compléter cette donnée manquante pour 4,6 millions de documents (16%) de l'EDS et donc d'améliorer la qualité des données.
— *Séjour embeddings* : l'agrégation des documents par séjour permettrait d'aider à la codification de la tarification à l'acte de l'hôpital grâce à l'implémentation d'un classifieur basé sur le diagnostic principal de la terminologie CIM-10, et des diagnostics secondaires.
— *Patient Embeddings* : la création d'un vecteur pour chaque patient permettrait de rapprocher automatiquement les patients ayant des antécédents médicaux similaires. Cela pourrait permettre d'implémenter des outils d'aide à l'inclusion dans des cohortes pour la recherche médicale, mais aussi des outils d'aide à la prise en charge en retrouvant les patients similaires à un patient donné.

La première tâche que nous avons réalisée et dont nous présentons les résultats dans cet article est la classification du type de document grâce aux *document embeddings*.

Les 15,6 millions documents de L'EDS du CHU de Rouen sont de dix types distinct. En dehors des 4,62 millions de documents non typés, on compte 5,61 millions de compte-rendus d'acte, 3,06 millions ordonnances, 2,02 millions compte-rendus de séjour, 1,97 millions compte-rendus opératoires, 72 209 compte-rendus de chimiothérapie, 41 879 compte-rendus de consultation, 25 917 compte-rendus de biologie, 19 525 documents à visée légale, 8 652 compte-rendus de soins intensifs post-opératoires, 2 783 compte-rendus d'accouchement.

L'utilisation de doc2vec nous permet, après entraînement sur le corpus, d'obtenir un vecteur pour chaque document. Il existe de nombreux paramètres d'entraînements, et leurs valeurs optimales dépendent largement du corpus utilisé et ne peuvent pas être déterminées à l'avance. Afin d'obtenir un espace vectoriel de qualité optimale, nous avons utilisé un premier ensemble de 100 000 documents pour effectuer une optimisation de dix hyper-paramètres principaux. Pour chaque combinaison de paramètres, nous avons effectué une tâche de classification basée sur la méthode des "K plus proches voisins" (classification Kppv). Ce type de classification présente l'avantage de ne reposer que sur la distance entre les points de l'espace vectoriel et exploite donc les vecteurs produits sans ajouter un classifieur qui nécessiterait un entraînement et ajouterait une complexité non désirée à cette étape de l'analyse. La qualité des vecteurs était donc évaluée par le taux de bonne classification du type de document.

Après avoir obtenu les meilleurs paramètres (la combinaison offrant les meilleures performances de classification par Kppv), nous avons séparé les documents restants en un ensemble d'entraînement des *embeddings* (90%), un ensemble d'entraînement du classifieur (9%) et un ensemble de validation (1%). Les *embeddings* nécessitent un large corpus pour être capable d'apprendre la meilleure représentation des documents, simplifiant au maximum la classification, ce qui explique l'allocation de 90% du corpus dans cette étape.

Le réseau de neurones du classifieur est implémenté avec la bibliothèque python keras [1]. Sa structure est gardée volontairement simple, car les vecteurs obtenus après entraînement devraient permettre une classification facile ne nécessitant pas un classifieur complexe. Pour la validation de l'entraînement, 10% des documents dédiés à l'entraînement de ce réseau de neurones seront utilisés.

Les calculs ont été réalisés sur un serveur hébergé au CHU de Rouen, disposant de 194 cœurs et 1To de RAM.

1. `https://keras.io/`

3 Résultats

3.1 Optimisation des hyper-paramètres

L'optimisation des hyper-paramètres a révélé une grande sensibilité des performances à certains paramètres d'entraînement. Notamment, l'algorithme PV-DBOW (Distributed Bag Of Words version of Paragraphs Vectors) donnait de meilleurs résultats que PV-DM (Distributed Memory version of Paragraphs Vectors), avec une exactitude de 0,85 (SD=0,11) et 0,68 (SD=0.07) respectivement (test de student apparié : p-value $<$ 0,0001). Ce résultat, allant à l'encontre des résultats retrouvés initialement par (Le & Mikolov, 2014), montre l'importance de ce travail d'optimisation pour chaque jeu de données. Le travail de (Dynomant *et al.*, 2019a) retrouvait les mêmes conclusions sur les meilleures performances de PV-DBOW.

L'augmentation du nombre d'époques permettait d'améliorer les performances de manière fiable, et nous n'avons pas retrouvé de surapprentissage.

Le meilleur modèle utilisait PV-DBOW et des vecteurs de 200 dimensions. Les autres paramètres étaient : window=8, negative=20, hs=0, min_alpha=0,0025, min_count=20, sample=0.

Étant donné que le nombre d'époques montrait une hausse des performances de façon fiable, et étant donné que le nombre de documents du jeu d'entraînement est beaucoup plus important, nous avons décidé d'entraîner le modèle final avec 200 époques. Les autres paramètres sont disponibles en annexe.

3.2 Entraînement des *embeddings*

L'entraînement a duré 6 jours et 7 heures.

Afin de pouvoir visualiser les *embeddings* en 3 dimensions, nous avons réalisé une réduction de dimensionnalité par T-SNE, une méthode de projection permettant de réduire le nombre de dimensions tout en conservant au mieux les distances entre les différents points d'un espace vectoriel.

Cette étape ne fait pas partie du processus de classification des documents, mais la visualisation des *embeddings* peut nous donner des indices sur leur structure. Afin de limiter le temps de calcul, nous avons sélectionné un sous-ensemble de 300 000 documents de façon aléatoire, et calculé la projection en trois dimensions de nos vecteurs de 200 dimensions, puis coloré chaque document selon son type (**Figure 1**).

On constate que les types de documents sont bien séparés en différents clusters, et donc que doc2vec semble avoir été capable d'identifier de façon automatique ces différents types de documents.

3.3 Entraînement du classifieur

Le réseau de neurones utilisé est constitué de deux couches denses de 32 unités avec une fonction d'activation ReLu (Rectified Linear unit), chacune suivie d'une couche de dropout à 29%. La dernière couche est une couche dense de 10 unités avec activation softmax permettant d'obtenir le type de document prédit, et a été entraîné pendant 200 époques. Il prend en entrée les différents vecteurs générés par doc2vec, puis retourne les types de documents prédits.

Il n'y avait pas de signe de surentraînement (pas de diminution du taux de bonne réponse (exactitude) de l'ensemble de validation), et l'évolution de l'exactitude était asymptotique et stabilisée à la fin de l'entraînement du réseau de neurones (**Figure 2**).

La visualisation des *embeddings* par T-SNE nous permettait de supposer qu'un classifieur simple devrait parvenir à exploiter les vecteurs facilement. En effet, nous n'avons pas obtenu de meilleurs résultats en augmentant le nombre de couches ou d'unités, et avons conservé le modèle le plus simple qui exploitait au mieux l'information des *document embeddings*.

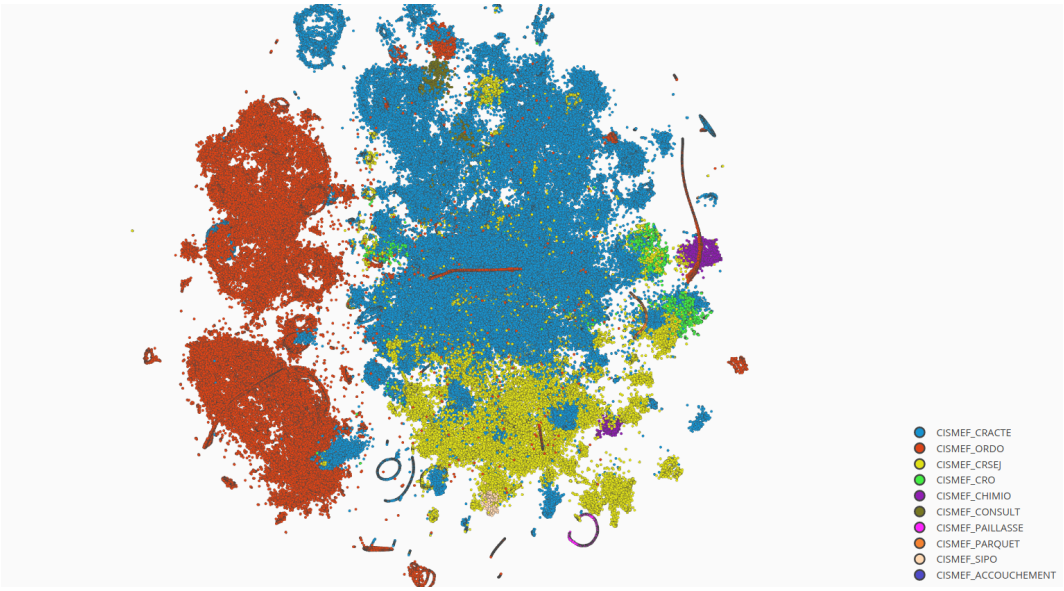

FIGURE 1 – Représentation 3D T-SNE des vecteurs de documents obtenus. Les couleurs superposées représentent les types de documents. On peut observer que les documents de même type se trouvent proches dans l'espace vectoriel, ce qui nous indique que doc2vec semble avoir été capable de séparer les documents de façon autonome.

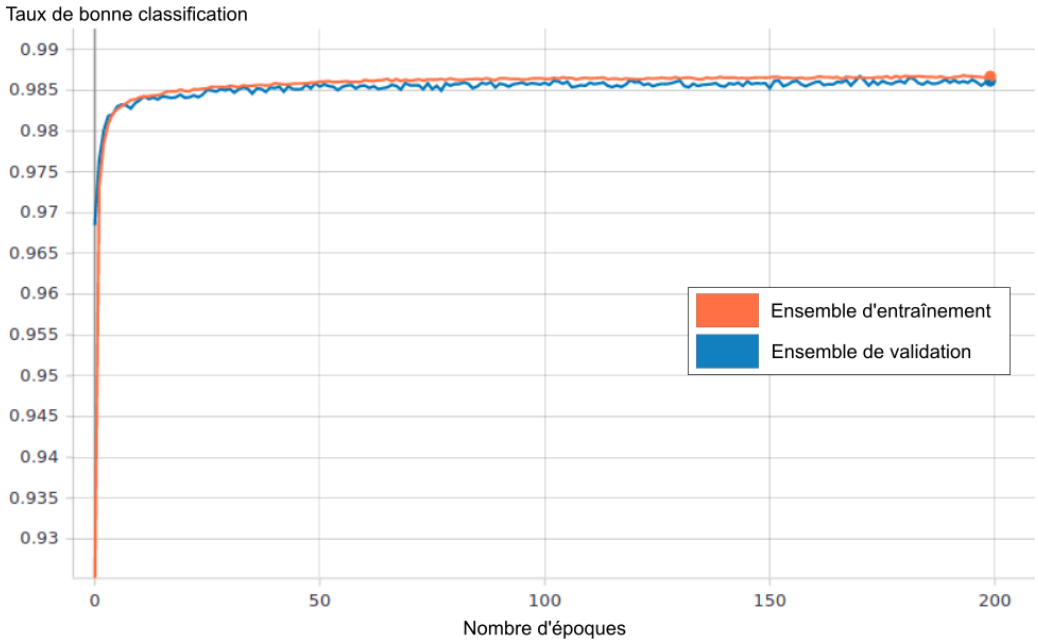

FIGURE 2 – Évolution de l'exactitude de l'ensemble d'entraînement et de validation, en orange et en bleu respectivement. En abscisse figure le nombre d'époques écoulées, en ordonnée l'exactitude.

3.4 Performances de classification

Le taux de bonne classification était de 99,07% sur les 110 481 documents évalués.
Le pourcentage de bonne classification était variable selon le type réel du document.

L'exactitude moyenne était de 97,54% (SD=2,35%) avec un minimum de 95,31% (61/64) pour les documents de soins intensifs post-opératoires, et un maximum de 100% (241/241) pour les documents issus des laboratoires de biologie.

Les principales erreurs étaient la prédiction "compte-rendu d'acte" alors qu'il s'agissait d'un compte-rendu de séjour, et inversement. La matrice de confusion des classifications est représentée sur la **Figure 3**.

Exactitude: 99.07 % (110481 Documents)

pred \ actual	ACCOUCHEMENT	CHIMIO	CONSULT	CRACTE	CRO	CRSEJ	ORDO	PAILLASSE	PARQUET	SIPO
ACCOUCHEMENT	25	0	0	0	0	0	0	0	0	0
CHIMIO	0	705	0	0	0	14	0	0	0	0
CONSULT	0	0	353	59	0	0	0	0	0	0
CRACTE	0	0	30	56001	36	162	31	0	1	0
CRO	0	0	0	10	1967	2	0	0	0	0
CRSEJ	1	24	0	601	4	19643	2	0	0	3
ORDO	0	0	0	36	0	2	30277	0	0	0
PAILLASSE	0	0	0	0	0	0	0	241	0	0
PARQUET	0	0	0	4	0	0	0	0	178	0
SIPO	0	0	0	0	0	8	0	0	0	61
Exactitude par type	96.15	96.71	92.17	98.75	98.01	99.05	99.89	100	99.44	95.31

FIGURE 3 – Table de confusion de la classification des documents de l'ensemble de validation. En colonne figurent les types de références dans l'entrepôt, en ligne les types prédits par le réseau de neurones.

Afin d'explorer les erreurs de classification, il est possible d'utiliser le niveau de confiance du réseau de neurones. Si le réseau de neurones fait plus d'erreurs de classification lorsqu'il présente un faible niveau de confiance dans le type prédit, l'exactitude sera plus élevée si on ne garde que les documents pour lesquels la confiance de prédiction était élevée.

Le niveau de confiance peut être approximé en utilisant les composantes de *dropout* du réseau de neurones (Gal *et al.*, 2016) : en effet, ces couches mettent à zéro une fraction de leurs neurones à chaque document fourni en entrée, de façon aléatoire. Elles sont habituellement utilisées afin de réduire le surapprentissage en servant de couche de régularisation, et dans ce cadre elles sont inactivées lorsque que l'entraînement est terminé.

Mais il est possible de garder ces couches actives après l'entraînement et d'introduire de ce fait une composante non déterministe dans les prédictions réalisées. Si l'on répète un grand nombre de fois une même prédiction, on peut agréger les résultats obtenus et calculer un indice de confiance (proportion de la classe prédite majoritaire / nombre de tentatives).

La **Figure 4** représente l'évolution de l'exactitude et du nombre de documents restants en fonction du seuil minimum de confiance jugé acceptable. Le niveau de confiance moyen était de 0,993 (DS = 0,039).

La confiance était en moyenne très élevée, et il y a très peu de documents ayant une confiance de prédiction inférieure à 98%. On constate cependant que l'exactitude augmente de 99,06% à 99,60% avec un seuil de confiance minimal fixé à 98%, et en ayant éliminé 7 326 (6,6%) documents ayant une confiance inférieure.

On peut cependant en conclure que pour de nombreux documents mal catégorisés, l'indice de confiance était élevé.

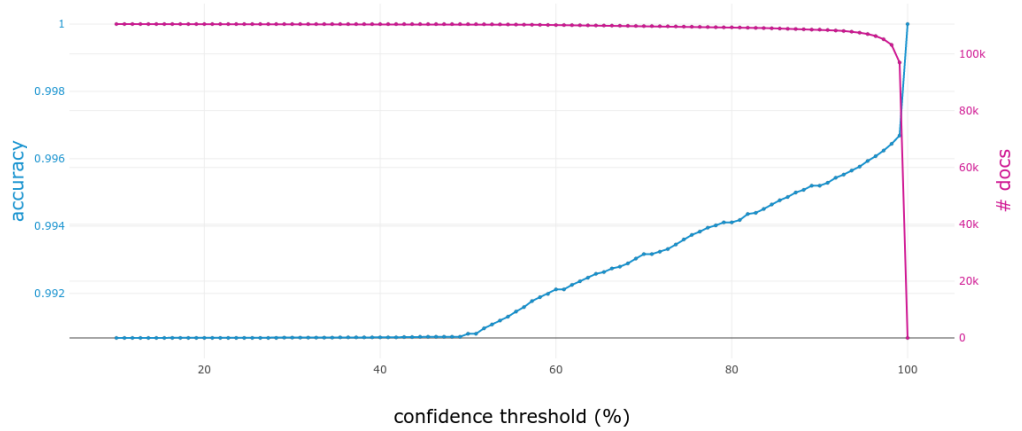

FIGURE 4 – Graphique de rejet du classifieur. En abscisse figure la certitude du réseau de neurones (de 0% à 100%) minimale pour calculer l'exactitude. En ordonnée en bleu (axe de gauche, représenté entre 99% et 100% d'exactitude pour la lisibilité de la figure) figure l'exactitude correspondante, et en rose (axe de droite) figure le nombre de documents considérés pour le calcul de l'exactitude. On constate que la confiance du modèle dans ses prédictions était forte, et il y a très peu de documents exclus avant le seuil de 98%. Au seuil de 99% de confiance, l'exactitude n'a que légèrement augmenté, indiquant que, lorsque le réseau était peu confiant, il se trompait plus fréquemment, mais que la plupart du temps le classifieur était certain de ses résultats.

4 Discussion

Les résultats de l'optimisation des hyper-paramètres pour l'entraînement de doc2vec nous permettent de constater l'importance de cette étape lors de l'utilisation des *embeddings* pour effectuer du traitement automatique de la langue.

Après avoir identifié les meilleurs paramètres pour notre jeu de données, nous avons pu obtenir de bons résultats de classification, avec un taux d'erreur inférieur à 1%, avec un classifieur relativement peu complexe.

Ceci confirme le fait que doc2vec est capable de créer des *embeddings* ayant des propriétés intéressantes et exploitables pour répondre à des problématiques concrètes.

Les erreurs de classifications existaient principalement entre compte-rendu d'acte et compte-rendu d'hospitalisation. Cela pourrait être causé par le lien fréquent qui existe entre ces documents, en effet, de nombreux patients ayant été hospitalisés ont aussi bénéficié d'un acte médical, et le déroulement de cet acte est repris dans le compte-rendu d'hospitalisation.

Les erreurs de typage peuvent être partiellement corrigées en excluant les documents pour lesquels la certitude du classifieur est plus faible. Cela n'empêche pas une mise en application pratique dans laquelle les documents non catégorisés seraient soumis à un évaluateur humain pour une classification manuelle, la majorité des documents étant traités automatiquement.

Pour les documents mal catégorisés malgré une grande certitude, on ne peut exclure la possibilité d'un mauvais typage manuel du document lors de sa création, et la suite de ce travail inclut une évaluation humaine des documents mal classifiés afin de déterminer précisément la cause de ces erreurs. Si ces erreurs sont avérées, notre outil pourrait ainsi être utilisé pour corriger les erreurs existantes dans l'EDS, et donc améliorer encore la qualité des données.

Ce premier travail nous conforte dans l'idée que cette technologie est capable d'apprendre sur des documents médicaux rédigés en français, et nous permet de continuer à l'explorer pour des tâches a priori plus complexes.

5 Conclusion et perspectives

Dans cet article nous avons étudié de façon approfondie une application des *embeddings* aux documents textuels médicaux, la classification des types de documents.

Nous avons démontré que les *embeddings* peuvent être utilisés sur des documents médicaux rédigés en français, et être utilisés avec une bonne fiabilité, qui sera confirmée par une évaluation manuelle d'un échantillon de documents.

Les autres applications des *embeddings* qui pourraient répondre à des problématiques de la recherche en santé, citées au début de cet article, seront implémentées lors de travaux futurs.

L'implémentation de l'annotateur hybride débutera par la création de *word embeddings*.

Pour la réalisation de ce travail nous nous baserons sur les résultats d'une étude précedemment réalisée au CHU de Rouen (Dynomant *et al.*, 2019b) , qui comparait word2vec, GloVe et Fasttext, des implémentation de *word embeddings* utilisant des algorithmes différents, sur un sous-ensemble des documents médicaux de l'EDS. Les résultats de cette étude montraient que word2vec avec l'architecture skip-gram présentait les meilleures performances, et nous débuterons l'implémentation en utilisant cette solution.

Cette étape nous permettra de plus d'explorer des implémentations récentes de *word embeddings*, notamment ALBERT (Lan *et al.*, 2019), une amélioration de BERT (Devlin *et al.*, 2018) développée en décembre 2019 par Google. Biobert (Lee *et al.*, 2019) est un modèle pré-entraîné à la fois sur un corpus généraliste (Wikipedia) et sur un corpus biomédical (abstracts de pubmed et texte des articles de PMC). Les performances de ce modèle sont intéressantes, mais le corpus d'entraînement n'étant pas en langue française, il ne peut pas être utilisé dans notre cas. Il sera intéressant d'utiliser des modèles pré-entraînés sur des corpus en langue française tels que camemBERT (Martin *et al.*, 2019). Ce modèle a été entraîné sur un très large corpus en français, sans orientation médicale. Il sera intéressant de comparer les performances à celles d'un modèle entraîné exclusivement sur des documents issus de l'EDS de Rouen.

La création de *Séjour embedding* sera différente du cas décrit dans cet article selon deux modalités : premièrement, un séjour est constitué d'un ensemble de documents médicaux produits au cours d'une hospitalisation. Il sera donc nécessaire de procéder à une agrégation afin de produire un vecteur par séjour. Une moyenne des différents vecteurs de documents est envisageable, mais il est aussi possible de concaténer les documents en amont afin d'obtenir un vecteur unique. Enfin, la variable à prédire est représentée par les codes de diagnostique de la CIM-10 : il existe donc un très grand nombre de modalités (environ 16 000). Ce facteur nécessitera de repenser la façon dont le classifieur est implémenté. La structure en arbre de la terminologie est un des éléments qui sera mis à contribution afin de rendre cette classification plus efficace.

Concernant la création de *patient embeddings*, il sera nécessaire de prendre en compte la dimension temporelle des différents séjours réalisés à l'hôpital pour ces patients. Une des pistes est l'utilisation de réseaux de neurones récurrents du type LSTM (Long Term - Short Term Memory).

Références

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding.

DYNOMANT E., DARMONI S. J., ÉMELINE LEJEUNE, KERDELHUÉ G., LEROY J.-P., LEQUERTIER V., CANU S. & GROSJEAN J. (2019a). Doc2vec on the pubmed corpus : study of a new approach to generate related articles.

DYNOMANT E., LELONG R., DAHAMNA B., MASSONNAUD C., KERDELHUÉ G., GROSJEAN J., CANU S. & DARMONI S. J. (2019b). Word embedding for the french natural language in health care : Comparative study. *JMIR medical informatics*, **7**.

GAL Y., YG279@CAM.AC.UK, GHAHRAMANI Z., ZG201@CAM.AC.UK & OF CAMBRIDGE U. (2016). Dropout as a bayesian approximation : Representing model uncertainty in deep learning.

LAN Z., CHEN M., GOODMAN S., GIMPEL K., SHARMA P. & SORICUT R. (2019). Albert : A lite bert for self-supervised learning of language representations.

LE Q. V. & MIKOLOV T. (2014). Distributed representations of sentences and documents. doc2vec.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2019). Biobert : pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv :1901.08746*.

MARTIN L., MULLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., ÉRIC VILLEMONTE DE LA CLERGERIE, SEDDAH D. & SAGOT B. (2019). Camembert : a tasty french language model.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space.

RAGHAVAN P., CHEN J. L., FOSLER-LUSSIER E. & LAI A. M. (2014). How essential are unstructured clinical narratives and information fusion to clinical trial recruitment ? *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, **2014**, 218–223.

SAKJI S., GICQUEL Q., PEREIRA S., KERGOURLAY I., PROUX D., DARMONI S. & METZGER M.-H. (2010). Evaluation of a french medical multi-terminology indexer for the manual annotation of natural language medical reports of healthcare-associated infections. *Studies in health technology and informatics*, **160**, 252–256.