Enrichissement linguistique et cognitif de modèles de langue contextualisés pour le domaine médical

Corentin Blanc

THESE de DOCTORAT DE
L'UNIVERSITE CLAUDE BERNARD LYON 1

Ecole Doctorale 341
Évolution, Écosystèmes, Microbiologie, Modélisation

Spécialité de doctorat :
Biostatistiques – Intelligence artificielle

Soutenue publiquement le 01/06/2023, par :
Corentin BLANC

Enrichissement linguistique et cognitif de modèles de langue contextualisés pour le domaine médical

---

Devant le jury composé de :

Chazard, Emmanuel PUPH Université de Lille	Président
Burgun, Anita PUPH Université Paris Descartes	Rapporteur
Névéol, Aurélie Directrice de Recherche CNRS	Rapporteur
Chazard, Emmanuel PUPH Université de Lille	Examinateur
Maucort-Boulch, Delphine PUPH Université Claude Bernard Lyon 1	Examinatrice
Michiels, Stefan Directeur de Recherche INSERM Université Paris Saclay	Examinateur
Roy, Pascal PUPH Université Claude Bernard Lyon 1	Directeur de thèse
Francis, Elie Directeur R&D Ever Team Software	Co-directeur de thèse
Jamal, Fadi Responsable Industriel izyCardio-Cardioparc	Invité
Wakim, Bechara Responsable Industriel Mediapps Innovation	Invité

Avant-propos

Le déchiffrement d'Enigma par Alan Turing au cours de la Seconde Guerre Mondiale. La victoire écrasante d'AlphaGo sur Lee Sedol en 2016 lors d'un match officiel de Go. La conception de l'agent conversationnel ChatGPT qui fascine le monde depuis des mois. Quel est le point commun entre tous ces progrès technologiques ? L'intelligence artificielle. Entre crainte et émerveillement, l'intelligence artificielle est désormais omniprésente au cœur de nos sociétés avec de nombreux domaines d'application comme les transports, la logistique ou encore la finance. Mais peut-elle un jour espérer se frayer un chemin dans le domaine très pointu qu'est la médecine où chaque mauvaise décision prise peut avoir des conséquences dramatiques et irréversibles pour le patient ? C'est là toute l'ambition du projet AI4Heart depuis sa création il y a cinq ans.

AI4Heart est né d'un consortium de plusieurs acteurs issus d'univers diamétralement opposés mais gravitant autour d'un objectif commun : l'amélioration de la prise en charge cardiologique en s'appuyant sur l'intelligence artificielle. La grande diversité des données de santé fait de AI4Heart un projet de grande envergure entremêlant de nombreuses branches de la recherche telle que l'apprentissage profond, le traitement automatique du langage naturel ou encore la vision assistée par ordinateur. Chaque acteur apporte ses propres connaissances et compétences au projet et se compte au nombre de trois.

L'éditeur de logiciels Everteam Software spécialisé dans la gouvernance de l'information, la gestion de contenu et l'archivage électronique. La maîtrise et l'intégration du traitement automatique du langage naturel dans plusieurs des produits phares d'Everteam Software par ses experts est indispensable au projet pour le traitement de données textuelles.

Acteur majeur de l'innovation en santé cardio-vasculaire, izyCardio propose des consultations nouvelle génération grâce à ses nombreux centres CardioParc disséminés sur le territoire Rhônalpin. En plus des connaissances médicales, la société izyCardio est le fournisseur de données du projet pour la conception des modèles d'intelligence artificielle.

Le Laboratoire de Biométrie et Biologie Évolutive (LBBE) expert de la modélisation mathématique et informatique en écologie évolutive, génomique et santé. En particulier, l'Équipe Biostatistiques Santé rattachée à cette unité et spécialisée dans la recherche en biostatistiques avec de nombreuses publications à son actif en fait un acteur essentiel.

Les travaux présentés dans ce manuscrit de thèse sont les premières pierres érigées dans le cadre du projet AI4Heart concernant la recherche en traitement automatique du langage naturel. Ils s'inscrivent dans une thèse CIFRE, sous la codirection du Pr Pascal Roy et d'Elie Francis, dont le sujet traite de l'enrichissement linguistique et cognitif de modèles de langue contextualisés pour le domaine médical. Ces travaux ont été financés par l'Association Nationale de la Recherche et de la Technologie (ANRT).

Remerciements

Paradoxalement, ces mots scellent trois années intenses de recherche et amorcent quelques heures de lecture pour les plus courageux. Cette thèse ne pourrait prendre sa forme actuelle sans la contribution, qu'elle soit évidente ou discrète, de nombreuses personnes que je tiens à remercier chaleureusement.

Et comment ne pas commencer par les personnes à l'origine du projet AI4Heart. Mes remerciements les plus chaleureux vont aux Drs Fadi Jamal et Bechara Wakim, ainsi qu'au Pr Pascal Roy, qui ont placé leur confiance en moi au cours de ces trois années pour ériger les fondations de ce projet ambitieux. Particulièrement, je tiens à exprimer ma gratitude envers mon directeur de thèse, le Pr Pascal Roy, pour son enthousiasme contagieux, sa bonne humeur quotidienne et les discussions, qu'elles soient d'ordre scientifique ou non, qui se sont avérées extrêmement enrichissantes.

Je tiens à exprimer ma profonde gratitude envers tous les membres éminents de mon jury de thèse, dont l'acceptation de mon invitation m'a honoré. Tout particulièrement, je souhaite adresser mes remerciements au Pr. Anita Burgun et au Dr. Aurélie Névéol, qui ont généreusement consacré leur temps à scruter chaque recoin de mon manuscrit, délivrant des retours d'une grande richesse. Leurs perspectives éclairantes ont ouvert de nouvelles voies de recherche pour mes futurs travaux, qui, je l'espère, seront très nombreux. Je tiens également à saluer les Prs. Emmanuel Chazard, Stefan Michiels et Delphine Maucort-Boulch, dont les échanges stimulants et les questions éclairées ont marqué ma soutenance d'une empreinte constructive et mémorable.

Un grand merci également aux membres de mon comité de suivi de thèse, à savoir les Prs Marc Cuggia et Stéfan Darmoni, ainsi que Dr. Grégoire Rey. Les contributions de ces différents comités ont indubitablement joué un rôle décisif dans l'orientation fructueuse de ma thèse jusqu'à sa forme actuelle. Et bien sûr, je ne saurais oublier de mentionner ma tutrice au sein de l'école doctorale, Dr. Caroline Leroux, dont le soutien m'a beaucoup aidé tout au long de ces trois années.

Je tiens à exprimer ma sincère gratitude envers l'ensemble de mes collègues chez Everteam Software pour leur accueil des plus chaleureux, leur constante bonne humeur et leur penchant pour les blagues toujours plus nombreuses. Plus particulièrement, je tiens à adresser mes remerciements à tous les membres du Lab, en commençant par Elie Francis et Thierry Guillotin, qui ont successivement assumé le rôle de co-directeurs de ma thèse. Leur mentorat avisé m'a guidé à travers cette première aventure professionnelle. N'oublions pas non plus Jean-Gaël Try et Johanna Simoens, qui apportent leur précieuse contribution au sein de cette équipe du Lab.

J'aimerais également exprimer ma gratitude envers toutes les personnes avec lesquelles j'ai eu le privilège de collaborer chez izyCardio. Une mention toute spéciale revient à Pierre Allilouch, dont le travail autour des données (recueil et étiquetage) a été un appui plus que précieux pour la composante applicative de ma thèse.

Je désire exprimer mes remerciements à l'ensemble de mes pairs au sein du service de biostatistiques du Laboratoire de Biométrie et Biologie Évolutive pour leur accueil. Une mention spéciale s'impose pour Jean Iwaz, dont l'assistance et les conseils avisés ont illuminé le chemin de la rédaction et des révisions de mes trois articles scientifiques. Et comment oublier Véronique Ficagna, dont l'aide a été essentielle dans la complexe chaîne de l'organisation de ma soutenance, une tâche plus ardue qu'il n'y paraît.

Et comment ne pas remercier Alexandre, ou plutôt, Dr. Bailly, mon partenaire de thèse qui a été à la fois un collaborateur scientifique et un ami. Collaborer avec toi durant ces trois années a été un réel plaisir, même si je dois admettre que certaines de tes blagues n'étaient pas toujours aussi drôles que prévu.

Clara, merci pour ton soutien inébranlable, tes encouragements et pour avoir toujours cru en moi tout au long de cette thèse. Merci pour ton avis et tes conseils toujours très éclairés et surtout d'avoir accepté de relire l'intégralité de mon manuscrit, ce qui n'a pas dû être une partie de plaisir.

Enfin, pour toute ma famille et tous mes amis, tout particulièrement mes parents et mes frères, un grand merci pour votre soutien indéfectible et votre présence le jour de ma soutenance. J'espère que désormais vous aurez une meilleure compréhension de ma vie professionnelle. Et si ce n'est pas le cas, j'espère au moins que les sièges étaient confortables ! Toutes ces années n'auraient jamais été possibles sans vous.

Production scientifique

Communications écrites

- Article en tant qu'auteur principal :

    1. Blanc C., Bailly A., Francis E., Jamal F., Roy P., Incorporating the ICD-10 Hierarchy into the Hypothesis Set and the Learning Algorithm for Hierarchical Text Classification. Soumis en mars 2023, en révision.

    2. Blanc C., Bailly A., Francis E., Guillotin T., Jamal F., Roy P., Corpus size considerations in continual pre-training of BioFlauBERT and BioCamemBERT. Soumis en mars 2023, en révision.

    3. Blanc C., Bailly A., Francis E., Guillotin T., Jamal F., Béchara W., Roy P., FlauBERT vs. CamemBERT : Understanding patient's answers by a French medical chatbot. *Artificial Intelligence in Medicine*, 2022.

    4. Bailly A., Blanc C., Guillotin T., Classification multi-label de cas cliniques avec CamemBERT. *Actes de la 28e conférence sur le Traitement Automatique des Langues Naturelles. Atelier Défi Fouille de Texte (DEFT)*, 2021.

- Article en tant que co-auteur :

    1. Bailly A., Blanc C., Francis E., Jamal F., Roy P., Importance of the number of classes and the proportion of labeled data in semi-supervised text classification. Soumis en février 2023, en révision.

    2. Bailly A., Blanc C., Francis E., Guillotin T., Jamal F., Roy P., Early vs. late data fusion in multimodal death-cause classification on text and structured data. Soumis en février 2023, en révision.

3. Bailly A., Blanc C., Francis E., Guillotin T., Jamal F., Béchara W., Roy P., Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Computer Methods and Programs in Biomedicine*, 2021.

Communications orales

1. Blanc C., Bailly A., Francis E., Guillotin T., Jamal F., Béchara W., Roy P., CamemBERT word-embedding for Information Extraction in a Biomedical Context. *International Society for Computational Biology (ISCB)*, 2021.
2. Blanc C., Bailly A., Francis E., Guillotin T., Jamal F., Béchara W., Roy P., FlauBERT vs. CamemBERT : Compréhension de la réponse du patient pour un chatbot français. *Les Mardis de la Commission Intelligence Artificielle. Hospices Civils de Lyon*, 2021.
3. Blanc C., Bailly B., Guillotin T., Classification multi-label de cas cliniques avec CamemBERT. *Traitement Automatique des Langues Naturelles. Atelier Défi Fouille de Texte (DEFT)*, 2021.

Table des matières

Liste des figures

Liste des tableaux

Chapitre 1

Introduction

À l'aube du 21ème siècle, la médecine vit une révolution sans précédent. Grâce aux nouveaux progrès technologiques, des quantités vertigineuses de données sont créées chaque jour si bien qu'en 2017 le volume de données de santé doublait tous les 73 jours [1]. Ces données peuvent être classées en deux catégories : les données structurées et non structurées. Une donnée structurée est une donnée précise, facilement exploitable et stockée dans un format prédéfini (e.g., pression artérielle ou taux de cholestérol d'un patient). Au contraire, une donnée non structurée est stockée dans son format d'origine sans jamais vraiment être exploitée après sa création (e.g., du texte brut comme des comptes-rendus médicaux ou des images comme des radiographies). Ces dernières pourraient offrir des possibilités immenses aux professionnels de santé si elles étaient correctement exploitées ; ce qui n'était malheureusement pas le cas en 2019 avec environ 80% des données de santé qui restaient non structurées [2]. L'exemple le plus frappant concerne les données textuelles pouvant contenir de nombreuses informations à propos des patients mais qui sont trop souvent ignorées ou abandonnées. C'est la raison pour laquelle les professionnels de santé se tournent peu à peu vers des outils de traitement automatique du langage naturel (TALN) afin d'exploiter cette mine d'or d'informations.

Le TALN est une branche pluridisciplinaire de la recherche à la croisée entre les sciences cognitives, l'intelligence artificielle, l'informatique et la linguistique. Cette branche vise à permettre aux ordinateurs de comprendre, d'interpréter et de produire du langage naturel de la même manière que les humains dans le but d'effectuer des tâches utiles du

quotidien. Généralement, le TALN se décompose en deux pans complémentaires : la compréhension et la génération du langage naturel. D'un côté, la compréhension du langage naturel consiste à extraire des informations à partir de données textuelles non structurées. Ce pan du TALN englobe des tâches telles que la reconnaissance d'entités nommées, la classification de textes, la traduction automatique et bien d'autres encore. D'un autre côté, la génération du langage naturel est l'art de produire du texte compréhensible par les humains à partir de données structurées ou non structurées. La synthèse de textes, la rédaction automatique, la paraphrase ou encore la simplification de textes sont des exemples de tâches de génération du langage naturel.

Au fil des années, le TALN a fortement évolué au point de se décrire en termes de trois vagues majeures (voir Figure 1.1) :

- Entre 1950 et 1990, la grande majorité des approches reposaient sur le TALN symbolique consistant à coder en dur un ensemble de règles complexes. L'exemple le plus connu était le système conversationnel ELIZA [3] capable de simuler une discussion avec un psychothérapeute en se basant principalement sur de la paraphrase. Ces approches étaient très spectaculaires à l'époque pour résoudre des problèmes étroitement définis mais manquaient de généralisation sur des problèmes beaucoup plus complexes.
- Entre 1990 et 2010, grâce à l'augmentation des données textuelles disponibles et de la puissance de calcul, les approches symboliques ont petit à petit laissé leur place à des approches basées sur l'apprentissage superficiel. L'idée était d'utiliser des méthodes mathématiques et statistiques afin de modéliser un problème bien défini en exploitant un corpus de données. Les modèles basés sur des chaînes de Markov étaient très utilisés afin de prendre en compte la contiguïté des mots au sein d'une phrase. Bien que se généralisant mieux que les approches symboliques, les approches statistiques absorbaient difficilement les énormes quantités de données disponibles et manquaient d'abstraction sur des problèmes très pointus.

- À partir des années 2010, l'avènement de l'apprentissage profond a ouvert la voie au TALN neuronal grâce à la notion de modèles de langue. De tels modèles étaient capables de calculer un plongement (i.e., représentation numérique abstraite encodée par les couches cachées d'un modèle de langue neuronal) pour chaque mot d'une phrase grâce à un pré-entraînement sur de grosses quantités de données textuelles. En 2013, la conception du modèle de langue Word2Vec [4] a fait grand bruit grâce à sa capacité à fournir des plongements de mots ayant une forme de similarité entre eux. En effet, les synonymes ou mots ayant une forte chance de se retrouver dans le même contexte avaient tendance à avoir des plongements similaires. Encore mieux, Word2Vec projetait les mots dans un espace vectoriel dans lequel certaines opérations avaient un sens notamment avec la très célèbre équation :

$$\overrightarrow{\text{Reine}} = \overrightarrow{\text{Roi}} - \overrightarrow{\text{Homme}} + \overrightarrow{\text{Femme}}$$

Cependant, Word2Vec et ses successeurs proches comme GloVe [5] souffraient de certaines limites. En plus d'une mauvaise gestion des phrases trop longues, leurs plongements étaient dits statiques ou non-contextuels. En d'autres termes, un mot avait toujours le même plongement, peu importe son contexte, et les homographes [1] ne pouvaient jamais être différenciés. En 2018, le modèle *Bidirectional Encoder Representations from Transformers* (BERT) [6] a totalement révolutionné le TALN. BERT était capable de produire des plongements de mots contextuels et d'encoder énormément d'informations grâce à une structure profonde basée sur des Transformeurs [7] ; devenant ainsi une référence dans le monde du TALN en repoussant l'état de l'art dans la quasi-totalité des tâches. De nombreuses extensions ont vu le jour dans une multitude de langues différentes, notamment comme pour le français avec FlauBERT [8] et CamemBERT [9].

1. Mots ayant la même écriture mais des significations différentes

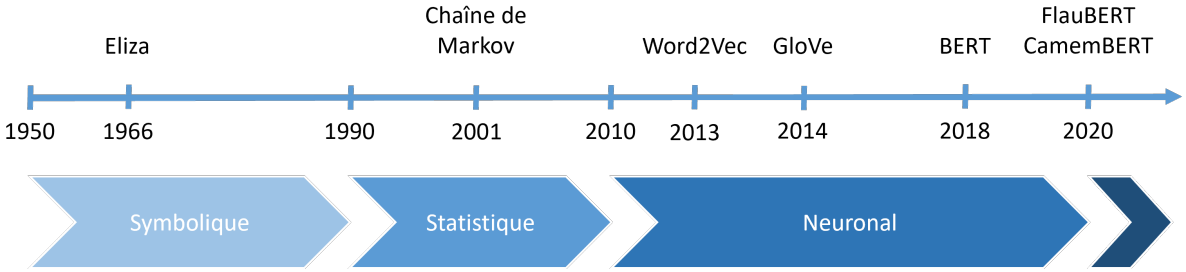


FIGURE 1.1 – Historique du Traitement Automatique du Langage Naturel.

Bien que très performants, ces modèles de langue n'en restent pas moins pré-entraînés sur des corpus de documents issus de tous horizons et peuvent présenter des lacunes sur des domaines bien spécifiques comme la médecine. En effet, la médecine possède son propre « langage » avec un vocabulaire unique et parfois une syntaxe et sémantique complètement différentes. Ainsi, est-il possible de donner à un modèle de langue la faculté de mieux comprendre ce langage médical ? De même que le langage, il existe bons nombres de connaissances propres à la médecine qui ne sont malheureusement aucunement prises en compte par les modèles de langue existants. N'est-il pas envisageable d'incorporer certaines de ces connaissances dans un modèle de langue afin de mieux performer sur des tâches TALN médicales ? Toutes ces interrogations ont mené à la réalisation de ce manuscrit de thèse qui se propose d'étudier les enrichissements linguistiques puis cognitifs [2] de modèles de langue contextuels à travers diverses d'applications de compréhension du langage naturel impliquant FlauBERT et CamemBERT.

Après une première description détaillée de la notion de modèles de langue contextuels (chapitre 2), le chapitre 3 met en situation FlauBERT et CamemBERT sur une tâche de TALN médicale avec comme toile de fond un chatbot faisant l'interface entre le patient et les professionnels de santé. Le but de cette mise en situation est d'étudier les performances de ces modèles sur le domaine médical, tout en les comparant et en déterminant la meilleure manière de les utiliser.

2. Qui concerne ici l'acquisition de connaissances

Le chapitre 4 introduit BioFlauBERT et BioCamemBERT, deux modèles de langue issus respectivement de FlauBERT et CamemBERT et capable de mieux appréhender le langage médical grâce à un enrichissement linguistique basé sur un corpus de données médicales. L'effet de la taille de ce corpus et différentes méthodes d'évaluation sont inspectés afin de contrôler au mieux ces enrichissements linguistiques et comparer les deux modèles résultants.

En plus de l'enrichissement linguistique, le chapitre 5 propose un enrichissement cognitif de BioFlauBERT afin de tenir compte de certaines connaissances médicales comme ici la classification internationale des maladies. Ces connaissances sont incorporées sous différentes formes à BioFlauBERT puis évaluées sur une tâche de prédiction hiérarchique de cause de décès.

Enfin, le chapitre 6 conclut ce manuscrit de thèse grâce à une application concrète effectuée pour l'acteur en cardiologie izyCardio. Les travaux présentés dans les chapitres précédents ont permis de concevoir un modèle capable de prédire de manière efficace les motifs de consultation en se basant sur des champs textuels préalablement remplis par les patients. À l'heure où ces lignes sont écrites, un modèle est en cours d'utilisation dans divers centres de cardiologie et d'autres versions sont à l'étude !

Chapitre 2

Modèle de langue contextualisé

Ces dernières années, les modèles de langue issus des Transformeurs sont devenus une référence dans le monde du TALN. Ces modèles se déclinent en plusieurs catégories :

- Ceux basés uniquement sur des encodeurs (BERT) qui sont efficaces pour des tâches de compréhension du langage naturel.
- Ceux basés sur des décodeurs (GPT, *Generative Pre-trained Transformer* [10–12]) ou sur les deux à la fois (BART, *Bidirectional and Auto-Regressive Transformers* [13]) qui sont privilégiés pour des tâches de génération de langage naturel.

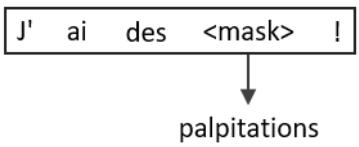
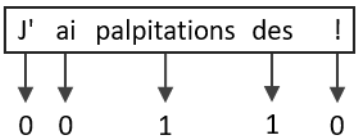
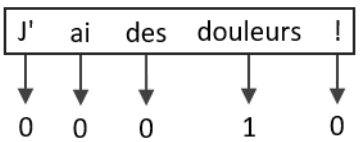
Du pré-entraînement, jusqu'à la structure, en passant par l'utilisation sur des tâches de compréhension du langage naturel, l'objectif de ce chapitre est de donner un bref éclaircissement de la notion de modèles de langue basés uniquement sur des encodeurs qui sera la seule catégorie traitée dans ce manuscrit de thèse.

Sommaire

2.1 Pré-entraînement

Le pré-entraînement d'un modèle de langue s'effectue par apprentissage auto-supervisé. Ce nouveau paradigme d'apprentissage, à mi-chemin entre l'apprentissage supervisé et non-supervisé, consiste à apprendre à partir de données pseudo-étiquetées qui sont automatiquement générées grâce à la définition d'une tâche de pré-entraînement. Dans le cadre du TALN, les tâches de pré-entraînement sont des tâches de modélisation du langage afin d'apprendre des connaissances linguistiques à partir d'un corpus de pré-entraînement constitué d'énormes quantités de textes. Une liste non exhaustive de tâches de modélisation du langage est donnée dans la Table 2.1. Enfin, pour effectuer le pré-entraînement, un réseau neuronal dont la structure est inhérente à la tâche de modélisation est connecté au modèle de langue puis l'ensemble des paramètres est ajusté de bout en bout.

TABLE 2.1 – Liste non exhaustive de tâches de modélisation du langage pour le pré-entraînement de modèles de langue. Pour chaque tâche, le pseudo-étiquetage correspondant est donnée sur la phrase « *J'ai des palpitations !* » à titre d'exemple.

Tâche de modélisation du langage	**Pseudo-étiquetage**
Modélisation du langage masqué : Prédiction de termes aléatoirement masqués dans une phrase en se basant sur le contexte non masqué.	
Modélisation du langage mélangé : Prédiction binaire pour chaque terme de la phrase afin d'identifier lesquels ont été aléatoirement mélangés.	
Modélisation du langage échangé : Prédiction binaire pour chaque terme de la phrase afin d'identifier lesquels ont été aléatoirement échangés par un autre mot.	

2.2 Structure

La structure d'un modèle de langue se décompose en deux étapes (Figure 2.1) : un prétraitement pour représenter numériquement la séquence d'entrée grâce à un générateur de jetons puis une contextualisation. Pour l'étape de contextualisation, les couches de plongement calculent des plongements initiaux non-contextuels. Puis une succession de L encodeurs contextualisent de plus en plus les plongements précédents jusqu'à l'obtention de plongements contextuels.

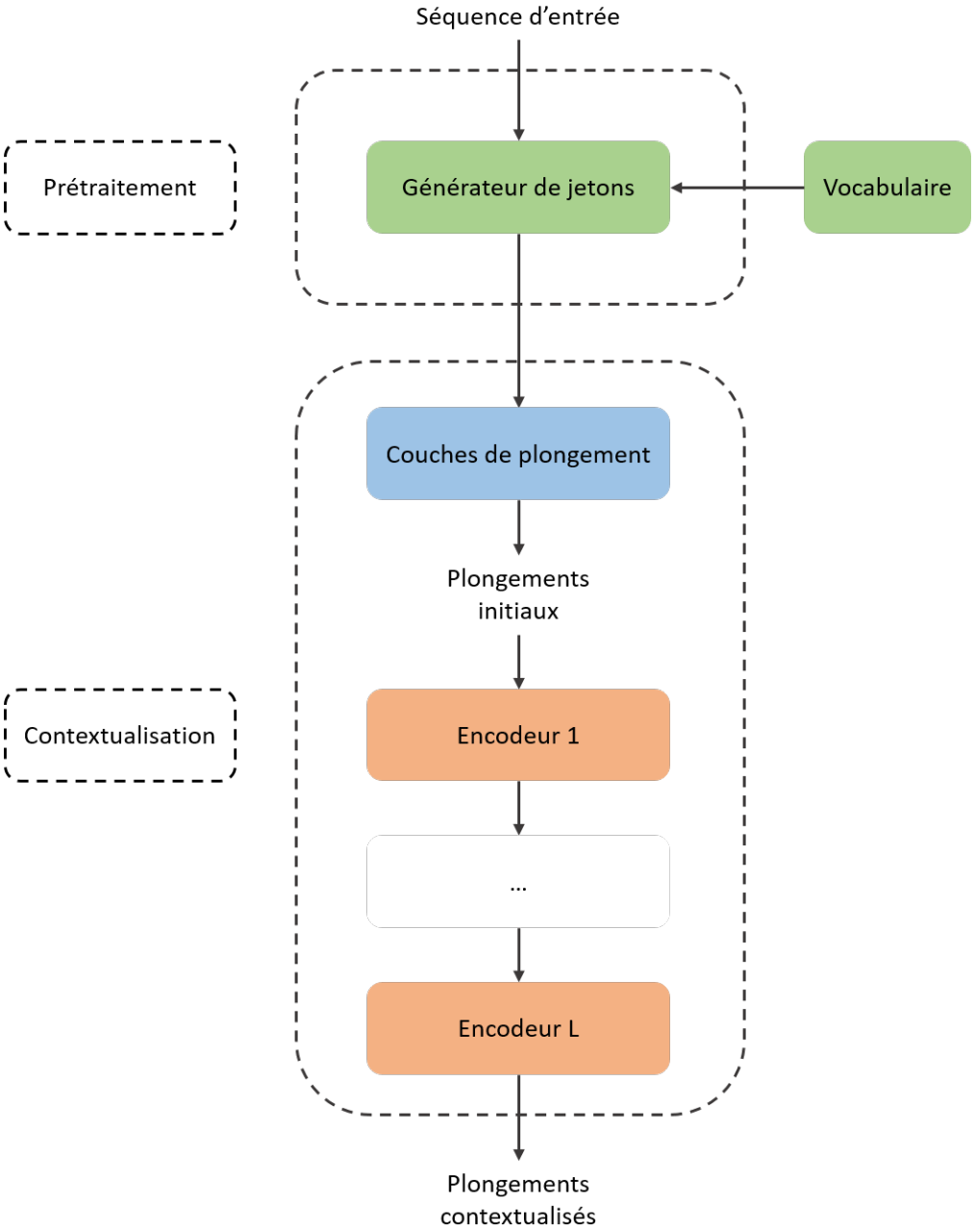


FIGURE 2.1 – Structure globale d'un modèle de langue. La séquence d'entrée est d'abord représentée numériquement grâce à un prétraitement et des couches de plongement puis une succession d'encodeurs calculent les plongements contextuels.

2.2.1 Prétraitement

Un modèle de langue prend en entrée une séquence composée d'un ou deux segments textuels (mot, fragment de phrase, phrase, paragraphe, etc.). Cette séquence est découpée en jetons (mots ou sous-mots) puis indexée à un vocabulaire préalablement défini grâce à un générateur de jetons. Le vocabulaire est composé de tous les caractères individuels de la langue naturel d'étude ainsi que des mots et sous-mots les plus fréquemment rencontrés. Ce choix permet de coller à l'idée fondamentale que les mots les plus fréquents devraient recevoir un indice unique, tandis que les mots rares ou hors vocabulaire devraient être décomposés en sous-mots. De plus, deux jetons artificiels sont ajoutés à la séquence : l'un au début pour obtenir un plongement de la séquence entière et l'autre à la fin de chaque segment pour les délimiter. Ainsi, une séquence d'entrée de longueur T (jetons artificiels compris) est représentée par (voir Figure 2.2) :

- un vecteur d'indices $x \in [\![1; V]\!]^T$ provenant d'un vocabulaire de taille V.
- un vecteur de positions $p \in [\![1; T_{max}]\!]^T$ permettant d'identifier la position et le voisinage de chaque jeton au sein de la séquence d'entrée de longueur maximale T_{max}. Cette limite a été fixée pour que le temps de réponse du modèle de langue soit raisonnable.
- un vecteur de segments $s \in [\![0; 1]\!]^T$ permettant d'identifier l'appartenance de chaque jeton à un segment de la séquence d'entrée (0 pour le premier, 1 pour le second).

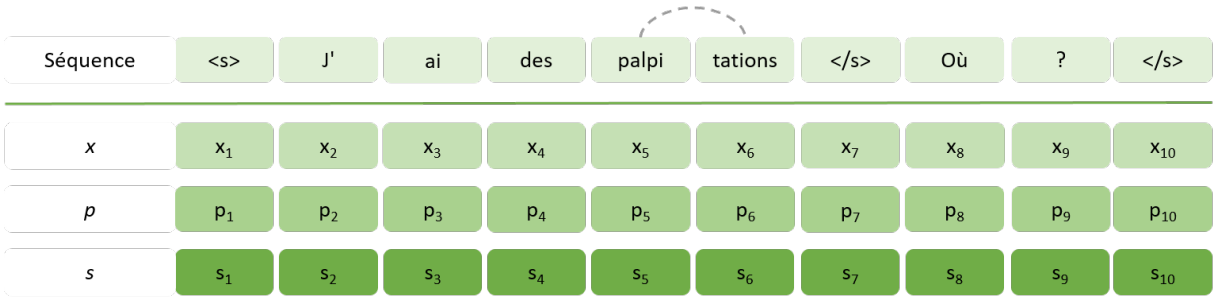


FIGURE 2.2 – Représentation d'une séquence d'entrée composée de deux segments. Dans cette séquence de longueur $T = 10$, le mot « *palpitations* » a été découpé en deux sous-mots « *palpi-* » et « *-tations* » grâce au générateur de jetons car il est hors vocabulaire. De plus, les jetons artificiels « *<s>* » et « *</s>* » ont été ajoutés pour respectivement obtenir un plongement de la séquence et délimiter les deux segments. Ainsi, les vecteurs x, p et s représentent respectivement les indices, positions et segments de chaque jeton.

2.2.2 Contextualisation

2.2.2.1 Couches de plongement

Les vecteurs d'indices, de positions et de segments précédemment obtenus par le générateur de jetons sont encodés à chaud [1] avant d'être linéairement projetés (sans biais) dans un espace de dimension D. Leur somme est finalement normalisée afin d'obtenir les plongements initiaux $Z^0 \in \mathbb{R}^{T \times D}$ donnés par (voir Figure 2.3) :

$$Z^0 \;=\; \text{Norm}\left(X \cdot W_x \;+\; P \cdot W_p \;+\; S \cdot W_s\right) \tag{2.1}$$

où $X \in \mathbb{R}^{T \times V}$, $P \in \mathbb{R}^{T \times T_{max}}$, $S \in \mathbb{R}^{T \times 2}$ sont les encodages à chaud des vecteurs d'indices, de positions et de segments ; $W_x \in \mathbb{R}^{V \times D}$, $W_p \in \mathbb{R}^{T_{max} \times D}$, $W_s \in \mathbb{R}^{2 \times D}$ les matrices de projection respectives ; et Norm la normalisation [14] définie par :

$$\text{Norm}(u) \;=\; \alpha \odot \frac{u - \mathbb{E}[u]}{\sqrt{\mathbb{V}[u] + \epsilon}} + \beta \quad \text{avec} \quad \begin{cases} \alpha, \beta \in \mathbb{R}^D & \text{les paramètres.} \\ \epsilon \in \mathbb{R} & \text{la stabilité numérique.} \\ \odot & \text{le produit d'Hadamard.} \end{cases} \tag{2.2}$$

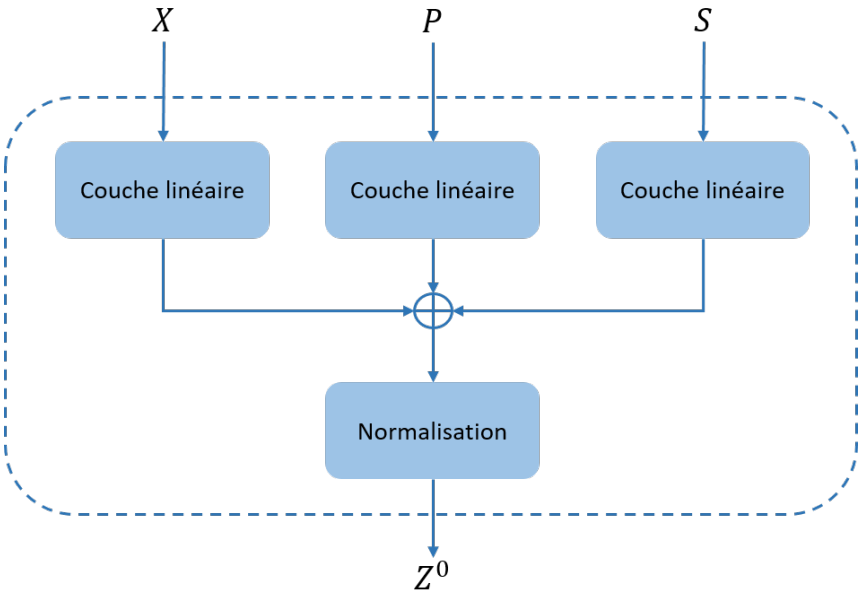


FIGURE 2.3 – Représentation des couches de plongement d'un modèle de langue. Chaque vecteur encodé à chaud (indice X, position P et segment S) est projeté linéairement puis leur somme (+) est normalisée pour obtenir les plongements initiaux Z^0.

1. Encodage binaire d'une variable catégorielle (1 si la catégorie est présente, 0 sinon.)

2.2.2.2 Encodeurs

Une fois encodés, les plongements initiaux sont contextualisés grâce à une succession de L encodeurs issus des Transformeurs. De tels encodeurs sont composés d'une couche d'auto-attention composée de H têtes qui déterminent sur quels jetons de la phrase se baser pour affiner la contextualisation des plongements puis d'un réseau de positions qui introduit une notion de non-linéarité. À noter que ces deux composants sont suivis d'une connexion résiduelle [15] puis d'une normalisation pour prévenir toute disparition ou explosion du gradient. La Figure 2.4 détaille le l-ème encodeur d'un modèle de langue qui calcule les plongements $Z^l \in \mathbb{R}^{T \times D}$.

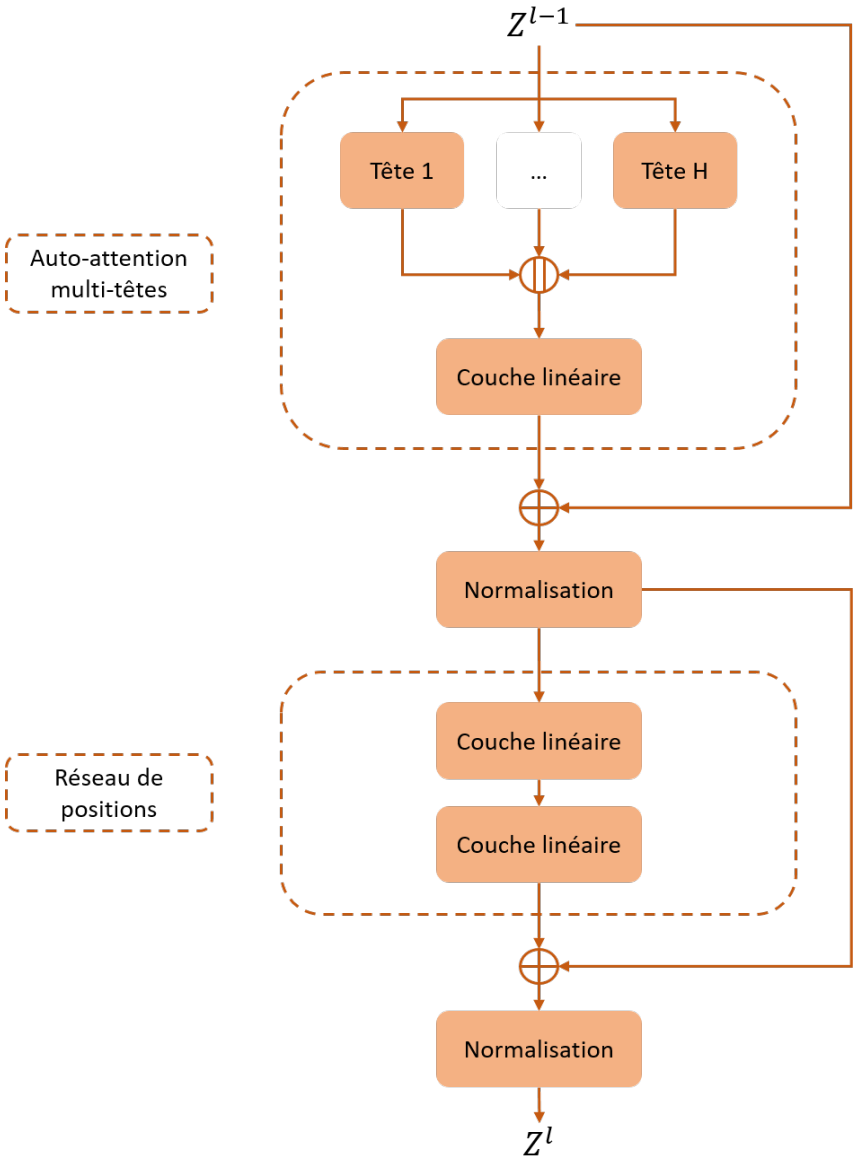


FIGURE 2.4 – Représentation d'un encodeur composé d'auto-attention multi-têtes et d'un réseau de positions. Ici, les opérateurs || et + représentent la concaténation et l'addition.

Auto-attention multi-têtes Pour la h-ème tête d'attention du l-ème encodeur, les plongements de la couche précédente sont transformés en trois matrices distinctes : les clés $K^{l,h} \in \mathbb{R}^{T \times D}$ référençant chaque jeton de la séquence, les valeurs $V^{l,h} \in \mathbb{R}^{T \times D}$ associées à chaque clé et les requêtes $Q^{l,h} \in \mathbb{R}^{T \times D}$ interrogeant chaque clé pour sélectionner celles qui correspondent le mieux à la demande. Clés, valeurs et requêtes sont données par :

$$K^{l,h} = Z^{l-1} \cdot W_k^{l,h} + b_k^{l,h} \tag{2.3}$$

$$Q^{l,h} = Z^{l-1} \cdot W_q^{l,h} + b_q^{l,h} \tag{2.4}$$

$$V^{l,h} = Z^{l-1} \cdot W_v^{l,h} + b_v^{l,h} \tag{2.5}$$

Ces trois matrices vont servir à calculer un produit scalaire d'attention $P^{l,h} \in \mathbb{R}^{T \times D}$ pour chaque tête :

$$P^{l,h} = \text{softmax}\left(\frac{Q^{l,h} \cdot {K^{l,h}}^T}{\sqrt{\frac{D}{H}}}\right) \cdot V^{l,h} \tag{2.6}$$

Enfin, l'ensemble des produits scalaires d'attention est concaténé puis linéairement projeté afin d'obtenir la sortie $A^l \in \mathbb{R}^{T \times D}$ de l'auto-attention multi-têtes :

$$A^l = \text{Norm}\left(Z^{l-1} + \left[P^{l,1} \,||\, \ldots \,||\, P^{l,H}\right] \cdot W_a^l + b_a^l\right) \tag{2.7}$$

Dans ces équations, $W_k^{l,h}$, $W_q^{l,h}$, $W_v^{l,h} \in \mathbb{R}^{D \times D}$ et $W_a^l \in \mathbb{R}^{HD \times D}$ sont les paramètres des différentes couches ; $b_k^{l,h}$, $b_q^{l,h}$, $b_v^{l,h}$ et $b_a^l \in \mathbb{R}^D$ les biais associés ; $||$ l'opérateur de concaténation ; et softmax la fonction permettant d'inférer une distribution de probabilités.

Réseau de positions Le réseau de positions est appliqué séparément et identiquement sur la sortie précédente de chaque jeton. Ce réseau est composé de deux couches linéaires associées à une activation non-linéaire de type GELU (*Gaussian Error Linear Units*) [16] afin de produire les plongements contextuels Z^l :

$$Z^l = \text{Norm}\left(A^l + \text{GELU}\left(A^l \cdot W_1^l + b_1^l\right) \cdot W_2^l + b_2^l\right) \tag{2.8}$$

où $W_1^l \in \mathbb{R}^{D \times D'}$, $W_2^l \in \mathbb{R}^{D' \times D}$ sont les paramètres du réseau de positions ; $b_1^l \in \mathbb{R}^{D'}$, $b_1^l \in \mathbb{R}^D$ les biais associés ; $D' \in \mathbb{R}$ la dimension temporaire des plongements ; et GELU la fonction d'activation définie par :

$$\text{GELU}(u) = \frac{1}{2}u\left(1 + \frac{2}{\sqrt{\pi}}\int_0^{\frac{u}{\sqrt{2}}} e^{-t^2}dt\right) \tag{2.9}$$

2.3 Utilisation

Une fois pré-entraîné, la stratégie la plus efficace pour utiliser un modèle de langue sur une tâche TALN est le réglage fin [6]. Très similaire à la procédure de pré-entraînement, cette stratégie consiste à connecter un réseau neuronal inhérent à la tâche puis d'ajuster très légèrement de bout en bout tous les paramètres grâce à un corpus relatif à la tâche (Figure 2.5). L'ajustement des paramètres doit impérativement être léger afin d'éviter le phénomène d'oubli catastrophique [17] qui se traduit par la perturbation ou suppression des connaissances pré-apprises par le modèle de langue.

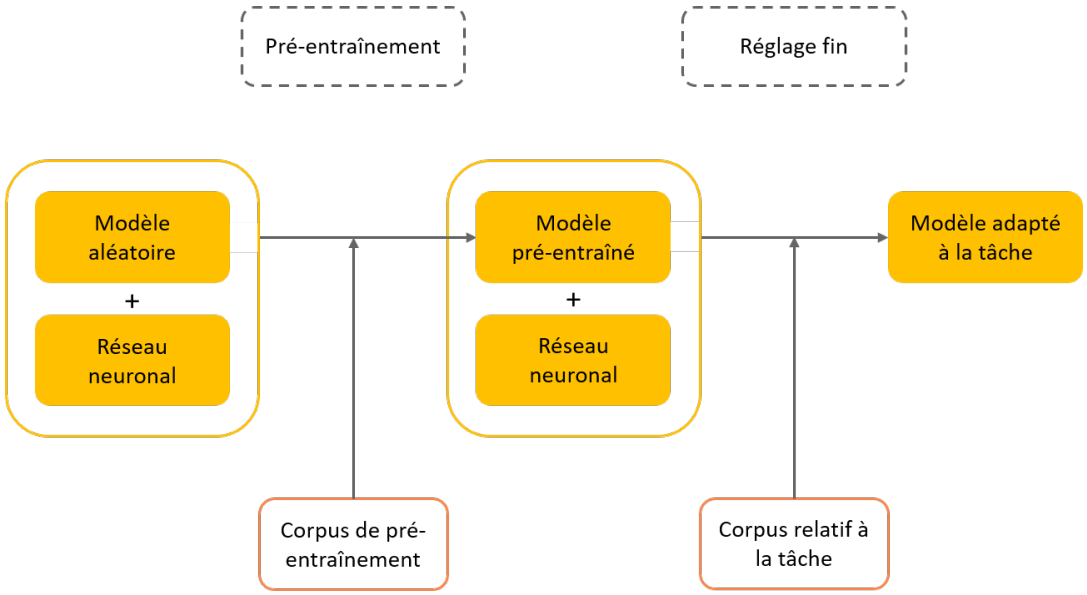


FIGURE 2.5 – Réglage fin d'un modèle de langue. Après une première étape de pré-entraînement, un réseau neuronal spécifique à la tâche est connecté au modèle de langue puis l'ensemble des paramètres est légèrement ajusté de bout en bout grâce à un corpus relatif à la tâche.

Chapitre 3

FlauBERT vs. CamemBERT

« *J'ai des palpitations* » ou encore « *Ma tension est anormalement élevée depuis plusieurs jours* » : deux phrases en apparence très simples à comprendre pour un humain mais qui peuvent poser énormément de problèmes à un chatbot interagissant avec des patients. Un tel chatbot servira de toile de fond à ce chapitre afin de comparer les deux modèles de langue français existants : FlauBERT [8] et CamemBERT [9]. Cette comparaison a fait l'objet d'une publication [18] dans la revue *Artificial Intelligence in Medicine* intitulée :

FlauBERT vs. CamemBERT :

Understanding patient's answers by a French medical chatbot

Sommaire

3.1 Contexte

Un chatbot médical est un programme informatique ayant la capacité de dialoguer avec un patient soit à l'oral soit par écrit. Sa fonction première est de récolter les informations d'un patient à travers une discussion auto-guidée. Ces informations seront par la suite utilisées par les professionnels de santé pour compléter les dossiers médicaux en amont et accélérer la prise en charge des patients. Un tel chatbot se compose de trois modules (voir Figure 3.1) :

- Un module de compréhension du langage naturel qui traite la déclaration du patient en créant une représentation formelle composée d'une intention et d'une ou plusieurs entités. L'intention correspond au type d'information présente dans la déclaration (e.g., un symptôme, une caractéristique sociodémographique, etc.), tandis qu'une entité est une donnée précise qu'un ou plusieurs mots ajoutent à l'intention de la phrase pour l'enrichir. Par exemple, la date de manifestation d'un symptôme est une entité qui contribue à l'intention symptôme. Un exemple de représentation formelle est donnée dans la Table 3.1.

TABLE 3.1 – Représentation formelle organisant la déclaration du patient « *J'ai eu de la tachycardie hier soir* » avec une seule intention et plusieurs entités.

Déclaration du patient	**Intention**	**Entités**
J'ai eu de la tachycardie hier soir !	Symptôme	Type (*tachycardie*) Date (*hier soir*)

- Un gestionnaire de dialogue qui oriente le dialogue en fonction des différentes intentions et entités extraites au fil de la conversation. En communication permanente avec une base de données contenant le dossier du patient, ce gestionnaire peut prendre la décision de demander des compléments, de nouvelles informations ou encore de stopper la conversation.
- Enfin, un générateur de texte qui produit une réponse en fonction de la décision préalablement prise par le gestionnaire de dialogue. La réponse générée est ensuite adressée au patient jusqu'à ce que le gestionnaire de dialogue prenne la décision de stopper la conversation.

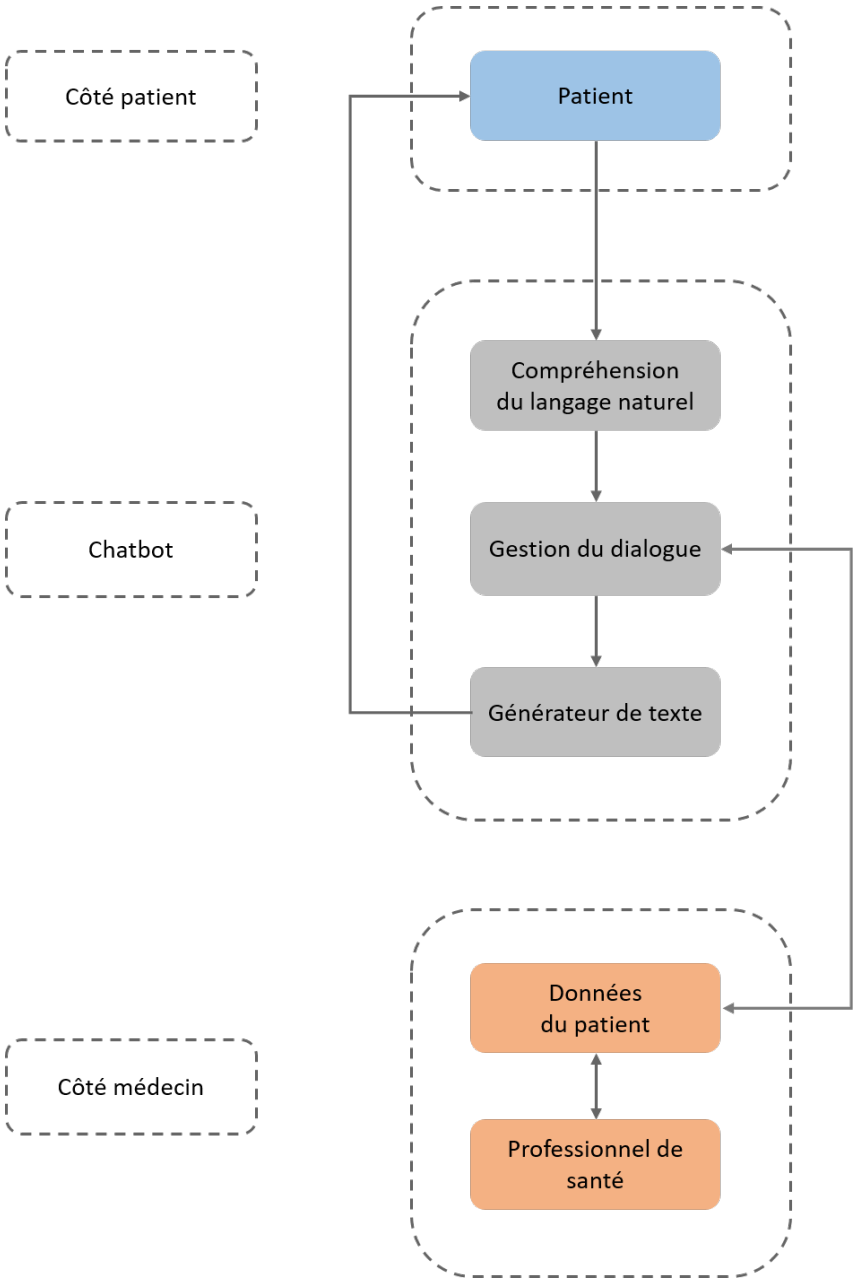


FIGURE 3.1 – Schéma d'un chatbot médical faisant l'interface entre un patient et les professionnels de santé.

Dans ce chapitre, les deux modèles de langue français FlauBERT et CamemBERT sont comparés sur une tâche de prédiction d'intentions et d'entités dans le cadre du module de compréhension du langage naturel d'un chatbot médical. Les performances en réglage fin des deux modèles sont comparées en testant différents réseaux neuronaux pour trouver la meilleure combinaison possible.

3.2 Méthodologie

3.2.1 Corpus

Un échantillon de 1 133 phrases a été sélectionné à partir du corpus CAS [19] ; une référence parmi les corpus médicaux français comprenant des milliers de rapports cliniques. Pour chaque phrase, une seule intention et une ou plusieurs entités ont été manuellement étiquetées selon la terminologie détaillée dans la Table 3.2. Les entités ont été étiquetées en utilisant le format BIO (*Beginning-Inside-Outside*). Dans ce format, le préfixe « *B-* » indique un jeton au début d'une entité, le préfixe « *I-* » un jeton au milieu ou à la fin d'une entité et le préfixe « *O* » un jeton n'appartenant à aucune entité. Enfin, le jeton artificiel « *</s>* », délimitant la fin des segments dans un modèle de langue, a servi à l'étiquetage de l'intention. À titre d'exemple, la phrase présentée dans la Table 3.1 (« *J'ai eu de la tachycardie hier soir ! </s>* ») est étiquetée : « *O O O O O B-TypePresent B-Date I-Date O Symptôme* ».

Table 3.2 – Terminologie des intentions et entités utilisées pour étiqueter l'échantillon de phrases issu du corpus CAS. Pour chaque intention, la distribution au sein du corpus est précisée, tandis que pour chaque entité la distribution relative à son intention est précisée.

Intentions et entités	**Description**	**Effectif (%)**
Patient		192 (17%)
Poids	Poids du patient	44 (14%)
Taille	Taille du patient	40 (12%)
Âge	Âge du patient	172 (53%)
Nom	Nom du patient	69 (21%)
Symptôme		203 (18%)
Type présent	Type de symptôme	377 (76%)
Date	Date d'apparition du symptôme	121 (24%)
Examen		566 (50%)
Type présent	Examen médical antérieur	639 (100%)
Antécédent		172 (15%)
Type présent	Type d'antécédent médical présent	218 (65%)
Type absent	Type d'antécédent médical absent	116 (35%)

3.2.2 Architecture proposée

3.2.2.1 Modèle de langue

Comme décrit dans le chapitre précédent, le modèle de langue calcule les plongements contextuels $Z^L = (z_t^L) \in \mathbb{R}^{T \times D}$ pour chaque jeton d'une phrase d'entrée $x = (x_t) \in [\![1; V]\!]^T$ issu de l'échantillon provenant du corpus CAS. Deux modèles de langue ont été comparés : FlauBERT et CamemBERT. Ces deux modèles ont des structures similaires mais diffèrent sur un certain nombre de points comme la taille du corpus de pré-entraînement ou encore les différentes variables mathématiques définies dans le chapitre 2. Les principales caractéristiques de FlauBERT et CamemBERT sont résumées dans la Table 3.3.

TABLE 3.3 – Principales caractéristiques de FlauBERT et CamemBERT.

Caractéristiques	**FlauBERT**	**CamemBERT**
Générateur de jetons	*Byte Pair Encoding* [20]	*SentencePiece* [21]
Taille du vocabulaire (V)	68 729	32 005
Taille du corpus	71 GB	132 GB
Nombre d'encodeurs (L)	12	12
Nombre de têtes d'attention (H)	12	12
Dimension des plongements (D)	768	768
Longueur maximale (T_{max})	512	512
Stabilité numérique (ϵ)	1e-12	1e-5
Dimension temporaire (D')	3 072	3 072
Nombre de paramètres	138M	110M

3.2.2.2 Réseaux neuronaux

À partir des plongements contextuels Z^L obtenus par le modèle de langue, quatre réseaux neuronaux ont été comparés afin d'inférer les scores $S = (s_t) \in \mathbb{R}^{T \times C}$ sur l'ensemble de taille C des intentions et entités possibles.

Couche linéaire La couche linéaire traite chaque plongement indépendamment des autres. Ainsi, le score inféré par la couche pour le t-ème jeton est donnée par :

$$s_t \;=\; z_t^L \cdot W_y + b_y \tag{3.1}$$

où $W_y \in \mathbb{R}^{D \times C}$ sont les paramètres de la couche et $b_y \in \mathbb{R}^C$ le biais associé.

Réseau récurrent Le réseau récurrent (RNN, *Recurrent Neural Network*) [22] traite chaque plongement un à un en tenant compte des précédents proches. Pour le t-ème jeton, la mémoire à court terme (ou état caché) du réseau $h_t \in \mathbb{R}^D$ est mis à jour grâce aux plongements z_t^L et la mémoire à court terme précédente h_{t-1} puis le score est calculé de la façon suivante :

$$h_t = \tanh\left(z_t^L \cdot W_z + b_z + h_{t-1} \cdot W_h + b_h\right) \tag{3.2}$$

$$s_t = h_t \cdot W_y + b_y \tag{3.3}$$

où W_z, $W_h \in \mathbb{R}^{D\times D}$ et $W_y \in \mathbb{R}^{D\times C}$ sont les paramètres du réseau ; b_z, $b_h \in \mathbb{R}^D$ et $b_y \in \mathbb{R}^C$ les biais associés ; et tanh la fonction tangente hyperbolique.

Réseau à mémoire à long et court terme Le réseau à mémoire à long et court terme (LSTM, *Long Short-Term Memory*) [23] traite chaque plongement un à un en tenant compte de tous les précédents. En plus d'une mémoire à court-terme $h_t \in \mathbb{R}^D$, un réseau LSTM intègre une mémoire à long-terme $c_t \in \mathbb{R}^D$. Les deux mémoires sont régies grâce à un mécanisme composé de trois portes permettant de réguler le flux d'information. Pour le t-ème jeton, la porte d'oubli choisit les informations $f_t \in \mathbb{R}^D$ à supprimer de la mémoire à long terme :

$$f_t = \sigma\left(z_t^L \cdot W_{zf} + b_{zf} + h_{t-1} \cdot W_{hf} + b_{hf}\right) \tag{3.4}$$

$$c_t = f_t \odot c_{t-1} \tag{3.5}$$

La porte d'entrée choisit d'abord les nouvelles informations $i_t, g_t \in \mathbb{R}^D$ à introduire dans la mémoire à long terme précédemment réduite :

$$i_t = \sigma\left(z_t^L \cdot W_{zi} + b_{zi} + h_{t-1} \cdot W_{hi} + b_{hi}\right) \tag{3.6}$$

$$g_t = \tanh\left(z_t^L \cdot W_{zg} + b_{zg} + h_{t-1} \cdot W_{hg} + b_{hg}\right) \tag{3.7}$$

$$c_t = c_t + i_t \odot g_t \tag{3.8}$$

La porte de sortie choisit les nouvelles informations $o_t \in \mathbb{R}^D$ à introduire dans la mémoire à court-terme en se basant sur la mémoire à long-terme précédemment mise à jour :

$$o_t = \sigma\left(z_t^L \cdot W_{zo} + b_{zo} + h_{t-1} \cdot W_{ho} + b_{ho}\right) \tag{3.9}$$

$$h_t = o_t \odot \tanh(c_t) \tag{3.10}$$

Une fois les deux mémoires mises à jour grâce aux trois portes, le score pour le t-ème jeton est donné par :

$$s_t = h_t \cdot W_y + b_y \tag{3.11}$$

Dans ces équations, W_{zf}, W_{hf}, W_{zi}, W_{hi}, W_{zg}, W_{hg}, W_{zo}, $W_{ho} \in \mathbb{R}^{D\times D}$ et $W_y \in \mathbb{R}^{D\times C}$ sont les paramètres du réseau ; b_{zf}, b_{hf}, b_{zi}, b_{hi}, b_{zg}, b_{hg}, b_{zo}, $b_{ho} \in \mathbb{R}^D$ et $b_y \in \mathbb{R}^C$ les biais associés ; $\odot$ le produit terme à terme ; et σ la fonction sigmoïde.

Réseau bidirectionnel à mémoire à long et court terme Le réseau bidirectionnel à mémoire à long et court terme (BiLSTM, *Bidirectional Long Short-Term Memory*) [24] va encore plus loin qu'un réseau LSTM car il traite chaque plongement un à un en se basant sur la totalité des autres plongements. Pour ce faire, un second LSTM est mis en parallèle pour traiter la phrase en sens inverse. Classiquement, le premier LSTM est dit à sens avant et le second à sens arrière. Les équations pour le sens avant sont données par :

$$f_t^f = \sigma\left(z_t^L \cdot W_{zf}^f + b_{zf}^f + h_{t-1}^f \cdot W_{hf}^f + b_{hf}^f\right) \tag{3.12}$$

$$i_t^f = \sigma\left(z_t^L \cdot W_{zi}^f + b_{zi}^f + h_{t-1}^f \cdot W_{hi}^f + b_{hi}^f\right) \tag{3.13}$$

$$g_t^f = \tanh\left(z_t^L \cdot W_{zg}^f + b_{zg}^f + h_{t-1}^f \cdot W_{hg}^f + b_{hg}^f\right) \tag{3.14}$$

$$o_t^f = \sigma\left(z_t^L \cdot W_{zo}^f + b_{zo}^f + h_{t-1}^f \cdot W_{ho}^f + b_{ho}^f\right) \tag{3.15}$$

et pour le sens arrière par :

$$f_t^b = \sigma\left(z_t^L \cdot W_{zf}^b + b_{zf}^b + h_{t-1}^b \cdot W_{hf}^b + b_{hf}^b\right) \tag{3.16}$$

$$i_t^b = \sigma\left(z_t^L \cdot W_{zi}^b + b_{zi}^b + h_{t-1}^b \cdot W_{hi}^b + b_{hi}^b\right) \tag{3.17}$$

$$g_t^b = \tanh\left(z_t^L \cdot W_{zg}^b + b_{zg}^b + h_{t-1}^b \cdot W_{hg}^b + b_{hg}^b\right) \tag{3.18}$$

$$o_t^b = \sigma\left(z_t^L \cdot W_{zo}^b + b_{zo}^b + h_{t-1}^b \cdot W_{ho}^b + b_{ho}^b\right) \tag{3.19}$$

Enfin, toutes les mémoires sont mises à jour puis le score pour le t-ème jeton est donné par :

$$c_t^f = f_t^f \odot c_{t-1}^f + i_t^f \odot g_t^f \tag{3.20}$$

$$c_t^b = f_t^b \odot c_{t-1}^b + i_t^b \odot g_t^b \tag{3.21}$$

$$h_t^f = o_t^f \odot \tanh(c_t^f) \tag{3.22}$$

$$h_t^b = o_t^b \odot \tanh(c_t^b) \tag{3.23}$$

$$s_t = \left[h_t^f \;||\; h_t^b\right] \cdot W_y + b_y \tag{3.24}$$

Toutes les matrices de paramètres et biais sont exactement les mêmes que dans le paragraphe 3.2.2.2 ; excepté $W_y \in \mathbb{R}^{2D \times C}$ pour gérer la concaténation des sorties des deux LSTMs.

3.2.2.3 Champ aléatoire conditionnel

Pour finir, un champ aléatoire conditionnel [25] a été ajouté en bout de chaîne afin de prédire conjointement les intentions et entités les plus probables tout en préservant au mieux le format BIO. Soit $y = (y_t) \in [\![1; C]\!]^T$ une séquence quelconque d'étiquettes possible pour x, l'énergie $E(y)$ induite pour cette séquence est donnée par :

$$E(y) = \sum_{t=1}^{T} s_{t,y_t} + a_{0,y_1} + a_{y_T,C+1} + \sum_{t=1}^{T-1} a_{y_t,y_{t+1}} \tag{3.25}$$

où $A = (a_{i,j}) \in \mathbb{R}^{(C+2)\times(C+2)}$ sont les paramètres du champ aléatoire. Ces paramètres correspondent au score de transition de la i-ème étiquette vers la j-ème, incluant deux transitions supplémentaires (en début et fin de séquence).

Finalement, un softmax sur l'ensemble des énergies de toutes les séquences d'étiquettes possibles $\mathcal{Y}$ pour x donne une probabilité pour la séquence y :

$$p_y = \frac{e^{E(y)}}{\sum_{\tilde{y} \in \mathcal{Y}} e^{E(\tilde{y})}} \tag{3.26}$$

La séquence d'étiquettes $\hat{y}$ obtenant la plus grande probabilité est inférée par le champ aléatoire conditionnel :

$$\hat{y} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}(p_y) \tag{3.27}$$

3.2.3 Fonction de perte

L'entropie croisée a été utilisée en tant que fonction de perte. Ici, elle évalue la discordance entre la distribution de probabilités prédites par le champ aléatoire conditionnel sur l'ensemble des séquences d'étiquettes possibles et celle observée dans le corpus :

$$L = -\sum_{\tilde{y} \in \mathcal{Y}} \mathbb{1}_{\tilde{y}} \cdot \log(p_{\tilde{y}}) \tag{3.28}$$

où $\mathbb{1}_{\tilde{y}}$ est une indicatrice (1 lorsque la séquence $\tilde{y}$ correspond à la séquence d'étiquette à prédire, 0 autrement) et $p_{\tilde{y}}$ la probabilité associée à la séquence $\tilde{y}$ par le modèle.

3.2.4 Hyperparamétrage

Pour entraîner conjointement l'architecture (modèle de langue, réseau neuronal et champ aléatoire conditionnel), plusieurs hyperparamètres doivent être configurés :

- Le nombre d'epochs (nombre de fois que le corpus est exploré durant l'apprentissage).
- La taille des lots (nombre de phrases propagées en même temps dans le modèle).
- L'écrêtage du gradient (réduction du gradient à chaque fois que sa norme dépasse une valeur préalablement fixée afin d'éviter toute explosion).

- L'algorithme de rétropropagation (algorithme itératif de mise à jour des paramètres grâce à la rétropropagation du gradient de la fonction de perte) implémenté avec :
  - ▷ Le taux d'apprentissage (magnitude de la mise à jour des paramètres du modèle) qui peut être variable au fil de l'apprentissage.
  - ▷ La décroissance des moments (pénalisation éventuelle des moments dans le cas d'algorithme avec momentum).
  - ▷ La décroissance des paramètres (pénalisation éventuelle des paramètres).
  - ▷ La stabilité numérique (valeur utilisée pour éviter toutes instabilités numériques dû à une division par zéro).

Afin de fixer une valeur pour ces hyperparamètres, une grille de recherche a été utilisée. Une telle grille consiste à sélectionner au préalable un ensemble de valeurs pour chaque hyperparamètre puis de tester toutes les combinaisons possibles afin de trouver la plus performante. Un récapitulatif de la combinaison optimale d'hyperparamètres utilisée dans ce chapitre est donné dans la Table 3.4.

TABLE 3.4 – Récapitulatif de l'hyperparamétrage utilisé pour entraîner conjointement le modèle de langue, le réseau neuronal et le champ aléatoire conditionnel pour la prédiction d'intentions et d'entités dans le cadre d'un chatbot médical.

Hyperparamètre	**Valeur**
Nombre d'epochs	10
Taille des lots	1
Écrêtage du gradient	1,0
Algorithme d'apprentissage	AdamW [26]
Taux d'apprentissage initial	2e-5
Variation du taux d'apprentissage	Linéaire
Taux d'apprentissage final	0,0
Stabilité numérique	1e-8
Décroissance du premier moment	0,9
Décroissance du second moment	0,999
Décroissance des paramètres	0,01

3.2.5 Critères d'évaluation

L'évaluation était portée par une méthode de bootstrap [27] consistant à tirer au hasard des phrases avec remise dans l'échantillon décrit dans la section 3.2.1 pour constituer un corpus d'entraînement. Le nombre de tirage effectué est égal à la taille de l'échantillon de phrases ; fixant constante la taille du corpus d'entraînement (1 133 phrases). Les phrases non tirées au sort constituaient le corpus de validation (théoriquement, environ 37% des phrases soit 419). Ce bootstrap était stratifié sur les entités et cinquante itérations ont été effectuées pour obtenir des moyennes du F1-score ; dans un premier temps localement pour chaque intention et entité. Pour les intentions, une matrice de confusion standard était calculée. Tandis que pour les entités, un vrai positif était une entité dont tous les jetons ont été correctement prédits, un faux positif était une entité dont tous les jetons n'ont pas été correctement prédits et un faux négatif était une entité qui n'a pas été prédit du tout. Enfin, un macro F1-score (i.e., moyenne de tous les F1-scores locaux) a été calculé globalement pour les intentions et entités.

3.3 Résultats

La Table 3.5 présente les macros F1-scores obtenus pour la prédiction globale d'intentions et d'entités. En se basant sur ces résultats, FlauBERT a surpassé CamemBERT pour la prédiction d'intentions et d'entités et ce quelque soit le réseau neuronal. Concernant la prédiction d'intentions, l'écart entre les macros F1-scores de FlauBERT et CamemBERT était environ de 2,0 ; tandis que pour la prédiction d'entités l'écart était beaucoup plus grand et pouvait atteindre les 7,0 de différences. Une explication plausible à ces meilleures performances est la différence de taille du vocabulaire qui permet à FlauBERT de mieux traiter les mots en entrée ; fournissant ainsi des plongements contextuels plus précis. De plus, cette différence entraîne une augmentation du nombre total de paramètres dans la structure de FlauBERT par rapport à CamemBERT pouvant être également à l'origine de ces résultats. En effet, Brown et al. [12] ont récemment démontré que le nombre de paramètres dans un modèle de langue a un fort impact et que plus ce nombre est élevé, meilleurs sont les résultats.

TABLE 3.5 – Macro F1-scores obtenus pour la prédiction d'intentions et d'entités.

Modèle	**Intention**	**Entité**
FlauBERT		
Couche linéaire	97,5	87,9
Réseau récurrent	97,2	88,3
Réseau LSTM	97,0	88,4
Réseau BiLSTM	97,2	88,3
CamemBERT		
Couche linéaire	95,7	83,6
Réseau récurrent	95,5	83,8
Réseau LSTM	95,3	80,5
Réseau BiLSTM	95,2	81,4

En termes de réseaux neuronaux, les performances étaient beaucoup plus hétérogènes et dépendantes du modèle de langue. Concernant la prédiction d'intentions, les quatre réseaux ont obtenu des macro F1-scores très proches avec FlauBERT et CamemBERT. Concernant la prédiction d'entités avec FlauBERT, les réseaux neuronaux obtenaient des performances similaires avec un léger avantage pour les réseaux à mémoire (récurrent, LSTM et BiLSTM). Le contraire a été observé avec CamemBERT : les réseaux avec peu

ou pas de mémoire (linéaire et récurrent) semblent légèrement supérieures. Malheureusement, dans les deux cas les différences ne sont pas flagrantes et conclure quant au choix du réseau neuronal semble toutefois hâtif ; cela nécessiterait une autre étude avec un corpus beaucoup plus grand.

Les Tables 3.6 et 3.7 présentent respectivement les F1-scores locaux obtenus pour chaque intention et entité. Au vu de ces F1-scores, les mêmes constatations faites au préalable étaient encore valables : FlauBERT a surpassé CamemBERT peu importe le réseau neuronal utilisé et aucune différence n'était notée entre les différents réseaux neuronaux. Cependant, les résultats obtenus pour les intentions médicales (*Symptôme*, *Examen* et *Antécédent*) et leurs entités associées (*Date*, *Type Présent* et *Type Absent*) étaient plus faibles ; notamment pour FlauBERT où les F1-scores dépassaient à peine la barre des 85% contre des F1-scores pouvant atteindre les 95% pour les entités non médicales. Une explication plausible à cette chute de performance est le pré-entraînement de FlauBERT et CamemBERT sur un corpus de documents issus de tous horizons. En effet, ce type de pré-entraînement est souvent moins performant dans des domaines spécialisés comme le médical en raison d'un grand nombre de termes et d'acronymes incompris et, parfois, d'une syntaxe ou d'une sémantique totalement différente [28, 29].

Table 3.6 – F1-scores calculés localement pour la prédiction de chaque intention.

Modèle	**Patient**	**Symptôme**	**Examen**	**Antécédent**
FlauBERT				
Couche linéaire	98,5	96,4	98,7	96,4
Réseau récurrent	98,7	96,0	98,3	95,7
Réseau LSTM	98,6	95,6	98,3	95,7
Réseau BiLSTM	98,8	95,5	98,7	95,9
CamemBERT				
Couche linéaire	98,2	94,1	97,4	93,4
Réseau récurrent	98,0	93,6	97,4	93,1
Réseau LSTM	97,8	93,4	97,0	93,3
Réseau BiLSTM	97,4	93,4	97,1	93,1

TABLE 3.7 – F1-scores calculés localement pour la prédiction de chaque entité.

Modèle	**Poids**	**Taille**	**Âge**	**Nom**	**Type présent**	**Date**	**Type absent**
FlauBERT							
Couche linéaire	93,8	92,2	94,9	94,9	84,1	77,1	78,9
Réseau récurrent	92,9	92,0	94,6	96,7	84,5	76,7	81,2
Réseau LSTM	94,4	93,4	93,6	94,6	83,5	78,8	80,7
Réseau BiLSTM	94,9	92,1	95,6	94,4	83,4	78,4	79,9
CamemBERT							
Couche linéaire	92,9	88,8	94,0	79,5	80,4	72,9	76,8
Réseau récurrent	91,5	84,8	96,0	79,5	81,2	74,1	79,2
Réseau LSTM	86,0	80,7	94,1	68,0	81,8	76,5	78,4
Réseau BiLSTM	85,3	75,7	96,5	74,3	81,6	76,3	80,1

3.4 Conclusion du chapitre

Dans cette comparaison entre FlauBERT et CamemBERT, réglés finement avec divers réseaux neuronaux sur une tâche de prédiction d'intentions et d'entités dans le cadre d'un chatbot orienté médical :

i. FlauBERT a surclassé CamemBERT ;

ii. les réseaux neuronaux les plus complexes n'ont pas dépassé de manière significative la couche linéaire qui semblait sur le papier être la plus faible ;

iii. les intentions et entités médicales étaient beaucoup plus difficiles à prédire.

En conclusion, pour un chatbot médical francophone, la combinaison idéale pour garantir une certaine fiabilité est FlauBERT avec une couche linéaire. Cette combinaison sera privilégiée dans la suite de ce manuscrit.

Publication dans la revue *Artificial Intelligence in Medicine* :

FlauBERT vs. CamemBERT :
Understanding patient's answers by a French medical chatbot

Artificial Intelligence In Medicine 127 (2022) 102264

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

FlauBERT vs. CamemBERT: Understanding patient's answers by a French medical chatbot

Corentin Blanc [a,b,c,d,e,*], Alexandre Bailly [a,b,c,d,e], Élie Francis [a], Thierry Guillotin [a], Fadi Jamal [f], Béchara Wakim [g], Pascal Roy [b,c,d,e]

[a] *Everteam Software, Lyon, France*
[b] *Université de Lyon, Lyon, France*
[c] *Université Lyon 1, Villeurbanne, France*
[d] *Service de Biostatistique-Bioinformatique, Pôle Santé Publique, Hospices Civils de Lyon, Lyon, France*
[e] *Équipe Biostatistique-Santé, Laboratoire de Biométrie et Biologie Évolutive, CNRS UMR 5558, Villeurbanne, France*
[f] *IzyCardio, Lyon, France*
[g] *Mediapps Innovation, Lyon, France*

ARTICLE INFO

Keywords:
Intent and slot prediction
FlauBERT
CamemBERT
Language models
Natural Language Understanding
Neural network architectures

ABSTRACT

In a number of circumstances, obtaining health-related information from a patient is time-consuming, whereas a chatbot interacting efficiently with that patient might help saving health care professional time and better assisting the patient. Making a chatbot understand patients' answers uses Natural Language Understanding (NLU) technology that relies on 'intent' and 'slot' predictions. Over the last few years, language models (such as BERT) pre-trained on huge amounts of data achieved state-of-the-art intent and slot predictions by connecting a neural network architecture (e.g., linear, recurrent, long short-term memory, or bidirectional long short-term memory) and fine-tuning all language model and neural network parameters end-to-end. Currently, two language models are specialized in French language: FlauBERT and CamemBERT. This study was designed to find out which combination of language model and neural network architecture was the best for intent and slot prediction by a chatbot from a French corpus of clinical cases. The comparisons showed that FlauBERT performed better than CamemBERT whatever the network architecture used and that complex architectures did not significantly improve performance vs. simple ones whatever the language model. Thus, in the medical field, the results support recommending FlauBERT with a simple linear network architecture.

1. Introduction

During a first patient's consultation or session of specialized care, the interview with a health care professional has to collect a non-negligible amount of data that are essential to establish a diagnosis and initiate or update a plan of care. A part of this essential task may be nevertheless time-consuming. A chatbot (i.e., a conversational interface able to interact with humans) might then help saving time and speeding patient management.

A medical chatbot interacts with a patient via natural language processing (NLP); i.e., an artificial intelligence technology that allows computers to process human speech. Such a chatbot is able to collect data from a patient's speech using an auto-guided dialogue. That chatbot's function involves three iterative tasks: understanding a patient's answer, choosing the next information to obtain according to the previous one, and generating the corresponding question. The present work focuses on the first task; i.e., the Natural Language Understanding (NLU).

NLU is a sub-field of NLP that deals only with understanding human natural language through building a formal representation of the meaning of speech. One possible formal representation aims to organize the information present in a simple sentence by splitting it into a single 'intent' and several 'slots'. The intent relates to the type of information (e.g., socio-demographic characteristic, symptom, etc.), whereas a slot is an accurate datum that one or more words add to enrich the intent of the sentence. For example, age is a slot that contributes to a socio-demographic intent and the date of symptom onset is a slot that contributes to the symptom intent. In such a formal representation, intent

* Corresponding author at: Everteam Software, 17 quai Joseph Gillet, F-69004 Lyon, France.
E-mail address: c.blanc@everteam.com (C. Blanc).

https://doi.org/10.1016/j.artmed.2022.102264
Received 10 June 2021; Received in revised form 15 February 2022; Accepted 23 February 2022
Available online 2 March 2022

and slot prediction consists in spotting in a simple sentence a single intent and one or more related slots. A very simple example is given in Table 1.

Over the past few years, language models pre-trained on huge amounts of data emerged. These models are able to encode each word by focusing on its context within a sentence. The well-known BERT [1] (Bidirectional Encoder Representation Transformers) achieves that coding in eleven tasks of which intent and slot prediction [1,2]. One way of using such a language model consists in connecting a simple neural network (NN) architecture, such as a linear neural network (LNN, a single layer) and fine-tuning all language model and NN parameters end-to-end [1]; however, other NN architectures may be used such as a recurrent NN (RNN) [3], a long short-term memory (LSTM) [4], or a bidirectional long short-term memory (BiLSTM) [5]. The success of BERT led the scientific community to extend it to other languages than English; in French, this extension generated two models: CamemBERT [6] and FlauBERT [7].

In the literature, few researchers have focused on intent and slot prediction with language models, a chatbot, and a French corpus. Between 2018 and 2020, for intent and slot prediction in the medical domain, Neuraz et al. used ELMo and FastText, two word embedding methods for representing sequences of words as corresponding sequences of vectors. Their three papers [8–10] showed a clear superiority of ELMo over FastText in terms of F1 score. Later, in 2020, CamemBERT and FlauBERT started being used for the same purpose (the present work) and others too. As the latter two language models are based on Transformers (one of the most powerful neural architectures today), they became the current references in NLP. Concomitantly, FlauBERT performed better than FastText in "diverse NLP tasks (text classification, paraphrasing, natural language inference, parsing, word sense disambiguation)" targeting of various content texts [7] and CamemBERT performed better than ELMo in processing sentiment analysis from texts from very various sources [11].

Since 2020, in the medical field, Anastasiadou et al. [12] compared BERT against a support vector machine (SVM) and a conditional random field (CRF) in a chatbot for diabetes management and, in 2021, Lei et al. [13] used BERT for COVID-19 patient monitoring. In both cases, the chatbots were domain-specific (diabetes and COVID-19). However, up to now, neither CamemBERT nor FlauBERT has been previously used in a medical chatbot on a much more extended medical corpus.

This study aims to compare CamemBERT and FlauBERT abilities to extract intents and slots from a French medical corpus and determine the best language model/neural network architecture combination able to help patients and health care professionals.

2. Materials and methods

2.1. The data

The present study used the French CAS corpus [14], a benchmark among French medical corpuses. CAS includes thousands of clinical reports extracted from the specialized literature. From that corpus, the study analyzed a sample of 1133 sentences. From each sentence, a single intent and one or several slots were manually tagged according to the taxonomy detailed in Table 2.

The slots were tagged using the IOB format. In that format, prefix 'B-' indicates a word at the beginning of a slot, 'I-' a word in the middle or the end of a slot. Prefix 'O' was used to tag words that do not belong to slots. Finally, '</s>' indicates the end of a sentence. For example, the sentence shown in Table 1 ('J'ai eu de la tachycardie hier soir </s>') would be tagged: O O O O O B-TypePresent B-Time I-Time Symptom.

Table 1
Formal representation of a sentence by intent and slots.

Speech sentence	Intent	Slots
J'ai eu de la tachycardie hier soir[a]	Symptom	Type present (*tachycardie*) Time (*hier soir*)

[a] I had tachycardia last night.

Table 2
Taxonomy and dataset description.

Intents and slots	Description	Frequency (%)
Patient		192 (17%)
Weight	Patient's weight	44 (14%)
Height	Patient's height	40 (12%)
Age	Patient's age	172 (53%)
Person	Patient's name	69 (21%)
Symptom		203 (18%)
Type present	Symptom description	377 (76%)
Time	Date of symptom onset	121 (24%)
Exam		566 (50%)
Type present	Past medical examination	639 (100%)
Risk		172 (15%)
Type present	Presence of risk factor	218 (65%)
Type absent	Absence of risk factor	116 (35%)

2.2. The proposed approaches

Intent and slot prediction was carried out in two steps (Fig. 1). First, the language model encoded each word of a sentence with an embedding layer and that encoding was contextualized with Transformer Encoder layers. Second, using the contextualized embeddings generated by the language model, a neural network architecture predicted intents and slots.

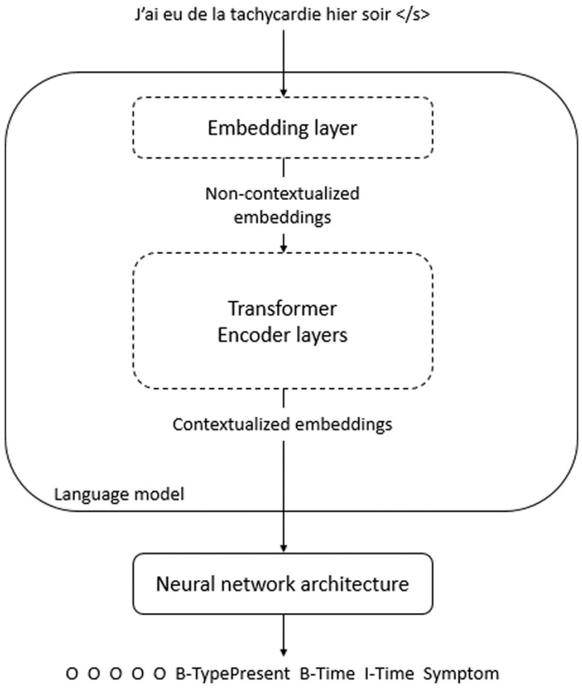


Fig. 1. Intent and slot prediction using a language model. At this figure top, tag </s> indicates the end of the sentence under analysis. First, the language model encodes the sentence non-contextually (with the embedding layer) then contextually (with the Transformer Encoder layers). Finally, the neural network architecture predicts the intent and the slots using the previously calculated contextualized embeddings. The slots are in IOB format: prefix 'B-' indicates a word at the beginning of a slot, 'I-' a word in the middle or the end of a slot, and 'O' a word that does not belong to a slot.

2.2.1. The language models

Language models analyze the text in segments of one or more sentences. In the present study, it is important to underline that each segment corresponded to a single sentence. Each word (sometimes part of the same slot; e.g., slot 'hier soir' derives from 'hier' (B-Time) and 'soir' (I-Time)) was indexed to a vocabulary composed of words and subwords (also called 'tokens'). A tokenizer allowed managing unknown words by cutting them up and then associating them with vocabulary tokens. The embedding layer returned, for each token, a non-contextualized embedding that corresponds to the sum of three different embedding (Fig. 2):

- A token embedding resulting from a linear projection of the tokens indexes.
- A position embedding resulting from a linear projection of tokens' positions in the segment (here, = sentence).
- A segment embedding resulting from a linear projection of the belonging of various tokens of the same sentence. Here, all tokens of a given sentence had the same segment embedding because each segment corresponded to a single sentence.

The non-contextualized embedding resulting from the embedding layer passed then through a succession of twelve Transformer Encoder layers [15] (Fig. 1). Each layer took as input the previous single output to refine the contextualization using a twelve-headed self-attention mechanism. In each of the twelve heads, the self-attention mechanism allowed each token of the sentence to find out other token needed to refine the contextualization. Finally, the language model returned a contextualized embedding for each token of the sentence.

Here, two language models were compared: FlauBERT and CamemBERT. These two models have similar structures but differ in a number of points: i) the tokenizer (Byte Pair Encoding [16] vs. SentencePiece [17], respectively); ii) the vocabulary size (50,000 vs. 32,000, resp.); iii) the number of parameter (138M vs. 110M, resp.); and, iv) the size of the training dataset (71 GB vs. 138 GB, resp.).

2.2.2. The neural network architectures

From a contextualized embedding and for each token, the NN architecture infers the logarithm of a probability distribution for each potential intent and slot using a LogSoftmax activation function. Finally, a CRF [18] was attached at the top of every NN architecture to predict the most likely intent or slot while preserving the IOB format.

Here, the results of using four NN architectures (each) associated with a CRF were compared:

- a Linear NN (LNN) that processes linearly each contextualized embedding.
- a Recurrent NN (RNN) that processes each contextualized embedding taking into account the previous one.
- a Long Short-Term Memory (LSTM) that processes each contextualized embedding taking into account all previous ones.
- a Bidirectional Long Short-Term Memory (BiLSTM) that processes each contextualized embedding taking into account all previous and subsequent ones.

2.3. The training parameters

To train jointly all language models and NN parameters end-to-end, the number of epochs (i.e., number of times the corpus is explored during training) was set to ten as determined by the convergence value of the negative log-likelihood. AdamW learning algorithm [19] was used for training with a learning rate initially set to 2e-5 (first epoch) but that decreased linearly until it came to 0 (last epoch). A gradient clipping [20] of 1.0 was used to reduce gradient-exploding effects; otherwise said, the gradient was scaled down whenever its norm exceeded 1.0 to avoid too large gradients. Finally, every NN architecture has only one hidden layer.

In this work, we used Python 3.8.2 as programming language and the following packages:

- Torchtext 0.9.1 to load and tokenize the CAS corpus.
- Transformers 3.1.0 from HuggingFace to apply CamemBERT and FlauBERT.
- PyTorch 1.8.1 to deal with the NN architecture, the CRF, and model training.

With an NVIDIA Graphics processing Unit of 16 GB, the processing time for the downstream task was about 20 ms per sentence and language model.

2.4. The evaluation criteria

Intent and slot predictions were separately evaluated with Macro F1 scores. A standard confusion matrix was calculated for intents. For slots, a 'true positive' slot was one whose every token was correctly predicted, a 'false positive' one whose not all tokens were correctly predicted, and a 'false negative' one that was not predicted at all. Combinations of intent and slot predictions were used to analyze a joint performance.

The evaluation used a bootstrap method [21]. That method consisted in drawing randomly sentences from the CAS corpus –with replacement

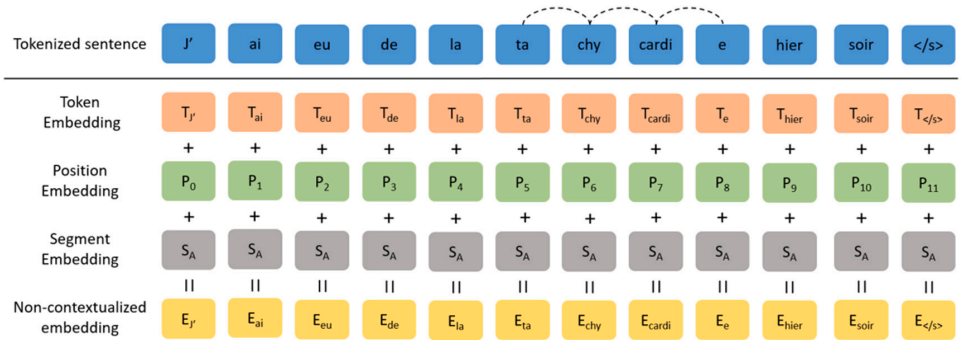


Fig. 2. Representation of the embedding layers. The tokenized sentence shows the tokens of the sentence (here, = segment) obtained with the tokenizer submitted to analysis. In this sentence, 'tachycardie' was cut into four tokens. The token embedding is a linear projection of the indexes given to the tokens of the sentence. The position embedding is a linear projection of the position of each token in the sentence. The segment embedding is a linear projection of the belonging of various tokens to the same sentence. Finally, the embedding layer returns a non-contextualized embedding that corresponds to the sum of the embeddings of the token, the position, and the segment.

to keep constant the size of the training set. Thus, the training set included 1133 sentences (just as the CAS corpus). Sentences not drawn constituted the test set. Theoretically, about 37% of the sentences of CAS corpus remained in the test set (nearly 419 sentences). This bootstrapping was slot-stratified and fifty bootstrap iterations were run to obtain means and standard deviations for the Macro F1 score.

3. Results

3.1. Performance of the language models

According to the Macro F1 scores, FlauBERT performed better than CamemBERT regardless of the NN architecture used (Table 3). Regarding intent prediction, FlauBERT and CamemBERT scores were rather very close (differences less than 0.02 according to the architecture). Regarding slot prediction, FlauBERT and CamemBERT scores were less close (differences less than 0.2 according to the architecture).

Regarding the combinations of intent and slot predictions, FlauBERT performed better than CamemBERT regardless of the NN architecture used (Table 4). FlauBERT and CamemBERT predicted correctly the intent and every slot with NN architecture. However, both language models were wrong about at least one slot in proportions very close to 0.19 with the former and 0.25 with the latter whatever the NN architecture. Anyway, the proportion of wrongly predicted intents never exceeded 0.33.

3.2. Performance of the NN architectures

According to the Macro F1 scores, the NN architectures had heterogeneous performance levels that depended on the language model used (Table 3). Regarding intent prediction, LNN, RNN, LSTM, and BiLSTM architectures had very close and not significantly different Macro F1 scores with either FlauBERT (circa 0.972) or CamemBERT (circa 0.957). Regarding slot prediction with FlauBERT, Macro F1 scores with RNN, LSTM, and BiLSTM architectures were slightly higher (though not significantly different) than with the LNN. Regarding slot prediction with CamemBERT, Macro F1 scores with a LNN or a RNN architecture were higher (though not significantly different) than with a LSTM or a BiLSTM architecture.

According to the combinations of intent and slot predictions, the NN architectures had heterogeneous performance levels that depended on the language model used (Table 4). All architectures (LNN, RNN, LSTM, and BiLSTM) predicted correctly the intent and every slot in proportions close to 0.785 with FlauBERT and to 0.710 with CamemBERT (the differences were not significant). LNN, RNN, LSTM, and BiLSTM architectures predicted correctly the intent but were wrong about at least one slot in proportions close to 0.190 with FlauBERT and 0.240 with CamemBERT (differences not significant). The proportion of wrongly predicted intents never exceeded 0.033.

Table 3
Macro F1 scores for intent and slot predictions.

Language model and architecture	Intent F1 score	Slot F1 score
CamemBERT		
LNN architecture	0.957	0.836[a]
RNN architecture	0.955[a]	0.838[a]
LSTM architecture	0.953	0.805[a]
BiLSTM architecture	0.952[a]	0.814[a]
FlauBERT		
LNN architecture	0.975	0.879[a]
RNN architecture	0.972	0.883
LSTM architecture	0.970	0.884[a]
BiLSTM architecture	0.972	0.883[a]

LNN: linear neural network. RNN: recurrent neural network. LSTM: long short-term memory. BiLSTM: bidirectional long short-term memory.
[a] Standard deviation range: 0.010–0.015. All other standard deviations range between 0.006 and 0.010.

Table 4
Distribution of intent and slot prediction combinations.

Language model and architecture	Intent true		Intent false	
	Slot true	Slot false	Slot true	Slot false
CamemBERT				
LNN architecture	0.714[a]	0.239[a]	0.014	0.033
RNN architecture	0.719[a]	0.236[a]	0.015	0.030
LSTM architecture	0.585[a]	0.264[a]	0.017	0.033
BiLSTM architecture	0.695[a]	0.258	0.014	0.033
FlauBERT				
LNN architecture	0.779[b]	0.197[b]	0.009	0.015
RNN architecture	0.786[b]	0.189[b]	0.010	0.015
LSTM architecture	0.787[a]	0.186[a]	0.010	0.017
BiLSTM architecture	0.785[a]	0.185[a]	0.012	0.018

All other standard deviations range between 0.006 and 0.010. LNN: linear neural network. RNN: recurrent neural network. LSTM: long short-term memory. BiLSTM: bidirectional long short-term memory.
[a] Standard deviation range: 0.015–0.020.
[b] Standard deviation range: 0.020–0.25.

4. Discussion

Given the above-shown results, FlauBERT performed better than CamemBERT regardless of the NN architecture used, whereas the NN architectures performed unequally depending on the language model. RNN, LSTM, and BiLSTM architectures outperformed slightly the LNN with FlauBERT but the LNN and RNN architectures outperformed slightly LSTM and BiLSTM architectures with CamemBERT. Undeniably, the best combinations were those that used FlauBERT.

One plausible explanation for FlauBERT better performance would be its extra 28M parameters vs. CamemBERT. Indeed, Brow et al. [22] have recently demonstrated that the number of parameters in a language model has a strong impact and that the higher is the number of parameters, the better are the results. Another explanation would be the 18,000-token vocabulary difference that make FlauBERT able to better deal with the input words and provide better contextual representations.

The NN architectures were difficult to compare given the non-significant differences in most Macro F1 scores. However, it seemed that, with FlauBERT, the architectures that compute the tokens and take into account the rest of the sentence (i.e., RNN, LSTM, and BiLSTM) performed better than the LNN. With CamemBERT, the opposite was seen: LSTM and BiLSTM architectures performed poorly vs. the LNN and the RNN architectures. Thus, for the moment, concluding about architecture performance seems very difficult; it requires another study on much more data (say, 10 times more).

One merit of the work is the addition to the current literature new results stemming from the use and comparisons of performance between CamemBERT and FlauBERT within the context of a medical chatbot in French language. Up to now, FastText and ELMo were the only language models used this way [8–10]; they showed much lower performance than those of CamemBERT and FlauBERT in most NLP tasks. The performances of the above-cited four word embedding methods are certainly worth being compared on the same medical corpus in a future dedicated work. This will be a natural and interesting extension of the present work.

Although CamemBERT and FlauBERT were used here with French language, it seems highly probable that the same kind of study could be conducted with other languages that have similar sentence structures (syntaxes) and may undergo similar contextual embeddings; e.g., Spanish, Italian, Portuguese. Moreover, the CAS corpus used [14] included nearly 200 medical case reports in nearly all medical specialties. This wide coverage is a non-negligible asset versus contexts with a single specialty or domain (e.g., diabetes [12] or COVID-19 [13]). A second merit of the work is the prediction of intents in addition to slots.

Intent prediction helps slot prediction and allows a better organization of the information contained in a sentence. A third merit is the manual labeling of the corpus. Manual intent and slot labeling provides a quality corpus analysis that accounts for the accuracy and sensitivity of the medical language. Furthermore, manual labeling allows adapting accurately different performant language models to the medical chatbot; thus, extracting relevant information for the health care professional. However, one inconvenience of manual labeling is that it may be tedious and time-consuming (depending on the corpus size). For the present work, the manual labeling of the whole corpus required nearly two weeks full-time work.

5. Conclusion

In this comparison of intent and slot prediction between CamemBERT and FlauBERT fine-tuned with different NN architectures in a medical chatbot for French-speaking patients, i) FlauBERT achieved a better performance regardless of the NN architecture; and, ii) the most complex architectures did not significantly outperform the LNN which seemed to be the most reliable. Thus, for a French medical chatbot, we would recommend FlauBERT with a LNN.

Declaration of competing interest

Authors CB, AB, EF, and TG are employed by Everteam Software. Authors FJ, BW, and PR have no interests to declare.

Acknowledgments

The authors are grateful to Jean Iwaz (Hospices Civils de Lyon) for augmenting, editing, proofreading, and formatting the latest versions of the manuscript.

Funding

This work was supported by Association Nationale de la Recherche et de la Technologie (ANRT) [grant number 2019/1374]. The sponsor had no role in the study design; the collection, analysis, and interpretation of the data; the writing of the report; and the decision to submit the article for publication.

References

[1] Devlin J, Chang M-W, Lee K, Toutanova K, BERT. Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference North American Chapter of the Association for Computational Linguistics: human language technologies. 1; 2019. p. 4171–86.
[2] Chen Q, Zhuo Z, Wang W. BERT for joint intent classification and slot filling. 2019. http://arxiv.org/abs/1902.10909.
[3] Elman JL. Finding structure in time. Cognit Sci 1990;14:179–211. https://doi.org/10.1016/0364-0213(90)90002-E.
[4] Greff K, Srivastava RK, Koutnik J, Steunebrink BR, Schimdhuber J. LSTM: a search space odyssey. IEEE Trans Neural Netw Learn Syst 2017;28:2222–32. https://doi.org/10.1109/TNNLS.2016.2582924.
[5] Schuster M, Paliwal K. Bidirectional recurrent neural networks. IEEE Trans Signal Process 1997;45:2673–81. https://doi.org/10.1109/78.650093.
[6] Martin L, Muller B, Suarez PJO, Dupont Y, Romary L, de la Clergerie V, Seddah D, Sagot B. CamemBERT: a tasty french language model. In: Proceedings of the 58th annual meeting of the Association for Computational Linguistics; 2020. p. 7203–19. https://doi.org/10.18653/v1/2020.acl-main.645.
[7] Le H, Vial L, Frej J, Segonne V, Coavoux M, Lecouteux B, Allauzen A, Crabbé B, Besacier L, Schwab D. FlauBERT: unsupervised language model pre-training for french. In: Proceedings of the 12th Language Resources and Evaluation Conference; 2020. p. 2479–90.
[8] Neuraz A, Looten V, Rance B, Daniel N, Garcelon N, Llanos LC, Burgun A, Rosset S. Do you need embeddings trained on a massive specialized corpus for your clinical natural language processing task? Stud Health Technol Inform 2019;264:1558–9. https://doi.org/10.3233/SHTI190533.
[9] Neuraz A, Rance B, Garcelon N, Llanos LC, Burgun A, S. Rosset S.. The impact of specialized corpora for word embeddings in natural language understanding. Stud Health Technol Inform 2020;270:432–6. https://doi.org/10.3233/SHTI200197.
[10] Neuraz A, Llanos LC, Burgun A, Rosset S. Natural language understanding for task oriented dialog in the biomedical domain in a low resources context. In: Machine Learning for Health (ML4H) Workshop at NeurIPS; 2018. https://arxiv.org/pdf/1811.09417.pdf.
[11] Habbat N, Anoun H, Hassouni L. LSTM-CNN deep learning model for french online product reviews classification. In: Saidi R, Bhiri B El, Maleh Y, Mosallam A, Essaaidi M, editors. International conference on advanced technologies for humanity (ICATH 2021). 110; 2022. p. 228–40. https://doi.org/10.1007/978-3-030-94188-8_22.
[12] Anastasiadou M, Alexiadis A, Polychronidou E, Votis K, Tzovaras D. A prototype educational virtual assistant for diabetes management. In: IEEE 20th international conference on bioinformatics and bioengineering (BIBE); 2020. p. 999–1004. https://doi.org/10.1109/BIBE50027.2020.00169.
[13] Lei H, Lu W, Ji A, Bertram E, Gao P, Jiang X, Barman A. Covid-19 smart chatbot prototype for patient monitoring. 2021. https://arxiv.org/abs/2103.06816.
[14] Grabar N, Claveau V, Dalloux C. CAS: French corpus with clinical cases. In: Proceedings of the ninth international workshop on health text mining and information analysis. Association for Computational Linguistics; 2018. p. 122–8. https://doi.org/10.18653/v1/W18-5614.
[15] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: The 31st conference on neural information processing systems; 2017. p. 1–11.
[16] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units. In: Proceedings of the 54th annual meeting of the association for computational linguistics; 2016. p. 1715–25. https://doi.org/10.18653/v1/P16-1162.
[17] Kudo T, Richardson J. In: Sentence piece: a simple and language independent subword tokenizer and detokenizer for neural text processing. Association for Computational Linguistics; 2018. p. 66–71. https://doi.org/10.18653/v1/D18-2012.
[18] Lafferty J, McCallum A, Pereira FCN. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML '01 proceedings of the 18th international conference on machine learning; 2001. p. 282–9.
[19] Loshchilov I, Hutter F. Decoupled weight decay regularization. In: 7th international conference on learning representations; 2019.
[20] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. In: Proceedings of the 30th international conference on international conference on machine learning. 28; 2013. p. 1310–8.
[21] Efron B. Bootstrap methods: another look at the jackknife. Ann Stat 1979;7:1–26. https://doi.org/10.1214/aos/1176344552.
[22] T.B. Brown B. Mann N. Ryder M. Subbiah J. Kaplan P. Dhariwal A. Neelakantan P. Shyam G. Sastry A. Askell S. Agarwal A. Herbert-Voss G. Krueger T. Henighan R. Child A. Ramesh D. M. Ziegler J. Wu C. Winter C. Hesse M. Chen E. Sigler M. Litwin S. Gray B. Chess J. Clark C. Berner S. McCandlish A. Radford I. Sutskever D. Amodei . Language models are few-shot learners. In: H. Larochelle M. Ranzato R. Hadsell M.F. Balcan H. Lin , Editors. Advances in neural information processing systems 33, Annual conference on neural information processing systems (NeurIPS 2020). https://arxiv.org/pdf/2005.14165.pdf.

Chapitre 4

Enrichissement linguistique

Bien que globalement performants, FlauBERT et CamemBERT ont présenté des lacunes dans le domaine médical ; probablement liées à leurs pré-entraînements non-médicaux. Combler une partie de ces lacunes sera l'objectif de ce chapitre grâce à un enrichissement linguistique de ces deux modèles. Ces enrichissements ont fait l'objet d'une soumission dans la revue *Computer Methods and Programs in Biomedicine* intitulé :

Corpus size considerations in continual pre-training of BioFlauBERT and BioCamemBERT

Sommaire

4.1 Contexte

Ces dernières années, différentes stratégies ont été envisagées pour la construction de modèle de langue médicaux :

- Le pré-entraînement à partir de zéro qui consiste à initialiser le modèle avec des paramètres aléatoires puis à le pré-entraîner grâce à un corpus contenant des documents médicaux, voir une combinaison de documents médicaux et non-médicaux (Figure 4.1). Le modèle de langue ainsi obtenu peut être réglé finement afin d'être adapté une tâche TALN médicale.

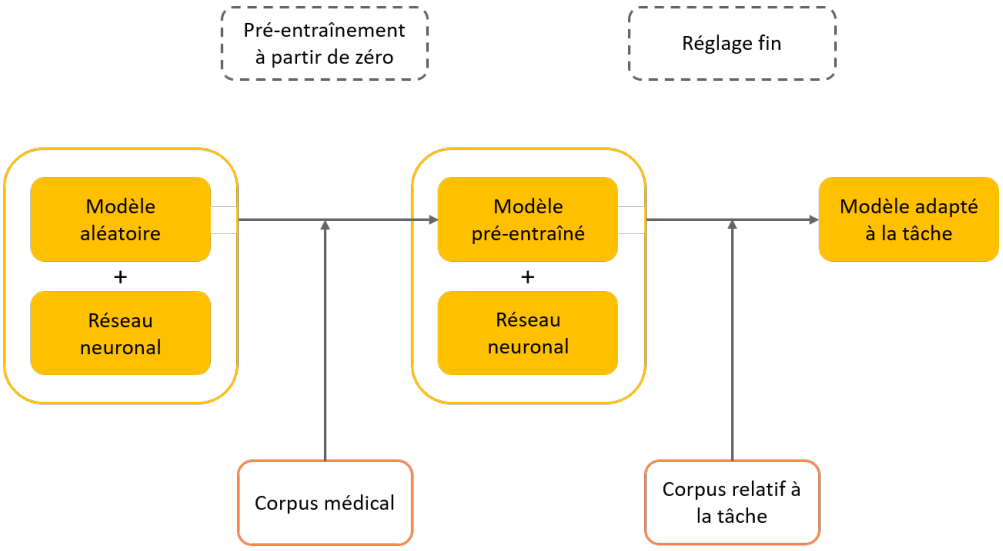


FIGURE 4.1 – Pré-entraînement à partir de zéro.

- Le pré-entraînement continu qui consiste à initialiser le modèle avec les paramètres d'un autre modèle déjà existant, puis à l'enrichir par un pré-entraînement supplémentaire grâce à un corpus contenant des documents médicaux (Figure 4.2).

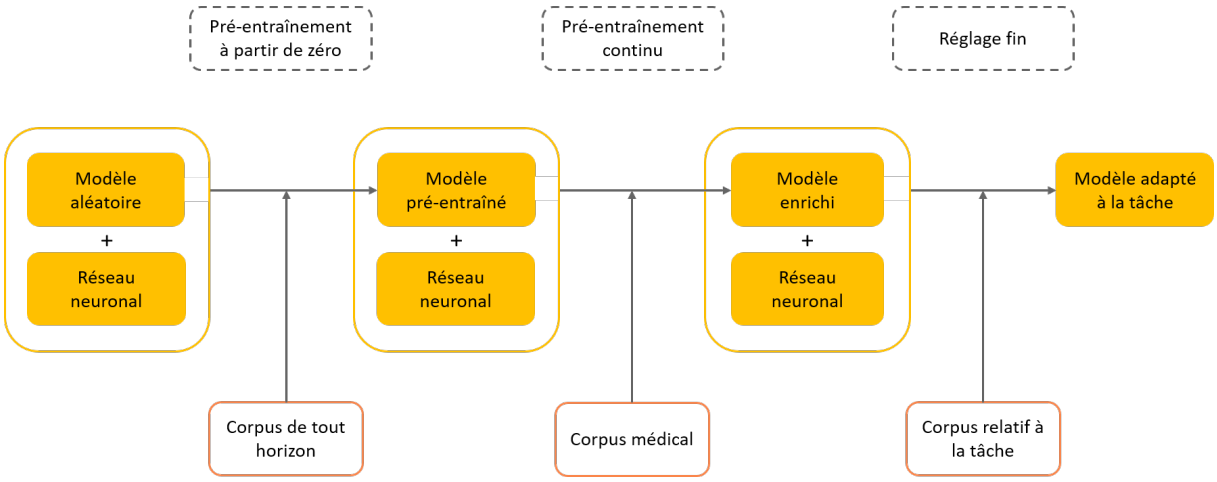


FIGURE 4.2 – Pré-entraînement continu.

Bien que ces deux approches soient efficaces pour concevoir un modèle de langue médical, le pré-entraînement continu reste la plus intéressante en termes de ressources numériques et textuelles. Par conséquent, ce chapitre vise à réaliser un enrichissement linguistique des modèles de langue FlauBERT et CamemBERT par pré-entraînement continu en inspectant l'effet de la taille du corpus médical. Plusieurs évaluations ont été mises en place afin de contrôler au mieux ces enrichissements, de déterminer la taille de corpus optimal et de comparer les deux modèles enrichis. Ces nouveaux modèles ont été prénommés BioFlauBERT et BioCamemBERT et ont pour vocation à être mis en libre accès sur la plateforme HuggingFace [30].

À ce stade, les travaux des chapitres 2, 3 et 4 de cette thèse ont servi à mettre en place toute la méthodologie TALN de deux articles scientifiques supplémentaires en cours de soumission dont je suis le second auteur :

- *Early vs. late data fusion in multimodal death-cause classification on text and structured data* traitant de la combinaison de données structurées et textuelles pour la prédiction de causes primaires de décès avec BioFlauBERT (Annexe A).
- *Importance of the number of classes and the proportion of labeled data in semi-supervised text classification* mettant BioFlauBERT en situation d'apprentissage semi-supervisé pour la prédiction de causes primaires de décès. (Annexe B).

4.2 Méthodologie

4.2.1 Corpus

Le corpus français CLEAR [31] contenant des extraits d'encyclopédies, de notices de médicaments et de résumés scientifiques, a été utilisé comme corpus médical pour le pré-entraînement continu. Dans l'optique d'enrichir FlauBERT et CamemBERT, les phrases de ce corpus ont été découpées en jetons à l'aide des générateurs *Byte Pair Encoding* et *SentencePiece*. Avec ce processus, les phrases trop courtes (< 16 jetons) et trop longues (> 64 jetons) pour les deux générateurs ont été filtrées pour limiter les silences et les bruits. Les phrases retenues ont été divisées en un corpus de pré-entraînement et un corpus d'évaluation interne composés respectivement de 1,5 million et 10 mille phrases. Enfin, quatre sous-corpus emboîtés de tailles 5, 50, 500 et 1 500 mille phrases ont été tirés au sort sans remplacement dans le corpus de pré-entraînement afin de faire varier sa taille.

4.2.2 Pré-entraînement continu

Tout d'abord, BioFlauBERT et BioCamemBERT ont été initialisés avec les paramètres respectifs de FlauBERT et CamemBERT. Pour les deux modèles, le pré-entraînement continu était porté par une tâche de modélisation du langage masqué consistant à prédire des jetons aléatoirement masqués au sein des phrases en se basant sur leurs contextes. Pour chaque phrase, la stratégie de masquage suivante a été effectuée au début de chaque epoch : 12% des jetons ont été aléatoirement masqués, 1,5% ont été aléatoirement échangés avec d'autres jetons du vocabulaire du modèle considéré et les jetons restants étaient inchangés. Pour renforcer cette stratégie, les jetons n'apportant aucun bénéfice médical à la phrase (e.g., préposition, article ou pronom) n'étaient jamais masqués pour se concentrer uniquement sur les jetons médicaux. Enfin, une couche linéaire composée de V neurones a été connectée au modèle pour modéliser le langage.

Pour le t-ème jeton, la distribution de probabilités $p_t \in [0;1]^V$ sur l'ensemble du vocabulaire est inférée à partir des plongements contextuels $Z^L = (z_t^L) \in \mathbb{R}^{T \times D}$:

$$p_t \;=\; \text{softmax}(z_t^L \cdot W_y + b_y) \tag{4.1}$$

où $W_y \in \mathbb{R}^{D \times V}$ sont les paramètres de la couche linéaire et $b_y \in \mathbb{R}^V$ le biais associé. Enfin, le jeton prédit $\hat{y}_t$ est donné par :

$$\hat{y}_t \;=\; \underset{1 \,\leq\, v \,\leq\, V}{\text{argmax}} \,(p_{t,v}) \tag{4.2}$$

où $p_{t,v}$ la probabilité que le v-ème jeton du vocabulaire soit le t-ème jeton de la phrase selon le modèle.

4.2.3 Fonction de perte

L'entropie croisée a été utilisée en tant que fonction de perte. Ici, elle évalue la discordance entre les distributions de jetons prédites par le modèle et celles observées dans le corpus. Pour la n-ème phrase de longueur T_n du corpus de pré-entraînement, l'entropie croisée est donnée par :

$$L \;=\; \frac{-1}{T_n} \sum_{t=1}^{T_n} \sum_{v=1}^{V} \; \mathbb{1}_{t,v} \cdot \log(p_{t,v}) \tag{4.3}$$

où $\mathbb{1}_{t,v}$ est une indicatrice (1 lorsque le v-ème jeton du vocabulaire est le t-ème jeton de la phrase, 0 sinon).

4.2.4 Hyperparamétrage

Pour entraîner conjointement BioFlauBERT et BioCamemBERT avec la couche linéaire de modélisation du langage, plusieurs hyperparamètres ont été sélectionnés grâce à une grille de recherche. Un récapitulatif complet de l'hyperparamétrage utilisé est donné dans la Table 4.1.

TABLE 4.1 – Récapitulatif de l'hyperparamétrage pour le pré-entraînement continu.

Hyperparamètre	**Valeur**
Nombre d'epochs	Convergence
Taille des lots	150
Écrêtage du gradient	Aucun
Algorithme d'apprentissage	AdamW
Taux d'apprentissage initial	2e-5
Variation du taux d'apprentissage	Aucune
Taux d'apprentissage final	2e-5
Stabilité numérique	1e-8
Décroissance du premier moment	0,9
Décroissance du second moment	0,999
Décroissance des paramètres	0,01

4.2.5 Évaluation interne

4.2.5.1 Entropie croisée

Le premier critère utilisé pour l'évaluation interne est la même entropie croisée que dans l'équation (4.3). Ce critère évalue la faculté du modèle à modéliser le langage masqué sur le corpus d'évaluation. En appliquant la même stratégie de masquage, l'entropie croisée est donnée par la moyenne de toutes les entropies croisées de chaque phrase.

4.2.5.2 Perplexité

Le second critère utilisé pour l'évaluation interne est la perplexité. Ce critère évalue l'hésitation moyenne du modèle de langue sur chaque jeton de chaque phrase du corpus d'évaluation d'interne sachant les précédents. De ce fait, plus la perplexité est proche de 1 et plus les phrases sont compréhensibles pour le modèle. En posant $x^n = (x_t^n)$ la n-ème phrase de longueur T_n du corpus d'évaluation de taille N, la perplexité PP s'écrit :

$$\mathrm{PP} \;=\; \exp\left(\frac{-1}{\sum_{n=1}^{N} T_n} \sum_{n=1}^{N}\left[\log \mathbb{P}(x_1^n) \;+\; \sum_{t=2}^{T_n} \log \mathbb{P}(x_t^n | x_1^n, \ldots, x_{t-1}^n)\right]\right) \tag{4.4}$$

4.2.6 Évaluation externe

L'évaluation externe évalue la faculté d'un modèle de langue à performer sur des tâches TALN. Dans cette étude, une compilation de quatre tâches de classification a été utilisée : deux d'entre elles étaient au niveau de la phrase (détermination d'intentions et similarité sémantique) et les deux autres au niveau des jetons (reconnaissance d'entités nommées et détection de négations). BioFlauBERT et BioCamemBERT ont été adaptés en réglage fin sur ces quatre tâches avec une couche linéaire associée à une fonction d'activation softmax et l'hyperparamétrage récapitulé dans la Table 4.2. Enfin, les performances ont été évaluées à l'aide d'une méthode de bootstrap à cinquante itérations pour obtenir des moyennes du macro F1-score.

4.2.6.1 Détermination d'intentions

La détermination d'intentions a été effectuée à l'aide du corpus QA dialogue [32] composé de 1 818 phrases extraites de conversations entre des patients et des professionnels de la santé. Les auteurs ont étiqueté manuellement chaque phrase d'une en fonction du type d'information (appelé intention dans le contexte d'un chatbot) selon les possibilités suivantes : motifs de consultation, données personnelles, antécédents médicaux, symptômes, habitudes de vie, traitements et autres. L'objectif était d'attribuer la bonne intention à chaque phrase du corpus.

4.2.6.2 Similarité sémantique

La similarité sémantique était portée par le corpus CLISTER [33] composé de 1 000 paires de phrases provenant de rapports de cas cliniques couvrant une grande diversité de données cliniques telles que des antécédents médicaux, des traitements ou encore des suivis. Chaque paire de phrases a été annotée d'un score de similarité compris entre 0 (aucune similarité) et 5 (similarité maximale). L'objectif était d'attribuer un score de similarité à chaque paire de phrases du corpus.

4.2.6.3 Reconnaissance d'entités nommées

La reconnaissance d'entités nommées a été réalisée en utilisant le corpus QUAERO [34] composé de 4 424 phrases médicales extraites d'informations sur des médicaments commercialisés par l'Agence Européenne des Médicaments, de titres d'articles de recherche indexés dans la base de données MEDLINE et de brevets enregistrés auprès de l'Office Européen des Brevets. Chaque phrase a été étiquetée à l'aide de dix catégories d'entités nommées : produits chimiques et médicaments, anatomie, troubles, dispositifs, procédures, zones géographiques, êtres vivants, objets, phénomènes et physiologie. L'objectif était de localiser les différentes entités nommées dans chaque phrase du corpus.

4.2.6.4 Détection de négations

La détection de négations et de portées a été effectuée à l'aide du corpus CAS [19] composé de 11 037 phrases provenant de rapports de cas cliniques extraits. S'agissant d'un cas particulier de reconnaissance d'entités nommées, l'objectif était de localiser les négations et les portées au sein de chaque phrase du corpus.

TABLE 4.2 – Récapitulatif de l'hyperparamétrage pour l'évaluation externe.

	Valeur			
Hyperparamètre	**QA Dialogue**	**CLISTER**	**QUAERO**	**CAS**
Nombre d'epochs	5	10	10	5
Taille des lots	1	8	8	8
Écrêtage du gradient	Aucun	Aucun	Aucun	Aucun
Algorithme d'apprentissage	AdamW	AdamW	AdamW	AdamW
Taux d'apprentissage initial	2e-5	2e-5	2e-5	2e-5
Variation du taux d'apprentissage	Linéaire	Linéaire	Linéaire	Linéaire
Taux d'apprentissage final	0,0	0,0	0,0	0,0
Stabilité numérique	1e-8	1e-8	1e-8	1e-8
Décroissance du premier moment	0,9	0,9	0,9	0,9
Décroissance du second moment	0,999	0,999	0,999	0,999
Décroissance des paramètres	0,01	0,01	0,01	0,01

4.3 Résultats

La Table 4.3a présente l'entropie croisée et la perplexité obtenues pour chaque modèle en fonction de la taille du corpus médical de pré-entraînement. En se basant sur ces deux critères, BioFlauBERT (resp. BioCamemBERT) a surpassé FlauBERT (resp. CamemBERT) et ce quelque soit la taille du corpus. Étonnamment, la modélisation du langage médical était déjà très efficiente avec seulement quelques milliers de phrases, avant de continuer d'augmenter progressivement avec la taille du corpus jusqu'à obtenir des minima grâce un pré-entraînement avec 1 500 000 phrases. Bien qu'elle ne garantisse pas de bons résultats sur des tâches TALN, l'évaluation interne reste primordiale pour estimer les hyperparamètres et l'ordre de grandeur du nombre optimal de phrases.

La Table 4.3b présente les macro F1-scores obtenus pour l'ensemble des tâches médicales composant l'évaluation externe. En se basant sur ces résultats, les modèles de langue enrichis ont de nouveau significativement surpassé leurs homologues non-médicaux mais cette fois-ci après quelques centaines de milliers de phrases. Cette supériorité était attendue au vu de la littérature et confirme l'intérêt d'un enrichissement linguistique lorsque l'on travaille dans un domaine bien spécifique comme la médecine. Néanmoins, les tailles de corpus utilisés restent très inférieures à celles de la littérature (plusieurs centaines de milliers de phrases en 2019 [28, 35] jusqu'à des centaines de millions en 2021 [36, 37]). Il serait intéressant de reproduire cette étude à l'avenir avec beaucoup plus de phrases pour voir quel niveau de performance peut être atteint asymptotiquement. Remarquablement, la diversité des tâches médicales a montré que BioFlauBERT (resp. FlauBERT) était le plus performant au niveau des jetons et BioCamemBERT (resp. CamemBERT) au niveau de la phrase. Une explication plausible pour la supériorité de BioFlauBERT sur les tâches au niveau des jetons serait son vocabulaire beaucoup plus étendu que celui de BioCamemBERT : augmentant le nombre de paramètres et fournissant de meilleurs plongements contextuels pour chaque jeton. Au contraire, l'ajout du couche linéaire de mise en commun dans la structure de BioCamemBERT (absente pour BioFlauBERT) semble garantir un meilleur plongement pour la globalité de la phrase. Infirmer ou confirmer cette hypothèse nécessiterait des recherches supplémentaires avec de nombreuses autres études.

TABLE 4.3 – Résultats obtenus pour l'évaluation interne et externe de BioFlauBERT et BioCamemBERT en fonction du nombre de phrases utilisées dans le corpus de pré-entraînement.

(a) Évaluation interne

Modèle de langue	**Entropie croisée**	**Perplexité**
FlauBERT	1,05	3,02
BioFlauBERT		
avec 5k phrases	0,14	1,54
avec 50k phrases	0,13	1,21
avec 500k phrases	0,06	1,11
avec 1,5M phrases	0,04	1,10
CamemBERT	2,75	3,24
BioCamemBERT		
avec 5k phrases	0,22	2,78
avec 50k phrases	0,20	2,52
avec 500k phrases	0,07	2,06
avec 1,5M phrases	0,06	1,92

(b) Évaluation externe

	Phrase		**Jetons**	
Modèle de langue	**QA Dialogue**	**CLISTER**	**QUAERO**	**CAS**
FlauBERT	84,6	91,0	68,2	93,4
BioFlauBERT				
avec 5k phrases	85,6	91,0	68,3	93,4
avec 50k phrases	87,0	91,4	68,1	93,6
avec 500k phrases	87,4	92,5	68,6	93,7
avec 1,5M phrases	87,7	92,3	68,9	93,8
CamemBERT	87,8	92,1	65,7	93,3
BioCamemBERT				
avec 5k phrases	88,0	91,9	65,9	93,3
avec 50k phrases	88,2	92,6	66,1	93,4
avec 500k phrases	88,3	92,8	66,2	93,4
avec 1,5M phrases	88,3	92,6	66,3	93,5

4.4 Conclusion du chapitre

Dans cette évaluation de l'effet de différentes tailles de corpus médical pour l'enrichissement linguistique de FlauBERT et CamemBERT :

i. BioFlauBERT (resp. BioCamemBERT) a surclassé FlauBERT (resp. CamemBERT) lors de l'évaluation interne (après quelques milliers de phrases) et externe (après quelques centaines de milliers de phrases) ;

ii. l'efficacité de l'enrichissement linguistique augmente avec la taille du corpus médical ;

iii. BioFlauBERT (resp. FlauBERT) semble être plus efficace pour les tâches au niveau des jetons ;

iv. BioCamemBERT (resp. CamemBERT) semble être plus efficace pour les tâches au niveau de la phrase.

En conclusion, choisir le bon modèle de langue est indispensable pour maximiser les performances sur des tâches TALN médicales. Conformément aux conclusions précédentes, BioFlauBERT et BioCamemBERT seront privilégiés dans la suite de ce manuscrit.

Soumission dans la revue *Computer Methods and Programs in Biomedicine* :

Corpus size considerations in continual pre-training of BioFlauBERT and BioCamemBERT

Corpus size considerations in continual pre-training of BioFlauBERT and BioCamemBERT

Corentin Blanc [a-e], Alexandre Bailly [a-e], Élie Francis [a], Thierry Guillotin [a], Fadi Jamal [f], Pascal Roy [b-e]

[a] Everteam Software, Lyon, France

[b] Université de Lyon, Lyon, France

[c] Université Lyon 1, Villeurbanne, France

[d] Service de Biostatistique-Bioinformatique, Pôle Santé Publique, Hospices Civils de Lyon, Lyon, France

[e] Équipe Biostatistique-Santé, Laboratoire de Biométrie et Biologie Évolutive, CNRS UMR 5558, Villeurbanne, France

[f] IzyCardio, Lyon, France

Corresponding author:

Corentin Blanc

Everteam Software

17 quai Joseph Gillet

F-69004, Lyon, France

blc.corentin@gmail.com

Abstract

Background and objective: Over the last decades, a tremendous number of various medical documents have been produced. Examining them, even for very specific purposes, requires a considerable amount of time. Natural Language Processing (NLP) tools might greatly shorten this time, especially through language models pre-trained (PTLMs) on massive volumes of text from general and specialized sources. Currently, FlauBERT and CamemBERT (PTLMs in French) underperform in the medical field. Continual pre-training (CPT) on medical texts is thus necessary. This work investigated the CPT corpus size needed to further develop and compare BioFlauBERT vs. BioCamemBERT (the medically-specialized versions of FlauBERT and CamemBERT).

Methods: Subcorpora of different sizes were used to pre-train BioFlauBERT and BioCamemBERT. The final performance was assessed through inner and outer evaluation using, respectively, perplexity and cross-entropy, then F1-scores on a set of four NLP tasks (intent determination, semantic similarity, named entity recognition, and negation and scope detection).

Results: BioFlauBERT and BioCamemBERT performed better than their counterparts FlauBERT and CamemBERT whatever the CPT corpus size or the type of evaluation (inner or outer) and performance increased with the CPT corpus size. Furthermore, choosing the right French PTLM depended on the level at which the classification task is made: BioFlauBERT was more efficient in token classification tasks, whereas BioCamemBERT was more efficient in sentence classification tasks.

Conclusion: Up to 1.5 million sentences, BioFlauBERT and BioCamemBERT perform better with higher pre-training corpus sizes. For optimal performance in medicine, BioFlauBERT and BioCamemBERT are respectively recommended for token and sentence classification tasks.

Keywords

BioFlauBERT; BioCamemBERT; Continual pre-training, Corpus size; Language model

1. Introduction

Over the last few decades, a tremendous number of medical documents have been produced (patient records, referral letters, medical visit or operative reports, test results, etc.) whose consultation by health care professionals for various purposes (annual reviews, setting up cohort studies, etc.) became arduous and time-consuming. To handle these documents in reasonable times, health care professionals started being interested in Natural Language Processing (NLP) tools; i.e., artificial intelligence technologies that allow computers to process human language.

Recent advances in NLP have been made with Deep Learning, especially, pre-trained language models (PTLMs) based on Transformer encoders [1] such as BERT (Bidirectional Encoder Representation Transformers) [2] for English language or FlauBERT [3] and CamemBERT [4] for French language. PTLMs stem from a new NLP paradigm that allows assigning weights to the language model then use them for downstream tasks. Language modeling (or pre-training) is carried out by a NLP task such as Masked Language Modeling (to predict randomly masked words in a sentence) or Next Sentence Prediction (to check whether two sentences are consecutive) using a massive volume of text from all-purpose documents. Afterwards, a PTLM undergoes two evaluations: an evaluation of language modeling (called inner or intrinsic evaluation) and an evaluation of the performance on downstream NLP tasks (called outer or extrinsic evaluation). However, a PTLM designed for all-purpose documents often underperforms in the medical domain [5, 6] because of a great number of misunderstood terms and acronyms and, sometimes, because of different syntaxes or semantics. This underperformance led the scientific community to build PTLMs specialized in the medical domain.

Building a PTLM for the medical domain may use either of two strategies: pre-training from scratch (PTS) and continual pre-training (CPT). PTS consists in attributing the language

model random weights and pre-training it on medical texts (with or without non-medical texts). However, PTS is computationally expensive and requires a great number of medical texts. For example, BioElectra [7] (a medical PTLM based on Electra structure [8]) was pre-trained from scratch using PubMed full-text articles. On the contrary, CPT consists in attributing the language model weights from a pre-existing model and then improving it with further pre-training on medical texts. Interestingly, CPT requires fewer medical texts and less computation resources. For example, BioALBERT [9] is a medical PTLM initialized with ALBERT's weights [10] then pre-trained on PubMed full-text articles. Although some studies have shown that PTS can be superior to CPT [11], CPT remains the most sober approach in terms of computational and textual resources.

In the literature, except for studies on multilingual models such as mBERT [12], no study has focused on the CPT of French PTLMs based on Transformer encoders (such as FlauBERT or CamemBERT), on their inner or outer evaluation, or on the effect of the CPT corpus size. Only Copora et al. [13], in 2020, introduced CamemBERT-Bio (or CamemBERT pre-trained on hundreds of millions of sentences from PubMed abstracts) and showed the superiority of CamemBERT-Bio over CamemBERT in terms of F1-score in a named entity recognition (NER) task, but these authors addressed neither the inner evaluation nor the effect of the CPT corpus size.

In English, several language models have been pre-trained in the medical domain then submitted to an outer evaluation. In 2019, Lee et al. [5] and Huang et al. [14] developed BioBERT and ClinicalXLNeT, respectively, using BERT and XLNet [15] pre-trained on one million sentences from PubMed articles and MIMIC-III corpus [16]. These two medical PTLMs achieved respectively better results than BERT (in terms of F1-score in a NER task) and XLNet (in terms of area under the curve in predicting > 7 days mechanical ventilation and 90-day mortality). Since 2020, CPT corpora started reaching colossal amounts of sentences. In

2020, Gururangan et al. [17] built BioRoBERTa using RoBERTa language model [18] and hundreds of millions of sentences from Semantic Scholar Open Research Corpus (S2ORC) [19]; and, in 2021, Phan et al. [20] built SciFive using T5 language model [21] and hundreds of millions of sentences from PubMed abstracts. Both BioRoBERTa and SciFive achieved better results than RoBERTa and T5, respectively, in various tasks such as sentence and document classification, relation extraction, or NER. Over several years, despite increases in CPT corpora sizes, the same result kept being reported: a domination of PTLMs over general models, but according to outer evaluation only. Besides, up to now, the effects of the CPT corpus sizes on the medical enhancement of FlauBERT and CamemBERT underwent neither inner nor outer evaluation, and no comparisons.

This work carried out an inner evaluation, an outer evaluation, and comparisons of the effects of the CPT corpus size on the performance of BioFlauBERT and BioCamemBERT, the FlauBERT-based and CamemBERT-based language models pre-trained in the medical domain.

2. Materials and methods

2.1. The continual pre-training

2.1.1. The data

The present study used for CPT the French CLEAR corpus [22] which contains extracts from encyclopedias, drug leaflets, and scientific summaries. From this corpus, sentences were cut into words and subwords (also called tokens) using FlauBERT's and CamemBERT's vocabularies and tokenizers (Byte Pair Encoding [23] and SentencePiece [24], respectively). With this process and both tokenizers, too short and too long sentences (< 16 and > 64 tokens, respectively) were filtered out to limit silence and noise. Thus, the retained sentences were split into a CPT of 1.5 million sentences and an inner evaluation corpus of 10 thousand sentences. Finally, different CPT subcorpora with sizes 5, 50, 500, and 1,500 thousand sentences were drawn at random (without replacement) and each of the latter three subcorpora included the sentences of the previous subcorpus.

2.1.2. The pre-training procedure

BioFlauBERT and BioCamemBERT were respectively initialized with FlauBERT's and CamemBERT's structures and weights, as in the work of Le et al. [3] and Martin et al. [4]. For both PTLMs, the CPT was carried out by a Masked Language Modeling task and the following masking strategy was applied: from each sentence, 12% of the tokens were randomly masked and 1.5% randomly exchanged with tokens from the considered PTLM's vocabulary to disrupt the pre-training and avoid overfitting; the other tokens were kept unchanged. To strengthen this random masking step, stop words (prepositions, articles, or pronouns–which hold no medical information) were not masked to focus only on the medical tokens. This way, a language-modeling layer of as many neurons as the size of the considered PTLM's vocabulary combined

to a softmax activation was attached at the top of the PTLM, then cross-entropy (CE) was used as loss function for backpropagation.

2.1.3. The hyperparameters

AdamW learning algorithm [25] was used for backpropagation and most hyperparameters were selected a priori using a grid search from sets of possible values: i) a fixed learning rate (i.e., the magnitude of weight update) from {2 x 10^{-4}, 2 x 10^{-5}, 2 x 10^{-6}}; ii) the batch size (i.e., the number of sentences propagated at the same time through the model) from {32, 64, 128, 150}.

The only hyperparameter set a posteriori was the number of epochs (i.e., the number of times the subcorpus was explored), which depends on the convergence of the CE. The masking strategy was repeated at the beginning of each epoch.

2.2. The inner evaluation

2.2.1. Cross-entropy

The first criterion used for inner evaluation was CE. For a given sentence, cross-entropy evaluates the degree of discordance between token distributions as predicted by a PTLM and token distributions as observed in the inner evaluation corpus. The same above-described masking strategy was applied to each sentence of the inner evaluation corpus.

When T_n denotes the length of the n-th sentence of an inner evaluation corpus of size N, the CE may be expressed:

$$L_{CE} = \frac{-1}{N} \sum_{n=1}^{N} \left[\frac{1}{T_n} \sum_{t=1}^{T_n} \sum_{v=1}^{V} \mathbb{1}_{t,v} \cdot log(p_{t,v}) \right]$$

where V is the vocabulary size, $\mathbb{1}_{t,v}$ a binary indicator (= 1 when the v-th token of the vocabulary is the t-th token of the sentence, 0 otherwise), and $p_{t,v}$ the probability that the v-th token of the vocabulary is the t-th token of the sentence according to the PTLM.

2.2.2. Perplexity

The second criterion used for inner evaluation was perplexity (PP). PP evaluates the average hesitation of a PTLM about each token of the inner evaluation corpus. Therefore, the closer PP is to 1, the more understandable is the text for the PTLM. When $x^n = (x_t^n)_{1 \le t \le T_n}$ denotes the n-th sentence of length T_n of an inner evaluation corpus of size N, PP may be expressed:

$$PP = \exp\left(\frac{-1}{\sum_{n=1}^{N} T_n} \sum_{n=1}^{N}\left[\log P(x_1^n) + \sum_{t=2}^{T_n} \log P(x_t^n \,|x_1^n, \ldots, x_{t-1}^n)\right]\right)$$

2.3. The outer evaluation

The outer evaluation used two sentence classification tasks (sentence level) and two token classification tasks (token level). For each task, both BioFlauBERT and BioCamemBERT were fine-tuned; i.e., the language-modeling layer used for pre-training was replaced by a new prediction layer then the PTLM and the prediction layer weights were trained end-to-end. AdamW learning algorithm [25] was used for backpropagation with a scheduled learning rate decreasing linearly from 2 x 10^{-5} (beginning of the first epoch) to 0 (end of the last epoch) to force convergence of the CE and avoid 'catastrophic forgetting' [26] (i.e., disruption or erasure of BioFlauBERT's knowledge during fine-tuning). All other hyperparameters were selected a priori using a grid-search from sets of possible values: the batch size from {1, 4, 8, 16, 32} and the number of epochs from {2, 5, 10}. Finally, performance was evaluated using a bootstrap method [27] consisting in drawing randomly sentences from a dataset, with replacement to keep constant the size of the training set. Sentences not drawn formed the test set. Theoretically, about 37% of the dataset sentences remained in the test set. This bootstrapping was stratified and fifty iterations were run to obtain means and standard deviations for the Macro F1-score.

2.3.1. Intent determination

Intent determination was carried out using the QA dialogue corpus [28] which consists in 1,818 sentences taken from conversations between patients and health care professionals. Each sentence was tagged manually regarding the type of information (called “intent” in the context of a virtual agent) according to the following possibilities: aim of the consultation, personal data, medical history, symptoms, lifestyle habits, treatments, etc. The final objective was to assign each sentence the right intent.

2.3.2. Semantic similarity

Semantic similarity was conducted on CLISTER corpus [29] which includes 1,000 pairs of sentences from case reports that include medical history, treatment, follow-up, etc. Each pair of sentences was annotated with a similarity score that ranged between 0 (no similarity) and 5 (maximum similarity). The final objective was to assign each pair of sentences a similarity score.

2.3.3. Named entity recognition

NER was carried out using QUAERO corpus [30] which contains 4,424 medical sentences extracted from information on marketed drugs from the European Medicines Agency, titles of research articles indexed in MEDLINE database, and patents registered with the European Patent Office. Each sentence was tagged using ten named entity categories (chemical and drugs, anatomy, disorders, devices, procedures, geographic areas, living beings, objects, phenomena, and physiology). The final objective was to locate named entities within sentences.

2.3.4. Negation and scope detection

Negation and scope detection were carried out using CAS corpus [31] which includes 11,037 sentences from case reports extracted from the specialized literature. Being a special case of NER, the final objective was to locate negations and scopes within a sentence.

2.4. Implementation details

This work used Python 3.6.8 as programming language and the following packages for specific tasks: Torchtext 0.11.0 to load and tokenize the corpora; Transformers 4.12.3 from HuggingFace to handle BioFlauBERT and BioCamemBERT; and PyTorch 1.10.0 to deal with the CPT, the inner evaluation, and the outer evaluation.

3. Results

3.1. The inner evaluation

According to the inner evaluation, both CE and PP decreased progressively with the increase of the CPT corpus size and reached their minimum values after pre-training with 1,500,000 sentences (0.04 CE and 1.10 PP for BioFlauBERT and 0.06 CE and 1.92 PP for BioCamemBERT (Table 1). After pre-training on only a few thousand sentences, BioFlauBERT performed better than FlauBERT (0.14 vs. 1.05 for CE and 1.54 vs. 3.02 for PP) and BioCamemBERT better than CamemBERT (0.22 vs. 2.75 for CE and 2.78 vs. 3.24 for PP).

Table 1 – Cross-entropy and perplexity stemming from the inner evaluation.

Language model	Cross-entropy *	Perplexity *
FlauBERT	1.05	3.02
BioFlauBERT		
with 5,000 sentences	0.14	1.54
with 50,000 sentences	0.13	1.21
with 500,000 sentences	0.06	1.11
with 1,500,000 sentences	0.04	1.10
CamemBERT	2.75	3.24
BioCamemBERT		
with 5,000 sentences	0.22	2.78
with 50,000 sentences	0.20	2.52
with 500,000 sentences	0.07	2.06
with 1,500,000 sentences	0.06	1.92

* Higher values indicate worse performance.

3.2. The outer evaluation

According to the outer evaluation, all Macro F1-scores increased progressively with the size of the CPT corpus and often reached their maxima at 500,000 or 1,500,000 sentences whatever the downstream task or corpus (87.7, 92.5, 68.9, and 93.8 for BioFlauBERT and 88.3, 92.8, 66.3, and 93.5 for BioCamemBERT) (Table 2). Unlike what was seen in the inner evaluation, the performance of each model depended on the level of the task. At the sentence level and whatever the CPT corpus size, BioCamemBERT outperformed BioFlauBERT (88.3 vs. 87.7 with QA dialogue and 92.8 vs. 92.3 with CLISTER). However, at the token level and whatever the corpus size, it was BioFlauBERT that outperformed BioCamemBERT (68.9 vs. 66.3 with QUAERO and 93.8 vs. 93.5 with CAS).

Table 2 – Macro F1-scores stemming from the outer evaluation.

Language model	Sentence level		Token level	
	QA dialogue	CLISTER	QUAERO	CAS
FlauBERT	84.6*	91.0	68.2	93.4
BioFlauBERT				
with 5,000 sentences	85.6*	91.0	68.3*	93.4*
with 50,000 sentences	87.0*	91.4	68.1	93.6*
with 500,000 sentences	87.4	92.5	68.6	93.7
with 1,500,000 sentences	87.7	92.3*	68.9*	93.8
CamemBERT	87.8	92.1*	65.7	93.3
BioCamemBERT				
with 5,000 sentences	88.0*	91.9	65.9*	93.3
with 50,000 sentences	88.2*	92.6*	66.1*	93.4
with 500,000 sentences	88.3	92.8*	66.2	93.4
with 1,500,000 sentences	88.3*	92.6*	66.3	93.5

* Standard deviation range: 0.3 – 0.5. All other standard deviations are lower than 0.2.

4. Discussion

As expected from the literature, in the medical domain and whatever the CPT corpus size or the type of evaluation (inner or outer), the enhanced PTLMs BioFlauBERT and BioCamemBERT achieved better results than their counterparts FlauBERT and CamemBERT and their performance increased together with the CPT corpus size.

In the inner evaluation, both CE and PP showed the efficiency of the enhanced PTLMs in medical language modeling. Surprisingly, this modeling started being very efficient with only a few thousand sentences then continued to increase progressively with the size of the CPT corpus until reaching minima obtained with 1 500 000 sentences. Though inner evaluation does not always guarantee good results on downstream tasks, it remains a useful, if not a necessary, criterion to estimate the hyperparameters and the order of magnitude of the optimal number of sentences.

In the outer evaluation, the enhanced PTLMs performed also better than their counterparts and their performance increased together with the CPT corpus size. Nevertheless, the corpus sizes used were much smaller than those used in other studies. Indeed, as medical corpus sizes were growing over the last few years (from several hundred thousand sentences in 2019 [5, 14] to hundreds of millions in 2021 [17, 20]), it would be interesting to reproduce this study in the future to see which level performance may reach.

The variety of medical downstream tasks showed that BioFlauBERT performed the best at the sentence level and BioCamemBERT at the token level. In the current literature, few studies comparing FlauBERT and CamemBERT do validate this choice based on the F1-score (see Supplementary Table 1). One plausible explanation for BioFlauBERT better performance in token-level classification tasks would be its larger vocabulary, which increases the number of parameters, deals better with the input sentences, and provides better contextual representations. On the contrary, the extra pooling linear layer into BioCamemBERT structure ensures a better

representation of a whole sentence and thus a better performance in sentence-level classification tasks. Confirming this choice requires additional investigations.

The effectiveness of medical language models applied to medical downstream tasks is no longer a concern but the performance of these models on downstream tasks in non-medical domains remains unexplored. In 2020, together with designing FlauBERT, Le et al. [3] introduced an evaluation called FLUE (French Language Understanding Evaluation) by grouping a set of tasks such as part-of-speech tagging, text classification, natural language inference, or word sense disambiguation. Thus, a future natural and interesting development of the present work will be assessing the performance of BioFlauBERT and BioCamemBERT using FLUE.

One merit of the present work is the addition to the current literature of new results relative to the performance of BioFlauBERT and BioCamemBERT in the medical domain. To the best of our knowledge, this kind of comparative study has been rarely conducted in French language; in fact, there are few medical PTLMs available whereas health care professionals are increasingly interested in new NLP technologies to support their research projects. Another merit is the finding of a new way for choosing the right medical PTLM for the right task according to the level at which the classification is made. At present, the latter finding is a suggestion because more studies are required to confirm it in various settings.

5. Conclusion

In this assessment of the effect of different CPT corpus sizes on the continual pre-training of BioFlauBERT and BioCamemBERT: i) BioFlauBERT and BioCamemBERT achieved better performance than FlauBERT and CamemBERT, respectively, regardless of the CPT corpus size and the type of evaluation (inner or outer); ii) the performance increased with the size of the CPT corpus; iii) BioFlauBERT seemed to be more efficient in token-level classification tasks,

whereas BioCamemBERT seemed to be more efficient in sentence-level classification tasks. This demonstrated that choosing the right PTLM is important to maximize performance in carrying out medical classification tasks.

Authors' contributions

Conceptualization: CB, AB, EF, TG, FJ, and PR.

Data curation: CB.

Formal analysis: CB.

Funding acquisition: EF, FJ, TG, and PR.

Investigation: CB.

Methodology: CB, AB, EF, TG, and PR.

Software: CB.

Supervision: EF, FJ, TG, and PR.

Validation: CB.

Visualization: CB.

Writing - original draft: CB.

Writing - review & editing: CB, AB, EF, TG, FJ, and PR.

Acknowledgments

The authors are grateful to Jean Iwaz (Hospices Civils de Lyon) for augmenting, editing, proofreading, and formatting the latest versions of the manuscript.

Funding

This work was supported by Association Nationale de la Recherche et de la Technologie (ANRT) [grant number 2019/1374]. The sponsor had no role in the study design; the collection, analysis, and interpretation of the data; the writing of the report; and the decision to submit the article for publication.

Conflicts of interest

None.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin. Attention Is All You Need. The 31st Conference on Neural Information Processing Systems, (2017), pp. 1–11. http://arxiv.org/abs/1706.03762.

[2] J. Devlin, M-W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1 (2019), pp. 4171-4186. http://aclweb.org/anthology/N19-1423.

[3] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, D. Schwab. FlauBERT: Unsupervised Language Model Pre-training for French. Proceedings of the 12th Language Resources and Evaluation Conference, (2020), pp. 2479-2490. https://aclanthology.org/2020.lrec-1.302.

[4] L. Martin, B. Muller, P.J.O. Suarez, Y. Dupont, L. Romary, V. de la Clergerie, D. Seddah, B. Sagot. CamemBERT: a Tasty French Language Model. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, (2020), pp. 7203-7219. https://doi.org/10.18653/v1/2020.acl-main.645.

[5] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang. BioBERT: a pretrained biomedical language representation model for biomedical text mining. Bioinformatics 2020; 4:1234-1240. https://arxiv.org/abs/1901.08746.

[6] I. Beltagy, A. Cohan, K. Lo. SciBERT: Pretrained Contextualized Embeddings for Scientific Text. Proceedings of the Second Workshop on Scholarly Document Processing, (2020), pp. 130–133, (2019). https://arxiv.org/abs/1903.10676.

[7] K. Kanakarajan, B. Kundumani, M. Sankarasubbu. BioELECTRA: Pretrained Biomedical text Encoder using Discriminators. Proceedings of the 20th Workshop on

Biomedical Language Processing, Association for Computational Linguistics, (2021), pp. 143–154. https://aclanthology.org/2021.bionlp-1.16.pdf.

[8] K. Clark, M.-T. Luong, Q. Le, C. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. International Conference on Learning Representations, (2020). https://arxiv.org/abs/2003.10555.

[9] U. Naseem, M. Khushi, V. Reddy, S. Rajendran, I. Razzak, J. Kim. BioALBERT: A Simple and Effective Pre-trained Language Model for Biomedical Named Entity Recognition. International Joint Conference on Neural Networks (2021). https://arxiv.org/abs/2009.09223.

[10] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. International Conference on Learning Representations, (2020). https://arxiv.org/abs/1909.11942.

[11] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing, (2021). https://arxiv.org/pdf/2007.15779.pdf

[12] T. Pires, E. Schlinger, D. Garrette. How multilingual is Multilingual BERT? Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, (2019). pp. 4996-5001. https://aclanthology.org/P19-1493/.

[13] J. Copara, J. Knafou, N. Naderi, C. Moro, P. Ruch, D. Teodoro. Contextualized French Language Models for Biomedical Named Entity Recognition. 6e conférence conjointe Journées d'Etudes sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL, 22e édition). Atelier Défi Fouille de Textes, ATALA, (2020), pp. 36–48. https://hal.archives-ouvertes.fr/hal-02784740.

[14] K. Huang, A. Singh, S. Chen, E. Moseley, C. Deng, N. George, C. Lindvall. Clinical XLNet: Modeling Sequential Clinical Notes and Predicting Prolonged Mechanical Ventilation, (2019), pp. 94-100. https://arxiv.org/abs/1912.11975.

[15] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. Proceedings of the 33rd International Conference on Neural Information Processing Systems, (2019), pp. 5753-5763. https://arxiv.org/abs/1906.08237.

[16] A. Johnson, T. Pollard, L. Shen, L. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Celi, R. Mark. MIMIC-III, a freely accessible critical care database, Scientific Data 3, (2016). https://www.nature.com/articles/sdata201635.

[17] S. Gururangan, A. Marasovic, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N. Smith. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, (2020), pp. 8342-8360. https://arxiv.org/abs/2004.10964.

[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. Proceedings of the 20th Chinese National Conference on Computational Linguistics, (2019), pp. 1218-1227. https://arxiv.org/abs/1907.11692.

[19] K. Lo, L. Wang, M. Neumann, R. Kinney, D. Weld. S2ORC: The Semantic Scholar Open Research Corpus. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, (2020), pp. 4969-4983. https://aclanthology.org/2020.acl-main.447.

[20] L. Phan, J. Anibal, H. Tran, S. Chanana, E. Bahadroglu, A. Peltekian, G. Altan-Bonnet. SciFive: a text-to-text transformer model for biomedical literature, (2021). https://arxiv.org/abs/2106.03598.

[21] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. J Mach Learn Res; 21:1-67 (2020). https://arxiv.org/abs/1910.10683.

[22] N. Grabar, R. Cardon. CLEAR – Simple Corpus for Medical French. Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA), Association for Computational Linguistics, (2018), pp. 3–9. https://hal.archives-ouvertes.fr/halshs-01968355.

[23] R. Sennrich, B. Haddow, A. Birch. Neural Machine Translation of Rare Words with Subword Units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, (2016), pp. 1715–1725. https://doi.org/10.18653/v1/P16-1162.

[24] T. Kudo, J. Richardson. Sentence piece: a simple and language independent subword tokenizer and detokenizer for neural text processing, Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations, Association for Computational Linguistics (2018), pp. 66-71. https://aclanthology.org/D18-2012/

[25] I. Loshchilov, F. Hutter. Decoupled Weight Decay Regularization. 7th International Conference on Learning Representations. (2019), https://openreview.net/forum?id=Bkg6RiCqY7.

[26] R. French. Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences, (1999), pp. 128–135.

[27] B. Efron. Bootstrap Methods: Another Look at the Jackknife. Ann Stat, 7 (1979), pp. 1–26. https://doi.org/10.1214/aos/1176344552.

[28] A. Fréjus. A. Laleye, G. Chalendar, A. Blanié, A. Brouquet, D. Behnamou. A French Medical Conversations Corpus Annotated for a Virtual Patient Dialogue System.

In Proceedings of the Twelfth Language Resources and Evaluation Conference, (2020) pp 574–580. https://aclanthology.org/2020.lrec-1.72/

[29] N. Hiebel, K. Fort, A. Névéol, O. Ferret. CLISTER : un corpus pour la similarité sémantique textuelle dans des cas cliniques en français. Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles, (2022). pp 287–296. https://aclanthology.org/2022.jeptalnrecital-taln.28.pdf

[30] A. Névéol, C. Grouin, J. Leixa, S. Rosset, P. Zweigenbaum. The QUAERO French medical corpus: A ressource for medical entity recognition and normalization. Proc of BioTextMining Work, (2014), pp. 24–30.

[31] N. Grabar, V. Claveau, C. Dalloux. CAS: French corpus with clinical cases. Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis, Association for Computational Linguistics, (2018), pp. 122–128. https://doi.org/10.18653/v1/W18-5614.

Supplementary material

Supplementary table 1 – Selected FlauBERT vs. CamemBERT comparisons in the literature.

Task	Level	Domain	Best model
Intent and slot prediction [1]	Token	Medical	FlauBERT
Text classification [2]	Sentence	Weather	CamemBERT
Text classification [3]	Sentence	Medical	CamemBERT
Text classification [4]	Sentence	Medical	CamemBERT
Sentiment analysis [5]	Sentence	Catering	CamemBERT
Sentiment analysis [5]	Sentence	Museum	FlauBERT
Text classification [6]	Sentence	E-commerce	CamemBERT
Paraphrasing [6]	Sentence	General	CamemBERT
Natural language inference [6]	Sentence	General	CamemBERT
Part-of-speech tagging [6]	Token	Newspaper	FlauBERT
Dependency parsing [6]	Token	Newspaper	FlauBERT
Word Disambiguation [6]	Token	General	CamemBERT
Text classification [7]	Sentence	Cooking	CamemBERT
Text classification [7]	Sentence	Cooking	CamemBERT

[1] C. Blanc, A. Bailly, Elie Francis, T. Guillotin, F. Jamal, B. Wakim, P. Roy. FlauBERT vs. CamemBERT: Understanding patient's answers by a French medical chatbot. Artif Intell Med, (2022).

[2] S. Cerna, C. Guyeux, D. Laiymani. The usefulness of NLP techniques for predicting peaks in firefighter interventions due to rare events. Neural Comput & Applic, (2022). pp. 10117–10132 (2022).

[3] G. Chenais, C. Gil-Jardiné, H. Touchais, M. Avalos Fernandez, B. Contrand, E. Tellier, X. Combes, L. Bourdois, P. Revel, E. Lagarde. Deep Learning Transformer Models for Building a Comprehensive and Real-time Trauma Observatory: Development and Validation Study. JMIR AI, (2023).

[4] G. Chenais, H. Touchais, M. Avalos, L. Bourdois, P. Revel, C. Gil-Jardiné, E. Lagarde. Performance of BERT models for French in the classification of textual data from emergency room visits. Plate-Forme Intelligence Artificielle (PFIA). (2021).

[5] A. Essebbar, B. Kane, O. Guinaudeau, V. Chiesa, I. Quénel, S. Chau. Aspect Based Sentiment Analysis using French Pre-Trained Models. International Conference on Agents and Artificial Intelligence. (2021).

[6] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, D. Schwab. FlauBERT: Unsupervised Language Model Pre-training for French. Proceedings of the 12th Language Resources and Evaluation Conference, (2020), pp. 2479-2490.

[7] E. Mohammadi, L. Marceau, E. Charton, L. Kosseim, L. Nerima, M-J. Meurs. Du bon usage d'ingrédients linguistiques spéciaux pour classer des recettes exceptionnelles. 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants

Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). (2020). pp.81-94.

Chapitre 5

Enrichissement cognitif

Les enrichissements linguistiques du chapitre précédent ont été un succès et ont permis le développement de BioFlauBERT et BioCamemBERT. Toutefois, la médecine est un domaine vaste et fourmillant de connaissances rarement prises en compte par les modèles existants. Pour remédier à ce manque, l'objectif de ce chapitre est d'enrichir cognitivement le modèle de langue BioFlauBERT. Cet enrichissement a fait l'objet d'une soumission dans la revue *International Journal of Medical Informatics* intitulée :

Incorporating the CIM-10 Hierarchy into the Hypothesis Set and the Learning Algorithm for Hierarchical Text Classification

Sommaire

5.1 Contexte

La classification hiérarchique de textes [38] est une tâche bien connue de TALN qui consiste à attribuer une ou plusieurs catégories issues d'une taxonomie [1] à un texte. Dans la pratique, cette classification est souvent effectuée grâce à une approche plate ou globale [39]. L'approche plate consiste à prédire une catégorie au plus bas niveau puis à accéder aux catégories de niveau supérieur grâce à la taxonomie ; garantissant une bonne cohérence hiérarchique. A contrario, l'approche globale consiste à entraîner un unique modèle à prédire toutes les catégories de tous les niveaux en même temps ; entraînant parfois des incohérences hiérarchiques. L'idée fondamentale derrière l'approche globale est que la prédiction simultanée de tous les niveaux prend implicitement en compte la taxonomie. Cependant, cette taxonomie n'est jamais explicitement intégrée dans le modèle comme une connaissance a priori. Plusieurs stratégies sont envisageables pour réaliser une telle incorporation dans un modèle de langue médical tel que BioFlauBERT [40] :

- L'incorporation dans l'ensemble d'hypothèses (Figure 5.1) qui consiste à faire coïncider la taxonomie dans la structure du modèle. De cette manière, un réseau neuronal basé sur des connaissances ayant la particularité d'avoir des connexions volontairement supprimées permet une telle incorporation.

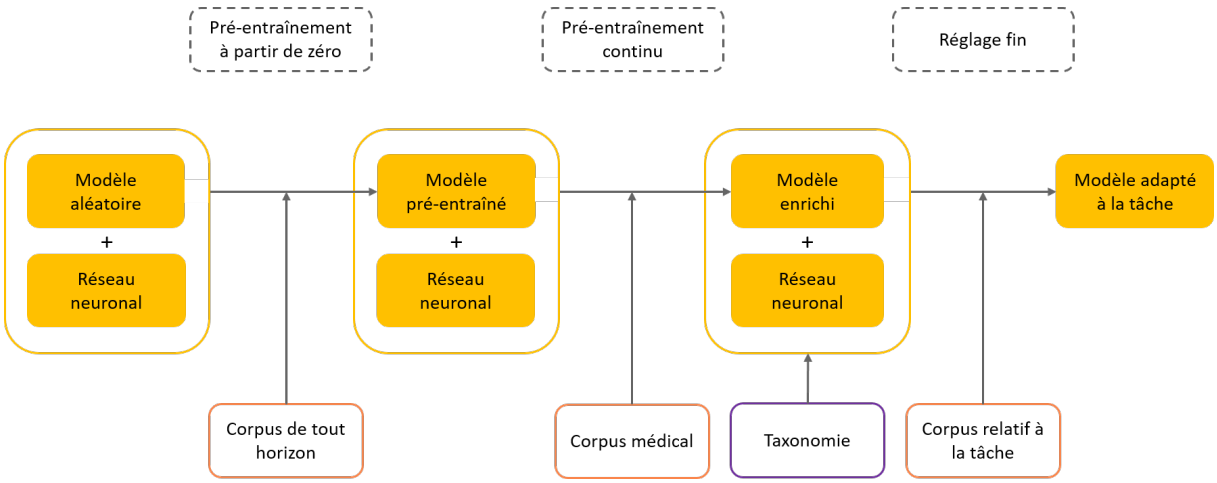


FIGURE 5.1 – Incorporation d'une taxonomie dans l'ensemble d'hypothèses du modèle.

1. Arborescence de catégories sur plusieurs niveaux

- L'incorporation dans l'algorithme d'apprentissage (Figure 5.2) qui consiste à ajouter un terme supplémentaire à la fonction de perte afin de pénaliser le modèle lorsque des incohérences hiérarchiques surviennent dans les prédictions. Souvent, un hyperparamètre d'incorporation est associé à ce nouveau terme afin d'ajuster sa contribution.

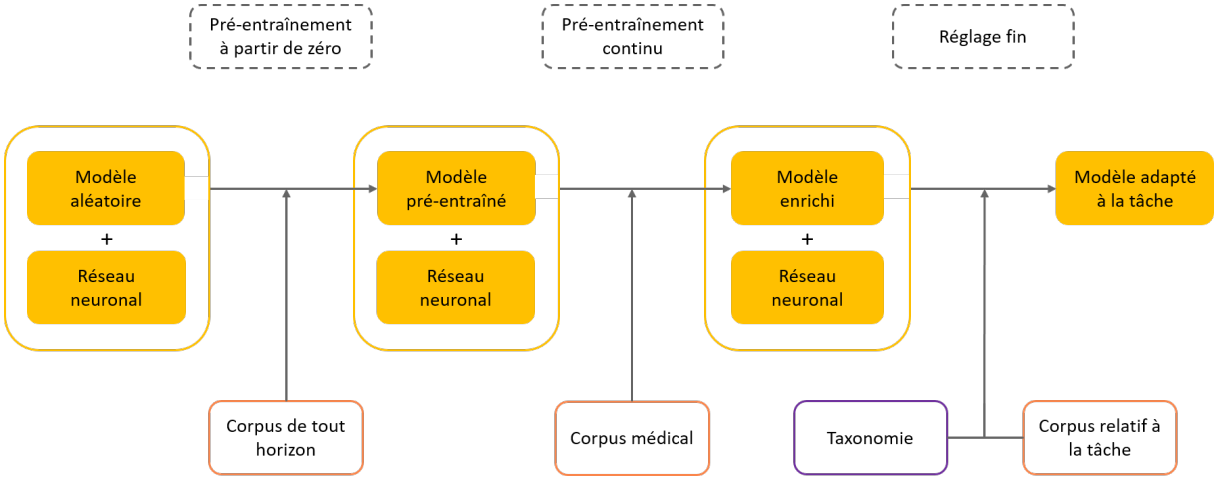


FIGURE 5.2 – Incorporation d'une taxonomie dans l'algorithme d'apprentissage.

- Une incorporation hybride consistant à fusionner les deux précédentes :

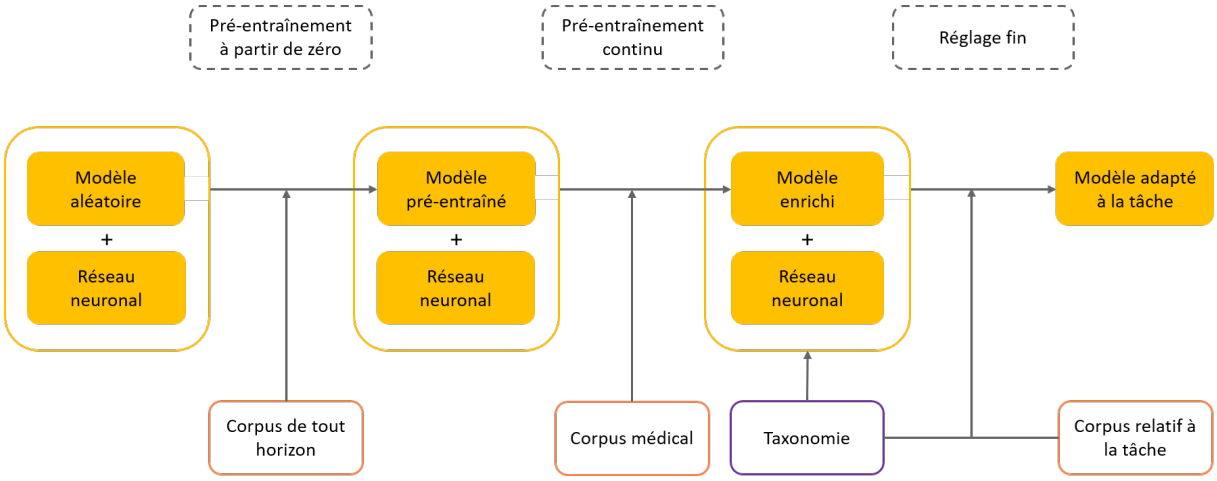


FIGURE 5.3 – Incorporation hybride d'une taxonomie dans l'ensemble d'hypothèses et l'algorithme d'apprentissage.

Dans le cadre d'une prédiction hiérarchique de causes primaires de décès, différentes approches incorporant ou non la taxonomie de la classification internationale des maladies (CIM-10) [41] sont comparés (Table 5.1) :

- dans l'ensemble d'hypothèses (à travers respectivement des réseaux neuronaux linéaires et basés sur des connaissances) ;
- dans l'algorithme d'apprentissage (à travers l'ajustement d'un hyperparamètre d'incorporation noté α) ;
- dans les deux.

Une comparaison a également été effectuée avec une approche plate jouant le rôle de base de référence.

TABLE 5.1 – Récapitulatif des incorporations étudiées.

Approche	**Réseau neuronal**	**Hyperparamètre**
Sans incorporation	Linéaire	$\alpha = 1,0$
Avec incorporation		
Dans le jeu d'hypothèse	Basé sur des connaissances	$\alpha = 1,0$
Dans l'algorithme d'apprentissage	Linéaire	$0,0 \leq \alpha < 1,0$
Dans les deux	Basé sur les connaissances	$0,0 \leq \alpha < 1,0$

5.2 Méthodologie

5.2.1 Corpus

Afin de prédire hiérarchiquement les causes primaires de décès, un corpus composé de 129 149 certificats de décès émis sur la période 2006-2015 par le CépiDC [42] a été utilisé (avec l'aimable autorisation de l'Institut National de la Santé et de la Recherche Médicale). Chaque certificat comprenait jusqu'à quatre lignes de texte libre et la cause primaire de décès spécifiée par un code de quatrième niveau dans la CIM-10. Avec ce code, il était possible d'accéder aux trois niveaux les plus élevés afin de réaliser une classification hiérarchique de causes primaires de décès. Cette accession a conduit à 2 394 codes de niveau 4, 935 codes de niveau 3, 154 codes de niveau 2 et 18 codes de niveau 1 (notés respectivement de C^4 à C^1). Un exemple simple des niveaux de la CIM-10 est donné dans la Figure 5.4.

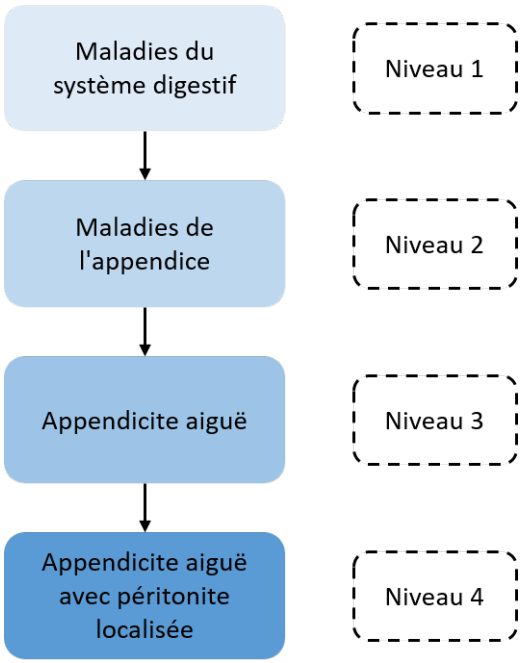


FIGURE 5.4 – Exemple des quatre premiers niveaux de la CIM-10.

5.2.2 Architecture proposée

L'architecture était composée de deux blocs principaux (Figure 2) : BioFlauBERT calculait le plongement contextuel $z_1^L \in R^D$ pour la globalité du certificat de décès (grâce au jeton artificiellement ajouté en début de séquence) puis une pile constituée de quatre réseaux neuronaux prédisait la cause primaire de décès à chacun des quatre niveaux ciblés de la CIM-10. Chaque réseau prenait en entrée le plongement du certificat de décès ainsi que les scores de sortie du réseau précédent (excepté le premier qui ne prenait en entrée uniquement le plongement). Ici, deux types de réseaux neuronaux différents ont été utilisés dans la pile : des réseaux linéaires et des réseaux basés sur des connaissances.

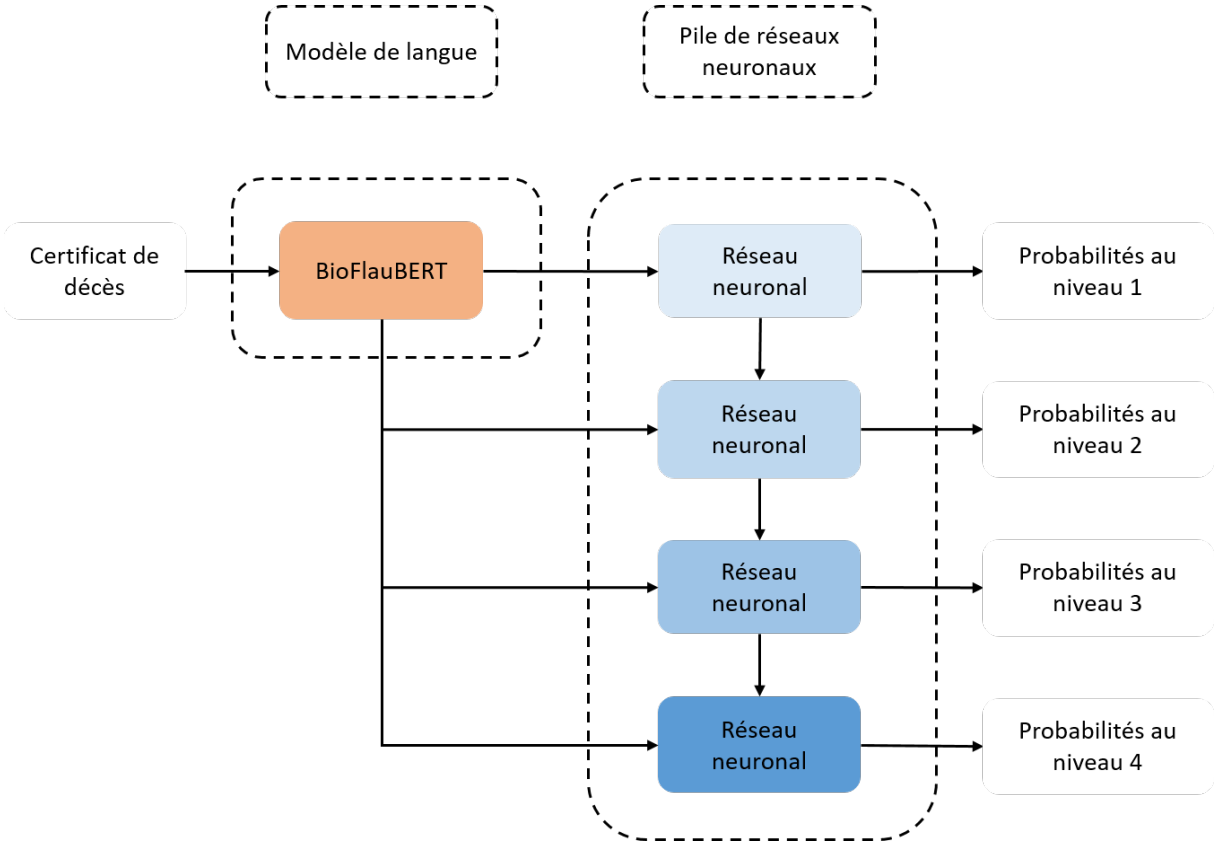


FIGURE 5.5 – Classification hiérarchique de causes primaires de décès abordée par une approche globale composée de BioFlauBERT et d'une pile de réseaux neuronaux. Tout d'abord, BioFlauBERT calculait le plongement contextuel du certificat de décès, puis les réseaux neuronaux prédisaient la cause primaire du décès à chacun des quatre niveaux ciblés de la CIM-10.

5.2.2.1 Réseau linéaire

Le réseau linéaire, défini ici comme une superposition de plusieurs couches linéaires sans fonction d'activation, ne tient pas compte de la CIM-10 dans l'ensemble d'hypothèses grâce à une interconnexion totale des neurones. Les scores $s^l \in \mathbb{R}^{C^l}$ pour chacun des quatre niveaux sont données par récurrence par :

$$\begin{cases} s^1 &= (z_1^L \cdot W_1^1 + b_1^1) \cdot W_2^1 + b_2^1 \\ s^l &= \left(\left[z_1^L \;||\; s^{l-1}\right] \cdot W_1^l + b_1^l\right) \cdot W_2^l + b_2^l \end{cases} \tag{5.1}$$

où $W_1^1 \in \mathbb{R}^{D \times D}$, $W_2^1 \in \mathbb{R}^{D \times C^1}$, $W_1^l \in \mathbb{R}^{(D+C^{l-1}) \times (D+C^{l-1})}$ et $W_2^l \in \mathbb{R}^{(D+C^{l-1}) \times C^{l-1}}$ sont les paramètres du réseau ; $b_1^1 \in \mathbb{R}^D$, $b_2^1 \in \mathbb{R}^{C^1}$, $b_1^l \in \mathbb{R}^{D+C^{l-1}}$ et $b_2^l \in \mathbb{R}^{C^{l-1}}$ les biais associés. Enfin, les probabilités $p^l \in [0;1]^{C^l}$ et les causes primaires de décès $\hat{y}^l$ de chaque niveau sont données par :

$$p^l = \text{softmax}(s^l) \tag{5.2}$$

$$\hat{y}^l = \underset{1 \leq c \leq C^l}{\text{argmax}} \, (p_c^l) \tag{5.3}$$

5.2.2.2 Réseau basé sur des connaissances

Contrairement au réseau linéaire, le réseau basé sur des connaissances permet de tenir compte de la CIM-10 en la faisant correspondre sur les connexions entre les réseaux neuronaux. Ainsi, les scores $s^l \in \mathbb{R}^{C^l}$ pour chacun des quatre niveaux sont données par :

$$\begin{cases} s^1 &= (z_1^L \cdot W_1^1 + b_1^1) \cdot W_2^1 + b_2^1 \\ s^l &= \left(\left[z_1^L \;||\; s^{l-1}\right] \cdot \left[M^l \odot W_1^l\right] + b_1^l\right) \cdot W_2^l + b_2^l \end{cases} \tag{5.4}$$

où $M^l \in \mathbb{R}^{(D+C^{l-1}) \times (D+C^{l-1})}$ est la matrice de correspondance du l-ème niveau (1 si la connexion dans la CIM-10 existe, 0 sinon) et $\odot$ le produit matriciel terme à terme. Enfin, les probabilités et les causes primaires de décès de chaque niveau sont données comme précédemment.

5.2.3 Fonction de perte

Une somme pondérée entre une perte hiérarchique L_h et une perte de cohérence L_c a été utilisée comme fonction de perte L dont l'écriture est donnée par :

$$L \;=\; \alpha \cdot L_h \;+\; (1-\alpha) \cdot L_c \tag{5.5}$$

où $\alpha \in [0;1]$ est l'hyperparamètre d'incorporation. Plus α est proche de 0 et plus la CIM-10 est incorporée dans l'algorithme d'apprentissage. Au contraire, elle est ignorée lorsque α est proche de 1.

5.2.3.1 Perte hiérarchique

La perte hiérarchique pénalise les discordances entre les distributions de probabilités prédites par l'architecture et celles observées dans le corpus aux quatre premiers niveaux de la CIM-10. Elle correspondait à une somme de quatre entropies croisées (une pour chacun des quatre niveaux). Pour un certificat de décès donné, cette perte est donnée par :

$$L_h \;=\; -\sum_{l=1}^{L}\sum_{i=1}^{C^l} \mathbb{1}_i^l \cdot \log(p_i^l) \tag{5.6}$$

où $\mathbb{1}_i^l$ est une indicatrice (1 lorsque la i-ème cause primaire de décès du niveau l correspond au certificat, 0 autrement) et p_i^l la probabilité que la i-ème cause primaire de décès du niveau l soit associée au certificat.

5.2.3.2 Perte de cohérence

La perte de cohérence pénalise les incohérences hiérarchiques entre tous les niveaux successifs de la CIM-10. Pour un certificat de décès donné, cette perte est donnée par :

$$L_c \;=\; \sum_{l=1}^{L-1}\sum_{i=1}^{C^l}\sum_{j=1}^{C^{l+1}} \mathbb{1}_{i,j} \cdot p_j^{l+1} \cdot (1-p_i^l) \tag{5.7}$$

où $\mathbb{1}_{i,j}$ est une indicatrice (1 lorsque la i-ème cause primaire de décès du niveau l et la j-ème cause de décès du niveau $l+1$ sont connectées dans la CIM-10, 0 sinon).

5.2.4 Hyperparamétrage

Pour entraîner conjointement la totalité de l'architecture, plusieurs hyperparamètres ont été sélectionnés grâce à une grille de recherche. En plus des hyperparamètres usuels, diverses valeurs de l'hyperparamètre d'incorporation α ont été testées tour à tour entre 0,0 et 1,0 inclus. Un récapitulatif complet de l'hyperparamétrage est donné dans la Table 5.2.

TABLE 5.2 – Récapitulatif de l'hyperparamétrage.

Hyperparamètre	**Valeur**
Nombre d'epochs	10
Taille des lots	16
Écrêtage du gradient	Aucun
Hyperparamètre d'incorporation	$[0,0;1,0]$
Algorithme d'apprentissage	AdamW
Taux d'apprentissage initial	2e-5
Variation du taux d'apprentissage	Linéaire
Taux d'apprentissage final	0,0
Stabilité numérique	1e-8
Décroissance du premier moment	0,9
Décroissance du second moment	0,999
Décroissance des paramètres	0,01

5.2.5 Critères d'évaluation

L'évaluation était portée par une méthode de bootstrap [27] à dix itérations afin d'obtenir les médianes des F1-scores pondérés pour chaque niveau ciblé de la CIM-10 et d'un score de cohérence hiérarchique correspondant au pourcentage de bonnes connexions entre les niveaux. Le F1-score pondéré a été choisi en raison du très grand nombre de causes primaires de décès ayant moins d'une dizaine de représentations dans le corpus et posant énormément de problèmes aux modèles de langue. Bien connu dans la littérature, cette faiblesse a été mise en avant dans la publication *Classification multi-label de cas cliniques avec CamemBERT* que j'ai moi-même présenté lors du Défi Fouille de Textes organisé sous l'égide de l'Association pour le Traitement Automatique des Langues (ATALA) (Annexe C).

5.3 Résultats

La Table 5.3 présentent les F1-scores pondérés et le score de cohérence hiérarchique obtenus pour la classification hiérarchique de causes primaires de décès avec les réseaux linéaires. Lorsque $\alpha = 1,0$, aucune incorporation n'était effectuée et la comparaison se réduisait à une approche plate vs. globale. Comme attendu dans la littérature [43,44], l'approche globale a surpassé l'approche plate en terme de F1-score grâce à la pile de réseaux linéaires permettant une prise en compte implicite de la CIM-10. Lorsque $0,0 \leq \alpha < 1,0$, la CIM-10 était incorporé uniquement dans l'algorithme d'apprentissage et l'hyperparamètre α jouait un rôle essentiel. Plus α diminuait de 1,0 vers 0,2 et plus les F1-scores et le score de cohérence hiérarchique augmentaient jusqu'à atteindre des maxima pour les F1-scores lorsque $\alpha = 0,2$. Au-delà de cette valeur, les F1-scores ont chuté à zéro en raison de la fonction de perte qui a totalement ignorée les prédictions et s'est focalisée uniquement sur la CIM-10 ; conduisant à toujours la même prédiction et à un score de cohérence hiérarchique parfait.

A l'instar de la table précédente, la Table 5.4 présente les F1-scores pondérés et les scores de cohérence hiérarchique obtenus pour la classification hiérarchique de textes avec cette fois-ci les réseaux basés sur des connaissances. Lorsque $\alpha = 1,0$ et $0,0 \leq \alpha < 1,0$, la CIM-10 était respectivement incorporée uniquement dans l'ensemble d'hypothèses du modèle et de manière hybride. Dans les deux cas, les performances étaient restées très similaires que dans le cas précédent. Pire, des instabilités plus marquées étaient présentes au vue des étendues observées (différences entre les maxima et les minima) probablement liées à la suppression de certaines connexions pour faire coïncider la CIM-10 dans les réseaux neuronaux. À noter cependant une petite hausse du score de cohérence hiérarchique.

TABLE 5.3 – F1-scores et scores de cohérence hiérarchique obtenus en utilisant l'architecture avec les réseaux linéaires.

		F1-score			
Réseau neuronal	**Cohérence hiérarchique**	**Niveau 1**	**Niveau 2**	**Niveau 3**	**Niveau 4**
Base de référence	100,0	86,7	83,0	78,3	73,9
Linéaire					
$\alpha = 0,0$	96,0	87,8	84,1	79,1	74,0
$\alpha = 0,1$	96,1	87,6	84,2	79,1	74,2
$\alpha = 0,2$	96,2	87,7	84,0	79,2	74,2
$\alpha = 0,3$	96,3	87,8	84,3	79,3	74,4
$\alpha = 0,4$	96,5	87,8	84,1	79,2	73,9
$\alpha = 0,5$	96,6	87,8	84,2	79,3	74,2
$\alpha = 0,6$	96,8	87,9	84,5	79,7	74,5
$\alpha = 0,7$	97,0	87,9	84,3	79,8	74,7
$\alpha = 0,8$	97,3	88,3	84,9	80,1	74,7
$\alpha = 0,9$	98,6	87,8	84,2	79,4	73,9
$\alpha = 1,0$	100,0	0,0	0,0	0,0	0,0

TABLE 5.4 – F1-scores et scores de cohérence hiérarchique obtenus en utilisant l'architecture avec les réseaux basés sur des connaissances.

		F1-score			
Réseau neuronal	**Cohérence hiérarchique**	**Niveau 1**	**Niveau 2**	**Niveau 3**	**Niveau 4**
Base de référence	100,0	86,7	83,0	78,3	73,9
Basés sur des connaissances					
$\alpha = 0,0$	96,4	87,7	84,2	79,2	74,4
$\alpha = 0,1$	96,4	87,8	84,3	79,5	74,5
$\alpha = 0,2$	96,5	88,0	84,5	79,7	74,6
$\alpha = 0,3$	96,7	88,0	84,6	79,6	74,4
$\alpha = 0,4$	96,8	88,0	84,5	79,9	74,8
$\alpha = 0,5$	96,9	87,8	84,4	79,6	74,6
$\alpha = 0,6$	97,1	87,9	84,5	79,9	74,8
$\alpha = 0,7$	97,3	88,1	84,7	79,8	74,9
$\alpha = 0,8$	97,7	88,1	84,7	80,2	75,0
$\alpha = 0,9$	99,9	18,3	18,0	26,2	26,9
$\alpha = 1,0$	100,0	0,0	0,0	0,0	0,0

5.4 Conclusion du chapitre

Dans cette enrichissement cognitif de BioFlauBERT, plusieurs méthodes d'incorporation de la taxonomie CIM-10 ont été comparées sur une tâche de prédiction hiérarchique de causes primaires de décès :

i. les F1-scores n'ont pas été influencés par le type de réseau neuronal utilisé ;

ii. l'incorporation de la CIM-10 dans l'algorithme d'apprentissage uniquement a conduit à de meilleurs résultats et une meilleure stabilité ;

iii. l'incorporation dans l'algorithme d'apprentissage était optimale avec un hyperparamètre fixé à environ 0,2.

En conclusion, l'incorporation dans l'algorithme d'apprentissage uniquement est recommandé pour la prédiction hiérarchique de causes primaires de décès à condition de soigneusement ajuster l'hyperparamètre α.

Soumission dans la revue *International Journal of Medical Informatics* :

Incorporating the CIM-10 Hierarchy into the Hypothesis Set and the Learning Algorithm for Hierarchical Text Classification

Incorporating the ICD-10 Hierarchy into the Hypothesis Set and the Learning Algorithm for Hierarchical Text Classification

Corentin Blanc[a-e], Alexandre Bailly[a-e], Élie Francis[a], Fadi Jamal[f], Pascal Roy[b-e]

[a] Everteam Software, Lyon, France

[b] Université de Lyon, Lyon, France

[c] Université Lyon 1, Villeurbanne, France

[d] Service de Biostatistique-Bioinformatique, Pôle Santé Publique, Hospices Civils de Lyon, Lyon, France

[e] Équipe Biostatistique-Santé, Laboratoire de Biométrie et Biologie Évolutive, CNRS UMR 5558, Villeurbanne, France

[f] IzyCardio, Lyon, France

Corresponding author:

Corentin Blanc

Everteam Software

17 quai Joseph Gillet

F-69004, Lyon, France

c.blanc@everteam.com

Abstract

Background and purpose: Yearly produced medical documents are tremendously numerous. Their synthesis requires Natural Language Processing tools and taxonomies. Currently, a 'global approach' is the reference for classifying documents according to a taxonomy but few prior knowledge about the taxonomy is incorporated. This work investigated the performance in predicting the causes of death from death certificates of two approaches: with and without incorporation the ICD-10 hierarchy into the hypothesis set and/or the learning algorithm.

Basic procedures: BioFlauBERT, a language model pre-trained on all-purpose and medical documents, was fine-tuned to predict hierarchically the primary cause of death at the first four levels of ICD-10. Two types of NNs were used: a fully connected neural network (FCNN) to disregard the ICD-10 hierarchy and a knowledge-based neural network (KBNN) to allow for it. A loss function corrected the sum of errors that occurred during training: i) a hierarchical loss due to prediction errors; and, ii) a dependency loss due to hierarchy-linked errors. A hyperparameter adjusted the contribution of each loss to the overall error.

Main findings: With either a FCNN or a KBNN, all F1-scores increased with the increase of the α hyperparameter until reaching maximal medians at $\alpha = 0.8$. From level 1 to 4, the maximal median F1-scores were respectively 88.3, 84.9, 80.1, and 74.7 with FCNNs and 88.1, 84.7, 80.2, and 75.0 with KBNNs. With $\alpha > 0.8$, all F1-scores decreased down to 0. The observed ranges of all F1-scores were slightly wider with KBNNs vs. FCNNs. The hierarchical consistency scores increased with the decrease of α; they were always slightly higher with KBNNs vs. FCNNs.

Conclusions: Incorporating the ICD-10 hierarchy into the hypothesis set did not significantly affect the prediction performance but hampered convergence, whereas incorporating it into the learning algorithm maximized performance, providing the hyperparameter is properly adjusted.

Keywords

Hierarchical text classification; Prior knowledge incorporation; ICD-10; BioFlauBERT

1. Introduction

Over the last decade, the number of medical documents has substantially increased such that their exploitation for any specific purpose has become very time-consuming. Sometimes, shortening that time and presenting these documents in a synthetic and practical way requires the use of a taxonomy. A taxonomy (such as the Medical Subject Headings, MeSH [1] or the 10th revision of the International Classification of Diseases, ICD-10 [2]) is a tree-like hierarchy of categories organized in several levels: the highest category (the root) being broad and the lower ones (the leaves) increasingly detailed.

The automatic or computer-assisted classification of a collection of medical documents according to a given taxonomy may use Natural Language Processing (NLP) tools (artificial intelligence technologies able to process human language). Within this context, hierarchical text classification (HTC) [3] is a well-known NLP task that consists in assigning a text one or several categories from a given taxonomy. In practice, HTC is often carried out using either a 'flat' or a 'global' approach [4]. The flat approach consists in predicting a category at a given level of a taxonomy then accessing higher-level categories in that taxonomy. Doing so ensures the right hierarchical order regardless of the prediction accuracy. On the contrary, the global approach consists in training a single model to predict the categories at all targeted levels without necessarily ensuring a hierarchical consistency. The main idea behind the global approach is that predicting all targeted levels at the same time is an implicit usage of the taxonomy.

Although the global approach outperforms the flat one [5, 6], it fails to incorporate explicitly a taxonomy as a useful, if not necessary prior knowledge. In fact, prior knowledge incorporation is becoming increasingly frequent because of specific needs for taxonomic hierarchy in several domains (medicine, pharmacology, etc.). In HTC with a global approach, a taxonomy may be incorporated into either the 'hypothesis set' or the 'learning algorithm'

[7]. The hypothesis set consists in a model structure (and several hyperparameters) in which the prior knowledge is to be incorporated. Within this context, two types of multilayer NNs may be used: a knowledge-based neural network (KBNN, a neural network with known deleted connections between layers) [8] allows incorporating a taxonomy directly into the hypothesis set, whereas a fully connected neural network (FCNN, neural network where all neurons are interconnected between layers) ignores that taxonomy. An incorporation is generally very efficient; however, whereas incorporating a complex prior knowledge into the hypothesis set is not always possible, incorporating it into the learning algorithm is easier; it consists in inserting into the loss function an additional term that penalizes for hierarchical error [9].

In the specialized literature, few researchers have focused on medical taxonomy incorporation either into the hypothesis set or into the learning algorithm. Focusing only on the hypothesis set, Masera et al. [10] introduced the AWX hierarchical layer for incorporating taxonomies directly into the model structure. Upon evaluation on benchmark datasets, this process outperformed all other non-incorporation approaches tested. Almost concurrently, focusing only on the learning algorithm, Wehrmann et al. [11] and Moons et al. [12] incorporated taxonomies using an additional term to the loss functions as penalization for hierarchical error. Both latter works used an additional term that stemmed from the differences between the probabilities of belonging to two successive levels and was weighted by a hyperparameter to adjust the contribution of this additional term; their results outperformed all other non-incorporation approaches tested. Nevertheless, up to now, incorporating prior knowledge simultaneously into the hypothesis set and the learning algorithm has seldom or never been reported.

This study aims to explore, in a medical HTC task, the efficiencies of approaches that incorporate or not the ICD-10 hierarchy into the hypothesis set (through KBNNs and FCNNs,

respectively), the learning algorithm (through an adjustment of a hyperparameter), or into both (Table 1). Comparisons were made with the efficiency of a flat approach as baseline.

Table 1 – Summary of the investigations

Approach	Neural network	α hyperparameter value
Without ICD-10 taxonomy incorporation	FCNN	1
With ICD-10 taxonomy incorporation		
Into the hypothesis set	KBNN	1
Into the learning algorithm	FCNN	0 to 0.9 with 0.1 increment
Into both	KBNN	0 to 0.9 with 0.1 increment

ICD-10: The International Classification of Diseases, 10th revision - FCNN: Fully connected neural network – KBNN: knowledge-based neural network

2. Materials and methods

2.1.The data

The present study used a part of the French CepiDC database [13] limited to a corpus of 129,149 death certificates issued over period 2006-2015 (courtesy of the Institute National de la Santé et de la Recherche Médicale, INSERM). Each certificate includes usually up to four lines of free text that specify the primary cause of death as per the ICD-10 [2]. The corpus used in the present study considered only certificates with four lines and that included a fourth-level ICD-10 code previously tagged by a CepiDC expert. With this code, it was possible to access the three highest levels of the ICD-10 hierarchy; this led to 2.394 level 4 codes, 935 level 3 codes, 154 level 2 codes, and 18 level 1 codes (below denoted C_4 *to* C_1). A simple example of the ICD-10 levels is given in Figure 1.

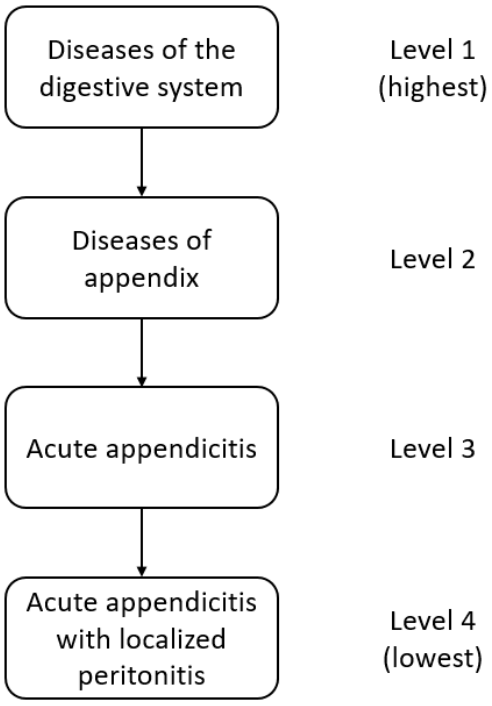


Figure 1 – Example of four highest levels in the ICD-10.

2.2. The proposed approaches

2.2.1. The baseline

The baseline for the comparisons was a flat approach that used BioFlauBERT [14]. BioFlauBERT is a medical French language model based on Transformers [15], pre-trained on a massive volume of text from all-purpose and medical documents, and able to encode a death certificate. BioFlauBERT was fine-tuned to predict the primary cause of death at the fourth level of the ICD-10; precisely, i) a prediction layer was connected; ii) BioFlauBERT and the prediction layer weights were trained end-to-end; then, iii) the primary cause of death at the highest ICD-level was accessed.

2.2.2. The architecture

The architecture was composed of two main blocks (Figure 2): BioFlauBERT encoded the death certificate into a vector $z \in \mathbb{R}^d$ of d hidden size (here, $d = 768$) and a stack of L neural networks (NNs) predicted the primary cause of death at each of the four highest levels of the ICD-10 ($L = 4$). Each NN takes, as input, BioFlauBERT's encodings of death certificates plus the output scores of the next higher level NN (the first NN that takes as input only BioFlauBERT's encoding). Here, two types of two-layer NNs were used: a FCNN that disregarded the ICD-10 hierarchy into the hypothesis set only and a KBNN that considered that hierarchy.

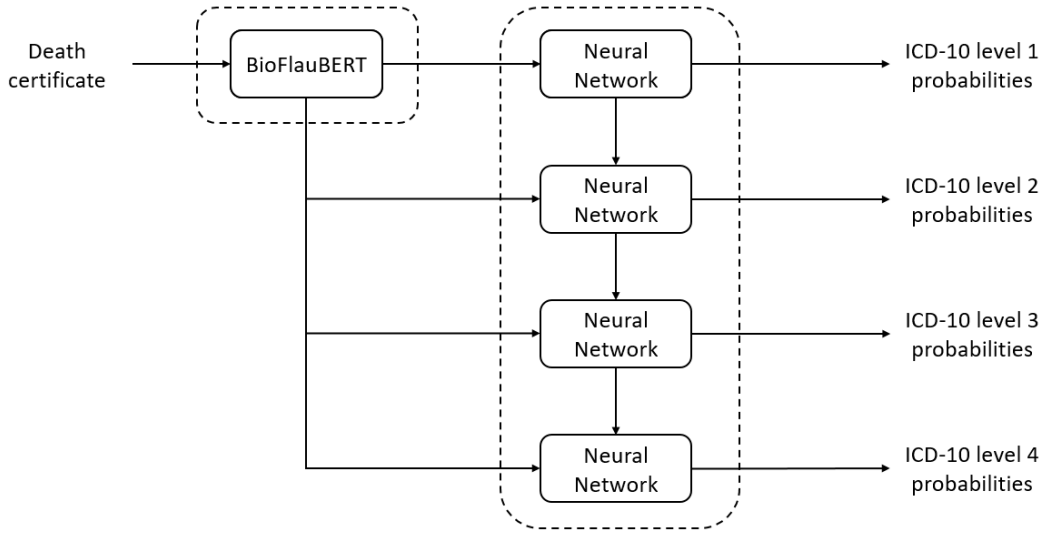


Figure 2 – Hierarchical text classification tackled by a global approach composed of BioFlauBERT and a stack of neural networks. First, BioFlauBERT encoded the death certificate, and then the neural networks predicted the primary cause of death at each of the four highest levels of the ICD-10 hierarchy.

2.2.2.1. Fully connected neural network

Let s^l be the output score computed by the l-th level NN. The score of level 1 (one) are then:

$$s^1 = (\mathrm{z} \cdot \mathrm{W}_1^1 + \mathrm{b}_1^1) \cdot \mathrm{W}_2^1 + \mathrm{b}_2^1$$

the score of the l-th level is:

$$\forall\, l \in \{2, \dots, \mathrm{L}\}, s^l = \left([\, \mathrm{z} \,||\, s^{l-1}\,] \cdot \mathrm{W}_1^l + \mathrm{b}_1^l\right) \cdot \mathrm{W}_2^l + \mathrm{b}_2^l$$

and the probability p^l for the l-th level is:

$$\forall\, l \in \{1, \dots, \mathrm{L}\}, p^l = \mathrm{softmax}(s^l)$$

In these equations, || is the concatenation operator; $\mathrm{W}_1^1 \in \mathbb{R}^{d \times d}$, $\mathrm{W}_2^1 \in \mathbb{R}^{d \times C_1}$, $\mathrm{W}_1^l \in \mathbb{R}^{(d+C_{l-1}) \times (d+C_{l-1})}$, and $\mathrm{W}_2^l \in \mathbb{R}^{(d+C_{l-1}) \times C_l}$ are the weight matrices; and $\mathrm{b}_1^1 \in \mathbb{R}^d$, $\mathrm{b}_2^1 \in \mathbb{R}^{C_1}$, $\mathrm{b}_1^l \in \mathbb{R}^{d+C_{l-1}}$, and $\mathrm{b}_2^l \in \mathbb{R}^{C_l}$ the associated biases.

2.2.2.2. Knowledge-based neural network

Contrary to a FCNN, a KBNN incorporates the ICD-10 hierarchy by mapping it onto the neuron connections. Thus, the output score computed by the level 1 (one) NN are then:

$$s^1 = (\mathrm{z} \cdot \mathrm{W}_1^1 + \mathrm{b}_1^1) \cdot \mathrm{W}_2^1 + \mathrm{b}_2^1$$

the score computed by the l-th level NN is:

$$\forall\, l \in \{2, \dots, \mathrm{L}\}, s^l = \left([\,\mathrm{z} \,||\, s^{l-1}\,] \cdot [M^l \odot \mathrm{W}_1^l] + \mathrm{b}_1^l\right) \cdot \mathrm{W}_2^l + \mathrm{b}_2^l$$

and the probability p^l for the l-th level is:

$$\forall\, l \in \{1, \dots, \mathrm{L}\}, p^l = \mathrm{softmax}(s^l)$$

In these equations, $M^l \in \mathbb{R}^{(d+C_{l-1}) \times (d+C_{l-1})}$ is the mapping matrix at level l containing 1 (one) when the connection exists in the ICD-10 (0 otherwise) and $\odot$ the element-wise product.

2.2.3. The loss function

For an end-to-end training of the whole architecture, the loss function $\mathcal{L}$ is the weighted sum of a hierarchical loss $\mathcal{L}_h$ and a dependency loss $\mathcal{L}_d$:

$$\mathcal{L} = (1 - \alpha) \cdot \mathcal{L}_h + \alpha \cdot \mathcal{L}_d$$

In this equation, $\alpha \in [0, 1]$ is a hyperparameter. The closer α is to zero, the more incorporated is the ICD-10 hierarchy into the learning algorithm. The ICD-10 hierarchy is ignored when α is close to one.

2.2.3.1. Hierarchical loss

The hierarchical loss evaluates the degree of discordance between the probability distributions as predicted by the architecture and the probability distributions as observed in the CepiDC corpus at all four first levels of the ICD-10. For any given death certificate, the hierarchical loss is:

$$\mathcal{L}_h = -\sum_{l=1}^{L}\sum_{i=1}^{C_l} \mathbb{1}_i^l \cdot \log(p_i^l)$$

In this equation, $\mathbb{1}_i^l$ is a binary indicator (= 1 when the death certificate belongs to the i-th class at level l, 0 otherwise), and p_i^l the i-th probability at level l.

2.2.3.2. Dependency loss

The dependency loss evaluates the hierarchical error using the dependencies between all four successive levels in the ICD-10. For any given death certificate, the dependency loss is:

$$\mathcal{L}_d = \sum_{l=1}^{L-1}\sum_{i=1}^{C_l}\sum_{j=1}^{C_{l+1}} \mathbb{1}_{i,j} \cdot p_j^{l+1} \cdot (1 - p_i^l)$$

In this equation, $\mathbb{1}_{i,j}$ is a binary indicator (= 1 when the i-th class at level l and the j-th class at level $l+1$ are connected in the ICD-10, 0 otherwise) (For more details, see the Appendix).

2.3.The hyperparameters

Most hyperparameters were selected a priori among the usual ones seen in the literature: i) the number of epochs (i.e., the number of times the corpus was explored) was set to 10; ii) the batch size (i.e., number of sentences propagated at the same time through the model) was set to 16; iii) AdamW [16] learning algorithm was used for backpropagation; iv) the learning rate (i.e., weight increment) was set to 2e-5 at the beginning of the first epoch but left to decrease linearly down to 0 at the end of the last epoch to force the convergence of the loss function and avoid the phenomenon of 'catastrophic forgetting' [17] (i.e., the disruption or erasure of BioFlauBERT's knowledge during fine-tuning). The hyperparameter α of the loss function was taken in turn from the following set of possible values {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}.

2.4.The evaluation method and criteria

The evaluation used a bootstrap method [18] that consisted in 129,149 random drawings, with replacement, of one death certificate from the study corpus; the drawn certificates constituted the training set. Undrawn death certificates formed a test set. Theoretically, about 37% of the corpus remained in the test set (47,785 death certificates). Ten bootstrap iterations were then run. The medians, minima, and maxima of the weighted F1-scores were calculated for each level and for the hierarchical consistency score (HCS) (i.e., percentage of right dependencies).

2.5.The implementation details

This work used Python 3.9.6 as programming language and the following packages for specific tasks: Torchtext 0.11.0 to load and tokenize the CepiDC corpus, Transformers 4.18.0 from Hugging Face to apply BioFlauBERT, and PyTorch 1.12.0 to deal with the FCNNs, the KBNNs, and architecture training.

3. Results

With either a FCNN or a KBNN, all F1-scores increased progressively with the decrease of the hyperparameter until reaching maximal medians with $\alpha = 0.8$. At the four successive levels (first to fourth), the maximal medians were respectively 88.3, 84.9, 80.1, and 74.7 with FCNNs (Table 2) and 88.1, 84.7, 80.2, and 75.0 using KBNNs (Table 3). However, with $\alpha > 0.8$ all F1-scores decreased down to zero (reached with $\alpha = 1.0$). In addition, the observed ranges (i.e., differences between the maximum and minimum) of all F1-scores were slightly wider with KBNNs than with FCNNs.

The HCSs increased progressively with the decrease of the hyperparameter until reaching a perfect median (i.e., 100.0) with $\alpha = 1.0$. The HCSs were always slightly higher with KBNNs than with FCNNs; e.g., 96.4 vs. 96.0 with $\alpha = 0.0$, 97.7 vs. and 97.3 with $\alpha = 0.8$, and 96.9 vs. and 96.6 with $\alpha = 0.5$. Generally, the baseline was overperformed when the hyperparameter α was properly adjusted.

Table 2 – Median [Minimum, Maximum] of HCSs and the F1-scores with the ICD-10 hierarchy incorporation into the learning algorithm only.

	HCS	F1-score			
Approach		Level 1*	Level 2	Level 3	Level 4
Baseline	100.0 [100.0, 100.0]	86.7 [86.3, 86.7]	83.0 [82.8, 83.2]	78.3 [78.2, 78.5]	73.9 [73.7, 74.0]
FCNN					
$\alpha = 0.0$	96.0 [96.0, 96.1]	87.8 [87.5, 87.8]	84.1 [84.0, 84.2]	79.1 [78.9, 79.1]	74.0 [73.9, 74.1]
$\alpha = 0.1$	96.1 [96.1, 96.3]	87.6 [87.6, 88.0]	84.2 [84.1, 84.4]	79.1 [79.0, 79.5]	74.2 [73.9, 74.3]
$\alpha = 0.2$	96.2 [96.0, 96.3]	87.7 [87.6, 87.7]	84.0 [84.0, 84.2]	79.2 [79.0, 79.3]	74.2 [74.0, 74.3]
$\alpha = 0.3$	96.3 [96.2, 96.4]	87.8 [87.6, 87.9]	84.3 [84.0, 84.3]	79.3 [79.1, 79.5]	74.4 [73.9, 74.5]
$\alpha = 0.4$	96.5 [96.4, 96.5]	87.8 [87.5, 88.1]	84.1 [84.0, 84.3]	79.2 [79.1, 79.3]	73.9 [73.9, 74.3]
$\alpha = 0.5$	96.6 [96.5, 96.7]	87.8 [87.7, 88.0]	84.2 [84.2, 84.4]	79.3 [79.2, 79.7]	74.2 [73.9, 74.6]
$\alpha = 0.6$	96.8 [96.7, 96.9]	87.9 [87.8, 88.1]	84.5 [84.3, 84.7]	79.7 [79.5, 79.8]	74.5 [74.2, 74.7]
$\alpha = 0.7$	97.0 [96.9, 97.1]	87.9 [87.8, 88.2]	84.3 [84.3, 84.7]	79.8 [79.4, 80.1]	74.7 [74.3, 74.9]
$\alpha = 0.8$	97.3 [97.2, 97.4]	88.3 [88.1, 88.4]	84.9 [84.6, 85.0]	80.1 [79.8, 80.4]	74.7 [74.4, 75.0]
$\alpha = 0.9$	98.6 [97.9, 99.8]	87.8 [18.5, 88.3]	84.2 [18.4, 84.8]	79.4 [26.6, 80.2]	73.9 [27.3, 74.9]
$\alpha = 1.0$	100.0 [100.0, 100.0]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]

*Highest level in the ICD-10 - FCNN: Fully-connected neural network - HCS: Hierarchical consistency score.

Table 3 – Median [Minimum, Maximum] of the HCSs and the F1-scores with the ICD-10 hierarchy incorporation into both the hypothesis set and the learning algorithm

	HCS	F1-score			
Approach		Level 1*	Level 2	Level 3	Level 4
Baseline	100.0 [100.0, 100.0]	86.7 [86.3, 86.7]	83.0 [82.8, 83.2]	78.3 [78.2, 78.5]	73.9 [73.7, 74.0]
KBNN					
$\alpha = 0.0$	96.4 [96.3, 96.5]	87.7 [87.6, 87.8]	84.2 [84.0, 84.3]	79.2 [79.4, 79.6]	74.4 [74.2, 74.6]
$\alpha = 0.1$	96.4 [96.3, 96.5]	87.8 [87.7, 87.9]	84.3 [84.2, 84.6]	79.5 [79.2, 79.9]	74.5 [74.3, 74.7]
$\alpha = 0.2$	96.5 [96.5, 96.6]	88.0 [87.7, 88.0]	84.5 [84.2, 84.6]	79.7 [79.4, 79.9]	74.6 [74.2, 75.0]
$\alpha = 0.3$	96.7 [96.7, 96.8]	88.0 [87.8, 88.2]	84.6 [84.3, 84.8]	79.6 [79.3, 80.1]	74.4 [74.2, 75.3]
$\alpha = 0.4$	96.8 [96.8, 96.9]	88.0 [87.7, 88.3]	84.5 [84.2, 84.9]	79.9 [79.4, 79.9]	74.8 [74.3, 74.9]
$\alpha = 0.5$	96.9 [96.8, 97.0]	87.8 [87.7, 88.2]	84.4 [84.1, 84.7]	79.6 [79.3, 79.9]	74.6 [74.3, 75.1]
$\alpha = 0.6$	97.1 [97.0, 97.2]	87.9 [87.8, 88.1]	84.5 [84.5, 84.7]	79.9 [79.8, 80.2]	74.8 [74.5, 75.1]
$\alpha = 0.7$	97.3 [97.3, 97.4]	88.1 [87.9, 88.2]	84.7 [84.5, 84.8]	79.8 [79.5, 80.2]	74.9 [74.4, 75.2]
$\alpha = 0.8$	97.7 [97.5, 97.7]	88.1 [87.7, 88.4]	84.7 [84.6, 84.9]	80.2 [79.6, 80.2]	75.0 [74.3, 75.2]
$\alpha = 0.9$	99.9 [99.8, 99.9]	18.3 [18.2, 20.1]	18.0 [16.9, 19.7]	26.2 [25.4, 27.6]	26.9 [26.0, 28.2]
$\alpha = 1.0$	100.0 [100.0, 100.0]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]

* Highest level in the ICD-10 - KBNN: Knowledge-based neural network - HCS: Hierarchical consistency score.

4. Discussion

In a hierarchical cause-of-death classification task, ignoring the ICD-10 hierarchy in the hypothesis set (through a FCNN) or taking it into account (through a KBNN) led to nearly the same results as per the F1-scores, but KBNNs performed slightly better than FCNNs as per the HCS. Further, incorporating the ICD-10 taxonomy into the learning algorithm seemed to be optimal around $\alpha = 0.8$ whatever the NN used; with $\alpha > 0.8$, all F1-scores headed to zero. Undeniably, incorporation into the learning algorithm was more effective than into the hypothesis set.

Regarding the incorporation into the hypothesis set, given the observed ranges (score maxima minus minima), the KBNNs were more unstable than the FCNNs (i.e., showed difficulty to converge to the minimum of the loss function) whatever the value of the incorporation hyperpameter α. This may be explained by the presence of known deleted connections in the KBNNs, which hinders convergence during backpropagation. Although performance was about the same, FCNNs were then more advantageous than KBNNs for ensuring convergence.

Regarding the incorporation into the learning algorithm, the hyperparameter α had an essential role whatever the NN used. Indeed, without ICD-10 incorporation (α approximately equal to 0.0), the backpropagation was ensured only by the hierarchical loss and the global approach overperformed the flat approach (baseline), as expected from the literature [5,6]. The more α increased from 0.0 to 0.8, the more the F1-scores and the HCSs increased. When $\alpha = 0.8$, the F1-scores reached their maximum. Finally, with a hyperparameter $\alpha > 0.8$, the F1-scores dropped to about zero because the loss function ignored totally the predictions and focused only on the ICD-10 hierarchy; this led to the same prediction and a perfect HCS. Thus, a carefully adjusted hyperparameter α maximized the F1-scores and made incorporating

the taxonomy into the learning algorithm was more advantageous than incorporating it into the hypothesis set.

According to the present and to previously published results, the interest of incorporating a taxonomy in a HTC task does not need to be proven. However, although a taxonomy can be seen as a set of logical rules, it can also be seen as a 'graph'; i.e., a set of vertices and a set of edges that connect some vertices together. Within this context, Zhou et al. [19] and Deng et al. [20] have recently incorporated taxonomies of news topics directly as features of a model to classify news documents. Both approaches encoded the taxonomy using Graph Neural Network (GNN) [21] and Graph Convolutional Network (GCN) [22] and then combined the resulting encodings with text encodings to achieve text classification. Thus, in a future work, it will be interesting to incorporate the ICD-10 hierarchy in a graph-theory approach and compare the results with the present ones.

One merit of the present work is enriching the current medical HTC literature with new results stemming from the use and performance comparison between ignoring and taking into account a taxonomy into the hypothesis set and the learning algorithm. To the best of our knowledge, few studies have focused on these two incorporations in medical HTC and performance comparisons have never been carried out. Another merit is investigating the importance of the hyperparameter. Although most values of this hyperparameter led to good predictions, a careful selection seems essential to maximize performance.

5. Conclusion

In this comparison between ignoring and taking into account the ICD-10 hierarchy into the hypothesis set and the learning algorithm, i) F1-scores were not influenced by the type of the neural network used; ii) the performance without incorporation into the hypothesis set led to a better convergence of the loss function; and, iii) the incorporation into the learning algorithm

was optimal with a hyperparameter set to about 0.8. This demonstrated that opting for an incorporation into the learning algorithm (through an additional term to the loss function) and adjusting the hyperparameter are optimal to maximize the prediction performance.

Acknowledgment

The authors are grateful to Jean Iwaz (Hospices Civils de Lyon) for his thoughtful comments and for editing the latest versions of the manuscript.

Funding

This work was supported by Association Nationale de la Recherche et de la Technologie (ANRT) [grant number 2019/1374]. The sponsor had no role in the study design; the collection, analysis, and interpretation of the data; the writing of the report; and the decision to submit the article for publication.

Ethical approval statement

No ethical issues were raised by this study.

Conflict of interest statement

None to declare.

Author statement

Conceptualization: CB, AB, EF, FJ, and PR.

Data curation: CB.

Formal analysis: CB.

Funding acquisition: EF, FJ, and PR.

Investigation: CB.

Methodology: CB, AB, EF, and PR.

Software: CB.

Supervision: EF, FJ, and PR.

Validation: CB.

Visualization: CB.

Writing - original draft: CB.

Writing - review & editing: CB, AB, EF, FJ, and PR.

Summary table

What Was Already Known On The Topic:

- Two strategies exist for incorporating a taxonomy as a prior knowledge: into the hypothesis set or the learning algorithm.
- The ICD-9 and 10 have been already successfully incorporated into the learning algorithm only.

What This Study Added To Our Knowledge:

- Incorporations the ICD-10 hierarchy into the hypothesis set, the learning algorithm, or both were tested.
- An incorporation into the learning algorithm only is worthwhile to maximize performance providing a weight adjustment of an additional term to the loss function.

References

[1] C. Lipscomb, Medical Subject Headings (MeSH), Bull Med Libr Assoc, (2000), pp 265. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC35238/.

[2] W. H. Organization, et al., ICD-10. International Statistical Classification of Diseases and Related Health Problems: 10th revision, Classification statistique internationale des maladies et des problèmes de santé connexes: Dixième révision, (1992).

[3] C. Silla, A. Freitas, A survey of hierarchical classification across different application domains, Data Min. Knowl. Discovery, (2011), pp. 31–72. https://link.springer.com/article/10.1007/s10618-010-0175-9.

[4] X. Qiu, X.-J. Huang, Z. Liu, J. Zhou, Hierarchical Text Classification with Latent Concepts, Proceedings of the 49th Annual Meeting of the Association for Computational, (2011), pp. 598-602. https://aclanthology.org/P11-2105/.

[5] D. Ghazi, D. Inkpen, S. Szpakowicz, Hierarchical Versus Flat Classification of Emotions in Text, Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, (2010), pp. 140–146. https://aclanthology.org/W10-0217.pdf.

[6] R. Babbar, I. Partalas, E. Gaussier, M. Amini, On Flat versus Hierarchical Classification in Large-Scale Taxonomies, Advances in Neural Information Processing Systems, (2013). https://dl.acm.org/doi/10.5555/2999792.2999816.

[7] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, J. Schuecker, Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems, IEEE Trans. Knowl. Data Eng., (2021). https://doi.org/10.1109%2Ftkde.2021.3079836.

[8] G. Towell, J. Shavlik, Knowledge-Based Artificial Neural Networks, Artif. Intell., (1994), pp. 119–165. https://www.sciencedirect.com/science/article/abs/pii/0004370294901058.

[9] M. Diligenti, S. Roychowdhury, M. Gori, Integrating Prior Knowledge into Deep Learning, 16th IEEE International Conference on Machine Learning and Applications, (2017), pp. 920–923. https://ieeexplore.ieee.org/document/8260755.

[10] L. Masera, E. Blanzieri, AWX: An Integrated Approach to Hierarchical-Multilabel Classification, Machine Learning and Knowledge Discovery in Databases, (2019), pp. 322–336. https://link.springer.com/chapter/10.1007/978-3-030-10925-7_20.

[11] J. Wehrmann, R. Cerri, R. Barros, Hierarchical Multi-Label Classification Networks, Proceedings of the 35th International Conference on Machine Learning, (2018), pp. 5075–5084. https://proceedings.mlr.press/v80/wehrmann18a.html.

[12] E. Moons, A. Khanna, A. Akkasi, M.-F. Moens, A Comparison of Deep Learning Methods for ICD Coding of Clinical Records, Applied Sciences, (2020). https://www.mdpi.com/2076-3417/10/15/5262.

[13] A. Névéol, A. Robert, F. Grippo, C. Morgand, C. Orsi, L. Pelikan, L. Ramadier, G. Rey, P. Zweigenbaum. CLEF eHealth 2018 Multilingual Information Extraction Task Overview: ICD10 Coding of Death Certificates in French, Hungarian and Italian. Conference and Labs of the Evaluation Forum, eHealth, CEUR, (2018), pp. 18. https://hal.archives-ouvertes.fr/hal-02276492.

[14] C. Blanc, A. Bailly, E. Francis, T. Guillotin, F. Jamal, P. Roy. Corpus size considerations in continual pre-training of BioFlauBERT and BioCamemBERT. 2023 (submitted on March 17, 2023 and under review).

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin. Attention Is All You Need. The 31st Conference on Neural Information Processing Systems, (2017), pp. 1–11. http://arxiv.org/abs/1706.03762.

[16] I. Loshchilov, F. Hutter. Decoupled Weight Decay Regularization. 7th International Conference on Learning Representations. (2019), https://openreview.net/forum?id=Bkg6RiCqY7.

[17] R. French. Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences, (1999), pp. 128–135.

[18] B. Efron. Bootstrap Methods: Another Look at the Jackknife. Ann Stat, 7 (1979), pp. 1–26. https://doi.org/10.1214/aos/1176344552.

[19] J. Zhou, C. Ma, D. Long, G. Xu, N. Ding, H. Zhang, P. Xie, G. Liu, Hierarchy-Aware Global Model for Hierarchical Text Classification, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, (2020), pp. 1106–1117. https://aclanthology.org/2020.acl-main.104.

[20] Z. Deng, H. Peng, D. He, J. Li, P. Yu, HTCInfoMax: A Global Model for Hierarchical Text Classification via Information Maximization, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (2021), pp. 3259–3265. https://arxiv.org/abs/2104.05220.

[21] F. Scarselli, M. Gori, A. Tsoi, M. Hagenbuchner, G. Monfardini, The Graph Neural Network Model, IEEE Trans. Neural Networks, (2009), pp. 61–80. https://ieeexplore.ieee.org/document/4700287.

[22] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, Proceedings of the 30th International

Conference on Neural Information Processing Systems, (2016), pp. 3844–3852. http://arxiv.org/abs/1606.09375.

[23] F. Bobillo, U. Straccia, A Fuzzy Description Logic with Product T-Norm, IEEE International Fuzzy Systems Conference, (2007), pp. 1–6. https://ieeexplore.ieee.org/abstract/document/4295443.

Appendix

Let $G = (V, E)$ be a taxonomy (e.g., the ICD-10) with L levels and a set of vertices V:

$$V = \bigcup_{l=1}^{L} V^l \ where\ V^l = \{ v_i^l \mid 1 \leq i \leq C_l \}$$

and a set of edges E:

$$E = \{ (v_j^{l+1}, v_i^l) \in V^{l+1} \times V^l, v_j^{l+1} \Rightarrow v_i^l \mid 1 \leq l \leq L \}$$

E corresponds to the prior knowledge to be incorporated into the learning algorithm. Each edge within the taxonomy can be rewritten

$$v_j^{l+1} \Rightarrow v_i^l = \neg(v_j^{l+1} \wedge \neg v_i^l)$$

and then transformed into an optimization constraint for the learning algorithm using a triangular norm (T-norm). A T-norm is a binary function T: $[0,1] \times [0.1] \rightarrow [0.1]$ bridging the gap between Boolean and fuzzy logic and satisfying the properties of commutativity, monotonicity, associativity, and with 1 acting as identity element. A popular T-norm is the product T-norm [23] defined by:

Product T-norm	$x \wedge y$	$x \vee y$	$\neg x$
$T(x, y)$	xy	$x + y - xy$	$1 - x$

Thus, using the product T-norm, each edge becomes an optimization constraint in the form of:

$$\Phi_{i,j}^l = T(p_j^{l+1}, p_i^l) = 1 - p_j^{l+1} \cdot (1 - p_i^l)$$

where $p_j^{l+1} = \mathbb{P}(v_j^{l+1})$ and $p_i^l = \mathbb{P}(v_i^l)$.

Finally, the dependency loss is:

$$\mathcal{L}_d = \sum_{l=1}^{L-1} \sum_{i=1}^{C_l} \sum_{j=1}^{C_{l+1}} \mathbb{1}_{i,j} \cdot (1 - \Phi_{i,j}^l) = \sum_{l=1}^{L-1} \sum_{i=1}^{C_l} \sum_{j=1}^{C_{l+1}} \mathbb{1}_{i,j} \cdot p_j^{l+1} \cdot (1 - p_i^l)$$

where $\mathbb{1}_{i,j}$ is a binary indicator (1 when the i-th class at level l and the j-th class at level $l + 1$ are connected in the taxonomy, 0 otherwise).

Chapitre 6

Prédiction de motifs de consultation

Ce dernier chapitre est la concrétisation des travaux menés ces trois dernières années. À travers une application effectuée pour l'acteur en cardiologie izyCardio, les modèles de langue précédemment étudiés pour le domaine médical ont été réglés finement afin de prédire efficacement les motifs de consultation de patients en se basant sur des champs textuels remplis en amont de la consultation. Ce dernier chapitre mettra un point d'honneur à expliciter la méthodologie derrière cette application. À l'heure où ces lignes sont écrites, un modèle est testé dans divers centres de cardiologie afin d'en éprouver la robustesse et de nouvelles versions sont à l'étude !

Sommaire

6.1 Contexte

Ces dernières années, les besoins de soins sont en constante augmentation mais les délais d'attente s'allongent notamment auprès de médecins spécialistes comme les cardiologues. Selon l'enquête réalisée en 2018 par la DREES (Direction de la Recherche des Études, de l'Évaluation et des Statistiques), il s'écoule en moyenne 50 jours entre la prise de contact et le rendez-vous auprès d'un cardiologue [45]. Délais pouvant excéder les trois mois dans les déserts médicaux. Malheureusement, ce phénomène s'amplifie jour après jour en raison du vieillissement de la population mettant sous tension les structures cardiologiques et allongeant dangereusement les délais de consultation.

Afin de proposer une solution à cette problématique, la société izyCardio a développé un outil en ligne intitulé « izyCardio Connect » permettant de faire l'interface entre le patient et les professionnels de santé en cardiologie. Avant chaque consultation, tous les patients remplissent un questionnaire médical à choix multiples afin que le cardiologue puisse préparer et personnaliser en amont la consultation. Encore mieux, ce questionnaire permettra également le calcul d'un « izyScore » reflétant le degré d'urgence de la consultation et déterminant la nécessité d'examens complémentaires et permettant une meilleure gestion des délais de consultation dans le contexte actuel de forte demande.

Pour aller encore plus loin, un champ textuel libre est mis à disposition du patient en complément des questions à choix multiples. L'intérêt de ce champ textuel est de laisser les patients s'exprimer avec leurs propres mots sur les raisons pour lesquelles ils souhaitent consulter un cardiologue. Bien que très informatif sur les motifs de consultation des patients, il n'est pour le moment pas pris en compte dans le calcul de l'izyScore. Par conséquent, la prédiction de motifs de consultation à partir de champs textuels pré-remplis par le patient est un cas d'usage idéal pour appliquer les différentes méthodes de TALN décrites dans les chapitres précédents et un premier pas vers sa prise en compte dans une possible nouvelle version de l'izyScore.

6.2 Méthodologie

6.2.1 Corpus

Afin de prédire les motifs de consultation, deux corpus composés de 4 213 et 1 054 champs textuels anonymisés et pré-remplis par des patients ont été émis par izyCardio. Le premier a servi à déterminer la structure du modèle et le choix des hyperparamètres grâce à un bootstrap répété 50 fois (phase de validation). Tandis que le second a servi à tester la généralisation sur de nouveaux champs textuels du modèle final entraîné sur la totalité du corpus de validation (phrase de test). Enfin, un expert d'izyCardio, avec l'assistance et la validation d'un cardiologue, a préalablement étiqueté chaque champ textuel avec zéro, un ou plusieurs motifs parmi $C = 17$ possibilités en se basant sur la terminologie décrite dans la Table 6.1.

Table 6.1 – Terminologie des motifs de consultation dans les corpus fournis par izyCardio. Pour chaque motif, la distribution est donnée dans les corpus de validation et de test.

	Corpus		
Motif de consultation	**Validation**	**Test**	**Total**
Douleurs thoraciques	827	199	1 026
Dépistage cardio-vasculaire	686	170	856
Hypertension artérielle	549	142	691
Dyspnée	533	118	651
Palpitations	499	129	628
Autre motif	446	117	563
Préopératoire	436	114	550
Cardiopathie rythmique	411	95	506
Malaise	338	95	433
Souffle	336	88	424
Coronaropathie	300	75	375
Cardio-oncologie	215	59	274
AVC/AIT	181	39	220
Insuffisance cardiaque	105	21	126
Assurance	94	23	117
Valvulopathie	90	15	105
Pré-thérapeutique	70	27	97

6.2.2 Architecture proposée

Au vu des conclusions présentées tout au long de ce manuscrit, BioCamemBERT a été privilégié en réglage fin avec une couche linéaire composée de C neurones pour prédire le motif de consultation. À la différence des chapitres précédents, la classification étant multi-étiquette (i.e., plusieurs motifs peuvent être attribués à un même champ textuel), une fonction d'activation sigmoïde a été utilisé en bout de chaîne pour obtenir les probabilités $p = (p_c) \in [0, 1]^C$ par motif :

$$p \;=\; \sigma(z_1^L \cdot W_y + b_y) \tag{6.1}$$

où $W_y \in \mathbb{R}^{D \times C}$ sont les paramètres de la couche linéaire et $b_y \in \mathbb{R}^C$ le biais associé. De ce fait, un seuil de décision $\theta \in [0, 1]$ a été ajouté à l'hyperparamétrage du modèle pour attribuer ou non les motifs en fonction de leurs probabilités. Ainsi, la prédiction $\hat{y_c}$ pour le c-ème motif est donnée par :

$$\hat{y_c} \;=\; \mathbb{1}_{p_c > \theta} \tag{6.2}$$

où $\mathbb{1}_{p_c > \theta}$ est une indicatrice (1 si la probabilité p_c est plus grande que le seuil θ, 0 autrement). À noter que CamemBERT, FlauBERT et BioFlauBERT ont également été testés en plus de BioCamemBERT à titre de comparaison.

6.2.3 Fonction de perte

Afin de prendre en compte le cas multi-étiquette, l'entropie croisée binaire a été utilisée comme fonction de perte. Ici, elle évalue pour chaque motif la discordance entre les probabilités prédites par le modèle et celles observées dans le corpus. Pour un champ textuel, cette perte est donnée par :

$$L \;=\; -\sum_{c=1}^{C} \; (\mathbb{1}_c \cdot \log(p_c) \;+\; (1 - \mathbb{1}_c) \cdot \log(1 - p_c)) \tag{6.3}$$

où $\mathbb{1}_c$ est une indicatrice (1 si le c-ème motif de consultation correspond au champ textuel, 0 autrement).

6.2.4 Hyperparamétrage

Pour entraîner conjointement le modèle de langue et la couche linéaire, plusieurs hyperparamètres ont été sélectionnés a priori grâce à une grille de recherche. En plus des hyperparamètres usuels, le seuil global de décision noté θ a été ajouté. Un récapitulatif complet de l'hyperparamétrage utilisé pour les quatre modèles de langue est donné dans la Table 6.2.

TABLE 6.2 – Récapitulatifs de l'hyperparamétrage utilisé pour la prédiction de motifs de consultation avec les différents modèles de langue.

(a) Seuil de décision

Modèle	**Seuil θ**
FlauBERT	0,7
CamemBERT	0,6
BioFlauBERT	0,7
BioCamemBERT	0,5

(b) Hyperparamètres usuels

Hyperparamètre	**Valeur**
Nombre d'epochs	15
Taille des lots	1
Écrêtage du gradient	Aucun
Algorithme d'apprentissage	AdamW
Taux d'apprentissage initial	2e-5
Variation du taux d'apprentissage	Linéaire
Taux d'apprentissage final	0,0
Stabilité numérique	1e-8
Décroissance du premier moment	0,9
Décroissance du second moment	0,999
Décroissance des paramètres	0,01

6.3 Résultats

La Table 6.3 présente respectivement les F1-scores obtenus par FlauBERT et CamemBERT pour la prédiction de motifs de consultation sur les phases de validation et de test. Pour les deux phases, FlauBERT et CamemBERT ont obtenu d'excellents résultats avec la quasi-totalité des F1-scores supérieurs à 90,00 (excepté pour le motif Pré-thérapeutique avec FlauBERT) et parfois frôlant la perfection (>99,0). Comme laissait entendre les conclusions des chapitres précédents, étant sur une tâche de classification au niveau de la phrase, CamemBERT était meilleur pour la grande majorité des motifs et donc globalement avec des F1-scores globaux respectivement de 96,8 et 96,5 pour les phases de validation et de test ; contre 95,6 et 95,7 pour FlauBERT.

Table 6.3 – F1-scores obtenus localement et globalement pour la prédiction de motifs de consultation avec les modèles FlauBERT et CamemBERT.

	FlauBERT		**CamemBERT**	
Motif de consultation	**Validation**	**Test**	**Validation**	**Test**
Douleurs thoraciques	95,4	97,1	96,8	98,0
Dépistage cardio-vasculaire	94,1	93,9	95,2	95,3
Hypertension artérielle	96,2	99,0	97,0	98,2
Dyspnée	97,1	97,2	97,6	98,8
Palpitations	97,3	95,9	98,1	96,5
Autre motif	94,3	94,2	95,1	96,6
Préopératoire	95,4	95,8	96,0	96,3
Cardiopathie rythmique	95,8	97,2	96,8	97,3
Malaise	98,1	97,2	98,4	96,8
Souffle	98,5	98,8	98,9	98,8
Coronaropathie	93,6	95,3	95,5	96,0
Cardio-oncologie	95,2	95,8	95,6	95,3
AVC/AIT	97,1	97,2	98,3	98,7
Insuffisance cardiaque	90,0	93,1	95,4	96,3
Assurance	99,2	95,3	99,2	97,8
Valvulopathie	99,3	96,8	99,1	94,0
Pré-thérapeutique	89,2	86,9	91,5	90,2
Global	95,6	95,7	96,8	96,5

La Table 6.4 présente respectivement les F1-scores obtenus par BioFlauBERT et BioCamemBERT pour la prédiction de motifs de consultation sur les phases de validation et de test. Sans surprise, les modèles enrichis ont mieux performé que leurs homologues non-médicaux pour la quasi-totalité des motifs avec encore une fois des F1-scores très élevés ; montrant à travers un nouvel exemple l'efficacité et la nécessité des enrichissements linguistiques des modèles de langue dans le domaine médical. Le meilleur de tous les modèles était BioCamemBERT et était le seul à dépasser la barre des 97,0 de F1-scores globaux (respectivement 97,0 et 97,3 pour les phases de validation et de test) ; confirmant une nouvelle fois les conclusions des chapitres précédents.

TABLE 6.4 – F1-scores obtenus localement et globalement pour la prédiction de motifs de consultation avec les modèles BioFlauBERT et BioCamemBERT.

	BioFlauBERT		**BioCamemBERT**	
Motif de consultation	**Validation**	**Test**	**Validation**	**Test**
Douleurs thoraciques	96,9	98,5	97,1	98,2
Dépistage cardio-vasculaire	95,0	94,1	95,6	95,4
Hypertension artérielle	97,2	98,0	96,9	97,5
Dyspnée	97,8	98,4	97,8	98,2
Palpitations	97,9	96,3	98,1	97,3
Autre motif	95,2	96,0	95,3	96,6
Préopératoire	95,8	95,4	96,0	96,6
Cardiopathie rythmique	96,8	97,4	97,0	97,6
Malaise	98,7	97,1	98,7	97,7
Souffle	98,7	98,8	99,1	98,8
Coronaropathie	95,5	96,0	96,5	96,4
Cardio-oncologie	96,2	96,3	95,5	98,2
AVC/AIT	97,8	99,3	98,5	98,6
Insuffisance cardiaque	95,2	95,0	96,2	97,6
Assurance	99,1	97,8	99,2	97,8
Valvulopathie	99,3	96,8	99,3	96,8
Pré-thérapeutique	90,4	91,8	91,8	94,0
Global	96,7	96,6	97,0	97,3

6.4 Conclusion du chapitre

Tous les travaux menés au cours de ces trois dernières années ont permis la conception d'un modèle capable de prédire efficacement les motifs de consultation d'un patient en se basant sur un champ textuel pré-rempli. À travers cette application, il est possible de retrouver les conclusions des chapitres précédents :

1. La supériorité des modèles enrichis (BioFlauBERT et BioCamemBERT) sur leurs homologues non-médicaux (FlauBERT et CamemBERT) ;
2. La supériorité de BioCamemBERT (resp. CamemBERT) sur BioFlauBERT (resp. FlauBERT) sur une tâche de classification au niveau de la phrase. Cette application permet encore une fois de confirmer l'hypothèse émise précédemment quant au choix du modèle de langue en fonction du niveau de la tâche.

Enfin, toutes les vérifications préliminaires ayant été effectuées avec succès, le modèle résultant a désormais été intégré en phase de production afin de le tester en condition réelle et avoir le retour de plusieurs cardiologues.

Chapitre 7

Conclusion

Face à la prolifération des données textuelles non structurées, il devient impératif d'exploiter ces données de manière efficace. Cette nécessité est d'autant plus pressante dans le domaine médical qui produit une quantité importante de ces données et où les enjeux sont immenses. En conséquence, la recherche en TALN a connu une croissance sans précédent ces dernières années surtout depuis l'avènement du modèle de langue contextualisé BERT. Grâce à ses mécanismes d'auto-attention multi-têtes et à son pré-entraînement sur de vastes quantités de données textuelles, BERT est capable de générer des représentations numériques de mots ou phrases. Après avoir repoussé les limites de l'état de l'art pour la quasi-totalité des tâches de TALN, BERT a été décliné à de nombreuses reprises et notamment en langue française avec la création de FlauBERT et CamemBERT. Toutefois, les comportements de tels modèles dans le domaine médical restent obscurs, et en particulier en langue française qui est très peu étudiée dans la littérature.

Sommaire

7.1 Conclusion générale

Tout au long de cette thèse, une attention particulière a été accordée à l'étude de l'efficacité des modèles de langue contextuels pour la compréhension du langage naturel en médecine et en langue française. Tout d'abord, FlauBERT et CamemBERT ont été mis à l'épreuve sur une tâche complexe : la prédiction d'intentions et d'entités médicales dans le cadre d'un chatbot servant d'interface entre les patients et les professionnels de santé. Les résultats de cette première étude ont confirmé l'intérêt d'utiliser ces modèles en médecine, en particulier FlauBERT qui a montré des performances supérieures à celles de CamemBERT. Néanmoins, ces deux modèles présentaient certaines lacunes lorsqu'ils étaient confrontés à des phrases complexes ou à un vocabulaire trop spécifique ; probablement lié à un pré-entraînement avec trop peu de données textuelles médicales.

Pour corriger en partie ces lacunes, FlauBERT et CamemBERT ont été enrichis linguistiquement grâce à un second pré-entraînement sur des données textuelles médicales. Ce processus a abouti à la création de deux nouveaux modèles de langue contextuels, BioFlauBERT et BioCamemBERT, qui ont été évalués sur une série de quatre tâches de TALN spécifiques au domaine médical. Les résultats obtenus lors de ces évaluations ont démontré l'efficacité de l'enrichissement linguistique puisque BioFlauBERT et BioCamemBERT ont surpassé leurs homologues non-médicaux avec des performances allant grandissant avec la taille du corpus de pré-entraînement. De plus, une règle de décision a été établie pour aider à choisir le modèle de langue optimal en fonction du niveau de la tâche : BioFlauBERT doit être privilégié au niveau des jetons, tandis que BioCamemBERT est plus adapté au niveau de la phrase.

Afin d'approfondir l'enrichissement linguistique de BioFlauBERT, cette dernière étude s'est concentrée sur un enrichissement cognitif dans le cadre d'une tâche de prédiction hiérarchique de causes primaires de décès. Lors du réglage fin de BioFlauBERT, ce processus a impliqué l'incorporation de la taxonomie CIM-10 dans la structure du modèle, dans la fonction de perte ou dans les deux simultanément. Les résultats ont clairement mis en évidence l'importance de l'enrichissement cognitif de BioFlauBERT avec des connaissances

propres au domaine médical, notamment avec l'incorporation dans la fonction de perte qui a obtenu les meilleurs résultats. Cette étude montre ainsi l'énorme potentiel d'enrichissement cognitif des modèles de langue pour la résolution de tâches complexes en médecine.

Enfin, tous ces travaux ont abouti à la conception d'un modèle capable de prédire des motifs de consultation à partir de champs textuels pré-remplis par des patients. Les résultats ont été remarquables pour la totalité des motifs de consultation ciblés et de nombreuses pistes d'amélioration sont à l'étude. Ce modèle est actuellement en phase de test dans plusieurs centres CardioParc pour éprouver sa robustesse auprès des cardiologues.

En définitive, cette thèse a éclairé la notion essentielle de modèles de langue contextuels et a révélé leur importance pour une compréhension plus fine et précise du langage naturel dans le domaine médical. Grâce à ces travaux, plusieurs modèles ont été développés et seront mis à disposition en libre accès à des fins de recherche, ainsi qu'une application pratique suffisamment fiable pour être testée en conditions réelles. Bien qu'ils soient encore imparfaits, il est vital de continuer à enrichir ces modèles pour qu'ils puissent mieux répondre aux exigences spécifiques et uniques de ce domaine en constante évolution, et ainsi permettre des avancées majeures dans la recherche médicale.

7.2 Perspectives

Les perspectives de ces travaux sont extrêmement prometteuses et ouvrent des portes à de nouvelles applications dans le domaine de la médecine et de la santé. Toutefois, il est important de souligner que des améliorations sont encore nécessaires, des pistes de recherche restent à explorer et des défis sont à relever pour étendre la recherche en TALN dans ce domaine. La suite de cette conclusion se propose d'explorer certaines de ces perspectives, en commençant par les possibilités à court terme, puis en évoquant les opportunités à long terme.

7.2.1 Court terme

Avant d'envisager de nouvelles applications, il est primordial de finaliser celle déjà en place : l'application de prédiction des motifs de consultation. Bien que ses résultats soient excellents, le modèle sous-jacent présente certaines limites notamment en termes de gestion des négations. De ce fait, le modèle peut attribuer le motif de consultation « Palpitations » à un patient qui indique « Je n'ai jamais eu de palpitations ». La gestion des négations est un défi difficile mais bien connu en TALN qui nécessite un corpus idéal comprenant autant de champs textuels affirmatifs que négatifs pour être résolu. Après avoir augmenté la taille des corpus de validation et de test, et les avoir séparés en fonction de la forme des champs textuels (affirmative ou négative), il serait judicieux de quantifier l'erreur commise sur les champs textuels négatifs. Une fois cette erreur quantifiée, le modèle pourrait être rendu plus robuste en incorporant les connaissances nécessaires, à l'image du chapitre 5, directement dans la fonction de perte, les données, ou a posteriori grâce à une correction post-hoc. Une piste d'amélioration supplémentaire consiste à optimiser le choix du seuil de décision pour l'attribution des motifs de consultation. Bien que l'application actuelle utilise un seuil de décision global, des seuils de décision locaux pourraient être plus adaptés et pertinents pour chaque type de motif. En poussant la réflexion encore plus loin, il serait même envisageable d'incorporer ces seuils en tant que paramètres du modèle, plutôt que comme des hyperparamètres, pour permettre un apprentissage automatique et précis des seuils lors du réglage fin.

7.2.2 Long terme

Avec l'essor des modèles de langue qui ont amélioré significativement les performances dans de nombreuses tâches de TALN, comprendre comment ces modèles effectuent leurs prédictions est devenu un défi plus important que jamais. Cela est d'autant plus vrai dans un domaine comme la médecine où des vies humaines peuvent être en jeu. Dans le cadre du TALN, ce défi est encore plus grand car les modèles de langue sont composés de plusieurs millions de paramètres et assimilables à d'énormes « boîtes noires » en raison de leur fonctionnement opaque très difficile à interpréter par l'utilisateur. Une première idée serait de visualiser les matrices des produits scalaires d'attention décrit dans l'équation

(2.6) du chapitre 2. Pour chaque jeton d'une phrase, ces matrices correspondent aux scores d'attention portés sur chacun des autres jetons de la phrase pour construire le plongement (incluant lui-même). Ainsi, l'exploration conjointe de tels matrices serait en mesure de mettre en évidence les jetons saillants sur lesquels le modèle de langue s'est focalisé pour effectuer la prédiction. Par ailleurs, d'autres chercheurs ont exploré des techniques telles que *Local Interpretable Model-Agnostic Explanations* (LIME) [46] et *SHapley Additive exPlanations* (SHAP) [47]. Cependant, ce défi reste encore très ouvert et des techniques beaucoup plus sophistiquées et fiables doivent être développées pour un jour comprendre intégralement les prédictions d'un modèle de langue.

Bien que cette thèse ait exploré l'utilisation des modèles de langue contextualisés pour la compréhension du langage naturel, il existe encore un vaste pan de recherche en TALN médical qui n'a pas été exploré : la génération du langage naturel. Pour une génération cohérente et pertinente de texte, il est probable que l'utilisation de modèles de langue basés sur des décodeurs comme le modèle de langue contextuel français BARThez [48] sera nécessaire. Les applications potentielles sont nombreuses notamment pour la génération automatique de documents médicaux tels que des comptes-rendus ou pour la création de chatbots dédiés à la santé. Alors que la médiatisation du célèbre chatbot ChatGPT [49] a ravivé l'engouement pour cette technologie, la construction complète d'un chatbot serait une extension passionnante des travaux proposés dans le chapitre 2. Avec la capacité d'offrir une assistance en temps réel et de répondre aux questions ou récolter les informations des patients, un tel chatbot pourrait alléger la charge de travail des professionnels de santé tout en améliorant l'expérience des patients.

Enfin, les données médicales ne se limitent pas aux seuls textes. Les variables numériques, qu'elles soient quantitatives ou qualitatives, ainsi que les images sont également des modalités importantes et en constante expansion. Chacune de ces modalités peut être traitée indépendamment, à l'aide de réseaux neuronaux pour les variables numériques (Annexe D) ou avec des architectures beaucoup plus complexes comme les réseaux convolutionnels [50] ou les *Vision Transformers* (ViT) [51] pour les images. L'actualité récente

est particulièrement riche en événements et confirme l'intérêt des approches multimodales avec notamment la sortie de GPT-4 [52]. Contrairement aux versions précédentes, cette dernière version de la série de modèle de langue GPT a la particularité d'être multimodale : en plus du texte, GPT-4 est capable de prendre des images en entrée pour générer du texte. Bien qu'il reste encore beaucoup à accomplir dans ce domaine, toutes ces modalités doivent être étudiées et surtout combinées grâce à l'intelligence artificielle afin d'offrir des outils puissants et durables pour les professionnels de santé.

Références

[1] H. Servy, "Big data : jouer au jeu de go pourrait-il donner du temps au médecin ?," *MISE AU POINT*, 2017. https://www.edimark.fr/Front/frontpost/getfiles/26015.pdf.

[2] H.-J. Kong, "Managing Unstructured Big Data in Healthcare System," *Healthcare Informatics Research*, 2019. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6372467/.

[3] J. Weizenbaum, "ELIZA — a computer program for the study of natural language communication between man and machine," *Association for Computing Machinery*, 1966. https://doi.org/10.1145/365153.365168.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proceedings of the 1st International Conference on Learning Representations*, 2013. https://arxiv.org/abs/1301.3781.

[5] J. Pennington, R. Socher, and M. Christopher, "GloVe : Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014. https://aclanthology.org/D14-1162/.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 conference North American Chapter of the Association for Computational Linguistics : human language technologies*, 2018. http://arxiv.org/abs/1810.04805.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Proceedings of the 31st conference on neural information processing systems*, 2017. https://arxiv.org/abs/1706.03762.

[8] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab, "FlauBERT : Unsupervised Language Model Pre-training for French," *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020. https://arxiv.org/abs/1911.03894.

[9] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, and B. Sagot, "CamemBERT : a Tasty French Language Model," *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, 2020. https://arxiv.org/abs/1911.03894.

[10] A. Radford and K. Narasimhan, "Improving Language Understanding by Generative Pre-Training," 2018. https://www.semanticscholar.org/paper/Improving-Language-Understanding-by-Generative-Radford-Narasimhan/cd18800a0fe0b668a1cc19f2ec95b5003d0a5035.

[11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019. https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe.

[12] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," 2020. https://arxiv.org/abs/2005.14165.

[13] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART : Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," 2019. https://arxiv.org/abs/1910.13461.

[14] J. Ba, J. Kiros, and G. Hinton, "Layer Normalization," 2016. https://arxiv.org/abs/1607.06450.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. https://arxiv.org/abs/1512.03385.

[16] Hendrycks, Dan and Gimpel, Kevin, "Gaussian Error Linear Units (GELUs)," 2016. https://arxiv.org/abs/1606.08415.

[17] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, 1999. https://www.sciencedirect.com/science/article/pii/S1364661399012942.

[18] C. Blanc, A. Bailly, Élie Francis, T. Guillotin, F. Jamal, B. Wakim, and P. Roy, "FlauBERT vs. CamemBERT : Understanding patient's answers by a French medical chatbot," *Artificial Intelligence in Medicine*, 2022. https://www.sciencedirect.com/science/article/pii/S093336572200029X.

[19] N. Grabar, V. Claveau, and C. Dalloux, "CAS : French Corpus with Clinical Cases," *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, 2018. https://aclanthology.org/W18-5614.

[20] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *Proceedings of the 54th annual meeting of the association for computational linguistics*, 2015. https://arxiv.org/abs/1508.07909.

[21] T. Kudo and J. Richardson, "Sentencepiece : A simple and language independent subword tokenizer and detokenizer for neural text processing," *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018. https://aclanthology.org/D18-2012/.

[22] J. L. Elman, "Finding structure in time," *Cognitive science*, 1990. https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1.

[23] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM : A search space odyssey," *IEEE transactions on neural networks and learning systems*, 2016. https://arxiv.org/abs/1503.04069.

[24] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, 1997. https://ieeexplore.ieee.org/document/650093.

[25] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data," *Proceedings of the 18th International Conference on Machine Learning*, 2001. https://dl.acm.org/doi/10.5555/645530.655813.

[26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *7th International Conference on Learning Representations*, 2019. https://arxiv.org/abs/1711.05101.

[27] B. Efron, "Bootstrap methods : another look at the jackknife," in *Breakthroughs in statistics*, Springer, 1992. https://link.springer.com/chapter/10.1007/978-1-4612-4380-9_41.

[28] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT : a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, 2019. https://arxiv.org/abs/1901.08746.

[29] I. Beltagy, K. Lo, and A. Cohan, "SciBERT : A Pretrained Language Model for Scientific Text," *Proceedings of the Second Workshop on Scholarly Document Processing*, 2019. https://arxiv.org/abs/1903.10676.

[30] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, *et al.*, "Huggingface's transformers : State-of-the-art natural language processing," 2019. https://arxiv.org/abs/1910.03771.

[31] N. Grabar and R. Cardon, "CLEAR - Simple Corpus for Medical French," in *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, 2018. https://aclanthology.org/W18-7002.

[32] F. A. A. Laleye, G. de Chalendar, A. Blanié, A. Brouquet, and D. Behnamou, "A French Medical Conversations Corpus Annotated for a Virtual Patient Dialogue System," *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020. https://www.aclweb.org/anthology/2020.lrec-1.72.

[33] N. Hiebel, K. Fort, A. Névéol, and O. Ferret, "CLISTER : A Corpus for Semantic Textual Similarity in French Clinical Narratives)," *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles*, 2022. https://aclanthology.org/2022.jeptalnrecital-taln.28.

[34] A. Névéol, C. Grouin, J. Leixa, S. Rosset, and P. Zweigenbaum, "The QUAERO French medical corpus : A ressource for medical entity recognition and normalization," *Proceedings of the Fourth Workshop on Building and Evaluating Ressources for Health and Biomedical Text Processing*, 2014. https://quaerofrenchmed.limsi.fr/.

[35] K. Huang, A. Singh, S. Chen, E. T. Moseley, C. Deng, N. George, and C. Lindvall, "Clinical XLNet : Modeling Sequential Clinical Notes and Predicting Prolonged Mechanical Ventilation," 2019. http://arxiv.org/abs/1912.11975.

[36] S. Gururangan, A. Marasovic, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't Stop Pretraining : Adapt Language Models to Domains and Tasks," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. https://arxiv.org/abs/2004.10964.

[37] L. N. Phan, J. T. A. andnHieu Tran, S. Chanana, E. Bahadroglu, A. Peltekian, and G. Altan-Bonnet, "SciFive : a text-to-text transformer model for biomedical literature," 2021. https://arxiv.org/abs/2106.03598.

[38] C. Silla and A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, 2011. https://link.springer.com/article/10.1007/s10618-010-0175-9.

[39] X. Qiu, X. Huang, Z. Liu, and J. Zhou, "Hierarchical Text Classification with Latent Concepts," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, 2011. https://aclanthology.org/P11-2105".

[40] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, and J. Schuecker, "Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems," *IEEE Transactions on Knowledge and Data Engineering*, 2021. https://ieeexplore.ieee.org/abstract/document/9429985.

[41] W. H. Organization, "International statistical classification of diseases and related health problems, 10th revision," 2009. https://apps.who.int/iris/handle/10665/44082?locale-attribute=en&.

[42] A. Névéol, A. Robert, F. Grippo, C. Morgand, C. Orsi, L. Pelikan, L. Ramadier, G. Rey, and P. Zweigenbaum, "CLEF eHealth 2018 Multilingual Information Extraction Task Overview : ICD10 Coding of Death Certificates in French, Hungarian and Italian," *Conference and Labs of the Evaluation Forum, eHealth*, 2018. https://hal.archives-ouvertes.fr/hal-02276492.

[43] D. Ghazi, D. Inkpen, and S. Szpakowicz, "Hierarchical versus flat classification of emotions in text," *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 2010. https://aclanthology.org/W10-0217.pdf.

[44] R. Babbar, I. Partalas, E. Gaussier, and M.-R. Amini, "On Flat versus Hierarchical Classification in Large-Scale Taxonomies," *Advances in Neural Information Processing Systems*, 2013. https://hal.archives-ouvertes.fr/hal-01118815.

[45] C. Millien, H. Chaput, and M. Cavillon, "La moitié des rendez-vous sont obtenus en 2 jours chez le généraliste, en 52 jours chez l'ophtalmologiste," *Études et résultats*, 2018. https://drees.solidarites-sante.gouv.fr/sites/default/files/2020-08/er1085-2.pdf.

[46] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You ?" : Explaining the Predictions of Any Classifier," 2016. https://arxiv.org/pdf/1602.04938.pdf.

[47] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," 2017. https://arxiv.org/abs/1705.07874.

[48] M. K. Eddine, A. J. P. Tixier, and M. Vazirgiannis, "Barthez : a skilled pretrained french sequence-to-sequence model," 2021. https://arxiv.org/pdf/2010.12321.pdf.

[49] C. Leiter, R. Zhang, Y. Chen, J. Belouadi, D. Larionov, V. Fresen, and S. Eger, "ChatGPT : A Meta-Analysis after 2.5 Months," 2023. https://arxiv.org/abs/2302.13795.

[50] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, 1995. http://yann.lecun.com/exdb/publis/pdf/lecun-bengio-95a.pdf.

[51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale," 2021. https://arxiv.org/abs/2010.11929.

[52] OpenAI, "GPT-4 Technical Report," 2023. https://arxiv.org/abs/2303.08774.

Annexes

Annexe A

Early vs. late data fusion in multimodal death-cause classification on text and structured data

Early vs. late data fusion in multimodal death-cause classification on text and structured data

Alexandre Bailly[1-5], Corentin Blanc[1-5], Élie Francis[1], Thierry Guillotin[1], Fadi Jamal[6], Pascal Roy[2-5]

[1] Everteam Software, Lyon, France

[2] Université de Lyon, Lyon, France

[3] Université Lyon 1, Villeurbanne, France

[4] Service de Biostatistique-Bioinformatique, Pôle Santé Publique, Hospices Civils de Lyon, Lyon, France

[5] Équipe Biostatistique-Santé, Laboratoire de Biométrie et Biologie Évolutive, CNRS UMR 5558, Villeurbanne, France

[6] IzyCardio - CardioParc, Lyon, France

Corresponding author:

Alexandre Bailly

Everteam Software

17 quai Joseph Gillet

F-69004, Lyon, France

a.bailly@everteam.com

Abstract

Background and objective In medicine, data of various modalities are daily collected and then analyzed for diagnostic and prognostic modeling. As their browsing may be difficult and/or time consuming, statistical methods were developed to deal with several modalities at the same time. According to the moment of data fusion, these multimodal methods fall into two categories: early fusion occurs before prediction and late fusion after it. Most previous studies on multimodal models with text and structured (categorical or numerical) data showed better performance than unimodal models with either type of data but reached no consensus about which fusion is more performant.

Method This study compared early vs. late fusion in a cause-of-death classification task on mixed (text and structured) data from death certificates and changes in the quantity of text to reduce the amount of information it brings. Text data were analyzed using BioFlauBERT, a French medical language model. Two early fusion approaches and two late fusion approaches were compared between them and with their corresponding unimodal models.

Results The unimodal model that used text data led to better results than the one that used structured data whatever the quantity of text considered. The additional information provided by the multimodal models decreased with the increase in the quantity of information in text data, leading to very few differences in case of high informative text. Considering the same number of death-certificate text lines, there was no clear performance difference between early and late fusion, but more lines led to better results with either unimodal or multimodal models whatever the fusion method.

Conclusions In this study, using multimodal models allowed performance improvement only when the text brought a limited amount of information. Whatever the information carried by the text, there were no differences between early and late fusion. In the presence of highly informative

modalities, multimodal methods did not prove necessary; but, in the presence of little informative modalities, either early or late fusion may be used to improve performance.

Keywords

Multimodal Analysis; Deep Learning; Text Classification; BioFlauBERT

1 Introduction

Medical consultations, procedures, and lab analyzes result in medical records with a great number of data about each patient. These data are of several types (or modalities) such as text data or structured data (categorical or numerical) and most, in either modality, are needed by the physician to establish a diagnosis or a prognosis. However, the analysis of information with different data modalities can be very complex. To help physicians, artificial intelligence (AI) methods were developed to process different data modalities either separately (with unimodal models) or jointly (with a single multimodal model). These AI models may be more or less complex according to the modalities to process. Indeed, structured data may be processed by machine learning or deep learning models [1], whereas, to be efficient, text data processing requires more complex deep learning models.

Over the last few years, the use of contextual language models based on Transformers [2] and pre-trained on large datasets led to performance improvement in most Natural Language Processing (NLP) tasks. This kind of language model may be used on downstream tasks such as text classification or named entity recognition by connecting an additional neural network after the model and applying an end-to-end training to all parameters of the language model and the additional neural network. In medicine, BERT, an English language model [3] showed a very good performance in medical NLP tasks [4, 5] but very few extensions of BERT to the French medical language were made. In 2022, Blanc et al. [6] compared two French language models called CamemBERT [7] and FlauBERT [8] and showed a slight superiority of FlauBERT over CamemBERT in medical Natural Language Understanding. Later works were conducted to improve the performance of FlauBERT in the medical field through continual pre-training; this resulted in BioFlauBERT, a medically-specialized French language model [9].

These works demonstrated that, with optimal training, BioFlauBERT outperformed FlauBERT in medical NLP tasks. However, despite this superiority, BioFlauBERT was still unable to process mixed modalities therefore the question of integrating structured data in addition to text with BioFlauBERT still arises.

One issue in medical AI is the joint analysis of text and structured data with multimodal models. According to the moment of modality fusion into the network structure, multimodal approaches fall into two main categories [10]: i) early fusion, whenever the fusion occurs after a few processing of the modalities and its result becomes the input of a single predictor [11] (Figure 1, upper panel); and, ii) late fusion, whenever the fusion includes the predictions obtained independently from all modalities as inputs of a final predictor [12] (Figure 1, lower panel). Within this context, published diagnostic studies of multimodal models with text and structured data showed better performance than unimodal models with single-type data (Jin et al. [13] and Khadanga et al. [14] for early fusion and Xu et al. [15] for late fusion). The results of these studies retained additional diagnostic contributions of each data modality, highlighting a specific effect of each modality data. However, no study has yet examined the interest of multimodal models according to the quantity of information provided by each modality. Furthermore, to the best of our knowledge, until very recently, apart from Xu et al. [16], no other authors compared early vs. late fusion with text and structured data. In 2021, these authors concluded that late fusion performed better than early fusion. This confirmed a previous conclusion by Snoek et al. [12] of studies on text, audio, and visual modalities, but disagreed with a much later conclusion by Mervitz et al. [17] that early fusion performs better than late fusion in predicting movie trailer classes with audio and video data as inputs. Thus, today, there is no consensus about the best fusion approach. However, in nearly all previous studies, various modalities contributed a meaningful quantity of information in unimodal

analyzes and, sometimes, one of the modalities was poorly informative and its corresponding unimodal model performed poorly. Anyway, no simulations were conducted to analyze the effect of the quantity of information brought by each modality in early and late fusion multimodal models.

This study aims to compare the performance in classifying the primary causes from death certificates of unimodal and multimodal models (early and late fusion) with a mix of text and structured data and varying the quantity of information brought by text data.

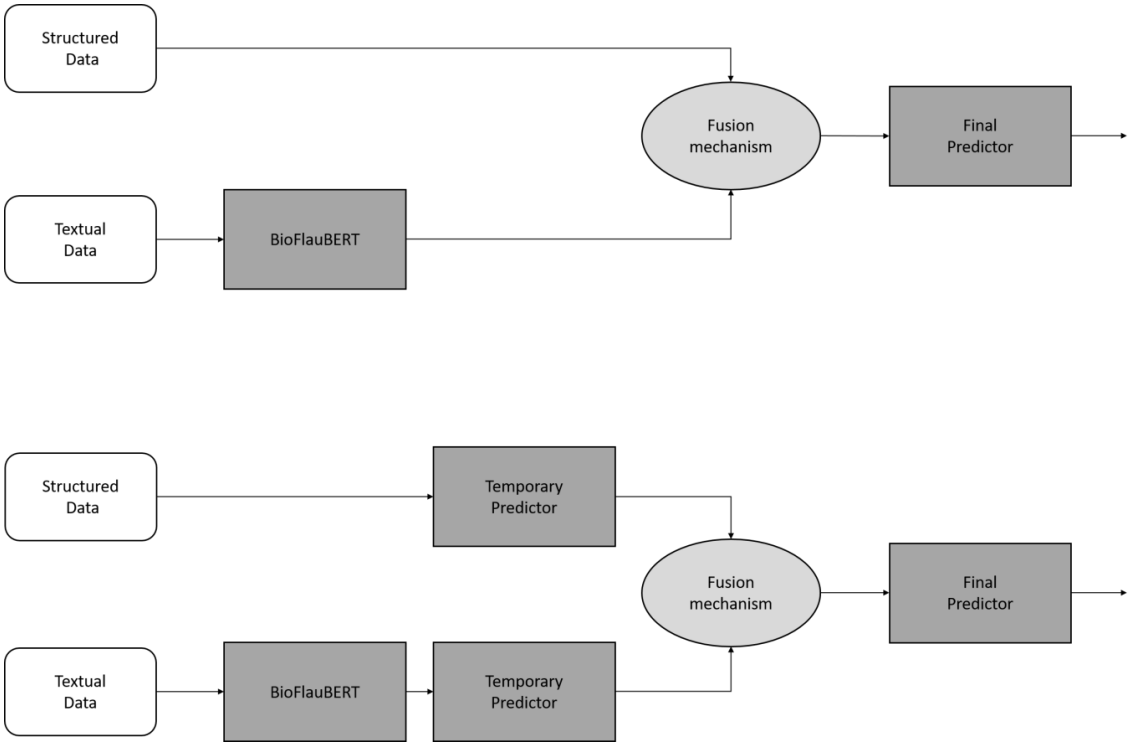


Figure 1 - Early and late fusion structures

2 Materials and methods

2.1 The data

The data used in this study were the set of French death certificates collected from 2006 to 2015 by the *Institut National de la Santé et de la Recherche Médicale* (INSERM) [18]. Each certificate contains: i) one to four lines of free text: the first indicates the main cause of death and the others the etiologies of that cause, the last one corresponding usually to the primary cause of death; ii) some structured data: sex, age at death, place of death (home, hospital, private clinic, public place, retirement home or other); iii) the primary cause of death as attributed by an expert of the *Centre d'épidémiologie sur les causes médicales de décès* (CépiDC) and labelled according to the chapters of International Classification of Diseases, 10th revision (ICD-10). Thus, in this dataset, each certificate was associated with one of 18 chapters of ICD-10.

The present study considered only certificates with four lines filled out; this kept for analysis 20,538 certificates. In the following, the ICD-10 chapters were the classes to predict. To vary the quantity of information stemming from the text data, four subdatasets were built by considering only a part of the four lines: the first included only the first line of each certificate; the second only the two first lines, and so on.

2.2 The unimodal models

2.2.1 The structured data modality

To deal with a vector of structured data, a neural network with two linear layers of 25 and 18 neurons was used. As activation functions, a sigmoid function was used for the first layer and a Softmax function for the second layer so as to obtain class probability vectors.

Let s_i be the vector of structured data of certificate i, σ the Softmax function, $sigm$ the sigmoid function, and L a linear layer defined as $L(X) = \Omega X + \beta$, with Ω a matrix of weights and β a vector of bias. Then, with h_i the output of the first layer, $\hat{p}_i$ the predicted class probability vector for any certificate i is:

$$h_i = sigm(L(s_i)) \quad (1)$$

$$\hat{p}_i = \sigma(L(h_i)) \quad (2)$$

2.2.2 The text data modality

The French language model BioFlauBERT was used to analyze the text data [9]. The text was first tokenized and indexed by Byte Pair Encoding (BPE) tokenizer [19] (i.e., the sentences were split into words and subwords, then matched to words from BioFlauBERT's vocabulary). Using this language model led to a vector t_i of size 768 that represented the text of the certificate i. This vector was then passed through a linear layer of 18 neurons with Softmax activation function to obtain $\hat{p}_i$. With this method, all the parameters of BioFlauBERT and those of the linear layer were trained end-to-end. The predicted class probability vector for any certificate i is:

$$\hat{p}_i = \sigma\big(L(t_i)\big) \quad (3)$$

2.3 The multimodal models

2.3.1 The early fusion models

In this work, two multimodal models that use early fusion were implemented and compared. The first is based on a weighted summation of a vector computed from text data and another vector computed from structured data; the second is based on attention gates used to merge the two vectors.

The projection and weighted summation multimodal method was introduced by Gu et al. [20]. To analyze simultaneously text and structured data, the text was first vectorized (using BioFlauBERT) to obtain a representation t_i as in the approach with textual data (see section 2.2.2). In parallel, the vector of structured data was passed through a linear layer to obtain a vector of the same size as that of the output of the language model. Next, a weighted summation of the two vectors provided a single vector v_i (Figure 2, upper panel). Vector v_i was then calculated according to the below-shown formula (W_s being the vector of weights associated with the structured data) and then passed through a linear layer with a Softmax activation function to provide $\hat{p}_i$ probabilities.

$$v_i = t_i + W_s L(s_i) \quad (4)$$

$$\hat{p}_i = \sigma\big(L(v_i)\big) \quad (5)$$

The attention gates based multimodal method was inspired by Rahman et al. [21] (Figure 2, lower panel). The text was first vectorized using the language model, then a gated summation of BioFlauBERT's output and the vector of structured data was performed. The concatenation of BioFlauBERT's output and the vector of structured data was passed through a linear layer with a ReLU (Rectified Linear Unit function) activation function to provide a new vector g_i. With $[.\,;.]$ as concatenation operator, g_i was obtained as follows:

$$g_i = ReLU\big(L([t_i; s_i])\big) \quad (6)$$

With this vector, a new vector h_i was created using the following formula:

$$h_i = g_i \odot L(s_i) + B_h \quad (7)$$

where B_h is a bias vector and $\odot$ the element wise product. Finally, a vector was computed using a weighted summation of t_i and h_i so as to create a multimodal vector v_i:

$$v_i = t_i + \alpha h_i \quad (8)$$

$$\alpha = \min\left(\frac{||t_i||_2}{||h_i||_2}\beta, 1\right) \quad (9)$$

with β as hyperparameter (fixed to 0.2) and $||\cdot||$ as L_2 norm. Vector v_i was finally passed through a linear layer with Softmax activation function to obtain the probabilities; i.e. $\hat{p}_i = \sigma\big(L(v_i)\big)$.

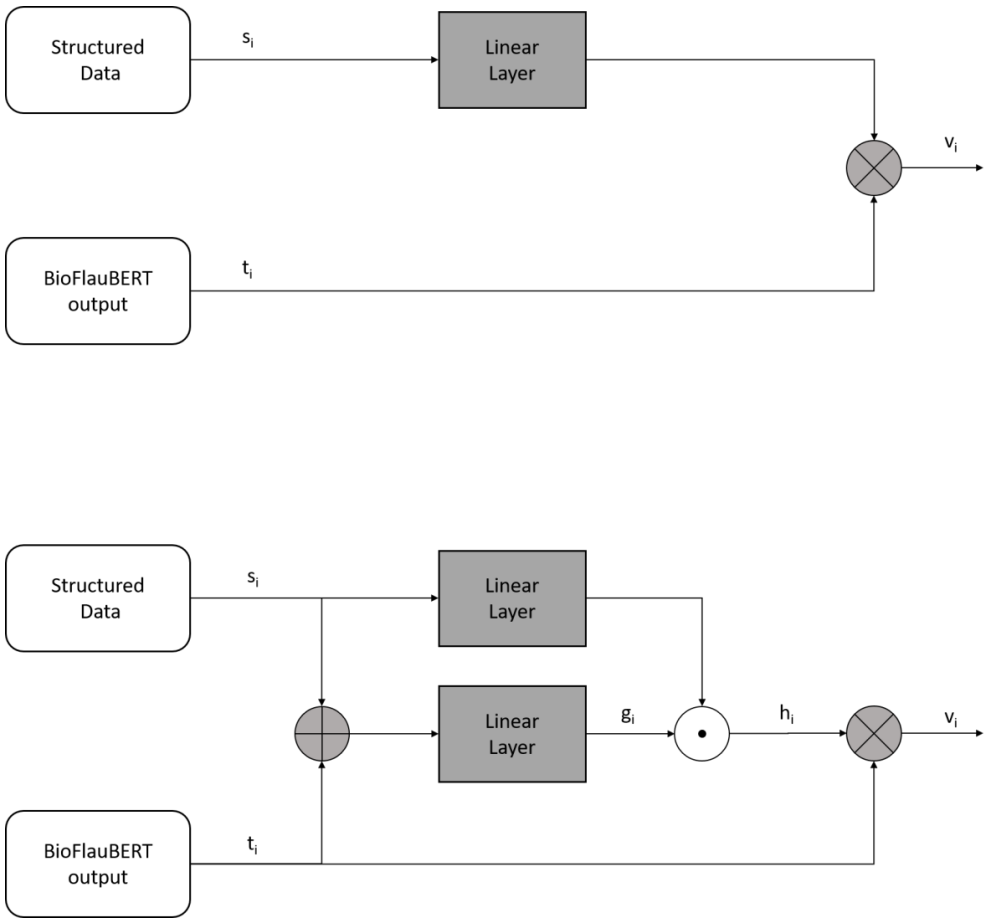


Figure 2 - Early fusion. Mechanism with weighted summation (upper panel) and mechanism with attention gate (lower panel). ⊗ weighted summation - ⊕ concatenation operator - ⊙ element-wise product operator.

2.3.2 The late fusion models

Two late fusion methods were used. In the first method, two unimodal models processed separately structured data and text data to yield two vectors of probabilities $\hat{p}_i^s$ and $\hat{p}_i^t$. Then, a weighted summation of these two vectors with a Softmax function led to a final vector of probability $\hat{p}_i$. In that case, $\hat{p}_i = \sigma(W_s\hat{p}_i^s + W_t\hat{p}_i^t)$, W_s and W_t being two vectors of weights updated during the training process and σ being the Softmax function.

In the second method, vectors $\hat{p}_i^s$ and $\hat{p}_i^t$ were computed in the same way but the fusion was carried out by a linear layer that took as input the concatenation of the two vectors and a Softmax function was used to obtain a probability vector $\hat{p}_i = \sigma(L([\hat{p}_i^s; \hat{p}_i^t]))$.

2.4 The training process

In this work, different hyperparameters were chosen among classical values in the literature. The number of epochs (i.e., the number of times the corpus is seen by the model) was set to 5, the batch size (i.e., the number of observations seen at the same time) to 16, and the learning rate (i.e., the weight increment) to 2^{e-5} in the language model trained by transfer learning and 1^{e-2} in the other layers trained from scratch. The learning rate was left to decrease linearly down to 0, a value reached at the end of the last epoch.

Furthermore, all the parameters of the linear layers were initialized with Xavier normal function [22] that improves the training of the neural network with sigmoid activation functions. All models were trained by backpropagation using AdamW learning algorithm [23] with a gradient clipping [24] of 1.0 to avoid a gradient-exploding effect. The loss function used for the backpropagation process was the Cross Entropy. In a classification task, Cross-Entropy CE for N observations and C classes is defined as below, $\hat{p}_i^c$ being the probability that

observation i belongs to class c and y_{ic} a binary indicator equal to 1 when the observation belongs to class c and 0 otherwise:

$$CE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log\left(\hat{p}_{ic}\right) \quad (10)$$

2.5 Evaluation

Model performance evaluation used the weighted F1-score. This score was computed for each class and the weighted mean score calculated. Doing so, the ponderation of each class corresponds to its prevalence.

As recommended, a bootstrap sampling on death causes was used rather than a data splitting approach. The bootstrap was stratified to ensure the conservation of label repartition. A training set was randomly drawn with replacement among the observations of each stratus; undrawn observations formed the test set. By construction, about 37% of the dataset remained in the test set. This data splitting was done 50 times. Then each model was trained on the training sets and evaluated on the corresponding test sets. Finally, the median, the minimum, and the maximum of the weighted F1-scores were computed. Bootstrap samples in which the loss function converged to a local minimum (when the cross-entropy value was higher than the others) were not taken into account.

2.6 Implementation details

In this work, Python 3.8.2 was the programming language. Package Torchtext 0.9.1 was used to load and tokenize the corpus; package Transformers 3.1.0 from HuggingFace to apply BioFlauBERT; package PyTorch 1.8.1 to deal with the NN architecture, the late fusion approaches, and model training; package Multimodal-transformers 0.1.2a0 [20] to implement early fusion approaches; and package Scikit-learn 1.0.2 to compute the F1-scores.

3 Results

The unimodal model with structured data led to a low predictive performance; the median of the F1-score was only 14.2 (Table 1). In comparison, the unimodal model with text data and only one line from death certificates led to a higher performance (F1-score: 28.8). An increase in the F1-score was observed when more lines were considered in a unimodal model with text data; the F1-score increased up to 82.3 in four-line certificates.

Table 1 - F1-scores obtained with unimodal models on 50 bootstraps.

Modality	Median F1-score [min, max]
Structured data	14.2 [13.7, 15.9]
Text data	
Only the first line	28.8 [27.5, 29.8]
First 2 lines	39.8 [39.0, 41.1]
First 3 lines	56.7 [55.9, 57.5]
All 4 lines	82.3 [81.3, 83.1]

Concerning multimodal models that associate structured and text data, with only one line both early fusion methods led to similar F1-scores: 31.3 with weighted summation and 32.4 with attention gate (Table 2). This performance similarity was also observed increasing the number of considered lines of text data. With both models, the performance increased the same way along the number of lines considered; the F1-score increased up to 82.3 and 82.2, respectively.

The same similarities between late fusion methods with linear layer and weighted summation were observed and the same effect of the number of lines considered.

Table 2 - F1-scores obtained with various methods for multimodal models and various numbers of death-certificate lines.

	Number of text lines from death certificates			
Type of multimodal fusion	1	2	3	4
Early fusion methods				
Weighted summation	31.3 [30.3, 32.6]	40.9 [39.5, 42.2]	56.9 [56.3, 58.4]	82.3 [81.6, 83.0]
Attention gate	32.4 [31.6, 33.3]	41.4 [40.3, 42.4]	57.4 [55.9, 58.0]	82.2 [81.6, 83.1]
Late fusion methods				
Linear layer	33.3 [32.2, 34.8]	42.2 [41.3, 43.0]	57.6 [56.7, 59.7]	82.5 [81.5, 83.3]
Weighted summation	33.2 [30.9, 34.0]	42.0 [40.6, 43.0]	57.7 [56.8, 59.1]	82.6 [81.6, 83.6]

The results are the medians [min, max] of 50 bootstraps

4 Discussion

The unimodal model with structured data led to a poor predictive performance. This seems to indicate that, here, structured data may not be essential for the prediction process. On the contrary, the unimodal model with text data led to much better performance even with only one written line from the death certificates. This was expected because death certificate texts usually include most of the information about the cause of death, which is the specific information requested. Furthermore, with all multimodal models, adding more lines (that is, more text data) increased performance. The improvement of performance of the unimodal model with text data along with an increased number of lines was also expected because an extended description of death circumstances increases the probability of determining the primary cause of death.

In comparison with the unimodal text model, the additional information provided by multimodal models decreased when increasing the number of lines considered, leading to very few differences for three lines and no differences for four. This would indicate that multimodal methods capture most of the information from text data alone and that structured data bring no additional relevant information able to improve prediction. This could be explained by the huge information provided here by the four lines of the death certificate, and by the little additional one provided by the structured data.

With the same number of lines, early fusion and late fusion led to close F1-scores whatever the fusion method. In this cause-of-death classification task, the type of fusion considered seemed to have no importance, the performance of all trained multimodal models being similar with the same number of lines considered. This result is in line with the lack of consensus in the literature regarding the importance of late vs. early data fusion [12, 16, 17].

Even if the impact of the information provided by structured and text data was underlined in this study, the specific contribution of each modality as well as the contribution of their association in the multimodal model is not easily identifiable. Studying the contribution of each modality in multimodal models is a field of AI called 'explainable AI' (XAI) [10]. Such a study may be the object of future works on the same data.

One asset of this study was building models with various quantities of information extracted from text data through setting the number of certificate lines examined. This allowed considering very poor as well as very informative texts and comparing the interest of using multimodal techniques in each case. However, one limitation was that the cause of death was highly correlated with the text input and only few structured data were available. With more correlated structured data, the quantity of information extracted from them can then be varied and studied in the same way as text data here.

5 Conclusions

In this application, unimodal models performed better with text than with structured data whatever the quantity of text. Examining the same number of death certificate lines, early fusion did not perform clearly better than late fusion. Considering more lines improved the results of both unimodal and multimodal models. However, taking into account at least 3 text lines, the performance of multimodal models did not still exceed those of the unimodal model. Further studies with a higher quantity of structured data should be carried out to evaluate the interest of multimodal models in classifying the causes of death from death certificates.

Conflict of interest statement

The authors have no financial or personal relationships with other people or organisations that could inappropriately influence (bias) their work.

Funding

This work was supported by Association Nationale de la Recherche et de la Technologie (ANRT) [grant number 2019/1373]. The sponsor had no role in the study design; the collection, analysis, and interpretation of the data; the writing of the report; and the decision to submit the article for publication.

Acknowledgments

The authors are grateful to Jean Iwaz (Hospices Civils de Lyon) for augmenting, editing, proofreading, and formatting the latest versions of the manuscript.

References

[1] Alexandre Bailly, Corentin Blanc, Élie Francis, Thierry Guillotin, Fadi Jamal, Béchara Wakim, and Pascal Roy. Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Computer Methods and Programs in Biomedicine*, 213:106504, 2022, doi: 10.1016/j.cmpb.2021.106504.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of Deep bidirectional transformers for language understanding, 2018, doi: 10.48550/ARXIV.1810.04805.

[4] Matthew Tang, Priyanka Gandhi, Md Ahsanul Kabir, Christopher Zou, Jordyn Blakey, and Xiao Luo. Progress notes classification and keyword extraction using attention-based deep learning models with bert, 2019, doi: 10.48550/ARXIV.1910.05786.

[5] Stefano Silvestri, Francesco Gargiulo, Mario Ciampi, and Giuseppe De Pietro. Exploit multilingual language model at scale for icd-10 clinical text classification. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7, 2020, doi: 10.1109/ISCC50000.2020.9219640.

[6] Corentin Blanc, Alexandre Bailly, Élie Francis, Thierry Guillotin, Fadi Jamal, Béchara Wakim, and Pascal Roy. Flaubert vs. camembert: Understanding patient's answers by a French medical chatbot. *Artificial Intelligence in Medicine*, 127:102264, 2022, doi: 10.1016/j.artmed.2022.102264.

[7] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suarez, Yoann Dupont, Laurent Romary, Eric de la Clergerie, Djamé Seddah, and Benoît Sagot. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, doi: 10.18653/v1/2020.acl-main.645.

[8] Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France, May 2020. European Language Resources Association, doi: 10.48550/ARXIV.1912.05372.

[9] Corentin Blanc, Alexandre Bailly, Élie Francis, Thierry Guillotin, Fadi Jamal, Pascal Roy. BioFlauBERT: a French medical language model pre-trained after corpus size selection. 2022 (submitted on September 15, 2022 and under review).

[10] Gargi Joshi, Rahee Walambe, and Ketan Kotecha. A review on explainability in multimodal deep neural nets. *IEEE Access*, 9:59800–59821, 2021, doi: 10.1109/ACCESS.2021.3070212.

[11] Kuan Liu, Yanen Li, Ning Xu, and Prem Natarajan. Learn to combine modalities in multimodal deep learning, 2018, doi: 10.48550/ARXIV.1805.11730.

[12] Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, page 399–402, New York, NY, USA, 2005. Association for Computing Machinery, doi: 10.1145/1101149.1101236.

[13] Mengqi Jin, Mohammad Taha Bahadori, Aaron Colak, Parminder Bhatia, Busra Celikkaya, Ram Bhakta, Selvan Senthivel, Mohammed Khalilia, Daniel Navarro, Borui Zhang, Tiberiu Doman, Arun Ravi, Matthieu Liger, and Taha Kass-hout. Improving hospital mortality prediction with medical named entities and multimodal learning, 2018, doi: 10.48550/ARXIV.1811.12276.

[14] Swaraj Khadanga, Karan Aggarwal, Shafiq Joty, and Jaideep Srivastava. Using clinical notes with time series data for ICU management. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6432–6437, Hong Kong, China, November 2019. Association for Computational Linguistics, doi: 10.18653/v1/D19-1678.

[15] Keyang Xu, Mike Lam, Jingzhi Pang, Xin Gao, Charlotte Band, Piyush Mathur, Frank Papay, Ashish K. Khanna, Jacek B. Cywinski, Kamal Maheshwari, Pengtao Xie, and Eric Xing. Multimodal machine learning for automated ICD coding, 2018, doi: 10.48550/ARXIV.1810.13348.

[16] Zhen Xu, David R So, and Andrew M Dai. Mufasa: Multimodal fusion architecture search for electronic health records. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10532–10540, 2021.

[17] Joseph H. Mervitz, Johan Pieter de Villiers, J.Pieter Jacobs, and Mauritz H.O. Kloppers. Comparison of early and late fusion techniques for movie trailer genre labelling. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–8, 2020, doi: 10.23919/FUSION45008.2020.9190344.

[18] Aurélie Névéol, Aude Robert, Francesco Grippo, Claire Morgand, Chiara Orsi, Laszlo Pelikan, Lionel Ramadier, Grégoire Rey, and Pierre Zweigenbaum. Clef ehealth 2018

multilingual information extraction task overview: Icd10 coding of death certificates in French, Hungarian and Italian. In *CLEF*, 2018.

[19] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics, doi: 10.18653/v1/P16-1162.

[20] Ken Gu and Akshay Budhkar. A package for learning on tabular and text data with transformers. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 69–73, Mexico City, Mexico, June 2021. Association for Computational Linguistics, doi: 10.18653/v1/2021.maiworkshop-1.10.

[21] Wasifur Rahman, Md Kamrul Hasan, Sangwu Lee, AmirAli Bagher Zadeh, Chengfeng Mao, Louis-Philippe Morency, and Ehsan Hoque. Integrating multimodal information in large pretrained transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2359–2369, Online, July 2020. Association for Computational Linguistics, doi: 10.18653/v1/2020.acl-main.214.

[22] Xavier Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, 9:249–256, 01 2010.

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017, doi: 10.48550/ARXIV.1711.05101.

[24] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *ArXiv*, abs/1211.5063, 2012.

Annexe B

Importance of the number of classes and the proportion of labeled data in semi-supervised text classification

Importance of the number of classes and the proportion of labeled data in semi-supervised text classification

Alexandre Bailly[a-e], Corentin Blanc[a-e], Élie Francis[a], Fadi Jamal[f], Pascal Roy[b-e]

[a] Everteam Software, Lyon, France

[b] Université de Lyon, Lyon, France

[c] Université Claude Bernard Lyon 1, Villeurbanne, France

[d] Service de Biostatistique-Bioinformatique, Pôle Santé Publique, Hospices Civils de Lyon, Lyon, France

[e] Équipe Biostatistique-Santé, Laboratoire de Biométrie et Biologie Évolutive, CNRS UMR 5558, Villeurbanne, France

[f] izyCardio - CardioParc, Lyon, France

Corresponding author:

Alexandre Bailly

Everteam Software

17 quai Joseph Gillet

F-69004, Lyon, France

ext-alexandre.bailly@chu-lyon.fr

Abstract

Artificial Intelligence models specialized in natural language processing were developed to help analyzing tremendous amounts of medical textual data. Training these models requires labeling texts, which is a tedious process. Training artificial intelligence models with semi-supervised learning using unlabeled and labeled text could be time-saving. A new semi-supervised approach called task-adaptive pretraining (TAPT) consists in a continuous pretraining of the language model on unlabeled data before fine-tuning it on labeled data. This paper investigates the influences of the number of classes and the proportion of labeled data on the performance of self-training, TAPT, and supervised learning approaches in a death-cause classification task. The text was processed by the medical French language model BioFlauBERT. Depending on the level of the ICD-10 cause of death, 109,212 death certificates could be associated with 14 to 321 classes and the proportion of labeled data ranged between 0.5% and 100% by 10 increments. Performance (F-1 scores) of various approach, number of classes, and percent labeling combinations were compared. With all approaches, increasing the proportion of labeled data increased performance. With low proportions of labeled data, self-training and TAPT (semi-supervised learning approaches) led to better performance than supervised learning. With few classes and same percent labelling, self-training and TAPT were equally performant; but TAPT performed slightly better with < 5% percent labeling. With a high number of classes, the gain in performance was higher with self-training than with TAPT whatever the proportion of labeled data. In conclusion, in a text classification task with few labeled data, semi-supervised learning might lead to more performant models than supervised learning. Furthermore, in classification tasks with high numbers of classes, self-training might lead to better models than TAPT.

Keywords

Self-training; Semi-supervised Learning; Task-Adaptive Pretraining; Text Classification

1 Introduction

Medical visits usually require collecting textual data about the patients. To help physicians analyze these data, Artificial Intelligence (AI) models specialized in Natural Language Processing (NLP) were developed. Designed for automatic text classification, these models are usually trained by supervised learning which requires text annotation with labels (i.e., defining classes). This text-labelling task requires the intervention of an expert and is tedious and time-consuming. Thus, using an unlabeled text together with a much shorter labeled one to train AI models could help saving precious time.

Unlabeled data as well as labeled data may be used in semi-supervised learning [1]. In this kind of learning, the information present in the unlabeled data was used to increase the performance of the model trained on labeled data. One way to use unlabeled data for training is to assign them classes by 'pseudo-labeling'; adding pseudo-labeled to labeled data increases the size of the training set on which the final model will be trained. In such a method, unlabeled data are used in a "task-specific way" [2]. Indeed, the larger is the training dataset the better is the predictive performance of the trained model [3]. One of the most known semi-supervised learning approaches is self-training [4]. In 2017, Pavlinek et al. [5] used a self-training approach for a text classification task and several datasets. They used very few labeled data (4 to 52, depending on the dataset considered), thousands of unlabeled data, and a number of classes varying between 4 and 52. They showed that self-training allowed improving the performance of the trained model in comparison with a simple supervised learning with labeled data only.

Over the last few years, new contextual language models based on Transformers architecture [6] improved the performance of most NLP tasks. To learn representing the text efficiently, such models are pretrained on huge quantities of unlabeled text then usually

fine-tuned on a downstream task (i.e., adding a neural network after the language model and applying an end-to-end training to all parameters). With the rise of Transformers-based models, a new semi-supervised approach was developed. This approach, based on a method called task-adaptive pretraining (TAPT) [7] consists in a continuous pretraining of the model on unlabeled texts before fine-tuning on the labeled data. Here, the unlabeled data are used in a "task-agnostic way" [2]. The goal is to improve the representation of the text for the considered task; thus, the final prediction. In 2020, Gururangan et al. [8] used the TAPT based approach with a biomedical dataset with five classes for text classification. Using only 500 labeled and 180,000 unlabeled data (i.e., 0.3% labeled data), they showed that TAPT based approach provided better results (in terms of F1-scores) than supervised learning.

In 2021, Li et al. [9] used both TAPT based approach and self-training approaches for text classification tasks, each task considering few classes and various proportions of labeled data. They showed that, whatever the task, semi-supervised learning performed better than supervised learning. However, the gain in performance decreased with the increase of the proportion of labeled data. Whereas the influence of the number of classes was not discussed in Li's paper, Sulea et al. [10] demonstrated in 2017 that, in a text classification task, the number of classes influenced the trained model performance but used only supervised learning. To the best of our knowledge, no studies have investigated yet the influence of the number of classes on the performance of semi-supervised learning approaches for text classification.

This work aimed to study the influence of the number of classes and the proportion of labeled data on the performance of TAPT based approach and self-training algorithms in a death-cause classification task.

2 Materials and methods

2.1 The data

The dataset used in this study was a sample of 124,876 French death certificates issued 2006 to 2015 (courtesy of the *Institut National de la Santé et de la Recherche Médicale,* INSERM) [11]. Each certificate included one to four lines of free text to tell the cause of death. Besides, each certificate included the primary cause of death as assigned by experts of the *Centre d'épidémiologie sur les causes médicales de Décès* (CépiDC) according to the ICD-10 (International Classification of Diseases, 10th revision) [12]. In the analyzed certificates, the primary cause of death was coded with the most detailed level (i.e., Level 4) of the ICD-10 classification, which allowed accessing upper levels. When the number of death certificates with an identical Level 4 code was < 50, these certificates (precisely, 15,664) were excluded; this left 109,212 certificates that could be associated with 321 classes. The use of lower levels led to 210 Level 3, 64 Level 2, and 14 Level 1 classes.

2.2 The dataset splitting procedure

The death certificates kept for analysis were first split into two sets: a test set and a second set further split into a validation set and a training set. The latter was then divided into one labelled and another unlabeled set.

Each of the four above-defined sets was stratified according to the top four ICD-10 codes in a way that ensures the presence of the same percentage of ICD-10 codes in each set. A 10-fold cross-validation was performed to evaluate the performance of the various semi-supervised learning approaches. At each iteration of the cross-validation, 10% of the data were randomly drawn to form the test set, 10% of the remaining data (9% of the

dataset) drawn to form the validation set, and, finally, the remaining part of the dataset (81%) was randomly split to form several training sets with p% labeled data and $(100 - p)$% of unlabeled data (after masking the labels). The training set included thus nearly 88,500 certificates. Here, proportions p were 100, 75, 50, 35, 20, 10, 5, 2, 1, and 0.5%. The special case where $p = 100\%$ corresponds of a fully labeled dataset and was used as baseline in supervised learning.

2.3 The contextual language model

The text was processed by the medical French language model BioFlauBERT [13] with an additional linear layer at the end of the language model to provide the prediction. This model is a Transformers-based model [6] extended from FlauBERT [14]; it has proven to be the optimal model to treat French medical NLP tasks [13]. Pretrained on a huge amount of text data, this model is able to vectorize contextually a medical text. BioFlauBERT was used in a fine-tuning way; i.e., adding a linear layer at the end of the language model and updating all the parameters of the language model and the linear layer, end-to-end.

2.4 The training algorithms

2.4.1 Supervised learning on labeled data

Let $D_l = \{(x_i, y_i)\}_N$ be a set of N labeled data where x_i is a sequence of words and y_i the sequence's corresponding label. Besides, let $D_u = \{x_j\}_M$ be a set of M unlabeled sequences. Both D_l and D_u were randomly drawn without replacement from the training set of the cross-validation.

A supervised learning on the labeled data only was used as a baseline. The text $x_i \in D_l$ was first tokenized with BPE (Byte Pair Encoding) tokenizer [15] in order to split it into

words and sub-words and index it with BioFlauBERT's vocabulary. The tokenized text t_i was then passed through BioFlauBERT to obtain the corresponding vectorial representation v_i of size 768 which was passed through a linear layer associated with a Softmax to obtain a prediction vector $\hat{p}_i$ (which size depends of the level of ICD-10 considered). The resulting classification model of this approach is a classical fine-tuned BioFlauBERT.

2.4.2 Semi-supervised learning approaches

2.4.2.1 Self-training

Self-training is an iterative algorithm based on successive supervised learnings with augmentation of the training set. In a first step, BioFlauBERT was fine-tuned with D_l. Afterwards, the obtained model $\mathcal{M}_0$ was used to obtain a prediction for each $x_j \in D_u$. Whenever the output probability of the model for an observation was above a fixed threshold τ, the prediction was considered reliable and the predicted label was associated with the unlabeled data and added to D_l to form a new set T_i. This step allowed obtaining "pseudo-labeled" data. Then the model was retrained using both labeled data and pseudo-labeled data from all previous iterations. The model training was stopped either after obtaining < 100 pseudo-labeled data or after 10 runs.

The self-training algorithm is described in the below-shown algorithm, with $\hat{p}_{\mathcal{M}}(x)$ as predictive probabilities returned by model $\mathcal{M}$ and $\hat{y}_{\mathcal{M}}^{x}$ as predictive label of x by model $\mathcal{M}$. The resulting model of this approach is the one trained during the last iteration of the algorithm.

Require D_l, D_u

$N\_iter \leftarrow 0$

1 **Repeat until** $N\_iter < 10$ and $n > 100$

2 $n \leftarrow 0$

3 $\mathcal{M} \leftarrow train\_model(D_l)$

4 **For** $x \in D_u$ **do**

5 **If** $max(\hat{p}_{\mathcal{M}}(x)) > \tau$ **then**

6 $D_l \leftarrow D_l \cup \{(x, \hat{y}_{\mathcal{M}}^{x})\}$

7 $D_u \leftarrow D_u \backslash \{x\}$

8 $n \leftarrow n + 1$

9 * $N\_iter \leftarrow N\_iter + 1$

2.4.2.2 Task-Adaptive Pretraining (TAPT)

In this semi-supervised learning approach, the first step was performed by continuous pretraining of the language model on D_u with the same process used to construct BioFlauBERT. This allowed obtaining a language model adapted to the corpus which will be secondarily used for the downstream task. This pretraining was conducted with a mask language modeling task where the model had to predict a randomly masked token (word or subword) in text.

The same masking strategy than the one used to develop BioFlauBERT was used for each text: 12% of the tokens were randomly masked, 1.5% were randomly swapped with other tokens of the vocabulary, and the others remained unchanged. A linear layer of 68,729 neurons attached at the top of BioFlauBERT was used for prediction and a cross-entropy loss was used for backpropagation. For a sentence n of length T_n , the latter was defined by:

$$L = \frac{-1}{T_n} \sum_{t=1}^{T_n} \sum_{c=1}^{C} \mathbb{1}_{t,c} \cdot log\left(\frac{e^{z_{t,c}}}{\sum_{c=1}^{C} e^{z_{t,c}}}\right)$$

In this formula, C is the number of classes (here, 68,729), $\mathbb{1}_{t,c}$ a binary term to indicate whether the t-th token belongs to the c-th class, and $z_{t,c}$ the c-th BioFlauBERT output for the t-th token.

The second step consisted in fine-tuning the obtained language model on D_l by adding a linear layer and then applying the supervised training as in section 2.4.2. The process of TAPT based approach is illustrated in Figure 1. The model resulting from this approach is the one obtained after the fine-tuning step.

Figure 1 - Representation of the Task-Adaptive Pretraining (TAPT)

2.5 The training parameters

All semi-supervised learning approaches were based on supervised learning steps. Indeed, in this work, various hyperparameters were selected among classical values to optimize the training. The learning rate was fixed to 2^{e-5} and allowed to decrease linearly to 0 at the end of the last epoch. The number of epochs was fixed to 5, the batch size to 16 for the fine-tuning stages and 150 for the TAPT step, and the threshold for self-training fixed to 0.7. With all models, the cross-entropy [16] was used as loss function and minimized using the AdamW backpropagation algorithm [17]. Furthermore, a gradient clipping of 1.0 was used to avoid the exploding gradient effect [18]

2.6 Evaluation

For each test set of the cross-validation, the F1-score of each resulting model was computed for each class and the average F1-score weighted by the prevalence of the class. The mean of the ten F1-scores was then computed to obtain a single value per approach.

3 Results

Whatever the level of ICD-10 coding considered and with all approaches (supervised learning, self-training, and TAPT based), increasing the proportion of labeled data increased the performance of each trained model (Table 1). Furthermore, at each level, the best model (that with the highest F1-score) was the one trained by supervised learning with 100% labeled data, the F1-scores at Levels 1 to 4 were 92.2, 90.0, 87.6, and 84.7, respectively.

At all levels of ICD-10 coding and only with low proportions of labeled data, semi-supervised learning led to better performance than supervised learning. At Level 1 and up to 10% labeled data, self-training brought better performance improvement than supervised learning; the gain in F1-score ranged between 1.0 and 4.3 when the proportion of labelled data ranged between 10 and 0.5% (Table 1). Up to 5% of labeled data, TAPT based approach performed better than supervised learning; the gains ranged between 1.1 and 5.6 for 0.5% to 5% labeled data. At Levels 2 and 3, self-training led to performance gain vs. supervised learning only up to 35% labeled data (minimum gain of 1.0 with Level 2 codes and 35% labelled data and maximum 31.8 with Level 3 codes and 1% labeled data), whereas TAPT based approach led to performance gain only up to 10% labeled data (minimum gain of 1.0 with Level 2 codes and 10% labelled data and maximum 11.7 with Level 2 codes and 0.5% labeled data) (Table 1). Finally, at Level 4 and up to 50% labeled data, self-training led to performance gain vs. supervised learning (minimum gain of 1.2 with 50% labelled data and maximum 18.2 with 0.5% labeled data), whereas TAPT based approach led to performance gain only up to 10% labeled data (1.0 and 11.7 with 10% and 0.5% labeled data, respectively) (Table 1).

At Level 1, self-training and TAPT based approach were equally performant, except that TAPT based approach performed better with 0.5 and 1% labeled data. At Levels 2, 3,

and 4, and whatever the proportion of labeled data, the gain in performance was higher with self-training than with TAPT.

4 Discussion

With different ICD-10 levels and proportions of labeled data, supervised learning, self-training, and TAPT based approaches led to different of performance degrees. At a given level of ICD-10 coding and with a given approach, performance increased together with the increase in the proportion of labeled data. This was expected because an increase in labeled data means an increase in the size of the training set at the fine-tuning stage. Similar results were obtained by increasing the size of the training dataset when resulting models were fitted with supervised learning on simulated numerical and categorical data [3].

With the semi-supervised learning approaches (self-training and TAPT), the benefit depends of the proportion of labeled data. At any level of ICD-10, the performance gains obtained with semi-supervised learning approaches increased with the decrease in the proportion (thus, the number) of labeled data. With the self-training approach, this may be partially explained by a 'pseudo-increase' in the training set at the fine-tuning stage (the sum of labeled and unlabeled data being fixed). However, it was not possible to know whether the gain was due to an improvement in the language model, the linear layer, or both because all weights were simultaneously updated during fine-tuning. With the TAPT based approach, using unlabeled data allowed fine-tuning a language model that is more adapted to the corpus. However, in these cases with low proportions of labeled data, numerous sentences were used to adapt BioFlauBERT to a corpus of same size as the one used to create BioFlauBERT from FlauBERT; this raised the issue of future use of a BioFlauBERT version highly specialized in death certificate analysis.

One asset of this study was the change in the number of classes for the classification task. To the best of our knowledge, no previous study has focused on the influence of the number of classes with semi-supervised approaches on text classification tasks. Varying the number of classes on the same task allowed studying the influence that number has on the gain in performance with semi-supervised approaches.

As the ICD-10 structure is hierarchical, switching from one level to a lower one implies grouping neighboring entities into less numerous but larger entities concealing thus classification errors made between sub-entities. This is one reason for which, with a given proportion of labeled data, the performance of each approach decreased as the level of ICD-10 increased. Moreover, reducing the number of classes reduced the complexity of the classification task; this would explain why better performance degrees were reached by models with the lowest ICD-10 levels (with the same proportions of labeled data).

At Level 1 of ICD-10 coding, self-training and TAPT based approaches showed similar degrees of performance whatever the proportion of labeled data (despite slightly higher F1-scores for TAPT with labelled proportions up to 2%). However, at successively higher ICD-10 levels, the self-training approach performed progressively better than TAPT based approach. This does not agree with results of a binary classification task carried out by Li et al. [9] where TAPT based approach performed better than self-training. This disagreement may be explained by the use here of several rather than only two classes. In similar tasks, it seems that the false increase in the size of the training set (as in the self-training approach) adds more information to the final model than additional pretraining of the language model (as in the TAPT based approach).

One limitation of this study is that only proportions were considered for labeled and unlabeled data. Therefore, increasing the number of labeled data decreased automatically

the number of unlabeled data. In future works, a fixed number of labeled data and different sizes of unlabeled datasets may be used in assessing the respective influences of the numbers of labeled and unlabeled data with semi-supervised learning approaches.

5 Conclusions

In a classification task on data from death certificates, the use of either self-training or TAPT based approach led to performance gain in case of low proportions of labeled data (< 10% for Level 1 and <35% for Levels 2 to 4). However, this gain decreased with the increase of the proportion of labeled data. These results seems to indicate that in a case of text classification, with few labeled data and much many unlabeled ones, using semi-supervised learning allowed constructing more performant models. Furthermore, on classification tasks with high number of classes, self-training led to better models than TAPT.

Declaration of competing interest

The authors have no competing interests to declare.

Acknowledgments

The authors are grateful to Jean Iwaz (Hospices Civils de Lyon) for augmenting, editing, proofreading, and formatting the latest versions of the manuscript.

Funding

This work was supported by Association Nationale de la Recherche et de la Technologie (ANRT) [grant number 2019/1373]. The sponsor had no role in the study design; the

collection, analysis, and interpretation of the data; the writing of the report; and the decision to submit the article for publication.

References

[1] van Engelen JE, Hoos HH. A survey on semi-supervised learning. Machine Learning 2019;109:373–440. https://doi.org/10.1007/s10994-019-05855-6.

[2] Chen T, Kornblith S, Swersky K, Norouzi M, Hinton G. Big self-supervised models are strong semi-supervised learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Vancouver, Canada; 2020. https://proceedings.neurips.cc/paper/2020/file/fcbc95ccdd551da181207c0c1400c655-Paper.pdf

[3] Bailly A, Blanc C, Francis E, Guillotin T, Jamal F, Wakim B, et al. Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Comput Methods Programs Biomed 2022;*213:106504. https://doi.org/10.1016/j.cmpb.2021.106504

[4] Zhu X, Goldberg AB. *Introduction to Semi-Supervised Learning*. Springer International Publishing; 2009. https://doi.org/10.1007/978-3-031-01548-9

[5] Pavlinek M, Podgorelec V. Text classification method based on self-training and LDA topic models. Expert Syst Appl 2017;80:83–93. https://doi.org/10.1016/j.eswa.2017.03.020.

[6] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA; 2017. https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[7] Kalyan KS, Rajasekharan A, Sangeetha S. AMMU: A survey of transformer-based biomedical pretrained language models. J Biomed Inform 2022;126:103982. https://doi.org/10.1016/j.jbi.2021.103982.

[8] Gururangan S, Marasović A, Swayamdipta S, Lo K, Beltagy I, Downey D, et al. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics; 2020. https://doi.org/10.18653/v1/2020.acl-main.740.

[9] Li S, Yavuz S, Chen W, Yan X. Task-adaptive pre-training and self-training are complementary for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic; 2021. Association for Computational Linguistics, https://doi.org/10.18653/v1/2021.findings-emnlp.86.

[10] Şulea OM, Zampieri M, Vela M, van Genabith J. Predicting the law area and decisions of french supreme court cases. In *RANLP 2017 - Recent Advances in Natural Language Processing Meet Deep Learning*. Incoma Ltd. Shoumen, Bulgaria; 2017. https://doi.org/10.26615/978-954-452-049-6_092.

[11] Névéol A, Robert A, Grippo F, Morgand C, Orsi C, Pelikan L, et al. CLEF eHealth 2018 Multilingual Information Extraction Task Overview: ICD10 Coding of Death Certificates in French, Hungarian and Italian. *Conference and Labs of the Evaluation Forum, eHealth,* CEUR: Avignon, France; 2018.

[12] World Health Organization. International statistical classification of diseases and related health problems. 10th revision, fifth edition, 2016 edition.

[13] Blanc C, Bailly A, Francis E, Guillotin T, Jamal F, Roy P. BioFlauBERT: a French medical language model pre-trained after corpus size selection. 2022 (submitted on September 15, 2022 and under review).

[14] Le H, Vial L, Frej J, Segonne V, Coavoux M, Lecouteux B, et al. FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association: Marseille, France; 2020. https://aclanthology.org/2020.lrec-1.302.pdf

[15] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics; 2016. https://doi.org/10.18653/v1/p16-1162.

[16] Zhang Z, Sabuncu MR. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, Red Hook, NY, USA; 2018. https://proceedings.neurips.cc/paper/2018/file/f2925f97bc13ad2852a7a551802feea0-Paper.pdf

[17] Loshchilov I, Hutter F. Decoupled weight decay regularization. International Conference on Learning Representations (2017). https://doi.org/10.48550/ARXIV.1711.05101.

[18] Pascanu R, Mikolov T, Bengio Y. Understanding the exploding gradient problem. *CoRR* 2012 abs/1211.5063

Table 1 -Mean (SD) of the F1-scores with the 10-fold cross-validations

	Level 1			Level 2			Level 3			Level 4		
Labelled data	Supervised learning	Self-training	TAPT	Supervised learning	Self-training	TAPT	Supervised learning	Self-training	TAPT	Supervised learning	Self-training	TAPT
0.5%	60.5 (2.3)	64.8 (3.2)	66.1 (2.4)	38.0 (4.9)	56.2 (3.2)	49.7 (3.7)	3.3 (1.2)	5.8 (7.1)	5.2 (2.9)	2.0 (0.9)	4.6 (6.4)	2.8 (1.4)
1%	71.0 (1.6)	75.0 (1.6)	76.6 (1.5)	55.9 (1.2)	64.8 (1.7)	61.2 (2.0)	16.7 (4.2	48.5 (2.1)	21.9 (4.6)	10.7 (3.9)	39.0 (3.4)	17.8 (4.1)
2%	79.4 (0.5)	80.8 (0.6)	82.0 (0.6)	67.5 (2.2)	72.7 (1.2)	70.7 (0.9)	38.4 (4.2)	58.0 (1.4)	46.6 (3.2)	33.5 (2.9)	50.1 (1.2)	36.8 (2.5)
5%	84.5 (0.4)	85.8 (0.4)	85.6 (0.4)	78.1 (0.7)	80.6 (0.5)	79.4 (0.6)	64.1 (1.5)	71.0 (0.6)	67.0 (1.5)	54.4 (1.1)	63.0 (0.8)	57.6 (1.3)
10%	87.3 (0.3)	88.3 (0.3)	88.0 (0.2)	82.9 (0.3)	84.9 (0.3)	83.9 (0.4)	74.2 (0.7)	78.2 (0.6)	76.1 (0.8)	67.3 (0.8)	72.6 (0.7)	68.4 (0.8)
20%	89.4 (0.4)	89.9 (0.3)	89.8 (0.3)	86.1 (0.3)	87.2 (0.3)	86.5 (0.4)	80.8 (0.7)	83.4 (0.5)	81.6 (0.6)	75.8 (0.8)	79.0 (0.4)	77.0 (0.5)
35%	90.4 (0.3)	91.0 (0.2)	90.7 (0.3)	87.7 (0.3)	88.7 (0.2)	88.0 (0.3)	84.2 (0.3)	85.6 (0.3)	84.5 (0.4)	80.5 (0.5)	82.3 (0.3)	80.9 (0.3)
50%	91.1 (0.2)	91.4 (0.2)	91.3 (0.2)	88.7 (0.3)	89.2 (0.2)	88.9 (0.3)	85.5 (0.3)	85.3 (3.6)	85.7 (0.3)	82.4 (0.3)	83.6 (0.3)	82.5 (0.4)
75%	91.7 (0.2)	91.9 (0.2)	91.9 (0.2)	89.5 (0.2)	89.8 (0.3)	89.6 (0.3)	86.8 (0.4)	87.2 (0.3)	86.9 (0.2)	84.0 (0.3)	84.4 (0.3)	84.0 (0.2)
100%	92.2 (0.2)	---	---	90.0 (0.3)	---	---	87.6 (0.4)	---	---	84.7 (0.3)	---	---

TATP: Task-Adaptive Pretraining

Annexe C

Classification multi-label de cas cliniques avec CamemBERT

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Classification multi-label de cas cliniques avec CamemBERT

Alexandre Bailly[1, *] Corentin Blanc[1, *] Thierry Guillotin [1]
(1) Everteam Software, 17 quai Joseph Gillet, 69004 Lyon, France
(*) Contributions égales
a.bailly@everteam.com, c.blanc@everteam.com, t.guillotin@everteam.com

RÉSUMÉ

La quantité de documents textuels médicaux allant grandissant, la nécessité d'en extraire automatiquement des informations concernant des patients devient de plus en plus grande. La prédiction du profil clinique permet de gagner du temps pour le praticien tout en extrayant l'essentiel de l'information concernant un patient. Avec l'explosion du nombre de documents (médicaux ou non), des modèles pré-entraînés tels que BERT pour l'anglais ou CamemBERT pour le français ont émergé. L'utilisation de ces modèles permet d'encoder contextuellement du texte afin de l'utiliser dans des réseaux neuronaux pour notamment prédire des profils cliniques. Cet article vise à comparer différentes méthodes de prédiction de profil clinique en se basant sur l'utilisation de CamemBERT. Dans un premier temps, uniquement du texte provenant de documents médicaux a été utilisé. Dans un second temps, des entités nommées ont été injectées en plus du texte par concaténation ou par sommation pondérée. Les résultats ont montré un succès limité et dépendant de la prévalence des chapitres à prédire dans le corpus ainsi qu'une dégradation des performances lors de l'ajout des entités nommées.

ABSTRACT

Multi-label classification of clinical cases with CamemBERT

As quantity of textual medical data is increasing, the necessity to extract automatically information about patients increases accordingly. Predicting the clinical profile of a patient record allows to save time for praticians by exhibiting essential information concerning the patient. Together with the explosion in the number of documents (medical or not), pretrained models such as BERT for english or CamemBERT for french has emerged. Using these models allows to encode contextually a text to this encoded representation in neural networks notably for NLP tasks such as predicting clinical profiles. This article aims to compare different methods of clinical profile prediction based on CamemBERT. In a first time, only the text from medical documents was used. In a second time, named entities were injected in addition to the text by concatenation or pondered sum. Results show a limited success depending on the prevalence of the chapters to predict in the corpus as well as a decrease of performances with the use of named entitie types.

MOTS-CLÉS : Classification multi-label ; Fouille de texte ; CamemBERT.

KEYWORDS: Multi-label classification ; Data mining ; CamemBERT.

1 Introduction

Avec l'augmentation du nombre de consultations médicales, la quantité de documents textuels concernant les patients a considérablement augmenté. Les divers compte-rendus de consultation ou de

Atelier DÉfi Fouille de Textes (DEFT), pages 14–20.

prise en charge hospitalière forment une masse de données importante et riche en informations sur le patient. Ces informations sont très intéressantes pour les différents praticiens et leur récupération est un enjeu important. L'extraction du profil clinique d'un patient (l'ensemble des pathologies associées à son cas) à partir d'un document textuel peut prendre un temps important, au détriment du temps accordé au patient. Cette étape reste néanmoins indispensable et une automatisation de l'extraction est une bonne alternative pour obtenir les informations nécessaires.

Cette explosion du nombre de documents médicaux a poussé la communauté scientifique à créer de nouveaux modèles de langue facilitant leur traitement. Ces modèles pré-entraînés sur des quantités colossales de données permettent d'encoder une phrase ainsi que les mots la constituant en tenant compte de leur contexte. Le plus connu est BERT (Bidirectionnal Encoder Representation from Transformers) qui a permis d'améliorer l'état de l'art sur une grande majorité des tâches de Traitement Automatique du Langage Naturel (TALN) en anglais (Devlin *et al.*, 2019). Suite à ce succès, de nombreux autres modèles dérivés de BERT ont vu le jour comme CamemBERT (Martin *et al.*, 2020) pour le français.

Ce papier vise à étudier la prédiction du profil clinique d'un patient en utilisant à la fois du texte brut provenant de documents médicaux mais aussi différentes entités nommées qui ont été préalablement mises en évidence. Trois approches seront étudiées : le traitement du texte brut par CamemBERT dans un premier temps puis l'injection par concaténation et sommation pondérée des entités nommées au texte brut dans un second temps.

2 Matériel et méthodes

2.1 Données

Ces travaux se situent dans le contexte de la compétition DEFT-21 (Grouin *et al.*, 2021). Le corpus de DEFT 2021 était constitué de 275 cas cliniques répartis en un jeu de d'entraînement (167) et un jeu de test (108). Chaque cas clinique était composé d'un texte brut accompagné d'un certain nombre d'entités nommées préalablement identifiées parmi 19 types distincts comme le montre l'exemple sur la Figure 1.

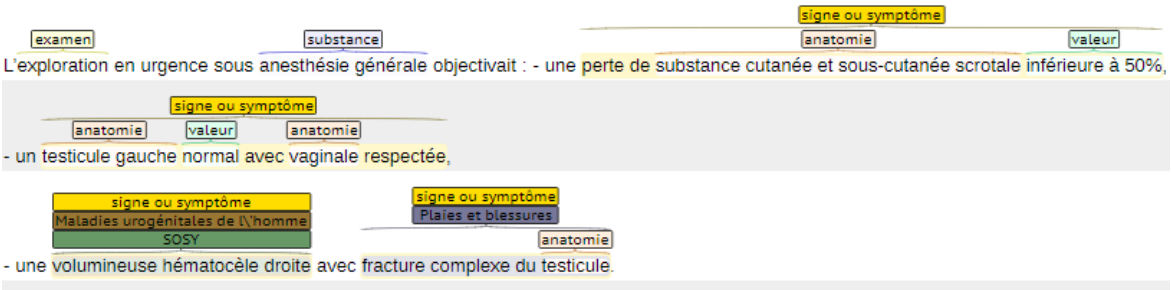


FIGURE 1 – Exemple de texte brut accompagné d'entités nommées d'un cas clinique

Un profil clinique constitué d'au moins un des 23 chapitres du MeSH a été attribué à tous les cas cliniques. Au sein du corpus, la distribution des chapitres était extrêmement déséquilibrée comme suit sur la Table 1.

Chapitre	Entraînement	Test	Total
Stomatognathique	3	3	6
Parasitaires	6	1	7
Virales	8	4	12
ORL	10	3	13
Oeil	11	9	20
Génétique	19	10	29
Immunitaire	20	11	31
Endocriniennes	22	14	36
Blessures	21	19	40
Osteomusculaires	21	22	43
Respiratoire	27	17	44
Peau	31	16	47
Nutritionnelles	29	23	52
Digestif	36	22	58
Hémopathies	34	25	59
Infections	33	27	60
Femme	33	32	65
Cardiovasculaires	39	27	66
Chimiques	45	22	67
Nerveux	41	46	87
Homme	63	36	99
Tumeur	80	51	131
Etatsosy	141	101	242

TABLE 1 – Distribution des chapitres du MeSH dans les données

Dans la suite, certains chapitres trop peu représentés ont été volontairement écartées des possibilités de prédiction afin d'améliorer celles des autres chapitres. Les chapitres écartés sont les suivants : *stomatognathiques* (3), *parasitaires* (6), *virales* (8), *ORL* (10) et *oeil* (11).

2.2 Approches proposées

CamemBERT est un modèle de langue pré-entraîné sur une énorme quantité de données permettant d'encoder le contexte d'une phrase. La meilleure manière d'appliquer un tel modèle à une tâche TALN est de connecter une couche de sortie pour réaliser la prédiction puis de régler finement tous les poids de bout en bout.

Dans la suite, chacune des méthodes introduites utilise le modèle de langue CamemBERT. Pour un cas clinique donné, toutes les phrases du texte brut ont été encodées par CamemBERT puis moyennées afin d'obtenir une unique représentation du cas clinique. Cette représentation a ensuite été utilisée afin d'effectuer la prédiction.

2.2.1 Texte brut

Dans un premier temps, une couche linéaire de 18 neurones avec une sigmoïde comme fonction d'activation a été connectée à CamemBERT afin de calculer une probabilité pour chacun des 18 chapitres précédemment retenus.

2.2.2 Injection des entités nommées au texte brut

Dans un second temps, les entités nommées extraites de chaque document ont été utilisées pour enrichir les entrées du modèle. Pour chaque cas clinique, les types d'entités nommées ont été encodées grâce à un vecteur binaire représentant la présence (1) ou l'absence (0) au sein du texte associé au cas clinique. Afin de les injecter au texte brut, deux méthodes ont été utilisées :

— Concaténation : l'encodage du texte brut était concaténé à celui des entités nommées.

— Somme pondérée : l'encodage du texte brut était sommé à une projection linéaire de l'encodage des entités nommées. La projection tout comme la pondération étaient apprises lors de la phase d'entraînement.

Ces deux méthodes ont permis de construire de nouvelles représentations du cas clinique qui ont ensuite été utilisées dans un couche linéaire de 18 neurones, couplée à une fonction sigmoïde, afin de calculer une probabilité pour chacun des chapitres retenus.

2.3 Paramètres d'entraînement

Pour toutes les approches, le nombre d'epochs a été fixé à 5 au vue de la convergence de la Binary Cross-Entropy Loss. L'optimisateur AdamW (Loshchilov & Hutter, 2019) a été utilisé avec un taux d'apprentissage fixé à 5e-3 sur la première epoch puis diminuant linéairement. Pour chacune des trois méthodes proposées, un seuil par chapitre a été recherché afin d'optimiser les résultats.

2.4 Évaluation

La recherche des paramètres d'entraînement a été effectuée par une méthode de bootstrap (Efron, 1979) à partir du corpus d'entraînement. Une fois les paramètres d'entraînement sélectionnés, les modèles ont été entraînés sur la globalité du corpus d'entraînement. Trois métriques ont été utilisées pour évaluer les performances sur chaque chapitre et de manière globale : le rappel, la précision et le f1-score. Les résultats présentés ci-après sont ceux obtenus sur le jeux de test.

3 Résultats

Évaluation par chapitre Pour chacun des différents modèles qui ont été entraînés, les performances obtenues pour les prédictions dépendent des chapitres et ne différent pas grandement d'une méthode à l'autre. En effet, comme il est visible dans la table 2, le f1-score pour les différents chapitres se situe entre 0.105 et 0.967. Les chapitres les moins représentés dans le corpus sont ceux qui présentent

	Rappel			Précision			F1		
	M1†	M2*	M3$	M1†	M2*	M3$	M1†	M2*	M3$
Stomatognathique	—	—	—	—	—	—	—	—	—
Parasitaires	—	—	—	—	—	—	—	—	—
Virales	—	—	—	—	—	—	—	—	—
ORL	—	—	—	—	—	—	—	—	—
Oeil	—	—	—	—	—	—	—	—	—
Génétique	**1.000**	0.700	0.400	**0.118**	0.091	0.103	**0.211**	0.161	0.163
Immunitaire	0.091	**0.364**	0.091	**0.250**	0.154	0.067	0.133	**0.216**	0.077
Endocriniennes	**1.000**	0.857	0.857	**0.140**	0.126	0.121	**0.246**	0.220	0.212
Blessures	0.158	**0.684**	0.158	0.188	0.171	**0.214**	0.171	**0.274**	0.182
Osteomusculaires	0.000	0.318	**0.409**	0.000	**0.250**	0.191	0.000	**0.280**	0.261
Respiratoire	0.235	**0.471**	0.176	**0.286**	0.167	0.075	**0.258**	0.246	0.105
Peau	0.312	0.688	**0.750**	**0.217**	0.125	0.124	**0.256**	0.212	0.212
Nutritionnelles	**1.000**	0.652	0.696	0.213	0.217	**0.229**	**0.351**	0.326	0.344
Digestif	**0.773**	0.591	0.545	**0.250**	0.197	0.200	**0.378**	0.295	0.293
Hémopathies	**0.520**	0.320	0.120	**0.371**	0.205	0.150	**0.433**	0.250	0.133
Infections	0.778	0.741	**0.926**	**0.292**	0.267	0.255	**0.424**	0.392	0.400
Femme	**0.844**	0.719	0.719	**0.386**	0.303	0.307	**0.529**	0.426	0.430
Cardiovasculaires	**1.000**	**1.000**	**1.000**	0.250	0.250	**0.260**	0.400	0.400	**0.412**
Chimiques	0.182	**0.273**	0.091	**0.571**	0.182	0.065	**0.276**	0.218	0.075
Nerveux	**0.457**	0.304	0.348	**0.636**	0.359	0.457	**0.532**	0.329	0.395
Homme	0.472	**0.722**	0.694	**0.500**	0.342	0.321	**0.486**	0.464	0.439
Tumeur	**0.941**	0.608	0.745	**0.623**	0.397	0.458	**0.750**	0.481	0.567
Etatsosy	**1.000**	0.931	**1.000**	**0.935**	0.931	**0.935**	0.935	0.931	**0.967**
Global	**0.683**	0.651	0.637	**0.370**	0.283	0.298	**0.480**	0.394	0.406

†Modèle textuel uniquement - * Modèle avec concaténation - $ Modèle avec pondération

TABLE 2 – Résultat obtenu pour chaque chapitre du MeSH

les f1-scores les plus faibles, et ce quel que soit le modèle. Le f1-score pour le chapitre *génétique*, qui est le moins représenté, est de 0.161 pour le modèle avec concaténation, de 0.163 pour celui avec pondération et de 0.211 pour le modèle n'utilisant que le texte brut. Au contraire, les chapitres les plus représentés dans le corpus présentent de bon résultats, notamment pour *etatsosy*, avec des f1-score de 0.935, 0.931 et 0.967 respectivement pour le modèle utilisant seulement le texte, le modèle avec la concaténation et le modèle avec la pondération. Le chapitre *chimiques* fait ici figure d'exception, avec des f1-scores de 0.276, 0.218 et seulement 0.075 respectivement, alors qu'il fait partie des chapitres avec les plus grandes prévalences. Les faibles performances en terme de f1-score pour les chapitres sous-représentés sont dues à une faible précision. En effet, pour le chapitre *endocriniennes* par exemple, le modèle avec pondération a obtenu un rappel de 0.857 mais une précision de seulement 0.121, ce qui explique alors le f1-score de 0.212.

Comparaison des modèles L'évaluation globale des modèles montre que quelque soit la métrique observée, le modèle n'utilisant que le texte obtient de meilleurs performances. En terme de f1-score, le modèle utilisant seulement le texte atteint 0.480 alors que les modèles utilisant la concaténation et

la pondération n'atteignent respectivement que 0.394 et 0.406. La comparaison de ces deux dernières valeurs semble indiquer que la pondération conduit à de meilleurs performances que la concaténation.

4 Discussion

La prédiction des différents chapitres est plus ou moins bonne selon leur prévalence dans le corpus d'entraînement. En effet, les chapitres les moins représentés ont tendance à être moins bien prédits. La faible prévalence de certains chapitres semble donc être un frein considérable pour tous les modèles comparés lors de l'apprentissage.

La moyenne des représentations des phrases d'un texte obtenues grâce à CamemBERT permet dans une certaine mesure d'attribuer les chapitres associés à ce même texte. Néanmoins, les performances de ce modèle restent limitées. Cela peut être notamment dû à la quantité limités de cas cliniques disponibles pour l'entraînement. En effet, l'utilisation de CamemBERT implique un grand nombre de poids à entraîner et donc nécessite beaucoup de données pour l'entraînement. Une autre explication pourrait être le fait que l'ensemble du texte a été considéré pour la prédiction, alors que l'information recherchée peut n'être présente que dans une partie seulement. Certaines phrases pourrait donc être à l'origine de bruit dans les données.

L'utilisation des entités nommées pour enrichir le texte était supposée apporter davantage d'informations et permettre d'améliorer la prédiction. Cependant, les modèles les incluant ont vu leurs performances se dégrader et ce peu importe la façon dont elles ont été injectées. L'information de la seule présence des entités dans le texte ne semble donc ne pas être suffisante pour améliorer les prédictions.

5 Conclusion

Dans une certaine mesure, l'utilisation du modèle pré-entraîné CamemBERT a permis de retrouver les chapitres du MeSH associés à différents cas cliniques à partir du texte. En revanche, la quantité de données présente n'a pas permis d'atteindre de bonne performances avec cette méthode. L'ajout de la présence de certaines entités nommées a eu pour seul effet de dégrader légèrement les performances initiales.

Références

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). Bert : Pre-training of deep bidirectional transformers for language understanding.

EFRON B. (1979). Bootstrap Methods : Another Look at the Jackknife. *Annals of Statistics*, **7**, 1–26.

GROUIN C., GRABAR N. & ILLOUZ G., Éds. (2021). *Actes de TALN 2021 (Traitement automatique des langues naturelles)*, Lille.

LOSHCHILOV I. & HUTTER F. (2019). Decoupled Weight Decay Regularization. *arXiv :1711.05101 [cs, math]*.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203–7219, Online : Association for Computational Linguistics.

Annexe D

Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models

Computer Methods and Programs in Biomedicine 213 (2022) 106504

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models

Alexandre Bailly[a,b,*], Corentin Blanc[a,b], Élie Francis[a], Thierry Guillotin[a], Fadi Jamal[c], Béchara Wakim[d], Pascal Roy[b]

[a] *Everteam Software, Research and Development Lab, 17 quai Joseph Gillet, Lyon, France*
[b] *Université de Lyon, Lyon, France; Université Lyon 1, Villeurbanne, France; Service de Biostatistique-Bioinformatique, Hospices Civils de Lyon, Lyon, France; Équipe Biostatistique-Santé, Laboratoire de Biométrie et Biologie Évolutive, CNRS UMR 5558 Villeurbanne, France*
[c] *izyCardio - CardioParc, Lyon, France*
[d] *Mediapps Innovation SA, Lyon, France*

ARTICLE INFO

Article history:
Received 1 April 2021
Accepted 24 October 2021

Keywords:
Deep learning
Machine learning
Data set size
Interactions

ABSTRACT

Background and objective: Machine learning and deep learning models are very powerful in predicting the presence of a disease. To achieve good predictions, those models require a certain amount of data to train on, whereas this amount i) is generally limited and difficult to obtain; and, ii) increases with the complexity of the interactions between the outcome (disease presence) and the model variables. This study compares the ways training dataset size and interactions affect the performance of those prediction models.

Methods: To compare the two influences, several datasets were simulated that differed in the number of observations and the complexity of the interactions between the variables and the outcome. A few logistic regressions and neural networks were trained on the simulated datasets and their performance evaluated by cross-validation and compared using accuracy, F1 score, and AUC metrics.

Results: Models trained on simulated datasets without interactions provided good results: AUC close to 0.80 with either logistic regression or neural networks. Models trained on simulated dataset with order 2 interactions led also to AUCs close to 0.80 with either logistic regression or neural networks. Models trained on simulated datasets with order 4 interactions led to AUC close to 0.80 with neural networks and 0.85 with penalized logistic regressions. Whatever the interaction order, increasing the dataset size did not significantly affect model performance, especially that of machine learning models.

Conclusion: Machine learning models were the less influenced by the dataset size but needed interaction terms to achieve good performance, whereas deep learning models could achieve good performance without interaction terms. Conclusively, with the considered scenarios, well-specified machine learning models outperformed deep learning models.

1. Introduction

In the last decades, Artificial Intelligence (AI) was largely used in a wide range of fields such as Pattern Recognition [1], Image Recognition [2], or Natural Language Processing [3]. In medicine, the main challenges have been assessing a disease risk, establishing a diagnosis, making a prognosis, or predicting the response to a treatment. All these challenges have been separately or jointly investigated in numerous medical specialties. Nevertheless, the superiority of AI methods over others was not always conclusive because of peculiarities of some methods, some diseases, some dataset contents or structure and, especially in prediction studies, because of the variety of metrics used to evaluate and compare model predictive performance; e.g., accuracy, sensitivity, specificity, area under the receiver operating curve (AUC), F1 score, etc.

Among the several hundreds of recent studies about the use of Deep Learning or Neural Networks for the analysis of medical data or, more specifically, prediction in medicine, a few selected works are briefly summarized herein.

* Corresponding author.
E-mail address: a.bailly@everteam.com (A. Bailly).

https://doi.org/10.1016/j.cmpb.2021.106504

In 2019, Ayon et al. [4] evaluated the performance of a deep neural network in predicting diabetes mellitus. With five-fold and ten-fold cross-validations on the Pima Indians Diabetes Dataset, the validations showed respectively 98.04 and 97.27% accuracy, 98.80 and 97.8% sensitivity, 96.64 and 96.27% specificity, 0.99 and 0.98 F1 score, and, finally, 0.96 and 0.94 Matthew's Correlation Coefficient. The deep neural network outperformed other current methods (e.g., with logistic regression, accuracy was only 78%).

In 2019 too, Tomita et al. [5] compared the performance of logistic regression, support vector machine, and deep neural network in making an initial diagnosis of adult asthma. The study included 566 adults with non-specific symptoms who visited a university hospital for the first time. Ten to 22 inputs were used with each model and performance assessed in terms of accuracy and AUC. With 10 symptom-physical signs as inputs, the accuracies were 65, 62, and 68%, respectively, but, with 22 inputs (10 + biochemical, functional, and other tests) the accuracies were 94, 82, and 98%, respectively, whereas the AUCs were 72.6, 54.5, and 63.2, respectively, with 10 inputs but 0.97, 0.83, and 0.99, respectively, with 22 inputs. Thus, with all inputs available, the neural network outperformed clearly the other methods in diagnosing adult asthma.

In 2020, Nazari et al. [6] used neural networks (with one and three hidden layers) and deep learning for the diagnosis of myeloid leukemia using microarray gene data from the Gene Expression Omnibus (GEO) database. The accuracies with a single-layer and three hidden layers were 63.33 and 96.67%, respectively, showing the power of the latter method.

In 2021, Lewis et al. [7] compared deep learning models with logistic regressions in predicting preventable acute care use and spending among 93,260 heart failure patients from a single large US insurer database. With all outputs predicted (preventable hospitalization, preventable ED visits and preventable costs), deep learning models showed the highest performance (e.g., concerning preventable hospitalization, the AUC was 0.75 for logistic regression vs. 0.77 for non-sequential neural network and 0.78 for sequential neural networks).

Machine Learning (ML) is a branch of AI based on learning from data using variables, also called 'features', to predict an 'outcome' (disease risk, diagnosis, prognosis, or response to treatment). Most ML models are trained with supervised learning, which implies a known outcome. Several models based on data training do exist (e.g., Linear Regression [8], Logistic Regression [9], Support Vector Machines [10], or Naive Bayes classifier [11]) in which it has been shown that increasing the amount of data improves performance [12]. In the field of classification, performance is the ability of a model to predict an observation's class using a test dataset. A bad performance of a classifier may have two causes: i) the features do not contain enough information to explain the outcome; and, ii) the model cannot deal with the complexity of the relationships between the features and the outcome. Generalized Linear Models are able to unravel part of this complexity by introducing interaction terms of various orders, complexity being considered as a deviation from an additive effect between the model's linear components. Unfortunately, in case of high complexity, ML has to consider high-order interaction terms during model building.

Deep Learning (DL) is a branch of ML that includes models with more elaborated architectures. The simplest DL model is the perceptron [13], an interconnection of artificial neurons. The perceptron may be extended by adding hidden layers to form a deeper network named 'multilayer perceptron'. A onelayer perceptron is able to approximate any continuous function; this was proven by Hornik et al. [14,15] with the Universal Approximation Theorem. One implication of that Theorem is that, in a fully connected NN, each neuron after the first one in the first hidden layer takes into account all interactions between the features. With a one-layer perceptron, DL models are thus able to deal with relationship complexity by including high-order interaction terms, especially that they do not have to be specified. However, DL models require large training datasets [16].

Some metrics were developed to evaluate the predictive performance of mod- els; e.g., accuracy, AUC, or F1 score. These metrics allow model comparisons of very different types (e.g., a ML model vs. a DL model). Concerning ML approaches, studies such as those of Tsangaratos et al. [12] compared model performance by varying both the number of observations and the number of features used in the models with real drug datasets. However, these studies did not deal with DL models. Interestingly, Korotcov et al. [17] compared the performance of DL and ML models in terms of accuracy and AUC. With all datasets used, DL models gave better results than ML models but the authors did not systematically explore various levels of complexity. Consequently, those results were specific to the datasets used and still have be confirmed. Van der Ploeg et al. [18] created simulated datasets from different cohorts to study the amount of data needed by Logistic Regression or DL models to predict binary outcomes. They showed that Logistic Regression models needed less data than DL models and suggested DL models be used only with large datasets. Overall, the latter works took into account the amount of data but not the interaction orders. Theoretical studies evaluating the abilities of various models to deal with high interaction orders and their effects on performance are still needed.

The aim of the present study was to compare the prediction performance of ML models vs. that of DL models according to the training dataset size and the complexity of the interactions between the variables and the outcome. The variables and the presence of disease were simulated in virtual patients and complexity was generated by introducing multiplicative interaction terms in the models.

This report presents first the data, the way variables and subject statuses were simulated, the way the scenarios were built, the models used for prediction, and then the criteria used for comparisons of models' predicting abilities. The results under various scenarios are then displayed before being discussed and finally summarized in a clear conclusion.

2. Materials and methods

2.1. Simulated data

The Framingham Study [19] is one of the longest and most important epidemiological studies in medical history. It was a population-based observational cohort study initiated in 1948 to investigate prospectively the epidemiology and risk factors for cardiovascular disease in nearly 5210 participants, all residents of Framingham (MA, USA). Among various findings, that remarkable study demonstrated the detrimental roles of cigarette smoking, elevated blood pressure, and high cholesterol level in the development of heart disease and their contributions to the risk of heart attack and stroke. The original and subsequent databases include (but are not limited to) information on hundreds of demographic, clinical, laboratory, and imaging variables. (For more details see: Boston Medical Center. Framingham Study. https://www.bmc.org/stroke-and-cerebrovascular-center/research/framingham-study).

The present work adopted a short list of risk factors identified by the Framingham Study and simulated corresponding data with various levels of interaction to simulate outcomes. These risk factors were the following: age (in years), total cholesterol (in mg/dL), HDL cholesterol (in mg/dL), systolic blood pressure (in mm Hg), treatment for hypertension, smoking status, and diabetic status (as binary variables) (Table 1).

Table 1
Description of the features considered.

Features	Mean	Standard deviation	Proportion
Continuous features			
Age	50	5.92	–
Total cholesterol	215	4.47	–
HDL cholesterol	45	5.00	–
Systolic blood pressure	130	12.25	–
Binary features			
Treatment for hypertension	–	–	0.1013
Smoking status	–	–	0.3522
Diabetes status	–	–	0.0650

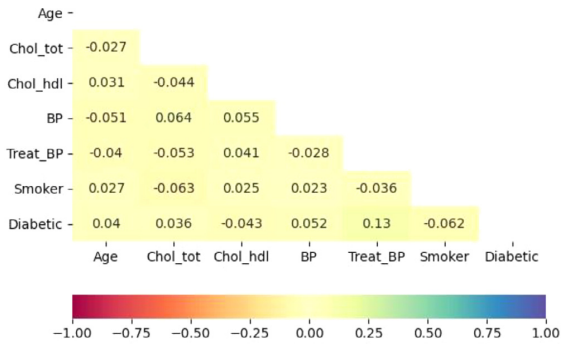


Fig. 1. Pairwise correlation between features.

2.1.1. Feature generation

Each continuous feature was randomly generated according to a Gaussian distribution having the same mean as in the Framingham study [19] and a standard deviation chosen according to common values seen in clinical practice (Table 1). Binary features were generated using binomial laws where each probability parameter corresponded to the distribution of the feature in the Framingham study (Table 1). All features were independently generated.

Fig. 1 shows pairwise correlations between features. Because several datasets were simulated, only the highest correlations (in absolute value) are shown. As the maximum correlation value was 0.13, we considered that there is almost no correlation between the features. This stems from the variable generation process where all the features were independently simulated.

2.1.2. Clinical status generation

Clinical statuses were dependent on the features and their possible interactions. Here, only multiplicative interactions between features were considered. Considering $P(X)$ as a polynomial created from a vector of features X, the probability of being a 'case' was computed with the following formula:

$$p = \frac{1}{1 + e^{-P(X)}}$$

Once a probability was computed for a given observation, a binomial law having this probability as parameter was used to assign a clinical status (case or non-case). In this process, the order of the interaction is the degree of P used to assign the clinical status.

2.1.3. Scenario generation

Different datasets were generated with sizes 1,000, 10,000, and 100,000 observations. Furthermore, three polynomials were used to assign cases to observations (see the Appendix): i) only the main effects considered (the features without interaction terms); ii) main effects and interaction terms of order 2 considered; and, iii) main effects and interaction terms up to order 4 considered. In this study, interactions within a single feature were not considered. Nine scenarios were thus generated (three sizes x three polynomials).

2.2. Models

2.2.1. Machine learning models

The first ML model was logistic regression (LR) [9]. LR is derived from the Linear Model where the outcome variable is binary. In a LR with k features, the log of the odd of the probability p is modeled from features $X = (x_1,\bullet\bullet\bullet,x_k)$ by a linear model:

$$log\left(\frac{p}{1-p}\right) = \sum_{j=1}^{k} \beta_j x_j$$

$$\Leftrightarrow \quad p = \frac{1}{1 + e^{-\sum_j \beta_j x_j}}$$

Parameters $\beta = (\beta_1,\bullet\bullet\bullet,\beta_k)$ were estimated by maximizing the log-likelihood function. With n observations, y_i and x_i being respectively the clinical status and the vector of features for the i-th observation, the log-likelihood function was given by:

$$l\beta \sum_{i=1}^{n} \left[y_i x_i \beta - log\left(1 + e^{x_i \beta}\right)\right]$$

Variable selection was made by comparing nested models using the likelihood ratio test. In a first step, the main effects were kept only when the associated p-values were < 0.05. In a second step, all order-2 interactions were successively tested by fitting models that included systematically all previously retained main effects. For interaction tests, p-values < 0.10 were considered. Furthermore, a model that included the previously retained main effects and all identified order-2 interactions was fitted; this led to consider only order-2 interactions with p-value still < 0.10. Starting from that model, order-3 then order-4 interaction terms were selected using similar approaches.

Two types of penalization were applied: Lasso and Ridge. The function to maximize was then:

$$l_\lambda^\omega(\beta) = \sum_{i=1}^{n} \left[y_i x_i \beta - \log\left(1 + e^{x_i \beta}\right)\right] - \lambda \sum_{j=1}^{k} \left|\beta_j\right|^\omega$$

In this formula, λ is the tuning parameter and $\omega = 1$ for Lasso or 2 for Ridge. In both situations, penalization led to shrinkage of parameters β. With Lasso penalization, a selection may be carried out by setting β to 0.

With each penalization method, a model that included all main effects was fitted first then a model that included all main effects and all order-2 interaction terms. Finally, a model that included all main effects and all order-2, order-3, and order-4 interaction terms was fitted. The tuning parameter was fixed to 1.

Hereafter, notations LR_i, $lasso_i$, and $ridge_i$ will indicate the different logistic regressions trained with the variables and their interactions up to order i.

2.2.2. Deep learning models

Neural networks (NNs) [1], just like a multilayer perceptron, are based on artificial neurons. These neurons are connected in many layers to create a network. The input layer includes as many neurons as there are features and the output layer as many neurons as there are output possibilities. Between these two layers, some other hidden layers may be included. Here, a sigmoid function was used as activation function for all layers, except the last one for which log-softmax function was used. In addition, between hidden layers, a dropout with probability 0.1 was considered to reduce overfitting. All NNs were trained by backpropagation [20] to

Table 2
Metrics relative to the main effects.

	Dataset size: 1000			Dataset size: 10000			Dataset size: 100000		
Model	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC
LR_1	0.70	0.48	0.70	0.71	0.49	0.71	0.71	0.49	0.71
LR_2	0.76	0.54	0.74	0.78	0.58	0.78	0.79	0.58	0.78
LR_4	0.76	0.55	0.75	0.79	0.59	0.78	0.80	0.60	0.79
$lasso_1$	0.89	0.69	0.80	0.89	0.70	0.80	0.89	0.69	0.79
$lasso_2$	0.89	0.69	0.79	0.89	0.70	0.80	0.89	0.69	0.79
$lasso_4$	0.88	0.68	0.78	0.89	0.70	0.80	0.89	0.69	0.79
$ridge_1$	0.89	0.70	0.80	0.89	0.70	0.80	0.89	0.69	0.79
$ridge_2$	0.88	0.68	0.79	0.89	0.69	0.80	0.89	0.69	0.79
$ridge_4$	0.88	0.67	0.78	0.89	0.69	0.79	0.89	0.69	0.79
NN_1	0.88	0.67*	0.79	0.89	0.68	0.78	0.89	0.67	0.78
NN_3	0.88	0.66†	0.77	0.88	0.66	0.77	0.89	0.68	0.78
NN_5	0.88	0.67†	0.79	0.88	0.65	0.77	0.88	0.67	0.78
NN_7	0.86	0.63‡	0.77*	0.88	0.65	0.77	0.88	0.66	0.77

†The SD for this value was 0.04 -
* the SD for this value was 0.05 -
‡ The SD for this value was 0.08 -All other SDs were $\leq$ 0.03

minimize the cross-entropy loss function [21] given, in a binary context, by:

$$L(y, p) = -(ylog(p) + (1 - y)log(1 - p))$$

In this equation, p is the probability returned by the model and y the clinical status (1 for 'case', 0 for 'non-case'). For the backpropagation, Adam optimizer was used to update the parameters of the model [22]. Furthermore, to optimize the backpropagation process, the weights of the NNs were initialized using Xavier normal function, which is recommended for NNs with sigmoid activation functions [23]. The training process needed some other hyperparameters that were selected from sets of possible values using cross-validation. Doing so, four hyperparameters were fixed for each model in each scenario: i) the number of epochs (25, 50, or 100); ii) the batch size (32 or 64); iii) the learning rate used for backpropagation (0.01 or 0.1); and, iv) the number of neurons in each hidden layer (3, 5, or 7). Here, for simplicity, all layers had the same number of neurons and four different NNs with 1, 3, 5, or 7 hidden layers were considered. Hereafter, NN_i will denote a NN with i hidden layers.

2.3. Performance evaluation

2.3.1. Metrics

To evaluate the predictive performance of each model, three metrics were considered: i) accuracy; i.e., the proportion of well-classified observations; ii) the AUC; i.e., the probability of ranking a random positive observation higher than a random negative observation; and, iii) F1 score, a harmonic mean between precision and recall; i.e., between the proportion of positive identifications actually correct = TP/(TP+FP) and the proportion of actual positives correctly identified = TP/(TP+FN) (T, F, P, and N stand respectively for True, False, Positives, and Negatives).

2.3.2. Evaluation of the results

Ten-fold cross-validations were used to simulate an internal validation. Theoretically, to avoid random sample fluctuations inherent to dataset generation, a large number of datasets in each scenario should be simulated with their corresponding outcomes. Here, only five datasets were used to reduce computing times and because the observed fluctuations were very small. The results show, in each scenario, the mean value of each metric as computed by cross-validation on the five datasets.

3. Results

3.1. Main effects

In scenarios with only main effects used to assign cases, LR results were similar whatever the order of the interaction terms used for training. Thus, with 1,000 observations, LR1 had an AUC of 0.70, whereas LR_2 and LR_4 had AUCs of 0.74 and 0.75, respectively (Table 2). Table 2 shows also that using penalization (either Lasso or Ridge) led to an improvement of performance. Indeed, the AUCs were around 0.80 with Lasso and Ridge whatever the order of interaction terms used for training. Furthermore, Table 2 shows that increasing the dataset size did not improve performance whatever the penalization technique (AUC = 0.80 with 10,000 observations vs. 0.79 with 100,000 observations). The AUC was around 0.78 with non-penalized LRs, especially LR_2 and LR_4.

Concerning NNs, the AUCs were quite good: 0.79 (SD: 0.03) for NN_1, 0.77 (SD: 0.02) for NN_3, 0.79 (SD: 0.03) for NN_5, and 0.77) for NN_7 (Table 2). When only the main effects were considered, the number of hidden layers had only a limited effect on performance.

Table 2 shows that increasing the dataset size did not increase the values of the metrics; it only stabilized the AUC around value 0.77 for 10,000 observations and 0.78 for 100,000 observations.

Similar results were observed for accuracy and F1 score.

3.2. Interactions of order 2

When interactions of order 2 were used to assign cases, differences in performance appeared between different LRs. Table 3 shows that with 1,000 observations, the AUC was 0.71 for LR_1, 0.80 for LR_2, and 0.79 for LR_4. Penalization improved slightly the results of all classical LRs. Indeed, Table 3 shows that the AUCs with Lasso and Ridge were 0.75 when only the main effects were used for training, 0.82 otherwise. Increasing the dataset size improved slightly AUC values. Table 3 shows also AUC values of 0.71, 0.81, and 0.82 for LR_1, LR_2, and LR_4, respectively with dataset sizes $>$ 1,000. With dataset sizes 10,000 and 100,000, penalization improved the performances regardless of the order of interaction terms used for training: the AUC was 0.77 for ridge and lasso trained without interactions and 0.84 for ridge and lasso trained with interactions.

Concerning NNs, Table 3 shows similar performances whatever the number of hidden layers. With 1,000 observations, the AUC was 0.79 for NN_1, 0.81 for NN_3, 0.80 for NN_5, and 0.78 for NN_7.

Increasing the dataset size stabilized the AUC for all NNs but did not improve performance. Table 3 shows that the AUC was 0.81

Table 3
Metrics relative to order-2.

	Dataset size: 1000			Dataset size: 10000			Dataset size: 100000		
Model	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC
LR_1	0.72	0.50	0.71	0.72	0.49	0.71	0.72	0.49	0.70
LR_2	0.80	0.62	0.80	0.81	0.63	0.81	0.82	0.63	0.81
LR_4	0.80	0.61	0.79	0.83	0.65	0.82	0.84	0.66	0.82
$lasso_1$	0.88	0.64[$]	0.75 [†]	0.89	0.67	0.77	0.89	0.66	0.76
$lasso_2$	0.90	0.74	0.82	0.92	0.77	0.84	0.92	0.77	0.84
$lasso_4$	0.90	0.74	0.82	0.92	0.77	0.84	0.92	0.77	0.84
$ridge_1$	0.88	0.64[$]	0.76 [†]	0.89	0.67	0.77	0.89	0.65	0.76
$ridge_2$	0.91	0.74	0.82	0.92	0.77	0.84	0.92	0.77	0.84
$ridge_4$	0.90	0.73	0.82	0.92	0.77	0.84	0.92	0.77	0.84
NN_1	0.89	0.69 [†]	0.79	0.90	0.73	0.81	0.90	0.72	0.81
NN_3	0.89	0.71	0.81	0.90	0.73	0.81	0.91	0.73	0.81
NN_5	0.88	0.69[*]	0.80	0.90	0.73	0.81	0.90	0.72	0.81
NN_7	0.87	0.66[+]	0.89[*]	0.90	0.72	0.80	0.90	0.72	0.81

[†] The SD for this value was 0.04 -
[*] The SD for this value was 0.05
[$] The SD for this value was 0.06 -
[+] The SD for this value was 0.07 - All other SDs were $\leq$ 0.03

Table 4
Metrics relative to order-4 interactions.

	Dataset size: 1000			Dataset size: 10000			Dataset size: 100000		
Model	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC
LR_1	0.73	0.49[*]	0.71	0.73	0.49	0.70	0.73	0.49	0.70
LR_2	0.80	0.59	0.78	0.81	0.61	0.79	0.81	0.61	0.79
LR_4	0.80	0.60	0.79	0.83	0.65	0.82	0.83	0.65	0.83
$lasso_1$	0.89	0.64[$]	0.76 [†]	0.89	0.66	0.76	0.89	0.66	0.76
$lasso_2$	0.90	0.71 [†]	0.80	0.90	073	0.81	0.91	0.73	0.81
$lasso_4$	0.91	0.74 [†]	0.82	0.92	0.78	0.85	0.92	0.79	0.85
$ridge_1$	0.89	0.64[$]	0.76	0.89	0.66	0.77	0.89	0.66	0.76
$ridge_2$	0.90	0.70 [†]	0.80	0.90	0.73	0.81	0.91	0.73	0.81
$ridge_4$	0.91	0.74	0.83	0.92	0.78	0.85	0.92	0.79	0.85
NN_1	0.89	0.66 [†]	0.77	0.90	0.71	0.80	0.90	0.71	0.80
NN_3	0.89	0.70	0.80	0.90	0.72	0.81	0.90	0.73	0.81
NN_5	0.88	0.67 [†]	0.79	0.90	0.72	0.81	0.90	0.72	0.81
NN_7	0.89	0.69	0.80	0.90	0.71	0.80	0.90	0.71	0.81

†The SD for this value was 0.04
[*] The SD for this value was 0.05
[$] The SD for this value was 0.06 - All other SDs were $\leq$ 0.03

for all NNs, except NN_7 with 10,000 observations (AUC = 0.80). Similar results were observed for accuracy and F1 score.

3.3. Interaction of order 4

When interactions of order 4 were used to assign cases, the performance of LR differed according to the order of interaction terms used during the training.

Table 4 shows that for 1,000 observations, the AUC was 0.71 for LR_1, 0.78 for LR_2, and 0.79 for LR_4. When the relationship between the features and the outcome required one or more interactions of order 4, penalization improved performance. Table 4 shows that the AUCs with Lasso and Ridge were above 0.76, 0.80, and 0.82 for penalized LR with main effects, main effects and order-2 interactions, main effects and all interactions up to order 4, respectively. Increasing the dataset size improved performance (Table 4). This improvement concerned especially penalized and non-penalized LRs with order-4 interaction terms. Indeed, the AUC for $lasso_4$ and $ridge_4$ peaked at 0.85 with 10,000 or 100,000 observations. Furthermore, the AUCs for LR4 were 0.82 and 0.83 for 10,000 and 100,000 observations, respectively. Nevertheless, for LRs without order-4 interaction, increasing the dataset size did not improve performance.

Concerning NNs, the number of hidden layers had no important impact on performance (Table 4). The AUCs were 0.77 for NN_1, 0.80 for NN_3, 0.79 for NN_5, and 0.80 for NN_7.

Table 4 shows also that increasing the dataset size stabilized the metric values rather than increasing them. Indeed, the AUC was around 0.80 for all NNs whatever the number of hidden layers.

Similar results were obtained for accuracy and F1 score.

4. Discussion

This work aimed at investigating the effects of one form of complexity (i.e., logistic interactions) and dataset size on logistic regression and neural network predicting abilities. The former effect was investigated with simulated data. Assuming that neural networks can approach any function in a complex space, low and high interaction orders were simulated. Another form (namely, model complexity) was used to compare the prediction ability of logistic regression (with various levels of interaction used as inputs and various penalization functions) vs. neural networks (with different numbers of hidden layers).

In all studied scenarios, penalization of the LR improved performance of all models. In addition, very few changes were seen by varying the size of the dataset, except for NNs. Unsurprisingly, LR with interaction terms provided better results than other models, especially with datasets with order-4 interactions. NNs provided good results with almost all datasets and had nearly the same level of performance than LR with datasets with interaction of order 1in scenarios without interaction terms. With higher interaction orders, NNs performed less well than LRs that

used the right interaction term orders. Despite this, when simulated data integrated interaction terms, NNs outperformed LRs that did not include such terms. With order-4 interactions, NNs provided similar results to LRs trained with main effects and order-2 interactions.

The present results show that almost all scenarios with Lasso or Ridge penalization improved performance. This was not surprising, especially with interactions of high orders. Indeed, in LRs, increasing the interaction order increases the number of variables used as inputs. Term selection during training classical LRs led to optimism (overfit with the training dataset leading to worse predictions with the test dataset) [24]. This problem may be solved using penalization techniques, either Lasso or Ridge. With penalization, the predictive performance improved but neither penalization method outperformed the other.

Concerning NNs, the performance was good but less than that of wellspecified LRs. One may recall that optimizing NNs is more complex than optimizing LRs. A high number of hyperparameters must be set to train NNs. This study used a restricted number of possible values for each hyperparameter.

More combinations of hyperparameters may lead to better optimization of NNs and, thereby, to better performance. Furthermore, training of NNs requires more computation time than training LRs.

Neural networks and logistic regression are subject to uncertainty. There are two types of uncertainty: aleatory uncertainty (the uncertainty inherent to the data and that the modeler cannot reduce) and epistemic uncertainty (the uncertainty due to the model design and that can be reduced by the modeler) through adding knowledge [25]. Though the effects of interaction terms on uncertainty were not studied yet, we expect that increasing the dataset size would decrease epistemic uncertainty (for a recent technical point on the techniques able to quantify uncertainty, see the comprehensive review of Abdar et al. [26]).

Two peculiarities of the present study is that, by construction, the data were idealized; i.e., free from bias. The lack of bias may inflate NN performance in cases of large datasets. Besides, the data were homoscedastic; i.e., generated as if they were drawn from a single population. The latter fact might not be true in other real-world datasets or in multicenter datasets. In case of several data sources, these points should be taken into account during model building. Moreover, the features were independently simulated (i.e., without any correlation) whereas correlations between features make interaction term influence model's prediction. Thus, correlations between features should be thoroughly examined. Furthermore, the effects of bias, heteroscedasticity, and correlation on prediction may be investigated in more extended studies.

One major asset of this study is the use of simulated data. Indeed, simulating the data allowed a better knowledge of their structure and complexity. In most cases, with real datasets, the complexity of the data structure is not known beforehand; consequently, its impact on models' performance is difficult (if not impossible) to assess and counterweight. In the present work, complexity was restricted to logistic interactions. Working with simulated data allows investigating the way NNs behave according to various forms and levels of complexity.

Consequently, a future work will examine the performance of LRs and NNs with datasets that display more complexity; e.g., nonlinear relationship between the features and the outcomes or interactions of very high orders. The abilities of other models such as Random Forest, Support Vector Machines, or Kernel regressions to approximate that relationship may also be evaluated considering various complexity levels. Besides, within the spirit of a previous review, a specific work might be dedicated to the effect of 'interaction' or 'dataset size' the two types of uncertainty [26].

5. Conclusions

The present study investigated the ability of LRs and NNs to deal with multiplicative interactions. With all interaction orders, well-specified LRs provided the best results. Furthermore, penalized LRs outperformed regular LRs. NNs performed at least as well as LRs without the right interaction order. The study showed no significant impact of the dataset size. DL can be a powerful tool but well-specified classical ML approaches are likely to be more efficient in many biomedical applications.

Funding

This work was supported by Association Nationale de la Recherche et de la Technologie (ANRT) [grantnumber 2019/1373] and by Service de Biostatistique-Bioinformatique des Hospices Civils de Lyon.

Declaration of Competing Interest

The authors have no competing interests to declare. Authors AB, CB, EF, and TG are employed by Everteam Software.

Acknowledgment

We would like to thank Jean Iwaz (Hospices Civils de Lyon) for his constructive feedback, useful comments, and valuable suggestions.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2021.106504.

References

[1] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Inc., USA, 1995.
[2] K.J. Cios, I. Shin, Image recognition neural network: IRNN, Neurocomputing 7 (1995) 159–185, doi:10.1016/0925-2312(93)E0062-I.
[3] D. Li, L. Yang, Deep Learning in Natural Language Processing, 1st Edition, Springer Publishing Company, 2018 Incorporated.
[4] S.I. Ayon, M.M. Islam, Diabetes prediction: a deep learning approach, Int. J. Inf. Eng. Electr. Bus. 11 (2019) 21–27, doi:10.5815/IJIEEB.2019.02.03.
[5] K. Tomita, R. Nagao, H. Touge, T. Ikeuchi, H. Sano, A. Yamasaki, Y. Tohda, Deep learning facilitates the diagnosis of adult asthma, Allergol. Int. 68 (2019) 456–461, doi:10.1016/j.alit.2019.04.010.
[6] E. Nazari, A.H. Farzin, M. Aghemiri, A. Avan, M. Tara, H. Tabesh, Deep learning for acute myeloid leukemia diagnosis, J. Med. Life 13 (2020) 382, doi:10.25122/jml-2019-0090.
[7] M. Lewis, G. Elad, M. Beladev, G. Maor, K. Radinsky, D. Hermann, Y. Litani, T. Geller, J.M. Pines, N. l. Shapiro, J.F. Figueroa, Comparison of deep learning with traditional models to predict preventable acute care use and spending among heart failure patients, Sci. Rep. 11 (2021), doi:10.1038/s41598-020-80856-3.
[8] K.H. Zou, K. Tuncali, S.G. Silverman, Correlation and simple linear regression, Radiology 227 (2003) 617–628, doi:10.1148/radiol.2273011499.
[9] S. Sperandei, Understanding logistic regression analysis, Biochemia Medica 24 (2014) 12–18, doi:10.11613/bm.2014.003.
[10] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowl. Discovery 2 (1998) 121–167, doi:10.1023/A:1009715923555.
[11] I. Rish, et al., An empirical study of the naive bayes classifier, in: IJCAI 2001workshop on empirical methods in artificial intelligence, 3, 2001, pp. 41–46.
[12] P. Tsangaratos, I. Ilia, Comparison of a logistic regression and Naïve Bayesclassifier in landslide susceptibility assessments: The influence of models complexity and training dataset size, Catena 145 (2016) 164–179, doi:10.1016/j.catena.2016.06.004.
[13] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (1958) 386, doi:10.1037/h0042519.
[14] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366, doi:10.1016/0893-6080(89)90020-8.
[15] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Netw. 4 (1991) 251–257, doi:10.1016/0893-6080(91)90009-T.

[16] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M.M.A. Patwary, Y. Yang, Y. Zhou, Deep Learning Scaling is Predictable, Empirically (2017) arXiv:1712.00409 [cs, stat].

[17] A. Korotcov, V. Tkachenko, D.P. Russo, S. Ekins, Comparison of Deep Learning With Multiple Machine Learning Methods and Metrics Using Diverse Drug Discovery Data Sets, Mol. Pharm. 14 (2017) 4462–4475, doi:10.1021/acs.molpharmaceut.7b00578.

[18] T. van der Ploeg, P.C. Austin, E.W. Steyerberg, Modern modelling techniquesare data hungry: a simulation study for predicting dichotomous endpoints, BMCMed. Res. Methodol. 14 (2014), doi:10.1186/1471-2288-14-137.

[19] R.B. D'Agostino, S.V. Ramachandran, M.J. Pencina, P.A. Wolf, M. Cobain, J.M. Massaro, W.B. Kannel, General Cardiovascular Risk Profile for Use in Primary Care, Circulation 117 (2008) 743–753, doi:10.1161/circulationaha.107.699579.

[20] Y.A. LeCun, L. Bottou, G.B. Orr, K.-R. M üller, Efficient BackProp, in: G. Montavon, G.B. Orr, K.R.M. Üller (Eds.), Neural Networks: Tricks ofthe Trade, Second Edition, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 9–48, doi:10.1007/978-3-642-35289-8_3.

[21] Z. Zhang, M. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, in: Advances in neural information processing systems, 2018, pp. 8778–8788.

[22] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, arXiv:1412.6980 [cs] (2017).

[23] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y.W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Vol. 9 of Proceedings of Machine Learning Research, PMLR, Sardinia, Italy, Chia Laguna Resort, 2010, pp. 249–256.

[24] Y. Zhao, E. Dantony, P. Roy, Optimism Bias Correction in Omics Studies with Big Data: Assessment of Penalized Methods on Simulated Data, OMICS 23 (2019) 207–213, doi:10.1089/omi.2018.0191.

[25] A.D. Kiureghian, O. Ditlevsen, Aleatory or epistemic? Does it matter? Struct. Saf. 31 (2009) 105–112, doi:10.1016/j.strusafe.2008.06.020.

[26] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P.W. Fieguth, X. Cao, A. Khosravi, U.R. Acharya, V. Makarenkov, S. Nahavandi, A review of uncertainty quantification in deep learning: Techniques, applications and challenges, CoRR abs/2011.06225 (2020). https://doi.org/10.1016/j.inffus.2021.05.008.

Résumé :

Grâce aux nouveaux progrès technologiques, des quantités vertigineuses de données textuelles sont créées chaque jour. Ces dernières pourraient offrir des opportunités immenses aux professionnels de santé, mais leur exploitation complexe et fastidieuse reste très peu répandue. Pour résoudre ce problème, les professionnels de santé se tournent donc peu à peu vers des solutions basées sur des modèles de langue contextualisés, la référence en matière d'outils de traitement automatique du langage naturel. Cependant, leur comportement dans le domaine médical demeure méconnu et insuffisamment étudié en langue française. L'objectif de cette thèse est d'étudier le comportement de deux modèles français -FlauBERT et CamemBERT- puis de les enrichir linguistiquement et cognitivement afin de mieux répondre aux exigences spécifiques du domaine médical. Ces recherches ont abouti au développement de deux nouveaux modèles -BioFlauBERT et BioCamemBERT- qui ont une meilleure compréhension du langage naturel dans le domaine médical ainsi qu'à une application industrielle prometteuse pour prédire des motifs de consultation.

Mots-clé :

BioCamemBERT, BioFlauBERT, CamemBERT, FlauBERT, Incorporation de connaissances, Modèle de langue, Pré-entraînement continu, Traitement automatique du langage naturel

Abstract :

The exponential growth of textual data presents a golden opportunity for healthcare professionals, yet its potential remains vastly untapped due to its complex and tedious nature. In response, healthcare professionals are increasingly turning to cutting-edge language models as the gold standard for natural language processing. However, their behavior in the medical domain remains poorly understood and insufficiently studied in French. This thesis aims to delve into the behavior of two French language models -FlauBERT and CamemBERT- and enhance them linguistically and cognitively to cater to the unique needs of the medical field. The outcome of these studies led to the creation of two new innovative models -BioFlauBERT and BioCamemBERT- with a more profound comprehension of natural language in medicine, as well as a promising industrial application in predicting consultation reasons.

Keywords :

BioCamemBERT, BioFlauBERT, CamemBERT, Continual pre-training, FlauBERT, Knowledge incorporation, Language model, Natural language processing