ICDiscovery : Aide à l'annotation par une méthode de budget pour le codage CIM-9 des textes hospitaliers

Leonardo Moros[*,***], Jérôme Azé[*], Sandra Bringay[*,**]
Pascal Poncelet[*], Maximilien Servajean[*,**], Caroline Dunoyer[***,****]

[*] LIRMM UMR 5506, Université de Montpellier, CNRS, Montpellier, France
prenom.nom@lirmm.fr
[**] Groupe AMIS, Université Paul Valery, Montpellier, France
prenom.nom@univ-montp3.fr
[***] Département d'Information Médicale, CHU Montpellier, Montpellier, France
prenom.nom@chu-montpellier.fr
[****] IDESP, UMR UA11, INSERM - Université de Montpellier, Montpellier, France
prenom.nom@umontpellier.fr

Résumé. Le codage médical est une tâche liée à la facturation clinique, visant à annoter des textes, généralement non structurés, avec des codes décrivant les diagnostics et les traitements d'un patient. Cette tâche réalisée par des spécialistes du codage, est connue pour être très difficile, en raison de la grande quantité de codes et de la longueur des documents. ICDiscovery est un outil d'aide au codage basé sur une approche d'apprentissage automatique par budget, qui propose un nombre de codes différents à associer à des documents selon plusieurs stratégies et qui explique les prédictions en identifiant les parties de textes qui ont permis la prédiction des codes.

1 Introduction

Les professionnels de santé documentent minutieusement chaque rencontre avec les patients dans leurs dossiers médicaux. Ils produisent de nombreux documents structurés et semi-structurés contenant des informations sur les traitements, les procédures et les diagnostics effectués. Afin d'obtenir des financements, les établissements de santé doivent associer aux séjours des patients des codes de facturation, issus de la Classification Internationale des Maladies (CIM). Ce codage, initialement réalisé à des fins médico-économiques peuve avoir d'autres finalités que la facturation, telles que l'amélioration de la prise en charge du patient, la prédiction de l'évolution des soins, etc. Actuellement, cette tâche est effectuée manuellement par des spécialistes du codage, c'est une activité très complexe, fastidieuse, subjective, coûteuse, chronophage et sujette à erreurs.

De nombreux travaux ont proposé des systèmes automatiques pour cette activité de codage. Ces dernières années, les approches utilisant des modèles d'apprentissage profond ont obtenus les meilleurs résultats. Les réseaux neuronaux convolutifs (CNN) et récurrents (RNN) avec des mécanismes d'attention (Xie et Xing, 2018; Mullenbach et al., 2018; Vu et al., 2020) correspondent à l'état de l'art actuel.

Les performances de ces modèles de l'état de l'art sont actuellement insuffisantes pour mettre en pratique une approche complètement automatique dans les établissements de santé. En effet, ses approches sont évaluées sur un nombre limité de codes. Lorsque l'on considère tous les codes, elles obtiennent de bons scores sur les métriques micros agrégées mais des scores très bas sur les métriques macro agrégées. Par exemple, LAAT (Vu et al., 2020) obtient 57,5 avec la métrique micro F1 et 9,9 avec la métrique macro F1. Dans cet article de démonstration, nous proposons l'outil ICDiscovery qui, comme son nom l'indique, permet de "découvrir" des codes ICD [1] dans les textes médicaux. Il repose sur une approche semi-automatique où un modèle fait des propositions de codes que le codeur doit valider. L'objectif est triple : il s'agit 1) d'adapter le nombre de codes proposés avec une approche par budget, 2) de réaliser des prédictions à différents niveaux de la hiérarchie CIM et 3) d'expliciter les prédictions en guidant le codeur vers les parties de textes ayant impacté la prédiction. Le travail de validation du codeur est ainsi facilité car il n'a pas besoin de lire le document dans son intégralité.

Il est important de noter qu'une erreur du modèle provient soit de l'oubli d'un code, soit d'une prédiction erronée. Dans le premier cas, le codeur doit lire tout le document en ayant en tête la totalité des codes. Dans le second cas, le codeur se concentre sur les parties du document utilisées par le modèle pour faire sa prédiction. Il vaut donc mieux ajouter un code à tort que l'inverse. L'outil ICDiscovery repose sur une approche par budget qui va permettre de faire varier le nombre de codes proposés au codeur. Lapin et al. (2016) renvoient les K classes les plus probables pour chaque donnée d'entrée (Top-K). Lorieul et al. (2021) généralisent l'approche avec une méthode permettant d'avoir en moyenne K labels pour chaque document, en renvoyant tous les labels ayant un score supérieur à un seuil global calculé sur le jeu de test. Bien que ces approches aient été initialement conçues pour des tâches multiclasses, nous les avons adaptées au cas des problèmes multilabels.

Par ailleurs, les codes CIM appartiennent à une hiérarchie. Le modèle peut être certain des prédictions à un niveau plus général dans la hiérarchie (e.g. diabète) malgré de mauvaises performances au niveau des feuilles. L'objectif est donc de maximiser le rappel sous contraintes tout en s'adaptant à la hiérarchie CIM-9.

L'outil ICDiscovery inclut également une explication des prédictions avec la visualisation de l'attention issue du réseau de neurones. Cette visualisation de la participation des tokens à la prédiction se fait à l'aide de cartes de chaleur (Li et al., 2016a,b; Arras et al., 2017). Les utilisateurs peuvent ainsi facilement comprendre pourquoi le modèle fait ses choix et si les codes obtenus en sortie sont vraiment présents dans le texte.

Cet article est organisé de la manière suivante : dans la section 2, nous décrivons les besoins que nous avons identifiés pour un outil d'aide au codage puis, dans la section 3, la méthode par budget mise en œuvre. Dans la section 4, nous présentons l'outil ainsi qu'un cas d'étude montrant son utilisation avant de conclure dans la section 5.

2 Caractérisation du problème

Nous identifions dans la suite les questions exprimées par les codeurs médicaux et proposons une liste de besoins ayant guidé la conception de l'outil.

1. ICD est l'équivalent de CIM en anglais International Classification of Diseases.

2.1 Questions des utilisateurs

Selon les besoins exprimés par les codeurs, nous listons 5 questions pour lesquelles une réponse les aide à identifier les codes CIM et à interpréter le processus dont sont issues les prédictions de ces codes.

(Q_1) Pour un document donné, peut-on borner le nombre de prédictions en sortie du modèle pour limiter le travail de validation du codeur ?

(Q_2) Pour un document donné, quels sont les codes associés et quel est le niveau de confiance du modèle dans les codes prédits ?

(Q_3) Pour une prédiction donnée, quel est son parent dans la hiérarchie CIM ?

(Q_4) Pour une prédiction donnée, quels sont les mots dans le texte qui ont permis au modèle de choisir un code et dans quelles sections du document se trouvent-t-ils ?

(Q_5) Pour un mot et un code donné dans le texte, quelle est l'importance du mot par rapport au code ?

La question Q_1 est liée à la gestion du budget. Répondre à cette question permet aux codeurs de modifier le comportement du modèle pour qu'il s'adapte à la charge de travail souhaitée en fixant le nombre de codes proposés. La question Q_2 est liée à la hiérarchie et à la capacité du système à aider le codeur à naviguer dedans. Les questions Q_3, Q_4 et Q_5 sont directement liés à la génération des prédictions et à leurs explications. Répondre à ces questions permet aux codeurs de facilement valider les codes prédits.

2.2 Besoins identifiés pour l'outil

À partir des questions précédentes, nous identifions 5 besoins pour notre outil :

(B_1) Permettre la sélection de la méthode de gestion du budget et des paramètres associés : soit avec un budget fixe par document (Top-K), soit avec un budget moyen pour tous les documents (Average-K) ou soit avec un compromis entre ces deux approches (Hybride).

(B_2) Énumérer les codes prédits pour un document triés par ordre de confiance.

(B_3) Afficher les deux derniers niveaux de la hiérarchie CIM-9 correspondant aux prédictions.

(B_4) Naviguer facilement vers les différentes sections d'un document soit à partir d'une table des matières, soit à partir des mots importants pour une prédiction.

(B_5) Matérialiser dans le texte l'importance d'un mot par rapport à un code avec une carte de chaleur.

3 Méthode

Dans cette section, nous allons décrire brièvement l'approche par budget [2]. Nous montrerons comment nous prenons en compte la hiérarchie des codes CIM et comment nous donnons des explications des prédictions à l'aide de l'attention issue du réseau de neurones.

2. Un article portant spécifiquement sur cette approche par budget a été accepté en papier court à la conférence EGC 2023

3.1 Définir le nombre de codes avec un budget

Soit $\mathcal{X}$ l'espace d'entrée (les comptes rendus médicaux associés à chaque patient) et $\mathcal{Y}$ les nœuds de la hiérarchie CIM. L'espace produit $\mathcal{X} \times \mathcal{P}(\mathcal{Y})$ est un espace de probabilités avec une mesure de probabilité jointe $\mathbb{P}_{X,Y}$ où $Y \in \{0,1\}^L \sim \mathcal{P}(\mathcal{Y})$ est un vecteur binaire (représentant la hiérarchie CIM aplatie) qui indique, pour chaque code, s'il est présent ou absent. Nous voulons minimiser le risque suivant qui est l'inverse du rappel :

$$\mathcal{R}(\mathcal{S}) = \mathbb{E}_{X,Y}\left[\sum_{j=1}^{|\mathcal{Y}|} \mathbb{1}\left[Y_j = 1, Y_j \notin \mathcal{S}(X)\right]\right]$$

Nous rajoutons deux **contraintes de budget** :

1a) Entre K' et K codes sont retournés par document : $\forall x \in \mathcal{X},\ K' \leq |\mathcal{S}(x)| \leq K$

1b) K'' codes sont retournés au plus en moyenne : $\mathbb{E}_X\left[|\mathcal{S}(X)|\right] \leq K''$

Notre objectif est de construire une fonction $\mathcal{S} : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ satisfaisant certaines combinaisons des contraintes précédentes, répondant à différents besoins. La contrainte $1a$ (Top-K) permet d'avoir une méthode qui borne le nombre de codes par document entre K' et K. La contrainte $1b$ (Average-K) permet d'avoir une méthode adaptative qui retourne en moyenne K'' codes par document. Finalement, une combinaison des contraintes $1a$ et $1b$ (Hybride) permet d'avoir une méthode adaptative avec une borne supérieure pour éviter de renvoyer trop de codes pour un document donné.

3.2 Prendre en compte la hiérarchie CIM

Les prédictions doivent être cohérentes vis-à-vis de la hiérarchie. Si une feuille est associée au document, alors tous les nœuds parents doivent l'être.

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \forall \tilde{y} \in \text{ancestors}(y), y \in \mathcal{S}(x) \Rightarrow \tilde{y} \in \mathcal{S}(x)$$

Cette contrainte **de hiérarchie** peut être combinée aux deux contraintes par budget et elle garantit la cohérence de la méthode selon la hiérarchie CIM.

3.3 Visualiser les mots impactant la prédiction à l'aide de l'attention

Pour la visualisation des explications du modèle, nous utilisons les poids d'attention produits par LAAT. Le modèle donne en sortie un vecteur des poids ($x = x_1, ..., x_n$) pour chaque code, où un poids est assigné à chaque mot pour chaque code. Nous calculons z, un vecteur des poids normalisés :

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)}$$

Ensuite, afin de limiter le nombre de mots mis en relief pour les utilisateurs, nous gardons uniquement ceux ayant un poids normalisé supérieur à un seuil fixé par expérimentation à 0,2.

4 Outil d'aide au codage

Dans cette section, nous décrivons l'outil ICDiscovery avec ses fonctionnalités ainsi qu'un cas d'étude illustrant comment un codeur pourrait s'en servir pour associer des codes à un compte rendu d'hospitalisation.

4.1 Implémentation

Nous construisons un estimateur via un réseau de neurones. Nous avons choisi LAAT (Vu et al., 2020) avec les paramètres optimaux mentionnés dans leur article. Nous entraînons le modèle avec un taux d'apprentissage de 0,001 et une taille de lot de 8 pendant 50 époques. Nous utilisons l'arrêt anticipé en surveillant la micro F1, s'il n'y a pas d'amélioration après 5 époques consécutives, nous arrêtons l'apprentissage. Nous utilisons les plongements word2vec [3] entraînés sur tous les comptes rendus et un abandon de neurones de 0,3 entre les couches de plongement et le LSTM. Finalement, pour les pré-traitements des textes, nous avons supprimé tous les tokens ne contenant pas des caractères alphabétiques et nous avons mis tout le texte en minuscule. L'estimateur produit des approximations de la probabilité conditionnelle de chaque code. Nous les trions de façon décroissante, ce qui nous permet de renvoyer d'abord les codes les plus probables. Une fois l'estimateur construit, nous l'utilisons en combinaison avec les règles Top-K, Average-K et Hybride.

4.2 Description de l'outil ICDiscovery

Le premier écran de l'outil permet de choisir le texte à utiliser en entrée par le modèle, soit en l'écrivant directement dans un champ de texte, soit en téléchargeant un fichier au format texte. Une fois que le texte est traité par le modèle, l'interface de la figure 1 est affichée. Dans l'encadré bleu (1), nous trouvons le texte utilisé en entrée.

L'une de caractéristiques d'ICDiscovery est qu'il permet de se déplacer dans les différentes parties d'un document. Grâce à des expressions régulières, l'outil est capable de détecter toutes les sections présentes dans le texte et d'afficher une table des matières comme vu dans l'encadré rouge (2). En cliquant sur une des sections dans la table des matières, l'outil déplace le focus automatiquement vers la section choisie. Cette fonctionnalité répond au besoin B4.

L'encadré vert (3) contient des options pour configurer la gestion du budget. En fonction des besoins, nous pouvons sélectionner les contraintes Top-K, Average-K et Hybride ainsi que leurs paramètres de configuration. Pour Top-K, nous pouvons choisir la valeur de K, pour Average-K, nous pouvons choisir la valeur de K''. Pour la méthode hybride, nous pouvons choisir K'' qui se comporte comme le K de Average-K ainsi que des bornes inférieures et supérieures pour garantir qu'au moins K' codes et au maximum K codes soient proposés par document. Cette fonctionnalité répond au besoin B1. Nous trouvons également une case à cocher qui permet d'activer l'utilisation de la hiérarchie dans les prédictions. Cette fonctionnalité répond au besoin B3.

Finalement, dans l'encadré orange (4), nous trouvons tous les codes prédits par le modèle triés par ordre décroissant. Quand la hiérarchie est utilisée, le tri est fait d'abord par rapport aux scores des parents et ensuite par rapport à ceux des enfants. Cette fonctionnalité répond au

3. https://github.com/aehrc/LAAT/tree/master/data/embeddings

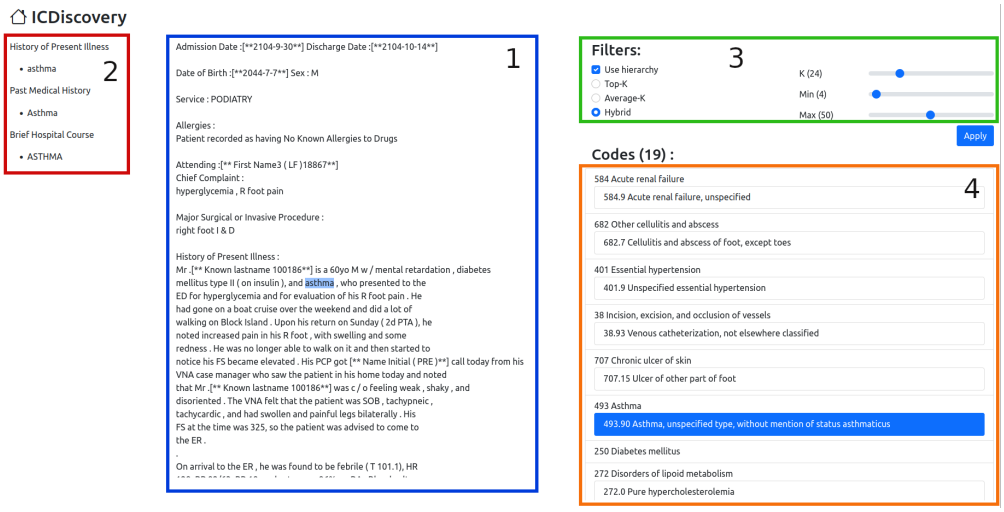

FIG. 1 – *Interface de l'outil.*

besoin B2. Cliquer sur une des prédictions permet de voir pourquoi le modèle l'a choisi. En effet, sous chaque section dans l'encadré rouge, une liste de mots importants pour la prédiction est affichée. Cliquer sur un des mots permet de se déplacer directement vers la partie du texte où le mot se trouve. Le texte est alors visualisé sous la forme d'une carte de chaleur où chaque mot prend une couleur plus ou moins foncée en fonction de l'importance de ce dernier par rapport au code sélectionné. Cette fonctionnalité répond au besoin B5.

4.3 Cas d'étude

Nous présentons un cas d'étude sur un document issu de la base MIMIC-III (Medical Information Mart for Intensive Care III) (Johnson et al., 2016). La plupart des études évaluent les approches sur la CIM-9, qui est la version précédente de la CIM-10 actuellement utilisée. Nous téléversons un compte rendu de sortie et utilisons la méthode Average-K en fixant K à 14 (la moyenne des codes par document dans notre jeu de test). Nous remarquons que l'outil propose 12 codes en sortie comme illustré sur la figure 1. Nous sélectionnons le code 493.90 (Asthma, unspecified) et remarquons que le modèle a trouvé trois mentions du mot "asthma" dans le document dans les sections "History of present Illness", "Past medical history" et "Brief hospital course". En cliquant sur les occurrences du mot dans la table de matières, nous naviguons directement vers les parties du texte où elles se trouvent. Ainsi, nous savons que le document traite d'un patient souffrant d'asthme et qu'il n'y a aucune information dans le texte précisant la nature de son asthme. Par conséquent, le code 493.90 semble correct. En suivant une approche similaire, nous pouvons valider la plupart des codes prédits.

Maintenant, concentrons-nous sur les codes 707.14 (Ulcer of heel and midfoot) et 707.15 (Ulcer of other part of foot). Ces deux codes sont très proches. Il est peu probable que le patient ait deux ulcères. Le modèle a donc du mal à détecter la bonne partie du pied à laquelle l'ulcère est associé. Pour le code 707.14, le modèle se concentre sur les mots "foot" et "ulcer" alors

que pour le code 707.15, le modèle se concentre sur les mots "foot", "toes" et "gangrene". En lisant les sections contenant ces mots, nous trouvons dans la section "Medical Condition" une mention de l'ulcère au niveau du talon, ce qui permet de valider le code 707.14. Dans ce cas, le modèle n'a pas été capable de donner un code précis au plus bas de la hiérarchie mais il nous a orienté vers la partie du document qui nous a permis de trouver la bonne réponse.

Finalement, nous choisissons d'utiliser la hiérarchie avec Average-K en fixant K à 24 (la moyenne des codes par document dans notre jeu de test en prenant en compte la hiérarchie). Nous remarquons que le code générique 250 (Diabetes mellitus) apparaît. Le modèle se concentre sur les mots "hyperglicemia", "metformin" (médicament pour le diabète) et "DKA" (acronyme pour diabetic ketoacidosis). La prédiction est donc correcte mais elle ne se situe pas au plus bas dans la hiérarchie. Nous décidons d'augmenter le budget (K=34) pour voir si le modèle nous propose un code plus précis. Nous obtenons le code 250.12 (Type II diabetes with ketoacidosis) qui correspond au bon type de diabète.

5 Conclusion

Dans cet article, nous avons présenté l'outil ICDiscovery, une interface visant à faciliter la tâche de codage médical. Les méthodes automatiques actuelles pour le codage CIM sont limitées par le grand nombre des codes. La méthode de budget proposée, la prise en compte de la hiérarchie et la visualisation de l'attention pour expliquer les prédictions facilitent le travail du codeur. Nous avons également présenté un cas d'étude illustrant le codage sur un document réel issu de la base MIMIC-III.

Notre approche est indépendante du modèle et nous souhaiterions en utiliser d'autres, comme LAAT entraîné avec la fonction de perte LDAM (Cao et al., 2019) conçue pour des jeux de données déséquilibrés. Nous pourrions aussi utiliser des *transformers* bien qu'ils n'aient pas encore dépassé l'état de l'art pour cette tâche, par exemple, le *Longformer* adapté aux longs documents (Beltagy et al., 2020). Concernant la visualisation pour expliquer les prédictions, nous utilisons les poids d'attention directement et obtenons des explications. Néanmoins, il existe des travaux qui questionnent cette utilisation des poids d'attention (Jain et Wallace, 2019). Nous pourrions utiliser d'autres méthodes telles que SHAP (Lundberg et Lee, 2017) par exemple. Il a été démontré que cette méthode produit des explications consistantes avec l'intuition humaine, ce qui est très intéressant pour une application comme la notre.

Références

Arras, L., G. Montavon, K.-R. Müller, et W. Samek (2017). Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Copenhagen, Denmark, pp. 159–168.

Beltagy, I., M. E. Peters, et A. Cohan (2020). Longformer : The long-document transformer. *ArXiv abs/2004.05150*.

Cao, K., C. Wei, A. Gaidon, N. Arechiga, et T. Ma (2019). Learning imbalanced datasets with label-distribution-aware margin loss. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 1567–1578.

Jain, S. et B. C. Wallace (2019). Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556.

Johnson, A., T. Pollard, L. Shen, L.-w. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Celi, et R. Mark (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data 3*, 160035.

Lapin, M., M. Hein, et B. Schiele (2016). Loss functions for top-k error : Analysis and insights. In *2016 IEEE Conference CVPR*, pp. 1468–1477.

Li, J., X. Chen, E. Hovy, et D. Jurafsky (2016a). Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pp. 681–691.

Li, J., W. Monroe, et D. Jurafsky (2016b). Understanding neural networks through representation erasure. *arXiv preprint arXiv :1612.08220*.

Lorieul, T., A. Joly, et D. Shasha (2021). Classification under ambiguity : When is average-k better than top-k ? *arXiv preprint arXiv :2112.08851*.

Lundberg, S. M. et S.-I. Lee (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, et R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774.

Mullenbach, J., S. Wiegreffe, J. Duke, J. Sun, et J. Eisenstein (2018). Explainable prediction of medical codes from clinical text. In *2018 Chapter of the ACL : Human Language Technologies, Volume 1*, pp. 1101–1111.

Vu, T., D. Q. Nguyen, et A. Nguyen (2020). A label attention model for icd coding from clinical text. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 3335–3341. Main track.

Xie, P. et E. Xing (2018). A neural architecture for automated ICD coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1)*.

Remerciements

Ce projet a été soutenu par le LabEx NUMEV (ANR-10-LABX-0020) intégré à l'I-Site MUSE (ANR-16-IDEX-0006) et le CHU de Montpellier.

Summary

Medical coding is a task related to clinical billing, aiming at annotating non structured medical reports with codes describing diagnoses and treatments. This task which is generally done by coding specialists is known to be extremely difficult because of the myriad of existing codes and the long length of these documents. ICDiscovery is a coding aid application based on a machine learning process with a budget approach. The tool proposes a customizable number of codes by documents using different strategies and shows the text excerpts that allowed the model to make its predictions.