Apprendre à classer des textes hospitaliers rédigés en anglais selon la classification CIM-9 avec une approche par budget

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-Valéry, 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 vise à annoter les comptes rendus médicaux selon les diagnostics et les traitements. Cette tâche est liée à la facturation clinique. Dernièrement, le codage automatique est devenu un domaine de recherche actif pour lequel de nombreux modèles, utilisant des architectures basées sur des réseaux neuronaux, ont été proposés. La plupart de ces approches sont validées sur le jeu de données MIMIC-III. Dans cet article, nous revenons sur la qualité de ce jeu et proposons un nouveau découpage. Puis, nous expérimentons un classificateur basé sur une approche par budget dont l'objectif est de faciliter ce codage en adaptant le nombre de codes à plusieurs combinaisons de contraintes.

1 Introduction

Les professionnels de santé documentent minutieusement chaque rencontre avec les patients dans des dossiers via des documents structurés et semi-structurés, contenant des informations sur les traitements, les procédures et les diagnostics. 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). De nombreux travaux ont proposé des systèmes (semi-)automatiques pour ce codage. Des chercheurs ont notamment exploré les méthodes d'apprentissage profond. 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. Dans ce travail, nous proposons un classifieur intégrant une approche par budget inspirée par les travaux de Lorieul et al. (2021) pour palier aux limites de ces derniers. Par ailleurs, la majorité des articles évaluent les approches sur la base MIMIC-III. Or, les découpages créés par Mullenbach et al. (2018) posent des problèmes de stratification, ce qui rend difficile l'évaluation des approches. Nous évaluons notre nouvelle architecture en la comparant à LAAT (Vu et al., 2020) sur un nouveau découpage du jeu de données MIMIC-III, garantissant la stratification des labels dans les échantillons.

Notation	Description
$\mathcal{X}$	l'ensemble des comptes rendus médicaux
$\mathcal{Y}$	l'ensemble des nœuds de la hiérarchie
L	le nombre de nœuds $\lvert\mathcal{Y}\rvert$
$[L]$	l'ensemble $\{1, \ldots, L\}$
$\eta_j(x)$	la probabilité conditionnelle $\mathbb{P}\big(Y_j = 1 \lvert X = x\big)$
$\hat{\eta}$	un estimateur de η ($\hat{\eta}(x) \approx \eta(x)$)
$\mathcal{P}(\mathcal{Y})$	l'ensemble des parties de $\mathcal{Y}$
$\mathcal{S}$	un prédicteur d'ensembles (Set-valued) $\mathcal{X} \to \mathcal{P}(\mathcal{Y})$
$\mathcal{R}$	le risque que l'on cherche à optimiser

TAB. 1 – *Notations utilisées.*

2 Méthode par budget

Problème, objectif et contraintes : Le tableau 1 présente les notations utilisées. Soit $\mathcal{X}$ l'espace d'entrée (les comptes rendus médicaux) et $\mathcal{Y}$ les nœuds de la hiérarchie CIM-9. L'espace produit $\mathcal{X} \times \mathcal{P}(\mathcal{Y})$ [1] 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 (la hiérarchie CIM-9 aplatie) qui indique, pour chaque label, s'il est présent. La probabilité conditionnelle associée est :

$$\eta_k(x) = \mathbb{P}\big(Y_k = 1 \lvert X = x\big)$$

Notre objectif est de construire une fonction $\mathcal{S} : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$, minimisant le risque suivant qui est l'inverse du rappel :

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

Cette fonction $\mathcal{S}$ doit également satisfaire des **contraintes de budget 1a et/ou 1b** :

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

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

Ainsi qu'une **contrainte de hiérarchie (2)**. Si une feuille est associée au document, alors tous les nœuds parents le sont.

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

Dans la suite, nous présentons les problèmes d'optimisation obtenus à partir des différentes combinaisons de contraintes, ainsi que leurs classifieurs de Bayes, qui est celui qui choisit les classes avec la probabilité d'occurrence la plus grande.

1. $\mathcal{P}(\mathcal{Y})$ est l'ensemble des parties de $\mathcal{Y}$.

Budget par document (Top-K) : Avec les contraintes 1a et 2, nous obtenons le problème d'optimisation suivant :

$$\begin{aligned} \min_{\mathcal{S}} \quad & \mathcal{R}(\mathcal{S}) \\ \text{s.t.} \quad & \forall x \in \mathcal{X},\ K' \leq |\mathcal{S}(x)| \leq K \\ & \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \forall \tilde{y} \in \text{ancestors}(y), y \in \mathcal{S}(x) \Rightarrow \tilde{y} \in \mathcal{S}(x) \end{aligned} \tag{1}$$

Pour minimiser ce risque, il suffit de prédire les K classes les plus probables pour n'importe quel document x. Soit σ une permutation de $[L] = \{1, \ldots, L\}$ telle que : $\eta_{\sigma_1(x)}(x) \geq \ldots \geq \eta_{\sigma_L(x)}(x)$. Un classifieur de Bayes pour le problème 1 est :

$$\mathcal{S}^\star(x) = \{\sigma_1(x), \ldots, \sigma_K(x)\}$$

En cas d'égalité sur les labels, on sélectionne les nœuds parents et, à un niveau donné de la hiérarchie, le choix est arbitraire. La probabilité d'un nœud parent est toujours supérieure ou égale à celle des descendants. La contrainte de hiérarchie est toujours satisfaite par ce classifieur.

Budget en moyenne (Average-K) : Avec les contraintes 1b et 2, nous avons le problème d'optimisation suivant :

$$\begin{aligned} \min_{\mathcal{S}} \quad & \mathcal{R}(\mathcal{S}) \\ \text{s.t.} \quad & \mathbb{E}_X\big[|\mathcal{S}(X)|\big] \leq K'' \\ & \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \forall \tilde{y} \in \text{ancestors}(y), y \in \mathcal{S}(x) \Rightarrow \tilde{y} \in \mathcal{S}(x) \end{aligned} \tag{2}$$

Reformulons le problème avec le multiplicateur de Lagrange λ et minimisons $\mathcal{R}_\lambda(\mathcal{S})$:

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

$$\begin{aligned} \mathcal{R}_\lambda(\mathcal{S}) &= \mathbb{E}_{X,Y}\left[\sum_{j=1}^{|\mathcal{Y}|} \mathbb{1}\big[Y_j = 1\big]\mathbb{1}\big[Y_j \notin \mathcal{S}(X)\big]\right] + \lambda \mathbb{E}\left[\sum_{j=1}^{|\mathcal{Y}|} \mathbb{1}\big[Y_j \in \mathcal{S}(X)\big]\right] \\ &= \mathbb{E}_{X,Y}\left[\sum_{j=1}^{|\mathcal{Y}|} \mathbb{1}\big[Y_j = 1\big]\mathbb{1}\big[Y_j \notin \mathcal{S}(X)\big]\right] + \lambda \mathbb{E}\left[|\mathcal{Y}| - \sum_{j=1}^{|\mathcal{Y}|} \mathbb{1}\big[Y_j \notin \mathcal{S}(X)\big]\right] \\ &= \lambda|\mathcal{Y}| + \mathbb{E}_X\left[\sum_{j=1}^{|\mathcal{Y}|} \mathbb{1}\big[Y_j \notin \mathcal{S}(X)\big]\big(\eta_j(X) - \lambda\big)\right] \end{aligned} \tag{3}$$

Le risque est minimisé lorsqu'un code est retourné uniquement si sa probabilité conditionnelle est supérieure au multiplicateur de Lagrange λ. On obtient le classifieur de Bayes :

$$\mathcal{S}^\star_\lambda(x) = \{j \in \mathcal{Y} : \eta_j(x) > \lambda\}$$

Pour un λ fixé, le nombre moyen de codes retournés est :

$$G(\lambda) = \sum_k \mathbb{E}_X\Big[\mathbb{1}\big[\eta_k(X) > \lambda\big]\Big] = \sum_k \mathbb{P}\big(\eta_k(X) > \lambda\big)$$

Son inverse généralisée est donnée par :

$$G^{-1}(K'') = \inf\big\{\lambda \in [0;1] \ : G(\lambda) \leq K''\big\}$$

avec un budget pour que le nombre moyen de labels par document soit K''. Le modèle doit retourner tous les codes avec une probabilité conditionnelle supérieure à $G^{-1}(K'')$.

Dans certains cas, il peut arriver que ce ne soit pas possible de fixer un λ qui garantisse $\mathbb{E}_X\big[|\mathcal{S}^\star(X)|\big] = K''$. Cela arrive quand l'ensemble $\{j : \eta_j(x) = G^{-1}(K'')\}$ n'est pas vide. Le classifieur de Bayes prenant en compte ce cas est donné par :

$$\mathcal{S}^\star(x) = \mathcal{S}^\star_{G^{-1}(K'')}(x) \cup \mathcal{E}(x)$$

où $\mathcal{E}(x) \subseteq \big\{j : \eta_j(x) = G^{-1}(K'')\big\}$ est un sous-ensemble arbitrairement choisi pour garantir $\mathbb{E}_X\big[|\mathcal{S}^\star(X)|\big] = K$ [2]. La contrainte hiérarchique est aussi automatiquement vérifiée ici.

Budget hybride : Dans certains cas, il se peut que le budget en moyenne renvoie trop de labels pour certains documents. Nous étudions maintenant une combinaison des contraintes par document 1a, en moyenne 1b et la contrainte hiérarchique 2 pour borner la quantité des prédictions par document. Nous rajoutons aussi une borne inférieure pour garantir qu'il n'y ait pas de documents sans prédictions. Nous obtenons le problème d'optimisation suivant :

$$\begin{aligned} \min_{\mathcal{S}} \quad & \mathcal{R}(\mathcal{S}) \\ \text{s.t.} \quad & \mathbb{E}_X\big[|\mathcal{S}(X)|\big] \leq K'' \\ & \forall x \in \mathcal{X},\ K' \leq |\mathcal{S}(x)| \leq K \\ & \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \forall \tilde{y} \in \text{ancestors}(y), y \in \mathcal{S}(x) \Rightarrow \tilde{y} \in \mathcal{S}(x) \end{aligned} \tag{4}$$

Minimisons $\mathcal{R}_\lambda(\mathcal{S})$ en satisfaisant average-K'' et en retournant entre K' et K éléments :

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

Le classifieur de Bayes obtenu via l'équation 3 s'étend au cas présent :

$$\tilde{\mathcal{S}}_\lambda(x) = \{j \in \mathcal{Y} : \eta_j(x) > \lambda, \sigma_j(x) \leq K | \sigma_j(x) < K'\}$$

La taille moyenne des prédictions pour cette règle de décision est donnée par :

$$\tilde{G}(\lambda) = \mathbb{E}\left[\sum_j \mathbb{1}\{\sigma_j(X) \leq K, \eta_j(X) > \lambda | \sigma_j(X) \leq K'\}\right]$$

2. Une preuve de l'existence de cette solution est donnée par Lorieul et al. (2021).

Le seuil est donné par $\lambda = \tilde{G}^{-1}(K'')$. Dans certaines configurations, les contraintes sont incompatibles. Si $K' > K''$, il n'y a pas de solution. Par ailleurs, nous considérons le score :

$$s_j(x) = \begin{cases} 1 & \text{si } \sigma_j(x) < K' \\ \eta_j(x) & \text{si } K' < \sigma_j(x) \leq K \\ 0 & \text{sinon.} \end{cases}$$

Avec l'égalité suivante si $\lambda < 1$ [3], la règle de décision devient :

$$\mathbb{1}\{\sigma_j(X) \leq K, \eta_j(X) > \lambda | \sigma_j(X) \leq K'\} = \mathbb{1}\{s_j(X) > \lambda\}$$

$$\tilde{\mathcal{S}}_\lambda(x) = \{j \in \mathcal{Y} : s_j(x) > \lambda\}$$

La contrainte hiérarchique : Considérons la proposition suivante :

Proposition 1 *Soit une hiérarchie telle que* $\forall k, j \in \mathcal{Y}$ *où* $k \in$ *ancestors*(j)*, nous avons* $Y_j = 1 \Rightarrow Y_k = 1$*. Alors, la contrainte suivante est toujours satisfaite pour les règles Top-*K *et Average-*K *:*

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

En supposant que les labels sont indépendants conditionnellement à x (i.e. la présence d'un label n'apporte pas plus d'informations que le document sur les autres labels), nous avons :

$$\eta_k(x) = \mathbb{P}\left(Y_k = 1 | X = x\right) = 1 - \prod_{j \in \text{descendant}(k)} \left(1 - \eta_j(x)\right)$$

Nous estimons avec un réseau de neurones la probabilité des feuilles de la hiérarchie et calculons la probabilité des nœuds parents en supposant l'indépendance conditionnelle garantissant ainsi la contrainte hiérarchique.

3 Expérimentations

Jeu de données : MIMIC-III (Johnson et al., 2016) est une base de données cliniques accessible librement. La plupart des études utilisent deux découpages créés par Mullenbach et al. (2018). Le premier contient les 50 codes CIM-9 les plus fréquents (11 368 documents) et le deuxième les 8 929 codes CIM-9 (52 722 documents) (voir le tableau 2). La distribution des étiquettes est très déséquilibrée. Par exemple, le label *567.2* est présent 211 fois et le label *276.5* est 6 fois plus présent avec 1 294 exemples. Par ailleurs, il n'y a aucune garantie de trouver tous les codes dans les échantillons d'apprentissage, de validation et de test. Par exemple, le code *276.5* apparaît dans 1 293 exemples en apprentissage, 1 fois en test et n'apparaît pas en validation. Avec cette répartition, il est difficile d'évaluer des modèles. Nous avons donc construit un découpage sur les 1 000 codes les plus fréquents. Nous avons appliqué une stratification (Sechidis et al., 2011) pour garantir que chaque label soit représenté dans les mêmes

Découpage	App	Val	Test	Total
MIMIC-III-50	8 066	1 573	1 729	11 368
MIMIC-III-1000 [4]	44 592	2 716	5 327	52 635
MIMIC-III-Full	47 719	1 631	3 372	52 722

TAB. 2 – *Quelques statistiques sur les découpages de MIMIC-III.*

proportions dans les trois jeux. De plus, les patients apparaissant dans l'apprentissage n'apparaissent pas en test/validation (voir tableau 2 (MIMIC-III-1000)).

Implémentation : Nous construisons un estimateur $\hat{\eta}$ via le réseau de neurones LAAT (Vu et al., 2020) avec les paramètres optimaux de 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, sans amélioration après 5 époques consécutives, nous arrêtons l'apprentissage. Nous utilisons les plongements word2vec [5] 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 mis tout le texte en minuscule. Une fois l'estimateur construit, nous l'utilisons avec les contraintes Top-K, Average-K et Hybride. Nous estimons le seuil $G^{-1}(K)$ pour Average-K sur l'ensemble de test mais nous aurions pu calculer $G^{-1}(K)$ sur n'importe quel ensemble de documents sans avoir les labels à partir des scores de prédiction du modèle pour calculer le seuil.

Évaluation : Nous utilisons notre découpage MIMIC-III-1000. Comme métrique, nous utilisons le rappel hiérarchique. Nous dessinons des courbes qui montrent le compromis entre la taille du budget et le rappel micro agrégé. Nous avons testé les 3 méthodes de budget sur deux configurations : 1) les labels sont les feuilles de la hiérarchie et 2) les labels sont les feuilles et leurs parents $Y_{aug} = Y \cup ancestors(Y)$.

Résultats : La figure 1 présente les résultats des approches par budget. En ordonnée, nous avons le rappel et en abscisse la taille du budget. La ligne rouge représente le rappel atteint pour la baseline et la ligne verte le nombre de labels en moyenne dans le jeu des données. **Dans le graphe de gauche**, nous montrons les résultats sur les feuilles. Les deux méthodes obtiennent de meilleurs résultats que la baseline. Pour Top-K, à partir d'un budget de 14, la méthode améliore la baseline et pour Average-K, à partir d'un budget de 12. En moyenne, Average-K obtient un rappel supérieur de 5.6% à celui de la méthode Top-K. **Le graphe de droite** correspond à un test avec la hiérarchie. Nous avons supprimé les feuilles sans frères ni parents. Pour se comparer à une méthode qui ne prend pas en compte la hiérarchie, nous représentons les courbes où le budget est complètement utilisé sur les feuilles et ajoutons les parents associés aux feuilles choisies. À partir d'un budget de 24, Average-K améliore la

3. Si $\lambda = 1$, aucun label n'est retourné et il faut faire un choix arbitraire en respectant les contraintes.
4. Découpage crée dans le cadre de cet article. `https://github.com/leo90v/MIMIC-1000`
5. `https://github.com/aehrc/LAAT/tree/master/data/embeddings`

baseline et pour Top-K, il faut un budget de 27. En moyenne, Average-K obtient un rappel 5.6% supérieur à Top-K.

Les résultats de la méthode hybride sont présentés dans la figure 2. La hiérarchie n'ayant pas amélioré les résultats, nous avons testé cette méthode uniquement sur les feuilles. Nous dessinons les courbes Top-K et Average-K qui représentent la borne inférieure et supérieure des performances pour la méthode hybride. Nous fixons K en fonction de K''. **Dans le graphe de gauche**, nous utilisons le micro rappel. Par exemple, avec $1.2K$, les performances sont supérieures en moyenne de 3.2% par rapport à Top-K et 2.25% inférieures à Average-K. Avec $1.8K$, les courbes hybride et Average-K se superposent. Ces résultats montrent qu'avec une borne relativement petite, les performances sont similaires à Average-K, sans avoir de documents avec un nombre des prédictions trop élevé pour être utilisables. **Dans le graphe de droite**, nous utilisons le macro rappel. Le comportement est similaire mais avec les performances inférieures. Avec 1.5K le micro rappel est 22% supérieure en moyenne au macro rappel. Ceci est dû au déséquilibre du jeu des données. Même avec des données stratifiées, le modèle à des difficultés avec certains labels.

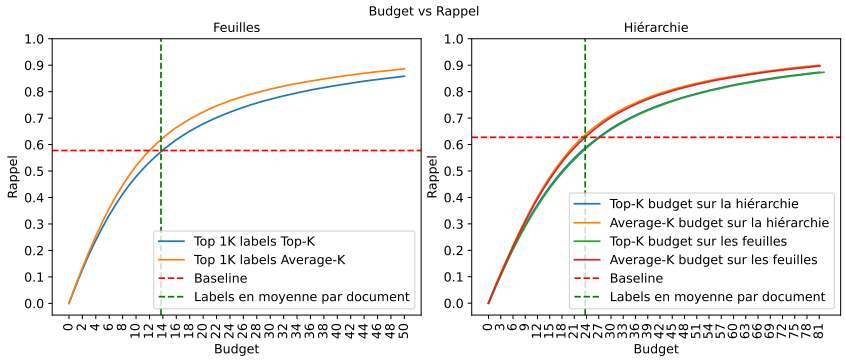

FIG. 1 – *Top-K vs Average-K.*

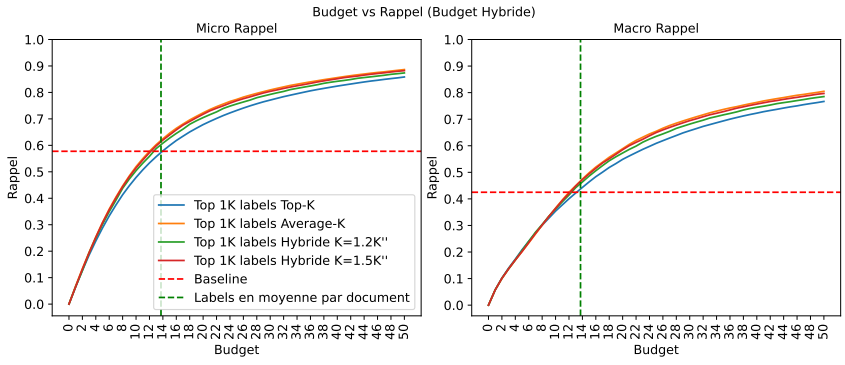

FIG. 2 – *Méthode hybride (évaluée avec le micro rappel à gauche et le macro rappel à droite).*

4 Conclusion

Nous avons présenté une nouvelle approche pour l'aide au codage médical. Via un prédicteur adaptatif, nous prédisons plus ou moins de labels par document et à différents niveaux de la hiérarchie. Notre solution est applicable à n'importe quel classifieur. Nous prévoyons donc d'utiliser d'autres modèles, tel LAAT entraîné avec la fonction de perte LDAM (Cao et al., 2019) conçue pour des jeux de données déséquilibrées. Nous pensons aussi aux approches avec un budget pondéré pour donner plus d'importance à certains labels. Nous pourrions aussi utiliser des *transformers* bien qu'ils n'aient pas encore dépassé l'état de l'art pour cette tâche.

Références

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.

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.

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

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.

Sechidis, K., G. Tsoumakas, et I. Vlahavas (2011). On the stratification of multi-label data. In *Machine Learning and Knowledge Discovery in Databases*, pp. 145–158.

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

Clinical coding is a task related to clinical billing, aiming at annotating medical reports with codes describing diagnoses and treatments. Recently, automatic coding has become a very active research area for which many models, using neural network-based architectures, have been proposed. Most of these approaches are validated on the MIMIC-III dataset. In this paper, we review the quality of this dataset and propose a new one, and then we experiment with a new classifier based on a budget approach, which aims to facilitate this coding.