Ré-entraîner ou entraîner soi-même ? Stratégies de pré-entraînement de BERT en domaine médical

Hicham El Boukkouri

Actes de la 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)
Nancy, France, 08-19 juin 2020

Ré-entraîner ou entraîner soi-même ? Stratégies de pré-entraînement de BERT en domaine médical

Hicham El Boukkouri
Université Paris-Saclay, CNRS, LIMSI, 91400, Orsay, France
hicham.elboukkouri@limsi.fr

RÉSUMÉ

Les modèles BERT employés en domaine spécialisé semblent tous découler d'une stratégie assez simple : utiliser le modèle BERT originel comme initialisation puis poursuivre l'entraînement de celui-ci sur un corpus spécialisé. Il est clair que cette approche aboutit à des modèles plutôt performants (e.g. BioBERT (Lee *et al.*, 2020), SciBERT (Beltagy *et al.*, 2019), BlueBERT (Peng *et al.*, 2019)). Cependant, il paraît raisonnable de penser qu'entraîner un modèle directement sur un corpus spécialisé, en employant un vocabulaire spécialisé, puisse aboutir à des plongements mieux adaptés au domaine et donc faire progresser les performances. Afin de tester cette hypothèse, nous entraînons des modèles BERT à partir de zéro en testant différentes configurations mêlant corpus généraux et corpus médicaux et biomédicaux. Sur la base d'évaluations menées sur quatre tâches différentes, nous constatons que le corpus de départ influence peu la performance d'un modèle BERT lorsque celui-ci est ré-entraîné sur un corpus médical.

ABSTRACT

Re-train or train from scratch ? Pre-training strategies for BERT in the medical domain

BERT models used in specialized domains all seem to be the result of a simple strategy : initializing with the original BERT then resuming pre-training on a specialized corpus. This method yields rather good performance (e.g. BioBERT (Lee *et al.*, 2020), SciBERT (Beltagy *et al.*, 2019), BlueBERT (Peng *et al.*, 2019)). However, it seems reasonable to think that training directly on a specialized corpus, using a specialized vocabulary, could result in more tailored embeddings and thus help performance. To test this hypothesis, we train BERT models from scratch using many configurations involving general and medical corpora. Based on evaluations using four different tasks, we find that the initial corpus only has a weak influence on the performance of BERT models when these are further pre-trained on a medical corpus.

MOTS-CLÉS : plongements de mots, plongements contextualisés, BERT, domaine médical, biomédical, domaine spécialisé, adaptation au domaine.

KEYWORDS: word embeddings, contextualized embeddings, BERT, medical domain, biomedical, specialized domain, domain adaptation.

1 Introduction

Les dernières années ont été témoins de l'apparition de nombreuses approches d'apprentissage par transfert en traitement automatique des langues (TAL). Ayant initialement connu un grand succès en traitement de la parole (Wang & Zheng, 2015) et en vision par ordinateur (He *et al.*, 2017), ces

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 29–42.
HAL : hal-02786184.

approches ont rapidement trouvé leur application en TAL avec notamment ULMFiT (Howard & Ruder, 2018) qui a pu montrer l'efficacité des modèles de langue pré-entraînés une fois adaptés pour des tâches de classification de texte. Peu de temps après, Radford *et al.* (2018) ont étendu ces résultats à d'autres tâches classiques du TAL (implicature textuelle, compréhension de texte…).

Suite à l'engouement autour de l'apprentissage par transfert, les modèles de plongement de mots ont eux aussi connu une importante rupture, avec l'apparition de modèles dits « contextualisés » (ELMo (Peters *et al.*, 2018), BERT (Devlin *et al.*, 2018)) capables de produire des représentations de mots qui dépendent du contexte. Malgré les gains en performance apportés par ces modèles (cf. classement GLUE), la complexité de leurs architectures implique un coût d'entraînement nettement plus important [1] [2] que pour des approches plus classiques (Word2vec (Mikolov *et al.*, 2013), Glove (Pennington *et al.*, 2014), fastText (Bojanowski *et al.*, 2017)). Ainsi, la tendance générale est d'utiliser les versions *pré-entraînées* plutôt qu'entraîner soi-même ces modèles.

Aujourd'hui, le choix le plus populaire en matière de plongements contextualisés semble être celui du modèle BERT, pour lequel on trouve aussi bien des versions pré-entraînées pour le domaine dit « général » que pour des domaines spécialisés (e.g. BioBERT (Lee *et al.*, 2020) et BlueBERT (Peng *et al.*, 2019) pour le domaine médical, SciBERT (Beltagy *et al.*, 2019) pour le domaine scientifique). Contrairement aux versions générales qui sont intégralement entraînées sur des corpus généraux, ces versions spécialisées semblent toutes être issues d'une procédure standard : partir du modèle BERT entraîné pour le domaine général puis poursuivre l'entraînement de celui-ci sur des textes spécialisés. Cette stratégie apporte des gains en performance incontestables par rapport à l'usage direct d'un modèle général (Alsentzer *et al.*, 2019; Si *et al.*, 2019) mais il semble néanmoins légitime de vouloir comparer ces modèles à des versions entraînées directement sur des textes spécialisés, sans passer par un entraînement en domaine général.

Dans le cadre de ce travail, nous nous concentrons sur le domaine médical pour lequel nous étudions l'impact de trois paramètres sur la performance finale du modèle BERT : le domaine du vocabulaire utilisé (général vs. médical), le corpus d'entraînement initial (général vs. médical vs. mélange des deux) et le corpus de spécialisation (aucun vs. médical). Pour une comparaison plus équitable, nous entraînons nous-mêmes tous les modèles en utilisant exactement les mêmes hyper-paramètres, puis nous évaluons ces modèles sur un ensemble varié de tâches classiques du domaine biomédical : détection de concepts médicaux (i2b2/VA 2010 (Uzuner *et al.*, 2011)), implicature textuelle (MEDNLI (Romanov & Shivade, 2018)) et extraction de relations (ChemProt (Krallinger *et al.*, 2017), DDI (Herrero-Zazo *et al.*, 2013)). Toutes les expériences sont effectuées en langue anglaise.

Nos contributions sont les suivantes :

— nous effectuons une analyse préliminaire sur l'impact du vocabulaire choisi sur la gestion des mots hors vocabulaire par BERT. Nous constatons ainsi une différence notable entre un vocabulaire général et médical pour traiter des termes techniques du domaine médical ;

— nous effectuons une comparaison équitable de plusieurs modèles BERT ayant des degrés de spécialisation différents. Nous observons alors que la stratégie standard consistant à ré-entraîner un modèle général obtient des performances similaires aux modèles entraînés directement sur des corpus médicaux ;

— nous partageons notre code afin de permettre la reproduction de nos résultats, et partageons nos modèles pré-entraînés pour le domaine médical.

Nous commencerons par introduire les principes de BERT (section 2), notamment ceux qui sous-

1. Entraîner un modèle BERT nécessite plusieurs cartes GPU sur une période pouvant atteindre plusieurs semaines.
2. Plusieurs benchmarks ont été effectués par Nvidia et sont consultables ici.

tendent les hypothèses que nous voulons tester (section 3), puis nous présenterons nos expériences (section 4) et leurs résultats (section 5) avant de conclure (section 6).

2 BERT

BERT (Devlin *et al.*, 2018) est un modèle neuronal de plongements lexicaux utilisant une succession de couches Transformer [3] (Vaswani *et al.*, 2017) afin de produire des représentations contextualisées. Ce modèle est entraîné sur deux tâches : une tâche de Modélisation du Langage Masquée (Masked Language Modelling - MLM) et une tâche de Prévision de la Phrase Suivante (Next Sentence Prediction - NSP). Cette section décrit l'architecture de BERT ainsi que la procédure employée pour l'entraîner. Dans tout ce qui suit, nous ferons la distinction entre deux phases :

— la phase de *pré-entraînement*, qui permet au modèle d'apprendre à produire des plongements contextualisés via les deux tâches MLM et NSP ;
— la phase d'*adaptation à une tâche*, où BERT est utilisé comme générateur de plongements au sein d'un modèle plus large qui est intégralement entraîné sur une tâche cible.

2.1 Description du modèle

2.1.1 Segmentation et représentation de l'entrée

Contrairement aux approches classiques à base de mots, BERT utilise un vocabulaire comprenant un mélange de mots et de sous-mots appelés *wordpieces* (Wu *et al.*, 2016). Cela lui permet de remédier au problème des mots hors vocabulaire en découpant chaque mot inconnu en une séquence de sous-mots faisant partie du vocabulaire [4].

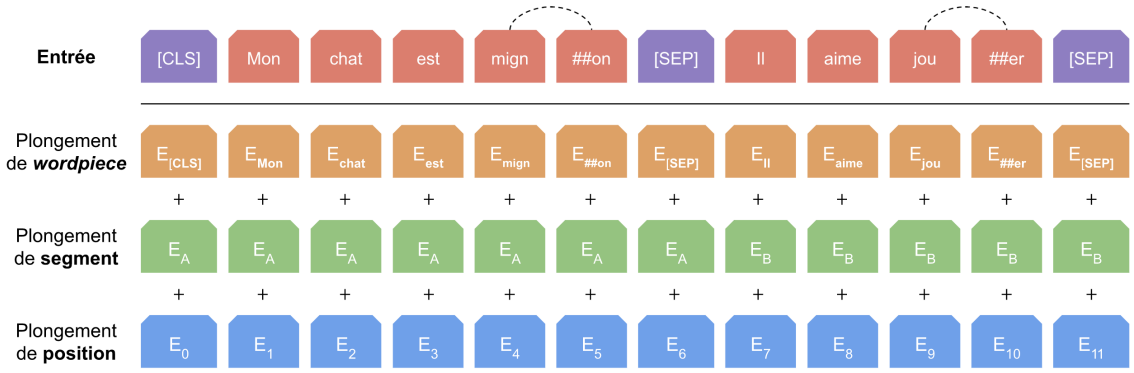

FIGURE 1 – Représentation d'une entrée dans BERT. Les plongements en entrée sont la somme de plongements de *wordpiece* (mot / sous-mot), de plongements de segment (phrase A / phrase B) et de plongements de position (Devlin *et al.*, 2018)

Lors de la phase de *pré-entraînement*, BERT prend systématiquement un couple de phrases en entrée. Ces phrases sont alors segmentées soit en mots, soit en sous-mots, avec un symbole spé-

3. En particulier, il s'agit de la partie « encodeur » de l'architecture du Transformer.
4. Si nécessaire BERT peut descendre jusqu'au niveau du caractère pour décomposer n'importe quel mot en entrée.

cial *[CLS]* en début de séquence et des symboles spéciaux *[SEP]* à la suite de chaque phrase. Chaque élément de l'entrée est alors représenté par un vecteur (*token embedding*) issu d'une matrice de plongement. Puis, afin d'injecter une notion de position et de distinguer plus facilement les éléments issus de chacune des phrases en entrée, on ajoute à ce vecteur initial un plongement de position (*position embedding*) ainsi qu'un plongement de segment (*segment embedding*). La vue complète de ces entrées est représentée en figure 1.

2.1.2 Contextualisation et couches Transformer

BERT transforme chaque paire de phrases en entrée en une séquence de vecteurs selon la procédure décrite précédemment. À ce niveau, la représentation de chaque élément est complètement indépendante de celle des autres. La prochaine étape est alors d'utiliser une série de L couches Transformer appliquant la même transformation [5] (cf. figure 2) afin de produire itérativement des représentations de plus en plus contextualisées.

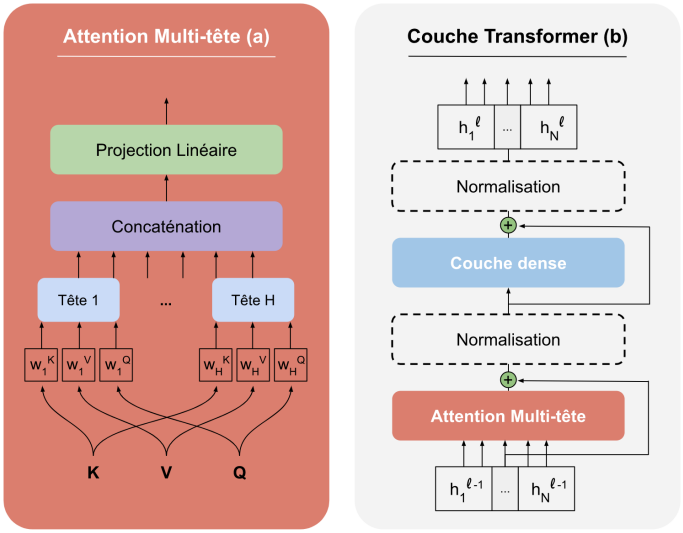

FIGURE 2 – Schéma d'une couche Transformer dans BERT. (a) Au sein du mécanisme d'attention multi-tête, les clés K, valeurs V et requêtes Q sont projetées pour chacune des têtes et servent au calcul d'un produit scalaire d'attention. L'ensemble des produits est ensuite concaténé pour enfin subir une projection linéaire. (b) Au sein de la couche Transformer, la sortie de la couche précédente sert de clé, valeur et requête pour le calcul de l'attention multi-tête

Attention Multi-tête Afin de prendre en compte le contexte global, la couche d'attention multi-tête pondère chaque représentation vis-à-vis du reste des représentations en entrée. Cette pondération dépend de paramètres propres à chaque couche d'attention qui sont ajustés à la tâche au moment de l'entraînement du modèle. De plus, afin de capter des signaux différents, la couche d'attention multi-tête se repose sur plusieurs « têtes » qui calculent à chaque fois une pondération différente des entrées. Enfin, l'ensemble des pondérations est concaténé puis projeté pour produire les représentations en sortie de la couche d'attention.

5. La transformation appliquée par chaque couche est identique mais dépend de paramètres entraînables différents.

2.2 Procédure d'entraînement

Afin d'entraîner BERT à produire des représentations contextualisées utiles pour une large gamme de tâches de TAL, on entraîne celui-ci sur deux tâches : une tâche de modélisation du langage masquée (MLM) et une tâche de prévision de la phrase suivante (NSP).

2.2.1 Modélisation du langage masquée (MLM)

Contrairement aux tâches de modélisation du langage classiques, où l'on cherche à prédire le mot suivant étant donné les mots observés précédemment, BERT est entraîné sur une tâche où l'on masque aléatoirement un mot du texte en entrée, l'objectif étant alors de prédire le mot masqué. Ainsi, grâce à la capacité de l'architecture Transformer à prendre en compte simultanément les contextes droit et gauche du mot cible, cette tâche permet a priori au modèle d'apprendre des représentations encore plus contextualisées que les modèles unidirectionnels tels qu'ELMo [6].

En pratique, les mots cible sont parfois remplacés par un symbole spécial *[MASK]*, parfois remplacés par un autre mot aléatoire et parfois conservés tels quels (cf. figure 3).

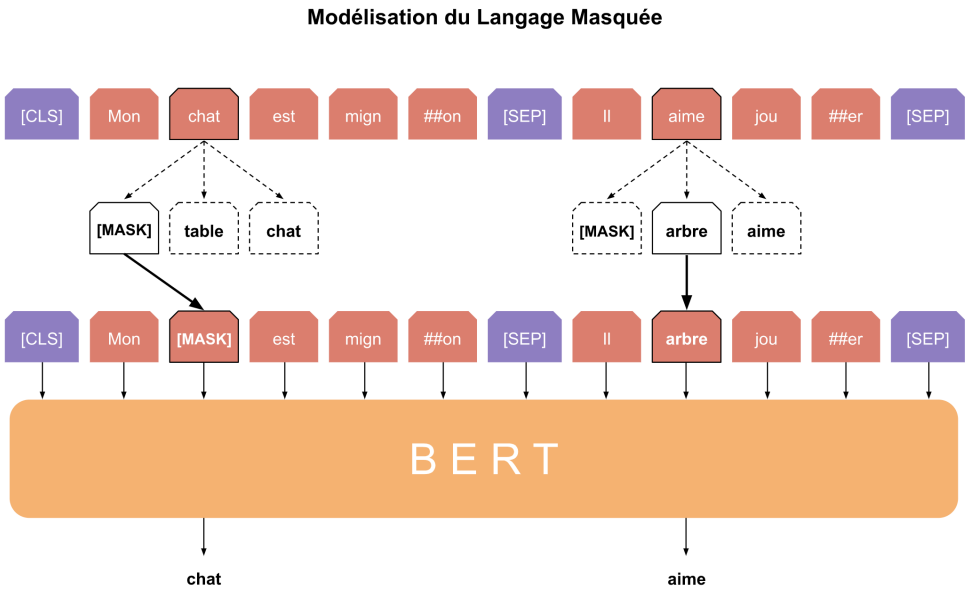

FIGURE 3 – MLM : dans un premier temps, le texte originel est perturbé en modifiant des mots choisis au hasard. Chaque mot est soit remplacé par un symbole spécial *[MASK]*, soit remplacé par un autre mot du vocabulaire ou bien conservé intact

2.2.2 Prévision de la phrase suivante (NSP)

BERT est également entraîné sur une tâche de prévision de la phrase suivante pour laquelle il faut décider si deux phrases en entrée sont consécutives. La justification de cette tâche est d'améliorer

6. Dans ELMo, les plongements sont issus d'une concaténation de représentations unidirectionnelles alors que dans BERT, celles-ci sont naturellement bidirectionnelles.

la performance du modèle sur des tâches où l'objectif est de qualifier la relation entre un couple de phrases (e.g. implicature textuelle). En pratique, la représentation du symbole spécial *[CLS]* est utilisée pour classifier chaque couple de phrases en entrée ainsi que pour toute autre tâche de classification une fois le modèle entraîné.

3 Importance du vocabulaire dans BERT

La segmentation appliquée par BERT se fait en deux étapes : d'abord une segmentation « classique » en mots, puis un découpage en sous-mots (*wordpieces*). Lors de cette seconde étape, BERT découpe autant de fois que nécessaire les mots hors vocabulaire jusqu'à retrouver des *wordpieces* connus. Ainsi, le choix du vocabulaire devrait directement influencer la qualité de cette décomposition, en particulier en domaine médical où le vocabulaire technique est très utilisé.

Pour nous en assurer, nous analysons le résultat de la segmentation d'un corpus médical [7] par deux vocabulaires : l'un du domaine général et l'autre du domaine médical (cf. figure 4). Nous observons alors que le vocabulaire médical a tendance à nettement moins découper les mots que le vocabulaire général, que l'on compte les occurrences dans le texte ou les types de mots distincts.

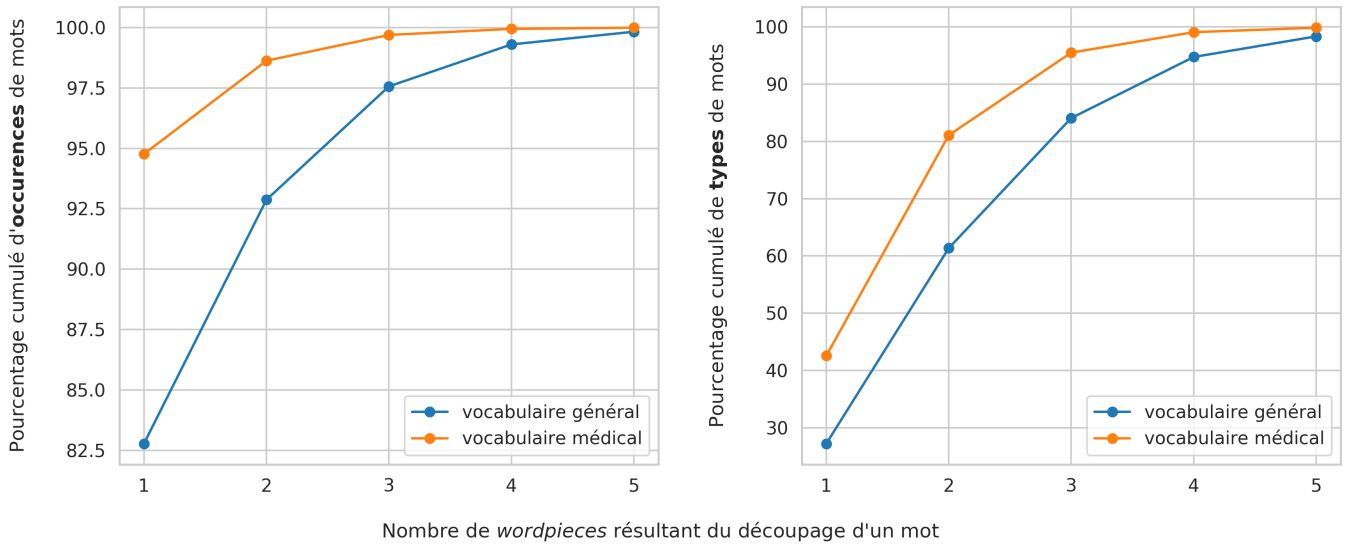

FIGURE 4 – Segmentation de textes médicaux par des vocabulaires de domaines différents

De plus, lorsque nous observons plus qualitativement cette segmentation pour des termes médicaux plus ou moins techniques, nous constatons que la qualité de la segmentation est elle aussi meilleure pour le vocabulaire médical (cf. table 1). En effet, le terme « paracetamol » est directement reconnu par le vocabulaire médical alors qu'il est divisé en *wordpieces* peu porteurs de sens en domaine général. Dans le second cas, « choledocholithiasis » est divisé en deux sous-mots par le vocabulaire médical (« choledoch » et « olithiasis »), tout deux correspondant à des notions médicales alors que le vocabulaire général divise le terme originel en un ensemble de sous-mots là encore peu porteurs de sens. Nous notons cependant que pour des termes plus rares, tels que « borborygmi », le vocabulaire médical semble également être inapte à segmenter le mot en unités porteuses de sens.

7. Il s'agit plus exactement d'un extrait du corpus médical présenté en section 4.1 de l'article.

Terme de référence	Vocabulaire médical	Vocabulaire général
paracetamol	[paracetamol]	[para, ##ce, ##tam, ##ol]
choledocholithiasis	[choledoch, ##olithiasis]	[cho, ##led, ##och, ##oli, ##thi, ##asi, ##s]
borborygmi	[bor, ##bor, ##yg, ##mi]	[bo, ##rb, ##ory, ##gm, ##i]

TABLE 1 – Segmentations en *wordpieces* issues de vocabulaires de domaines différents

4 Expériences

L'approche généralement adoptée est d'entraîner les versions spécialisées de BERT à partir du modèle originel (domaine général) en poursuivant simplement la procédure de pré-entraînement sur des textes spécialisés. Pour évaluer la pertinence de cette stratégie, nous entraînons plusieurs modèles en faisant varier les paramètres suivants : domaine du vocabulaire (général vs. médical), corpus initial (général vs. médical vs. mélange des deux) et corpus de spécialisation (aucun vs. médical).

Dans ce qui suit, nous utilisons l'architecture BERT (base, uncased) qui consiste en $L = 12$ couches Transformer avec pour chacune, $H = 12$ têtes. Nos modèles sont tous appris à partir de textes anglais en minuscule et produisent des plongements contextualisés de dimension 768.

4.1 Différentes configurations de BERT

Corpus	Composition	Nombre de documents	Nombre de mots
Général	Wikipedia (EN)	11 979 758	2 138 764 476
	OpenWebText	3 150 000	1 284 308 223
Médical	MIMIC-III	4 166 225	504 856 155
	PubMed	4 653 528	521 637 990

TABLE 2 – Détail des corpus utilisés pour le pré-entraînement de BERT

Nous notons chaque configuration par un triplet correspondant aux différentes valeurs des paramètres : (V = domaine du vocabulaire, C_1 = corpus initial, C_2 = corpus de spécialisation).

(V = **général,** C_1 = **général,** $\mathrm{C}_2 = \varnothing$) Pour une comparaison équitable, nous entraînons notre propre modèle pour le domaine général. Malgré la redondance que cela représente avec les modèles distribués par (Devlin *et al.*, 2018), entraîner ce modèle nous-même garantit une uniformité des conditions d'entraînement pour tous les modèles comparés. Cependant, nous utilisons le même vocabulaire que le modèle BERT originel : un vocabulaire construit à partir des corpus Wikipédia anglais et BookCorpus (Zhu *et al.*, 2015).

Lors du pré-entraînement, nous utilisons un corpus général (cf. table 2) constitué à partir de Wikipédia anglais ainsi qu'une partie du corpus OpenWebText (Gokaslan & Cohen, 2019) [8]. La portion de ce dernier est choisie de façon à aboutir à une taille de corpus comparable à celle utilisée originellement dans (Devlin *et al.*, 2018).

8. Étant donné que le corpus BookCorpus n'est plus disponible, nous avons remplacé celui-ci par le corpus OpenWebText qui cherche à reproduire WebText, un corpus utilisé pour entraîner le modèle GPT-2 (Radford *et al.*, 2019).

($V =$ **général,** $C_1 =$ **général,** $C_2 =$ **médical**) Nous cherchons ici à reproduire l'approche classique consistant à poursuivre l'entraînement d'un modèle du domaine général sur des textes spécialisés. Plus précisément, tout en gardant le vocabulaire général, nous poursuivons l'entraînement du modèle précédent sur un corpus médical constitué à partir de notes cliniques issues de MIMIC-III (Johnson *et al.*, 2016) et de résumés d'articles scientifiques médicaux issus de PubMed (Fiorini *et al.*, 2018).

($V =$ **médical,** $C_1 =$ **médical,** $C_2 = \varnothing$) Contrairement aux modèles précédents, cette version est directement entraînée sur des textes médicaux. De plus, nous utilisons ici un vocabulaire médical que nous construisons à partir du corpus médical (cf. table 2) en passant par la bibliothèque SentencePiece, qui implémente l'algorithme BPE (Sennrich *et al.*, 2015) [9].

($V =$ **médical,** $C_1 =$ **médical,** $C_2 =$ **médical**) À partir du modèle entraîné directement sur le corpus médical, nous effectuons un second entraînement complet sur ce même corpus afin d'aboutir à une version comparable en terme de durée d'entraînement du modèle ($V =$ général, $C_1 =$ général, $C_2 =$ médical).

($V =$ **médical,** $C_1 =$ **combinaison,** $C_2 = \varnothing$) Il est possible de s'interroger quant à l'intérêt de fusionner deux corpus, l'un général et l'autre médical, afin de pré-entraîner un modèle tel que BERT. En effet, s'il est raisonnable d'entraîner successivement sur les deux corpus, alors il peut être intéressant de considérer le cas où l'on entraîne simultanément sur ceux-ci.

Nous complétons alors notre analyse en entraînant un modèle sur la somme des deux corpus. Afin de garder l'accent sur le domaine médical, nous utilisons ici aussi un vocabulaire médical.

($V =$ **médical,** $C_1 =$ **combinaison,** $C_2 =$ **médical**) Afin d'étudier l'intérêt de partir non pas d'un modèle du domaine général mais d'une version hybride qui aurait vu les deux types de textes, nous poursuivons aussi l'entraînement du modèle précédent sur notre corpus médical.

4.2 Tâches d'évaluation

Nous évaluons nos modèles sur un ensemble varié de tâches biomédicales et cliniques comprenant : détection de concepts médicaux (détection d'entités), implicature textuelle et extraction de relations.

Détection de concepts médicaux Nos modèles sont évalués sur la tâche de détection de concepts médicaux d'i2b2/VA 2010 (Uzuner *et al.*, 2011). Cette tâche consiste en l'extraction de trois types de concepts médicaux : problème (PROBLEM, e.g. « migraine »), traitement (TREATMENT, e.g. « doliprane ») et test médical (TEST, « endoscopie »). Un exemple de note clinique annotée est donné en figure 5.

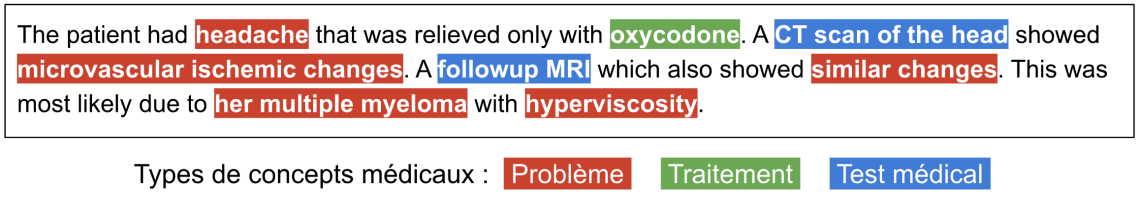

FIGURE 5 – Exemple issu de i2b2/VA 2010. Les entités sont annotées selon le format BIO.

9. Il faut noter que l'algorithme utilisé pour construire le vocabulaire du BERT original n'est pas disponible. Ainsi, pour construire son propre vocabulaire de wordpiece, il faut passer par des algorithmes similaires, tels que le BPE.

Implicature textuelle Nous évaluons également nos modèles sur la tâche d'implicature textuelle MEDNLI (Romanov & Shivade, 2018). Cette tâche consiste à classifier des couples d'extraits de notes cliniques selon 3 catégories : contradiction (CONTRADICTION), implicature (ENTAILMENT) et neutre (NEUTRAL). Des exemples sont donnés en figure 6.

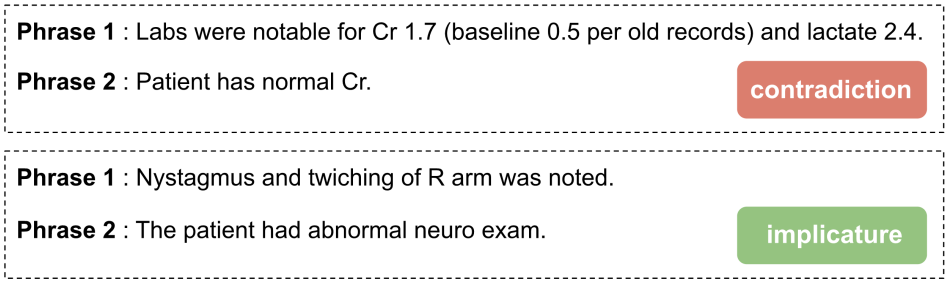

FIGURE 6 – Exemples issus de MEDNLI

Extraction de relations Afin de diversifier nos tâches d'évaluation, nous évaluons nos modèles sur deux tâches d'extraction de relations : ChemProt (Krallinger *et al.*, 2017), issue de la compétition BioCreative VI et DDI (Herrero-Zazo *et al.*, 2013), issue de SemEval 2013 - Tâche 9.2. Pour ChemProt, il s'agit de détecter la présence d'interactions entre composés chimiques et protéines en précisant le type de cette interaction : active (CPR:3), inhibe (CPR:4), agoniste (CPR:5), antagoniste (CPR:6) et substrat (CPR:9). Pour DDI, il s'agit de classifier des extraits de textes selon le type d'interaction entre médicaments qui s'y trouve : conseil (DDI-advise), effet (DDI-effect), mécanisme (DDI-mechanism) et interaction (DDI-int). Un exemple est donné pour chaque tâche en figure 7.

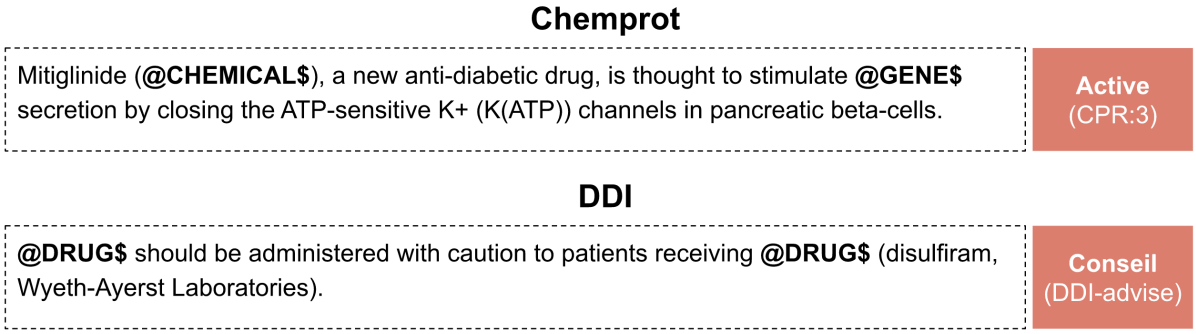

FIGURE 7 – Exemples issus de ChemProt et DDI

Le nombre d'exemples est rapporté pour chaque tâche dans la table 3.

	i2b2/VA 2010	MEDNLI	ChemProt	DDI
Entraînement	24 757	11 232	19 460	18 779
Validation	6 189	1 395	11 820	7 244
Test	45 404	1 422	16 943	5 761

TABLE 3 – Distribution du nombre d'exemples des différentes tâches d'évaluation

4.3 Paramètres des modèles

Afin de faciliter la reproduction de nos résultats, nous partagerons les paramètres utilisés lors du pré-entraînement de nos modèles BERT ainsi que lors de leur évaluation sur nos tâches médicales. De plus, nous mettrons à disposition l'ensemble de nos modèles BERT pré-entraînés ainsi que les codes ayant servi à l'entraînement et à l'évaluation [10].

4.3.1 Paramètres de pré-entraînement

Nous entraînons nos modèles BERT en utilisant 16 cartes graphiques de type Tesla V100-SXM2-16GB. L'implémentation et les paramètres choisis sont ceux fournis dans la base de code de NVIDIA [11]. Chaque entraînement consiste en deux phases :

- **Phase 1** Une série de 3 519 mises à jour (*updates*) sur des paquets (*batch size*) de 8 192 observations de taille 128 avec un taux d'apprentissage (*learning rate*) de 6.10^{-3}. Cette phase dure environ 17 heures.
- **Phase 2** Une série de 782 mises à jour (*updates*) sur des paquets (*batch size*) de 4096 observations de taille 512 avec un taux d'apprentissage (*learning rate*) plus faible valant 4.10^{-3}. Cette phase dure environ 9,5 heures.

L'optimisation a été effectuée via l'algorithme LAMB (You *et al.*, 2019) en employant un taux d'échauffement (*warmup rate*) de $0,01$ et un taux de dégradation (*weight decay*) de $0,01$.

4.3.2 Paramètres d'évaluation

Nous évaluons chaque modèle en effectuant 15 itérations (*epoch*) sur nos données d'entraînement par paquets (*batch size*) de 32 observations. À la suite de chaque itération, nous évaluons le modèle sur un jeu de validation propre à la tâche. Au bout de la dernière itération, le modèle ayant obtenu la meilleure performance sur le jeu de validation parmi les 15 itérations effectuées est retenu.

5 Résultats

Afin de prendre en compte les effets dus aux aspects aléatoires tels que l'initialisation des modèles ou l'échantillonnage des jeux de validation, nous effectuons systématiquement 10 évaluations avec à chaque fois une graine aléatoire (*random seed*) différente. Ainsi, la performance de chaque modèle est calculée en (moyenne$\pm$ écart-type). Les résultats sont présentés en table 4.

5.1 Analyse

Étant donné la complexité de la procédure de pré-entraînement de BERT, il est utile de comparer notre modèle du domaine général au modèle originel : BERT (base). Nous constatons alors que ces deux modèles ont des performances similaires, avec néanmoins un avantage pour BERT (base). Cependant,

10. https://github.com/helboukkouri/recital_2020
11. Plus exactement, nous adaptons ces scripts à nos besoins.

Modèle			Tâche d'évaluation			
V	**C_1**	**C_2**	**i2b2/VA 2010**	**MEDNLI**	**ChemProt**	**DDI**
général	général	∅	85,66 ± 0,18	77,31 ± 0,71	67,47 ± 0,99	75,81 ± 1,02
général	général	médical	89,00 ± 0,17	**84,91** ± 0,46	72,29 ± 0,58	78,82 ± 1,11
médical	médical	∅	88,80 ± 0,10	83,54 ± 0,43	71,30 ± 0,51	79,40 ± 1,15
médical	médical	médical	<u>89,20</u> ± 0,20	84,32 ± 0,73	**72,97** ± 0,46	**80,11** ± 0,79
médical	combinaison	∅	88,32 ± 0,17	82,20 ± 0,79	69,80 ± 0,51	78,90 ± 1,09
médical	combinaison	médical	**89,30** ± 0,11	84,35 ± 0,74	<u>72,80</u> ± 0,87	<u>80,04</u> ± 0,78
BERT (base) (Devlin *et al.*, 2018)			86,42 ± 0,31	77,85 ± 0,63	69,22 ± 0,56	77,89 ± 0,92
BlueBERT (base) (Peng *et al.*, 2019) [a]			88,70 ± 0,21	<u>84,53</u> ± 0,76	68,35 ± 0,61	77,89 ± 0,65

a. Il s'agit ici de la version de BlueBERT entraînée sur PubMed et MIMIC-III.

TABLE 4 – Résultat de l'évaluation des modèles. La performance de i2b2/VA 2010 est calculée en terme de F1 stricte sur les entités à détecter, celle de MEDNLI est calculée en terme de taux d'exemples corrects (*accuracy*) et enfin celles de ChemProt et DDI sont calculées en terme de micro-F1 mesure. La meilleure performance est affichée en gras, la deuxième meilleure est soulignée.

cette différence peut être due aux différences de corpus (Wikipédia et BookCorpus pour BERT (base) et Wikipédia et OpenWebText pour notre version) ou bien aux différences de paramètres de pré-entraînement. Ainsi, étant donné la proximité des performances des deux modèles, nous pouvons considérer notre procédure d'entraînement comme correcte et interpréter le reste des résultats.

Nous nous concentrons dans un premier temps sur les modèles entraînés sur un unique corpus ($C_2 = \varnothing$). Nous vérifions alors une idée intuitive : un modèle BERT entraîné sur un corpus médical avec un vocabulaire médical (V = médical, C_1 = médical, $C_2 = \varnothing$) obtient systématiquement de meilleurs résultats que son équivalent du domaine général. Par ailleurs, nous constatons que la combinaison de corpus (V = médical, C_1 = combinaison, $C_2 = \varnothing$) aboutit à une performance meilleure que celle du modèle général mais moins bonne que celle du modèle médical.

Pour ce qui est des modèles entraînés sur un second corpus, le constat principal est que globalement les performances sont très proches les unes des autres et qu'en tout état de cause, aucune configuration n'est systématiquement meilleure pour toutes les tâches, même celle maximisant le rattachement au domaine médical (V = médical, C_1 = médical, C_2 = médical). Cependant, nous pouvons remarquer de légères différences en faveur des modèles à vocabulaire médical qui semblent être favorisés dans le cas des tâches biomédicales (ChemProt et DDI). En effet, sur les quatre tâches évaluées, i2b2 et MEDNLI font partie du domaine/genre dit « clinique » (textes issus de MIMIC-III) alors que ChemProt et DDI sont du domaine/genre dit « biomédical » (textes issus de PubMed). Étant donné cette catégorisation, nous pouvons observer que l'espace entre le modèle général ré-entraîné sur du médical et le modèle entraîné deux fois sur notre corpus médical est légèrement plus marqué pour les tâches biomédicales, avec notamment une différence moyenne de 1,29 points de F1 sur DDI. Cependant, il nous est impossible de savoir si cet écart est dû aux paramètres étudiés (vocabulaire, corpus) ou bien au type de tâche (les deux tâches biomédicales sont aussi des tâches d'extraction de relations). Par ailleurs, nous notons que la combinaison de corpus n'aboutit jamais à une amélioration importante par rapport au modèle purement médical.

Enfin, nous observons que notre modèle entraîné d'abord sur un corpus général puis sur un corpus médical obtient de meilleures performances que BlueBERT sur les tâches biomédicales, bien que ce dernier soit entraîné dans des conditions similaires.

5.2 Discussion

Étant donné la proximité des performances de certains modèles, il est important de prendre en compte les valeurs moyennes des performances au regard des écarts-types associés. Ainsi, le résultat principal reste tout de même que l'ensemble des modèles atteignent des performances similaires au moment du second entraînement sur le corpus de spécialisation. Par ailleurs, les résultats impliquant notre corpus médical pourraient être amenés à varier en fonction de la taille de celui-ci. En effet, le corpus général est approximativement trois fois plus grand que le corpus médical. Ainsi, lorsque l'on compare des modèles ayant vu l'un ou l'autre, une partie de l'effet observé est sans doute due à la différence de domaine, mais une autre partie pourrait découler de la différence de taille des deux corpus. En particulier, compte tenu de la proximité des performances moyennes des modèles $(\mathrm{V} = \text{général}, \mathrm{C}_1 = \text{général}, \mathrm{C}_2 = \text{médical})$ et $(\mathrm{V} = \text{médical}, \mathrm{C}_1 = \text{médical}, \mathrm{C}_2 = \varnothing)$, il est possible que ce dernier surpasse l'approche classique en utilisant un corpus médical plus conséquent. Enfin, il est utile de préciser que nous avons fait attention à ce que les modèles utilisant une combinaison de corpus soient entraînés aussi longtemps que ceux utilisant un unique corpus. Ainsi, il est possible que ces versions ne soient pas entraînées suffisamment longtemps pour pouvoir observer leur réel potentiel.

6 Conclusion

Dans un contexte où les modèles de plongements « à la BERT » gagnent de plus en plus de terrain, nous avons voulu évaluer le mode d'utilisation standard de ces modèles en domaine spécialisé : ré-entraîner le modèle BERT d'origine sur un corpus spécialisé avant de l'adapter à la tâche d'intérêt. En nous concentrant sur le cas particulier du domaine médical, nous avons comparé plusieurs approches où le modèle initial est entraîné sur un corpus différent (général, médical, général + médical) avant d'être finalement ré-entraîné sur un corpus médical. Nous arrivons alors à la conclusion que, malgré les différences initiales des modèles suite au premier entraînement (médical > médical + général > général), toutes les configurations aboutissent à des performances sensiblement similaires. Nous pouvons alors en déduire, après factorisation des ressources et du temps nécessaires pour l'entraînement de chacun des modèles, que l'approche préférable demeure celle employée par défaut.

Remerciements

Ce travail a été financé par l'Agence Nationale de la Recherche (ANR) dans le cadre du projet ADDICTE (ANR-17-CE23-0001). Nous remercions également Junichi Tsujii ainsi le centre japonais de recherche en intelligence artificielle AIRC [12] pour nous avoir permis d'utiliser le cluster ABCI [13] afin d'effectuer l'ensemble de nos expériences.

12. https://www.airc.aist.go.jp/en/intro/
13. https://abci.ai/

Références

ALSENTZER E., MURPHY J. R., BOAG W., WENG W.-H., JIN D., NAUMANN T. & MCDERMOTT M. (2019). Publicly available clinical bert embeddings. *arXiv preprint arXiv :1904.03323*.

BELTAGY I., LO K. & COHAN A. (2019). SciBERT : A pretrained language model for scientific text. 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)*, p. 3606–3611.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). BERT : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

FIORINI N., LEAMAN R., LIPMAN D. J. & LU Z. (2018). How user intelligence is improving PubMed. *Nature biotechnology*, **36**(10), 937–945.

GOKASLAN A. & COHEN V. (2019). OpenWebText corpus. http://Skylion007.github.io/OpenWebTextCorpus.

HE K., GKIOXARI G., DOLLÁR P. & GIRSHICK R. B. (2017). Mask R-CNN. *CoRR*, **abs/1703.06870**.

HERRERO-ZAZO M., SEGURA-BEDMAR I., MARTÍNEZ P. & DECLERCK T. (2013). The DDI corpus : An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, **46**(5), 914–920.

HOWARD J. & RUDER S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv :1801.06146*.

JOHNSON A. E., POLLARD T. J., SHEN L., LI-WEI H. L., FENG M., GHASSEMI M., MOODY B., SZOLOVITS P., CELI L. A. & MARK R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific data*, **3**, 160035.

KRALLINGER M., RABAL O., AKHONDI S. A. *et al.* (2017). Overview of the BioCreative VI chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, p. 141–146.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2020). BioBERT : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **36**(4), 1234–1240.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*.

PENG Y., YAN S. & LU Z. (2019). Transfer learning in biomedical natural language processing : An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, p. 58–65.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). GloVe : Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543.

PETERS M. E., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. *arXiv preprint arXiv :1802.05365*.

RADFORD A., NARASIMHAN K., SALIMANS T. & SUTSKEVER I. (2018). Improving language understanding by generative pre-training. *URL https ://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, **1**(8), 9.

ROMANOV A. & SHIVADE C. (2018). Lessons from natural language inference in the clinical domain. *arXiv preprint arXiv :1808.06752*.

SENNRICH R., HADDOW B. & BIRCH A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv :1508.07909*.

SI Y., WANG J., XU H. & ROBERTS K. (2019). Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, **26**(11), 1297–1304.

UZUNER Ö., SOUTH B. R., SHEN S. & DUVALL S. L. (2011). 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, **18**(5), 552–556.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. In *Advances in neural information processing systems*, p. 5998–6008.

WANG D. & ZHENG T. F. (2015). Transfer learning for speech and language processing. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, p. 1225–1237 : IEEE.

WU Y., SCHUSTER M., CHEN Z., LE Q. V., NOROUZI M., MACHEREY W., KRIKUN M., CAO Y., GAO Q., MACHEREY K. *et al.* (2016). Google's neural machine translation system : Bridging the gap between human and machine translation. *arXiv preprint arXiv :1609.08144*.

YOU Y., LI J., REDDI S., HSEU J., KUMAR S., BHOJANAPALLI S., SONG X., DEMMEL J., KEUTZER K. & HSIEH C.-J. (2019). Large batch optimization for deep learning : Training BERT in 76 minutes. In *International Conference on Learning Representations*.

ZHU Y., KIROS R., ZEMEL R., SALAKHUTDINOV R., URTASUN R., TORRALBA A. & FIDLER S. (2015). Aligning books and movies : Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, p. 19–27.