Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Tri-apprentissage génératif : génération de données pour de la reconnaissance d'entitées nommées semi-supervisé

Hugo Boulanger Thomas Lavergne Sophie Rosset
Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, CNRS,
Université Paris-Saclay
prenom.nom@lisn.upsaclay.fr

RÉSUMÉ

Le développement de solutions de traitement automatique de la langue pour de nouvelles tâches nécessite des données, dont l'obtention est coûteuse. L'accès aux données peut être limité en raison de la nature sensible des données. La plupart des travaux récents ont exploité de grands modèles pré-entraînés pour initialiser des versions spécialisées de ceux-ci. La spécialisation d'un tel modèle nécessite toujours une quantité élevée de données étiquetées spécifiques à la tâche cible. Nous utilisons l'apprentissage semi-supervisé pour entraîner des modèles dans un contexte où le nombre d'exemples étiquetés est limité et le nombre de données non étiquetées est nul. Nous étudions plusieurs méthodes pour générer le corpus non étiqueté nécessaire à l'utilisation de l'apprentissage semi-supervisé. Nous introduisons les méthodes de génération entre les épisodes d'entraînement et utilisons les modèles entraînés pour filtrer les exemples générés. Nous testons cette génération avec le tri-apprentissage et l'auto-apprentissage sur des corpus Anglais et Français.

ABSTRACT

Generative Tri-training : Semi-Supervised NER with On-the-fly Generation of Unlabeled Data.

Developing Natural Language Processing solutions to new tasks or domains requires data, which is costly to collect and annotate. For particular domains, access to data is limited due to the sensitive nature of the data. These past years, most works have leveraged large general pre-trained models to initialize specialized versions of these. The specialization of such a model still requires a relatively high quantity of labeled data specific to the target task. We use semi-supervised learning to train models in a context where the number of labeled examples is limited, and the number of unlabeled data is zero. Since semi-supervised learning requires unlabeled examples, we study methods of generating the necessary unlabeled corpus. We introduce the generation methods between the training episodes and use the models trained to filter the generated examples. We test this on-the-fly generation with tri-training and self-training with corpora in English and French.

MOTS-CLÉS : REN, génération, faible ressources, semi-supervision.

KEYWORDS: NER, generation, low resource, semi-supervision.

1 Introduction

La reconnaissance d'entités nommées (REN) est une tâche d'étiquetage de séquences. L'entraînement de modèles de REN nécessite généralement une quantité importante de données étiquetées. L'accès aux données, étiquetées ou non, est compliqué pour certaines langues ou certains domaines. Par

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 68–79.

exemple, obtenir des données médicales prend des années car l'obtention de données diffusables et l'annotation par des spécialistes est un processus lent. Par conséquent, l'apprentissage dans un environnement à faibles ressources est un sujet en plein essor. Le but de cet article est de proposer une méthode permettant de réduire la quantité de données naturelles annotées nécessaires pour entraîner des modèles de REN.

Les techniques d'apprentissage par transfert (Ruder, 2019) sont un moyen d'utiliser les connaissances acquises depuis d'autres données pour améliorer les performances des modèles. L'apprentissage semi-supervisé est un autre paradigme d'apprentissage qui aborde le problème des ressources limitées (Van Engelen & Hoos, 2020). Dans ce paradigme, le contexte de faibles ressources fait généralement référence à une faible quantité de données étiquetées mais à une grande quantité de données non étiquetées disponibles. Contrairement à l'apprentissage semi-supervisé, le problème que nous cherchons à résoudre est de n'avoir qu'une petite quantité de données étiquetées et aucune donnée non étiquetée. Nous utilisons de grands modèles de langue génériques sans réglage fin pour générer les données non étiquetées utilisées dans l'apprentissage semi-supervisé. Nous n'affinons pas nos modèles de génération car nous ne disposons pas de suffisamment de données.

Nous évaluons plusieurs méthodes de génération utilisant des modèles de langue pré-entraînés. Premièrement, nous employons deux méthodes de génération fondées sur la modélisation classique de gauche à droite : la génération de la phrase suivante et la complétion de phrases. Ensuite, nous utilisons deux méthodes de modélisation séquence à séquence pour remplacer le contexte ou les mentions dans les phrases étiquetées. Nous évaluons notre méthode sur deux corpus bien connus, CoNLL (Sang & De Meulder, 2003) et I2B2 (Uzuner *et al.*, 2011). Nos principales contributions sont les suivantes [1] :

— L'*algorithme de tri-apprentissage génératif*, un algorithme qui ajoute la génération et la sélection des exemples non étiquetés à chaque épisode de l'algorithme de tri-apprentissage.
— Une analyse complète des méthodes de génération multiples pour l'algorithme de tri-apprentissage générative.

2 Contexte

L'apprentissage d'un modèle de REN nécessite une grande quantité de données étiquetées. L'augmentation a été utilisée pour améliorer les performances des modèles de traitement automatique de la langue. Des techniques telles que la traduction circulaire (Sennrich *et al.*, 2016), l'augmentation EDA (Wei & Zou, 2019) ou la génération de paraphrases en utilisant BART (Dopierre *et al.*, 2021) ont été utilisées pour améliorer la classification de phrases. Cependant, la paraphrase utilisant la rétro-traduction sur des données médicales pour une tâche d'étiquetage n'est pas efficace (Neuraz *et al.*, 2018). Par conséquent, nous ne l'utiliserons pas sous cette forme, mais l'utilisation de modèle de langue se rapproche de cette pratique. Une méthode utilisant des modèles de langue pour augmenter les données, DAGA (Ding *et al.*, 2020), s'est avérée efficace dans un contexte d'apprentissage supervisé et semi-supervisé. Cette méthode utilise les données d'apprentissage pour entraîner un BiLSTM afin de générer des données étiquetées ou non étiquetées. Nous avons choisi d'utiliser des modèles de langue pré-entraînés.

L'apprentissage semi-supervisé est un paradigme d'apprentissage visant à améliorer les performances des modèles entraînés en ajoutant des exemples non étiquetés à l'ensemble d'apprentissage (Van En-

1. Le code est disponible ici : https://github.com/HugoBoulanger/Tritraining-Gen

gelen & Hoos, 2020). Ce paradigme comporte plusieurs branches qui dépendent de la manière dont les données non étiquetées sont utilisées. L'utilisation de pseudo-étiquettes à différents stades de l'entraînement a fonctionné pour la REN (Wang *et al.*, 2021b). Les modèles aux stades d'apprentissage précédents génèrent des pseudo-étiquettes pour les étapes suivantes. Ces algorithmes dépendent du nombre de modèles entraînés et des modèles utilisés pour générer les pseudo-étiquettes. L'algorithme le plus simple est l'auto-apprentissage (Yarowsky, 1995), qui entraîne un modèle et l'utilise pour créer les pseudo-étiquettes. D'autres méthodes utilisant des ensembles de modèles ont été créées pour réduire les biais induits par la génération des pseudo-étiquettes par le même modèle. Le co-training (Blum & Mitchell, 1998) est une méthode d'entraînement de deux modèles dans laquelle chaque modèle génère les pseudo-étiquettes de l'autre. Une généralisation de cette méthode existe dans laquelle un ensemble de n modèles est entraîné, et les pseudo-étiquettes pour un modèle m_j sont produites en utilisant un système de vote à travers les n autres modèles. L'algorithme semi-supervisé que nous utilisons est une variante utilisant trois modèles appelée tri-apprentissage (Zhou & Li, 2005). Des résultats positifs existent avec cette méthode sur la tâche d'extraction de concepts cliniques. Une version optimisée pour les données de l'algorithme de tri-apprentissage est le tri-apprentissage avec désaccord (Søgaard, 2010). Cette version réduit la quantité de données nécessaires en ajoutant uniquement les données pseudo-étiquetées au jeu d'entraînement d'un modèle quand celui-ci est en désaccord avec les deux autres. Nous n'avons pas implémenté cette version du tri-apprentissage car nous voulons conserver autant de données générées pertinentes que possible. Nous proposons un algorithme de tri-apprentissage génératif, qui utilise le système de vote pour écarter les échantillons nouvellement générés sur lesquels il n'y a pas d'accord.

Nous utilisons des modèles de langue pour générer de nouvelles données non étiquetées nécessaires à l'apprentissage semi-supervisé. Nous utilisons les modèles GPT-2 (Radford *et al.*, 2019) et T5 (Raffel *et al.*, 2020) dans leur version à 1M paramètres. Ces données servent à entraîner nos étiqueteurs, qui utilisent des modèles BERT (Devlin *et al.*, 2018) pré-entraînés comme base de leur architecture.

3 Tri-apprentissage génératif

Tri-apprentissage Le tri-apprentissage (Zhou & Li, 2005) est un algorithme d'apprentissage semi-supervisé pour entraîner un ensemble de trois modèles. Chaque modèle est entraîné sur des données non étiquetées qui reçoivent des pseudo-étiquettes des deux autres modèles. Les phrases pseudo-étiquetées sont utilisées lorsque les deux modèles parviennent à un accord, ce qui permet à la fois l'ajout de pseudo-étiquettes et un filtrage des phrases non-adaptées. La génération d'un ensemble de données non étiquetées de taille fixe pour l'apprentissage semi-supervisé semble arbitraire. Il n'y a aucune garantie que les modèles utiliseront les données pour apprendre car il n'y a aucune garantie que les modèles parviendront à un accord sur les pseudo-étiquettes. En pratique, certaines des données générées ne sont jamais utilisées. Pour résoudre ce problème, nous avons conçu l'algorithme de tri-apprentissage génératif qui se sert des qualités de filtrage du tri-apprentissage pour sélectionner les phrases issues de la génération au fil des épisodes.

Pré-entraînement Le tri-apprentissage nécessite une étape d'entraînement pour initialiser les étiqueteurs afin qu'ils soient capables de produire des pseudo-étiquettes. Nous appelons cette étape l'étape de pré-entraînement. Elle est représentée entre les lignes 1 et 4 de l'Algorithme 1. L'étape de pré-entraînement est effectuée sur des sous-ensembles de données échantillonnés avec remise à

Algorithme 1 : Tri-apprentissage génératif

Entrées : S_n le sous-ensemble de données annotées, g la méthode de génération

1 **pour** $i \in [\![1\ ;\ 3]\!]$ **faire**
2 $\quad m_i^{-2} \leftarrow entrainement(echantillonnage(S_n), BERT)$
3 $\quad m_i^{-1} \leftarrow entrainement(S_n, m_i^{-2})$
4 **fin**
5 $t \leftarrow 0$
6 $U^0, L_1^{-1}, L_2^{-1}, L_3^{-1} \leftarrow \emptyset$
7 **tant que** *un* m_i *apprend toujours* **faire**
8 $\quad L_1^t, L_2^t, L_3^t \leftarrow generation\left(S_n \cup \bigcup_{i=1}^3 L_i^{t-1}, m_{i,i\in[\![1\ ;\ 3]\!]}^{t-1}, g\right)$
9 $\quad U^{t+1} \leftarrow U^t \cup enlever\_etiquettes\left(\bigcup_{i=1}^3 L_i^t\right)$
10 $\quad$ **pour** $i \in [\![1\ ;\ 3]\!]$ **faire**
11 $\quad\quad j, k \leftarrow [\![1\ ;\ 3]\!] - \{i\}$
12 $\quad\quad$ **pour** $x \in U^t$ **faire**
13 $\quad\quad\quad$ **si** $m_j^{t-1}(x) = m_k^{t-1}(x)$ *and* $(m_i^{t-1}(x) \neq m_j^{t-1}(x)$ *or* $drop(p > .5))$ **alors**
14 $\quad\quad\quad\quad L_i^t \leftarrow L_i^t \cup \{(x, m_j^{t-1}(x))\}$
15 $\quad\quad\quad$ **fin**
16 $\quad\quad$ **fin**
17 $\quad$ **fin**
18 $\quad$ **pour** $i \in [\![1\ ;\ 3]\!]$ **faire**
19 $\quad\quad$ **si** m_i *apprend toujours* **alors**
20 $\quad\quad\quad m_i^t \leftarrow entrainement(S_n \cup L_i^t, m_i^{t-1})$
21 $\quad\quad$ **fin**
22 $\quad$ **fin**
23 $\quad t \leftarrow t + 1$
24 **fin**

partir de l'ensemble étiqueté S_n (Ruder & Plank, 2018). Ces sous-ensembles font **la même taille que** S_n**.** Cependant, nous avons constaté que l'ajout d'un épisode d'apprentissage sur l'ensemble étiqueté complet S_n après l'ensemble échantillonné améliore considérablement les performances du tri-apprentissage. Cet ajout se trouve à la ligne 3 de l'Algorithme 1.

Tri-apprentissage génératif L'algorithme 1 et la figure 1 reflètent les modifications apportées à l'algorithme de tri-apprentissage. Avec le tri-apprentissage, nous entraînons un ensemble de trois modèles $m_i, i \in [\![1\ ;\ 3]\!]$. L'entraînement est divisé en épisodes au cours desquels nous entraînons les trois modèles. Pour un modèle m_i donné, l'entraînement épisodique s'arrête lorsque le score du modèle sur l'ensemble de validation est inférieur à celui de l'épisode précédent. Ce processus est décrit dans la Figure 2. Au début de chaque épisode (ligne 8 de l'algorithme 1), nous générons de nouveaux sous-ensembles de données pseudo-étiquetées L_i^t à partir des données annotées et pseudo-annotées de l'épisode précédent. Nous ajoutons les données générées sans annotation aux données non-annotées disponibles pour l'étape suivante ligne 9. Ces sous-ensembles sont générés à partir des ensembles étiquetés et précédemment pseudo-étiquetés. Nous ajoutons les exemples nouvellement générés L_i^t sans leurs étiquettes aux exemples précédemment générés U^t pour l'épisode suivant. Chacun de ces

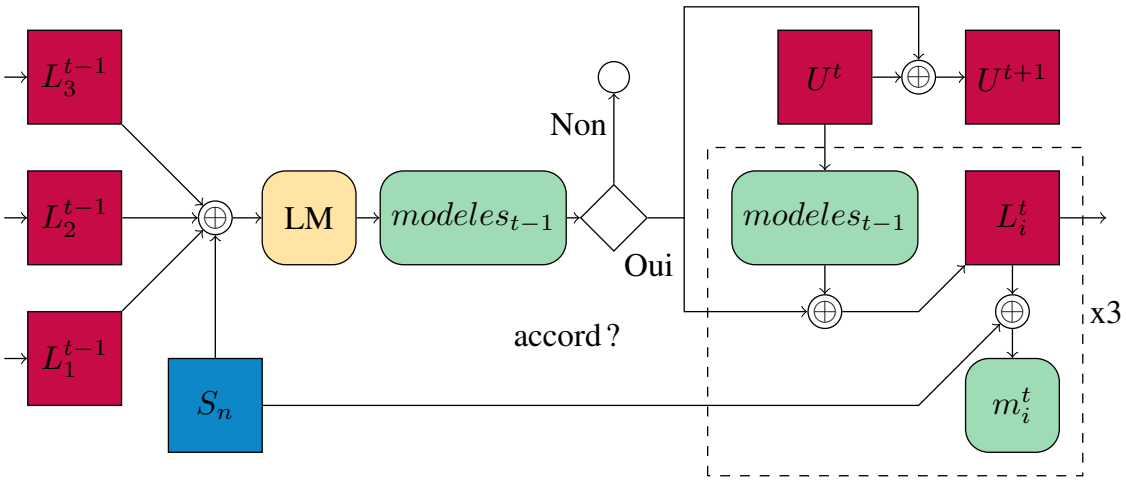

FIGURE 1 – Étape t du tri-apprentissage génératif. En rectangles les données, et avec les coins arrondis les modèles et ensembles de modèles.

ensembles pseudo-étiquetés L_i^t est ensuite augmenté à l'aide d'exemples précédemment générés U^t avec le mécanisme classique de tri-apprentissage (lignes 10 à 17 de Alg. 1). Enfin, entre les lignes 18 et 22, nous entraînons les modèles qui nécessitent encore un entraînement. Cet algorithme résout le problème des données inutilisées en rejetant les exemples fraîchement générés sur lesquels il n'y a pas d'accord, comme le montre la Figure 1. Nous utilisons une modification du désaccord avec une chance sur deux pour chaque modèle de conserver les exemples pseudo-étiquetés sur lesquels il y a un accord complet, comme le montrent les lignes 13 et 14. La motivation derrière ce changement est de conserver en parti les exemples vus par les modèles pour éviter l'oubli, tout en bénéficiant d'une quantité de données plus faible pour améliorer la vitesse d'apprentissage.

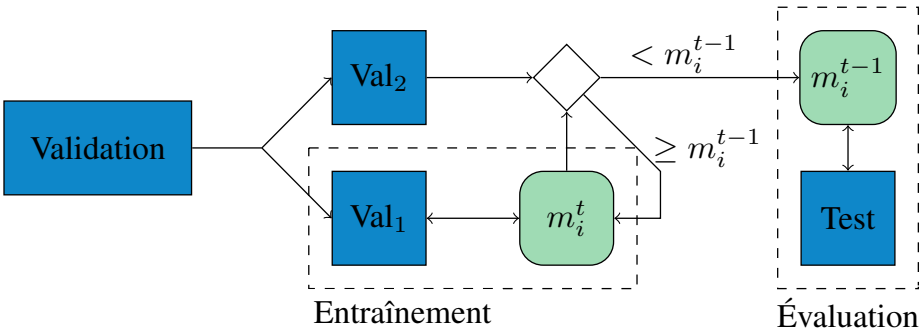

FIGURE 2 – Mécanisme de validation pour un modèle m_i. Nous découpons le jeu de validation du corpus naturel en deux, une partie servant durant l'entraînement, et une autre servant à comparer entre les épisodes.

Pour le tri-apprentissage, et l'aspect épisodique en général, nous avons besoin de deux ensembles de données de validation pour évaluer notre entraînement. Nous divisons l'ensemble de validation de nos corpus en deux. La première moitié sert d'ensemble de validation pour l'entraînement des modèles pendant l'épisode. La seconde moitié est utilisée pour comparer les modèles entre les épisodes et définir le moment où un modèle doit arrêter son apprentissage. Ce mécanisme est décrit dans la Figure 2.

4 Méthodes de génération

Les méthodes pour générer de nouveaux échantillons non étiquetés sont au cœur de cette étude. La modélisation de la langue est l'outil que nous utiliserons pour générer de nouveaux exemples. Il existe plusieurs façons de faire de la modélisation de la langue. La façon la plus traditionnelle de faire de la modélisation de la langue est de prédire les tokens de gauche à droite. Ces modèles utilisent le contexte à gauche pour prédire les tokens suivants à droite. Nous utiliserons ce type de modélisation pour nos deux premières méthodes de génération. Le modèle que nous utilisons pour ces méthodes de génération est GPT-2 [2] (Radford *et al.*, 2019). Il existe d'autres modèles entraînés sur d'autres tâches. Dans notre cas, nous utiliserons aussi T5 (Raffel *et al.*, 2020), qui a été entraîné en utilisant un objectif de remplacement de séquences. Nos deux dernières méthodes de génération utilisent cette capacité de remplacement pour éditer la mention ou le contexte d'une phrase étiquetée ou pseudo-étiquetée. Pour ces méthodes, nous utilisons T5 v1.1 [3]. Toutes les méthodes de génération sont appliquées à des phrases étiquetées ou pseudo-étiquetées. Chaque méthode est illustrée par un exemple pour lequel le rouge indique les portions provenant directement de la phrase d'origine, le bleu indique les portions générées, et le gras indique les portions annotées.

génération de phrase suivante : Pour la génération de phrase suivante, nous utilisons le texte d'une phrase étiquetée ou pseudo-étiquetée comme contexte gauche pour modéliser le texte suivant. Notre hypothèse est que les nouvelles phrases apportées à l'ensemble de données par cette méthode apporteront de la variété à l'ensemble tout en restant dans le domaine.

Ceci est un **exemple**. → Il permet de décrire la situation.

Complétion de phrase : La complétion est une méthode utilisant la modélisation de gauche à droite pour compléter la phrase. La position à partir de laquelle générer est choisie par échantillonnage entre les premiers tokens des mentions dans la phrase. Si la phrase ne contient aucune mention, la position est échantillonnée entre 25% et 75% de la longueur de la phrase. Nous supposons que cette méthode apportera de la variété aux mentions tout en conservant l'approche classique de modélisation.

Ceci est un **exemple**. → Ceci est un objet d'intérêt.

Remplacement de mentions : Notre objectif est d'apporter la diversité aux objets d'intérêt en remplaçant les mentions des phrases étiquetées. Si plus d'une mention est présente, nous échantillonnons la mention qui doit être remplacée. Nous ignorons les phrases sans mention. Notre hypothèse est que les modèles de langue ont appris des informations sur n'importe quel sujet pendant leur entraînement. Les sujets qui nous intéressent sont liés aux concepts que nous souhaitons étiqueter dans les phrases. Nous supposons qu'étant donné un contexte qui contient une instance d'un concept donné, le modèle sera capable de remplacer cette instance par d'autres instances du même concept. Nous pensons que cette méthode fonctionnera mieux en combinaison avec d'autres méthodes.

Cet **exemple** est plus adapté. → Cet objet d'intérêt est plus adapté.

Remplacement de contexte : Le remplacement du contexte fonctionne de manière similaire au remplacement des mentions, sauf que c'est le texte non étiqueté qui est remplacé. Nous créons une liste de séquences contenant toutes les séquences de contexte de la phrase. Nous échantillonnons la séquence qui sera remplacée. Nous ignorons les phrases sans mention. Notre hypothèse est que les

2. https://huggingface.co/gpt2-large
3. https://huggingface.co/google/t5-v1_1-large

modèles d'étiquetage apprennent à prédire à la fois à partir du contexte et des mentions. De cette hypothèse, nous déduisons que l'apprentissage à partir de contextes plus variés pourrait être bénéfique pour l'étiqueteur.

Ceci est un **exemple**. → Nous avons besoin d'un exemple.

5 Ressources

	nb. phrases train	nb. phrases test	BERT large	BioBERT	SOTA
CoNLL	14 986	3 683	**92,0**		94,6
I2B2	11 482	27 625	85,0	**86,6**	90,3

TABLE 1 – Performances des modèles utilisés en score F_1 quand ils sont entraînés sur le corpus naturel complet et comparaison avec l'état de l'art. Métrique calculée avec seqeval (Ramshaw & Marcus, 1995; Nakayama, 2018). État de l'art pour CoNLL (Wang *et al.*, 2021a) et I2B2 (Si *et al.*, 2019) d'après papers with code [5].

Données Nous testons notre méthode sur deux corpus, CoNLL 2003 English (Sang & De Meulder, 2003) et I2B2 (Uzuner *et al.*, 2011). CoNLL est un corpus de nouvelles Reuters en anglais dans lequel des personnes, des lieux, des organisations et diverses entités sont étiquetées. I2B2 est un corpus de dossiers médicaux en anglais dans lequel les traitements, les problèmes et les tests sont étiquetés. Notre travail vise à permettre l'entraînement de modèles à faibles ressources. Nous testons notre méthode sur des corpus disponibles afin d'obtenir des résultats pouvant être comparés aux méthodes de l'état de l'art. Pour simuler un environnement à faibles ressources, nous devons restreindre la quantité de données disponibles pour l'entraînement. Nous échantillonnons les données pour obtenir des ensembles d'entraînement réduits. Nous répétons nos expériences sur un ensemble de graines afin de prendre en compte les biais d'échantillonnage dans l'évaluation de la méthode.

Modèles L'architecture des modèles que nous utilisons est BERT avec un classificateur MLP, ce qui nous permet de bénéficier de la vaste gamme de modèles BERT pré-entraînés. Nous affinons les modèles sur les données d'apprentissage, que ce soit pour la baseline ou les méthodes d'augmentation testées avec les paramètres présenté dans BERT (Devlin *et al.*, 2018). Nous avons testé quelques modèles BERT pré-entraînés sur les corpus complets afin de sélectionner les modèles que nous utiliserons (voir Tableau 1). Nous avons opté pour BERT Large (Devlin *et al.*, 2018) pour CoNLL, et BioBERT (Lee *et al.*, 2020) pour I2B2.

6 Expériences

Nous avons conçu un ensemble d'expériences pour évaluer le tri-apprentissage génératif. Ces expériences comparent l'apprentissage supervisé au tri-apprentissage génératif et avec les données

5. https://paperswithcode.com/

de tâches disponibles. La génération avec semi-supervision est testée avec deux algorithmes d'apprentissage semi-supervisé : le tri-apprentissage et l'auto-apprentissage. Notre baseline est l'apprentissage supervisé sur des données étiquetées sans aucune semi-supervision. Notre **topline est le tri-apprentissage** avec des données naturelles comme données non étiquetées. Pour la topline, nous considérons un ensemble de 10 000 phrases de chaque corpus à partir duquel nous extrayons les corpus supervisés S_n, composé de 50 à 1 000 phrases, et le reste est utilisé comme l'ensemble non étiqueté servant à peupler les L_i^t. Dans ces cas plus ou moins extrêmes, la quantité de données n'est pas suffisante pour apprendre des modèles performants, comme le montrent les résultats de la baseline Table 2.

À chaque épisode d'apprentissage semi-supervisé, nous générons une quantité fixe de données. Nous visons à générer 5 000 exemples pour chaque épisode. Cette quantité est suffisamment importante pour apporter une quantité significative d'informations mais pas trop afin d'éviter de ralentir le processus d'apprentissage. Nous générons une plus petite quantité de données pour les premiers épisodes car la quantité de contexte disponible, les données étiquetées et pseudo-étiquetées, est plus petite. Cette petite quantité de données pourrait également voir son signal noyé si trop de données sont générées, comme on le voit avec la méthode topline. Nous facilitons les deux premières étapes en générant seulement 500 et 2 500 exemples au lieu de 5 000.

La génération se fait dans le contexte de l'apprentissage semi-supervisé. Pour le tri-apprentissage, les exemples générés sont répartis en trois ensembles, un pour chaque modèle entraîné dans le tri-apprentissage. Les nouveaux ensembles de données sont générés en appliquant la méthode de génération respective aux exemples échantillonnés de l'ensemble de données naturelles et aux ensembles pseudo-étiquetés de l'épisode précédent. Les trois modèles étiquettent ensuite le texte généré par la méthode de génération. Si deux modèles ou plus parviennent à un accord, les nouvelles données sont conservées avec les pseudo-étiquettes. Ces données pseudo-étiquetées sont envoyées soit au paquet du modèle qui n'est pas d'accord si deux modèles sont d'accord, soit à l'un des ensembles au hasard si les trois modèles sont d'accord. Si aucun accord n'est trouvé, les données sont rejetées. Nous continuons à générer de nouveaux exemples jusqu'à ce que nous atteignions la quantité attendue ou un nombre maximal d'opérations de génération autorisé que nous avons fixé à 10 000 pour éviter de générer indéfiniment. En pratique, cette limite n'est jamais atteinte. Chaque méthode de génération génère cinq nouvelles phrases pour chaque phrase d'entrée.

7 Résultats

Les résultats de la baseline et les Δ entre la méthode présentée et la baseline sont présentés dans le tableau 2. Par souci de place, nous n'avons pas indiqué les résultats obtenus sur des sous-ensembles de taille 1000, mais ils conservent les tendances observés. Les résultats de la topline sont bien supérieurs aux résultats du supervisé. Les résultats du supervisé vont de 60,0 F_1 avec des sous-ensembles de taille 50 à 84,6 F_1 avec une taille de 500 pour CoNLL avec des gains décroissant de 8,6 à 3,8 pour la topline. Pour I2B2, les résultats de la baseline sont plus petits que ceux de CoNLL, allant de 39,3 à 77,6 F_1 score avec des gains. Les gains de I2B2 vont de 8,9 à la taille 100 à 5,4 à la taille 500 et sont légèrement inférieurs pour 50 phrases, avec 8,3.

Les résultats décrits dans cette section se trouvent dans le Tableau 2 pour les ressources faibles, et dans le Tableau 4 pour les corpus complets. Pour le tri-apprentissage, chaque méthode affiche des gains positifs à chaque taille de sous-ensemble et certaines dépassent même la topline. C'est le cas

Crp.	CoNLL				I2B2			
Taille	50	100	250	500	50	100	250	500
bsl.	60,0 2,9	70,5 4,7	81,3 1,7	84,6 1,2	39,3 3,1	50,4 1,4	63,8 0,7	71,5 1,0
Δtl	+8,6 2,5	+5,6 2,0	+4,2 0,7	+3,8 1,0	+8,3 2,2	+8,9 2,2	+6,7 1,2	+5,4 0,7
	Tri-apprentissage							
Δfl	+8,9 2,6	+4,4 2,8	+3,1 0,4	+2,8 1,0	+6,3 2,6	+4,5 2,6	+4,2 1,6	+3,6 1,0
Δcp	+7,8 1,9	+4,7 4,0	+2,7 1,0	+2,9 0,7	+6,3 1,6	+5,6 2,8	+4,9 1,5	+4,3 1,7
Δm	+9,0 2,1	+4,1 2,6	+2,8 1,1	+2,6 0,8	+10,4 4,5	+7,4 2,9	+7,1 1,7	+4,6 1,0
Δcr	+7,3 1,9	+2,9 2,3	+1,2 0,4	+1,7 0,7	+5,6 1,5	+7,7 3,1	+6,0 1,2	+4,2 1,0
Δcb	+9,3 2,7	+4,6 2,6	+3,4 1,0	+2,9 1,0	+7,7 1,5	+6,4 1,7	+5,7 1,9	+4,4 1,2
	Auto-apprentissage							
Δfl	+4,8 2,9	+1,4 3,6	-0,1 1,1	+0,6 1,0	+0,8 1,7	+0,2 2,3	-0,7 1,3	-0,2 0,5
Δcp	+3,9 2,7	+0,8 2,0	-0,4 1,0	+0,6 0,9	-0,5 1,6	+0,3 2,9	-1,4 0,8	-0,9 1,8
Δm	+6,0 2,9	+2,5 2,8	+0,7 0,6	+0,8 0,4	+4,5 2,6	+2,9 1,8	+3,0 1,1	+2,0 1,1
Δcr	+4,3 2,0	-0,5 4,0	-1,1 1,2	-0,2 0,7	+3,6 1,8	+3,6 2,5	+3,9 0,9	+2,2 0,7
Δcb	+4,9 3,5	+1,4 3,1	-0,4 0,4	+0,9 1,2	+1,6 1,0	+0,7 1,7	-0,1 1,2	+0,6 1,6

TABLE 2 – Δ de F_1 des différentes méthodes comparées à l'apprentissage supervisé. Les écarts type sur les cinq seeds utilisées sont rapportés en plus petit. Bsl. et tl sont les résultats obtenus avec la baseline et la topline. Les abréviations fl, cp, m, cr, et cb décrivent la génération de phrase suivante, la complétion, le remplacement de mentions, le remplacement de contexte, et l'expérience combinée.

des méthodes de génération de phrase suivante et de remplacement de mention pour CoNLL avec des gains moyens de 8,9 et 9,0 F_1. C'est également le cas pour la méthode de remplacement de mention pour I2B2 avec des gains moyens de 10,4 et 7,1 F_1 score aux tailles 50 et 250, respectivement. Les gains suivent les mêmes tendances que la topline, avec des rendements décroissants plus la taille du sous-ensemble est élevée. Pour CoNLL, le remplacement de contexte est la méthode de génération la moins performante dans l'ensemble. Pour I2B2, avec un sous-ensemble d'apprentissage de taille 50, le remplacement de contexte est la méthode la moins performante. Néanmoins, avec des quantités de données plus importantes, elle devient la deuxième méthode la plus performante, et la génération de phrase suivante devient la moins performante. Le remplacement de mentions est la meilleure méthode de génération individuelle sur I2B2, avec les meilleurs ou les seconds meilleurs résultats pour chaque taille de sous-ensemble. Dans l'ensemble, les méthodes fondées sur GPT-2 donnent des résultats inférieurs aux méthodes fondées sur T5v1.1 sur I2B2.

Les gains avec l'auto-apprentissage sont inférieurs à leurs homologues du tri-apprentissage, avec quelques résultats négatifs. La génération de mentions reste la meilleure méthode individuelle, car c'est la meilleure méthode pour CoNLL et elle est en concurrence avec le remplacement de contexte en fonction de la taille du sous-ensemble pour I2B2.

La méthode combinée est réalisée en échantillonnant la méthode de génération parmi les méthodes disponibles chaque fois que nous voulons générer. La méthode présente deux comportements différents sur les deux corpus. Pour CoNLL, la méthode de génération combinée est la meilleure méthode de génération. Pour I2B2, la méthode de génération combinée s'apparente davantage à une méthode

	Suivante	Complétion	Mentions	Contexte	Combiné
Suivante		0.68	0.96	0.81	0.92
Complétion	**0.35**		0.79	0.63	0.69
Mention	**0.05**	**0.22**		**0.20**	**0.26**
Contexte	**0.21**	**0.38**	0.82		0.51
Combiné	**0.11**	**0.33**	0.76	0.51	

TABLE 3 – Score de l'Almost Stochastic Dominance (Dror *et al.*, 2019) du tri-apprentissage à faible ressources avec indice de significativité de 0.05. Nous testons si la méthode de la ligne A est dominante sur la méthode de la colonne B. A est dominante sur B si le score est inférieur à 0.5.

	baseline	complétion	suivante	mention	contexte	combiné
CoNLL	92,0	92,1	91,8	92,2	92,2	**92,3**
I2B2	86,6	87,5	**88,0**	87,7	87,6	87,5

TABLE 4 – F_1 scores des méthodes de génération sur les corpus complets. Une seed a été utilisée.

moyenne. Elle est par ailleurs inférieure ou égale aux méthodes de remplacement, à l'exception du remplacement de contexte sur le plus petit sous-ensemble. La génération combinée est supérieure aux méthodes de modélisation de gauche à droite pour I2B2. Dans l'ensemble, la méthode combinée a un facteur de corrélation de Pearson plus élevé pour les méthodes de suivi et de complétion avec 0,98 et 0,97 que pour la mention et le remplacement de contexte avec 0,92 et 0,80, respectivement. Ces coefficients ont été calculés avec les résultats rapportés dans le Tableau 2 et les résultats sur les sous-ensembles de taille 1000. Nous avons utilisé le test d'Almost Stochastic Dominance(ASD) (Dror *et al.*, 2019) pour déterminer quelle est la meilleure méthode de génération sur nos sous-ensembles de données avec un indice de significativité de 0.05. D'après les résultats montrés Table 3, le remplacement de mentions est dominant sur toutes les autres méthodes et est donc la méthode à privilégier. La complétion n'est dominante que sur la génération de phrase suivante, qui ne domine aucune méthode. Le remplacement de contexte et la méthode combiné sont équivalentes.

À but indicatif, nous avons calculé des résultats pour le tri-apprentissage génératif sur les corpus de taille complète Table 4. Pour CoNLL, nos meilleurs résultats sont avec la méthode combinée avec une amélioration de 0,3. Nous obtenons cependant une détérioration de 0,2 avec la génération de phrase suivante. Globalement, sur CoNLL grandeur nature, nous obtenons des résultats positifs mais non significatifs. Pour I2B2, toutes les méthodes augmentent significativement le score F_1 avec des améliorations allant de 0,9 pour la complétion et la méthode combinée à 1,4 pour le suivi.

8 Conclusion

Dans cet article, nous avons proposé et évalué une méthode permettant d'allier génération de données et semi-supervision dans le but de réduire la dépendance aux données pour l'apprentissage étiqueté. Nous avons montré que notre méthode fonctionnait bien avec le tri-apprentissage, et moins bien avec l'auto-apprentissage. Nous avons testé différentes méthodes de génération découlant de la modélisation de langue pour générer des données. Parmi ces méthodes, le remplacement de mention

par T5 est la méthode ayant montré les meilleurs résultats sur nos deux corpus de test et domine les autres méthodes présentées d'après le test ASD. Cependant, utiliser des données naturelles de la tâche comme données non-annotées produit presque toujours de meilleurs résultats. Il reste cependant beaucoup de questions auxquelles nous souhaiterions répondre dans de futurs travaux. Nous avons montré que la génération permettait de transformer une tâche supervisée en tâche semi-supervisée avec des gains de performances. Il peut être intéressant d'analyser l'impact des données produites de cette manière dans un cadre supervisé.

9 Remerciements

Ces travaux ont été financé sur le projet PSPC AIDA : 2019-PSPC-09 par BPI-France. Ces travaux ont bénéficié d'un accès aux ressources HPC de l'IDRIS sous l'allocation 2021-AD011013018 faite par GENCI. Ces travaux ont aussi bénéficié d'un accès aux ressources HPC de Saclay-AI au travers de la machine Lab-IA.

Références

BLUM A. & MITCHELL T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, p. 92–100.

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*.

DING B., LIU L., BING L., KRUENGKRAI C., NGUYEN T. H., JOTY S., SI L. & MIAO C. (2020). Daga : Data augmentation with a generation approach for low-resource tagging tasks. *arXiv preprint arXiv :2011.01549*.

DOPIERRE T., GRAVIER C. & LOGERAIS W. (2021). Protaugment : Intent detection meta-learning through unsupervised diverse paraphrasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 2454–2466.

DROR R., SHLOMOV S. & REICHART R. (2019). Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)* : Association for Computational Linguistics.

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.

NAKAYAMA H. (2018). seqeval : A python framework for sequence labeling evaluation. Software available from https ://github.com/chakki-works/seqeval.

NEURAZ A., LLANOS L. C., BURGUN A. & ROSSET S. (2018). Natural language understanding for task oriented dialog in the biomedical domain in a low resources context. *CoRR*, **abs/1811.09417**.

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

RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, **21**, 1–67.

RAMSHAW L. & MARCUS M. (1995). Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

RUDER S. (2019). *Neural Transfer Learning for Natural Language Processing*. Thèse de doctorat, National University of Ireland, Galway.

RUDER S. & PLANK B. (2018). Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1044–1054.

SANG E. T. K. & DE MEULDER F. (2003). Introduction to the conll-2003 shared task : Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, p. 142–147.

SENNRICH R., HADDOW B. & BIRCH A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 86–96.

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.

SØGAARD A. (2010). Simple semi-supervised training of part-of-speech taggers. In *Proceedings of the ACL 2010 Conference Short Papers*, p. 205–208.

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 : JAMIA*, **18**(5), 552.

VAN ENGELEN J. E. & HOOS H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, **109**(2), 373–440.

WANG X., JIANG Y., BACH N., WANG T., HUANG Z., HUANG F. & TU K. (2021a). Automated concatenation of embeddings for structured prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 2643–2660.

WANG Y., MUKHERJEE S., CHU H., TU Y., WU M., GAO J. & AWADALLAH A. H. (2021b). Meta self-training for few-shot neural sequence labeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, p. 1737–1747.

WEI J. & ZOU K. (2019). EDA : Easy data augmentation techniques for boosting performance on text classification tasks. 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. 6382–6388, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1670.

YAROWSKY D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, p. 189–196.

ZHOU Z.-H. & LI M. (2005). Tri-training : Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, **17**(11), 1529–1541.