Apprentissage actif multi-labels pour des architectures transformers

Maxime Arens

RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à l'Université Toulouse III - Paul Sabatier

Apprentissage actif multi-labels pour des architectures transformers

---

Thèse présentée et soutenue, le 30 mai 2024 par

Maxime ARENS

École doctorale
EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Spécialité
Informatique et Télécommunications

Unité de recherche
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Mohand BOUGHANEM et José MORENO

Composition du jury
M. Patrice BELLOT, Rapporteur, Aix-Marseille Université
M. Philippe MULHEM, Rapporteur, CNRS Alpes
Mme Haïfa ZARGAYOUNA, Examinatrice, Université Sorbonne Paris Nord
M. Guillaume CABANAC, Examinateur, Université Toulouse III - Paul Sabatier
M. Mohand BOUGHANEM, Directeur de thèse, Université Toulouse III - Paul Sabatier
M. José MORENO, Co-directeur de thèse, Université Toulouse III - Paul Sabatier

Membres invités
Mme Lucile Callebert, Synapse Développement
M. Charles Teissèdre, ODIA

Maxime Arens : *Apprentissage actif multi-labels pour des architectures de type transformers,* 2024

RÉSUMÉ

L'annotation des données est cruciale pour l'apprentissage automatique, notamment dans les domaines techniques, où la qualité et la quantité des données annotées affectent significativement l'efficacité des modèles entraînés. L'utilisation de personnel humain est coûteuse, surtout lors de l'annotation pour la classification multi-labels, les instances pouvant être associées à plusieurs labels. L'apprentissage actif (AA) vise à réduire les coûts d'annotation en sélectionnant intelligemment des instances pour l'annotation, plutôt que de les annoter de manière aléatoire. L'attention récente portée aux transformateurs a mis en lumière le potentiel de l'AA dans ce contexte. De plus, le mécanisme de *fine-tuning*, où seules quelques données annotées sont utilisées pour entraîner le modèle sur une nouvelle tâche, est parfaitement en accord avec l'objectif de l'AA de sélection des meilleures données à annoter. Nous étudions donc l'utilisation de l'AA dans le contexte des transformers pour la tâche de classification multi-labels. Hors, la plupart des stratégies AA, lorsqu'elles sont appliquées à ces modèles, conduisent à des temps de calcul excessifs, ce qui empêche leurs utilisations au cours d'une interaction humain-machine en temps réel. Afin de pallier ce problème, nous utilisons des stratégies d'AA plus rapides, basées sur l'incertitude. D'abord, nous mettons l'accent sur l'application de 6 stratégies d'AA différentes sur deux modèles transformers. Nos travaux mettent en évidence qu'un certain nombre de stratégies basées sur l'incertitude ne surpassent pas l'échantillonnage aléatoire lorsqu'elles sont appliquées aux modèles transformers. Afin d'évaluer si ces résultats sont dûs à un biais des stratégies basées sur l'incertitude, une approche de pré-clustering est introduite pour ajouter de la diversité dans la sélection des instances. Enfin, nous nous penchons sur les défis pratiques de la mise en œuvre de l'AA dans des contextes industriels. Notamment, l'écart entre les cycles de l'AA laisse du temps inutilisé aux annotateurs. Pour résoudre ce problème, nous étudions des méthodes alternatives de sélection d'instances, visant à maximiser l'efficacité de l'annotation en s'intégrant de manière transparente au processus de l'AA. Nous commençons par adapter deux méthodes existantes aux transformers, en utilisant respectivement un échantillonnage aléatoire et des informations de cycle d'AA périmées. Ensuite, nous proposons notre méthode novatrice basée sur l'annotation des instances pour rééquilibrer la distribution des labels. Notre approche atténue les biais, améliore les performances du modèle (jusqu'à 23% d'amélioration sur le score F1), limite les disparités dépendantes de la stratégie (diminution de près de 50% de l'écart-type) et réduit le déséquilibre des libellés (dimi-

nution de 30% du ratio moyen de déséquilibre). Nos travaux ravivent ainsi la promesse de l'AA en montrant que son intégration adaptée dans un projet d'annotation se traduit par une amélioration des performances du modèle final entraîné.

ABSTRACT

Data annotation is crucial for machine learning, especially in technical domains, where the quality and quantity of annotated data significantly impact the effectiveness of trained models. Human annotation is costly, particularly for multi-label classification tasks, as instances may be associated with multiple labels.

Active Learning (AL) aims to reduce annotation costs by intelligently selecting instances for annotation, rather than annotating randomly. Recent attention on transformers has highlighted the potential of AL in this context. Moreover, the fine-tuning mechanism, where only a few annotated data points are used to train the model for a new task, aligns well with the goal of AL to select the best data for annotation.

We investigate the use of AL in the context of transformers for multi-label classification tasks. However, most AL strategies, when applied to these models, lead to excessive computational time, hindering their use in real-time human-machine interaction. To address this issue, we employ faster AL strategies based on uncertainty.

First, we focus on applying six different AL strategies to two transformer models. Our work highlights that several uncertainty-based strategies do not outperform random sampling when applied to transformer models. To evaluate if these results stem from a bias in uncertainty-based strategies, we introduce a pre-clustering approach to add diversity to instance selection.

Lastly, we tackle the practical challenges of implementing AL in industrial contexts. Particularly, the gap between AL cycles leaves idle time for annotators. To resolve this, we explore alternative instance selection methods aiming to maximize annotation efficiency by seamlessly integrating with the AL process. We start by adapting two existing methods to transformers, using random sampling and outdated AL cycle information, respectively. Then, we propose our innovative method based on instance annotation to rebalance label distribution. Our approach mitigates biases, improves model performance (up to 23% improvement on the F1 score), reduces strategy-dependent disparities (nearly 50% decrease in standard deviation), and decreases label imbalance (30% decrease in the mean imbalance ratio).

Our work thus revives the promise of AL by demonstrating that its adapted integration into an annotation project results in improved performance of the final trained model.

PUBLICATIONS

Article publié dans une conférence nationale
Arens Maxime, Teissèdre Charles, Callebert Lucile, Moreno José G.et Boughanem Mohand. *Impact de l'apprentissage multi-labels actif appliqué aux transformers.* (Article long) Dans : Actes de CORIA-TALN 2023. Actes de la 18e Conférence en Re- cherche d'Information et Applications (CORIA).

Article publié dans une conférence internationale
Arens Maxime, Callebert Lucile, Moreno José G.et Boughanem Mohand. *Rebalancing Label Distribution while Eliminating Inherent Waiting Time in Multi Label Active Learning applied to Transformers.* (Article long) Dans : The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) (*A paraître.*)

Great hope can come from small sacrifices.

— George Lucas

REMERCIEMENTS

J'écris ces remerciements au lendemain de ma soutenance de thèse, que j'ai défendue avec succès. La raison pour laquelle je n'ai pas rédigé cette partie plus tôt est que je souhaitais dédier cette thèse à la mémoire de deux personnes qui me sont très chères. Cependant, je n'étais pas certain que mon travail serait à la hauteur de leur mémoire. Grâce aux retours du jury et au recul que j'ai pu prendre, je réalise maintenant que je suis fier de ce manuscrit, du travail accompli et des épreuves surmontées.

Je dédie donc cette thèse à mes deux grands-parents, André Arens et Josiane Béduer Eymard, qui ont été de grands modèles pour moi. Ils m'ont toujours poussé à faire de mon mieux, à être la meilleure version de moi-même et surtout à prendre soin de mes proches et de mon prochain.

Je souhaite adresser mes premiers remerciements à ma conjointe, Charlotte Lagoutte-Druez, mon plus grand soutien et la première correctrice de mes étourderies. Ses épaules ont également porté une partie de la charge mentale de cette thèse. Son amour et sa présence ont été une île de stabilité dans l'océan parfois turbulent de ces trois dernières années.

Je remercie ensuite Jesús Lovón, mon ami et collègue de bureau au laboratoire, qui a réalisé sa thèse en même temps que moi. Ses conseils, son expérience, mais aussi sa présence et son amitié, font partie des plus belles choses que je retire de cette aventure.

Je remercie également l'entreprise Synapse-Développement de m'avoir fait confiance pour cette thèse, et plus particulièrement mes différents collègues et amis là-bas, qui m'ont permis de garder un lien avec le monde industriel pendant toutes ces années. Leur soutien m'a souvent offert une bouffée d'air frais loin de mes expériences et rédactions. Une mention particulière à mon ami Vincent Erb, qui m'a accordé son oreille attentive et sa patience durant mes moments de ronchonneries.

Un grand merci également à tous mes proches, ma famille et mes amis, qui ont toujours cru en moi et m'ont soutenu de manière constante. Vous avez été présents jusqu'à la soutenance, où votre présence a compté énormément pour moi, et après, où votre émotion a rendu la célébration d'autant plus importante.

Merci à mon jury, dont la bienveillance a contribué à faire de ma soutenance un moment dont je suis très fier. Merci de m'avoir reconnu et accueilli en tant que pair. Merci au président du jury, Guillaume

Cabanac, d'avoir présidé de manière magistrale et d'avoir rendu ce moment d'autant plus mémorable. Grâce à cette expérience, je considère désormais des chemins que je n'avais pas envisagés jusqu'à présent.

Enfin, merci à mon équipe d'encadrement de thèse. Merci à Mohand Boughanem pour avoir dirigé ma thèse, tes connaissances et ta tranquillité ont été précieuses durant ces années. Merci à José Moreno pour ton investissement dans ce projet ; j'aime à penser que nous avons appris l'un de l'autre. Merci à Lucile Callebert pour toutes tes relectures, mais surtout pour ton soutien juste et inflexible. Merci à Charles Teissedre pour avoir cru en moi du début jusqu'à la fin, d'avoir été à l'origine de ce sujet et d'avoir endossé le rôle de mentor. Cette thèse ne serait pas la même sans vous.

TABLE DES MATIÈRES

1 INTRODUCTION 1
- 1.1 Contexte 1
- 1.2 Problématique 4
- 1.3 Contributions 6
- 1.4 Organisation de la thèse 7

1 SYNTHÈSE DES TRAVAUX DE L'ÉTAT DE L'ART 9

2 APPRENTISSAGE ACTIF MULTI-LABELS 11
- 2.1 Apprentissage actif 11
  - 2.1.1 Notations 15
  - 2.1.2 Stratégies basées sur l'informativité 15
    - 2.1.2.1 Stratégies basées sur l'incertitude 15
    - 2.1.2.2 Stratégies basées sur le désaccord 17
    - 2.1.2.3 Stratégies basées sur la prédiction de performance 19
  - 2.1.3 Stratégies basées sur la représentativité 19
    - 2.1.3.1 Stratégies basées sur la densité 20
    - 2.1.3.2 Stratégies discriminantes 20
    - 2.1.3.3 Stratégies basées sur la diversité 21
  - 2.1.4 Apprentissage actif par lot de données 21
  - 2.1.5 Combinaisons d'approches 23
  - 2.1.6 L'humain au centre du processus 24
- 2.2 Classification multi-labels 27
  - 2.2.1 Catégorisation d'approches pour solutionner la problème de classification multi-labels 28
  - 2.2.2 Particularités de la tâche de classification multi-labels 29
    - 2.2.2.1 Fonction d'activation multi-labels 29
    - 2.2.2.2 Déséquilibre entre labels 30
    - 2.2.2.3 Corrélation entre labels 32
    - 2.2.2.4 Labels manquants 34
  - 2.2.3 Stratégies d'apprentissage multi-labels étudiées 36
- 2.3 Apprentissage actif sur la tâche de classification multi-labels (AAML) 39
  - 2.3.1 AAML : informativité vs représentativité 40
  - 2.3.2 AAML : instance individuelle vs lot de données 42
  - 2.3.3 AAML : instance vs paire instance-label 43
- 2.4 Discussion et conclusion 43

3 APPRENTISSAGE PROFOND ACTIF 45
- 3.1 Apprentissage actif avec des réseaux neuronaux 45

3.1.1 Catégorisation et stratégies d'apprentissage actif avec des réseaux neuronaux 46
3.1.1.1 Stratégies basées sur les données 47
3.1.1.2 Stratégies basées sur le modèle 48
3.1.1.3 Stratégies basées sur les prédictions 49
3.2 Apprentissage actif avec des transformers 50
3.2.1 Application de l'apprentissage actif avec des transformers 51
3.2.2 Comparaison avec des modèles de classification antérieurs 52
3.2.3 Application à la tâche de classification multi-label 53
3.3 Mitiger l'impact de la taille des modèles 54
3.3.1 Utilisation de la distillation 55
3.3.2 Temps d'attente durant cycle d'apprentissage actif 57
3.4 Paradigmes voisins de l'apprentissage actif 58
3.4.1 Apprentissage en continu, tout-au-long de sa vie et en ligne 58
3.4.2 Apprentissage suivant un curriculum 59
3.4.3 Apprentissage actif dans l'éducation 60
3.5 Discussion et conclusion 60

2 CONTRIBUTIONS 62

4 AA MULTI-LABELS BASÉ SUR L'INCERTITUDE APPLIQUÉ AUX TRANSFORMERS 63
4.1 Contexte et motivations 63
4.1.1 Travaux connexes 64
4.2 Définition du problème et contexte expérimental 65
4.2.1 Méthodologie 65
4.2.2 Mise en œuvre 67
4.2.3 Jeux de données 67
4.2.3.1 Jigsaw toxic comment classification (Jigsaw_Toxic) 68
4.2.3.2 Go_Emotions 68
4.2.3.3 EUR_Lex57K (EUR_Lex) 69
4.2.3.4 UNFAIR - Terms of Services (UNFAIR-ToS) 71
4.2.4 Références et modèles 71
4.2.4.1 Oracle 71
4.2.4.2 Modèles et références 72
4.2.4.3 Paramètres et détails d'implémentation 72
4.2.5 Évaluation 73
4.2.6 Résultats et analyses 73
4.2.7 Bilan 77
4.3 Réduction de la redondance des instances sélectionnées 80

4.3.1 Architecture pré-clustering 81
4.3.2 Résultats et analyses 81
4.4 Discussion et conclusion 82
5 VALORISATION DU TEMPS D'ATTENTE DES ANNOTATEURS DURANT L'AA 86
5.1 Contexte et motivations 86
5.1.1 Travaux connexes 87
5.1.2 Mise en oeuvre d'un outil d'annotation en entreprise 89
5.1.3 Retour d'expérience : mise en pratique de l'AA 92
5.2 Définition du problème 94
5.3 Contexte expérimental 95
5.3.1 Mise en oeuvre 96
5.3.1.1 Référence 96
5.3.1.2 Evaluation 96
5.4 Optimisation du temps d'attente de l'annotateur 97
5.4.1 Lot d'entraînement hybride : apprentissage actif et échantillonage alternatif 97
5.4.1.1 Apprentissage actif et annotation aléatoire 98
5.4.1.2 Apprentissage actif et scores d'incertitudes périmés 98
5.4.1.3 Apprentissage actif et équilibrage des labels annotés 100
5.4.2 Résultats et Analyses 101
5.4.2.1 Variations d'alpha 106
5.5 Discussion et conclusion 109

3 CONCLUSION 111
6 CONCLUSION 112

BIBLIOGRAPHIE 117

4 ANNEXES 147
A ANNEXES 148
A.1 Brèves définitions de réseaux neuronaux 148
A.2 Définition des transformers 153

TABLE DES FIGURES

Figure 1 Premier niveau de catégorisation hiérarchique des stratégies d'AA. Les catégories en gris seront moins explorées dans cette thèse. 11
Figure 2 Schéma du cycle d'AA basé sur un réservoir de données. 13
Figure 3 Deuxième et troisième niveau de catégorisation hiérarchique des stratégies d'AA. Les catégories en gris seront moins explorées dans cette thèse. 14
Figure 4 Exemple de sélections d'instances par informativité ou par représentativité dans un problème de classification binaire 23
Figure 5 Schéma explicatif des différents types de classification. 28
Figure 6 Exemple d'une matrice de corrélation 33
Figure 7 Légende des symboles de la Figure 5 prenant en compte l'absence de labels 34
Figure 8 Exemple de diverses mesures d'incertitude dans l'AAML. Le ① correspond à l'instance sélectionnée par apprentissage actif. 41
Figure 9 Catégorisation des stratégies d'AA par phase de prise de décision. 47
Figure 10 Architecture du modèle encodeur + réseau neuronal pour classification multi-labels. 54
Figure 11 Visualisation de la croissance en nombre de paramètres des modèles de langues. [1] 55
Figure 12 Nombre d'instances par label dans le jeu de données Jigsaw_Toxic. 69
Figure 13 Nombre d'instances par label dans le jeu de données Go_Emotions. [2] 69
Figure 14 Distribution de la fréquence des libellés EUR_Lex [102]. Tous les labels sont répartis en trois groupes (dans chaque groupe, F indique la fréquence du libellé, P est la proportion que les libellés dans le groupe représentent par rapport à l'ensemble complet des libellés). 70
Figure 15 Nombre d'instances par label dans le jeu de données UNFAIR-ToS. 71
Figure 16 Performances M_{iF1} suivant les différentes stratégies d'AA pour chaque transformers 75
Figure 17 *Sim_batch* suivant les différentes stratégies d'AA pour chaque transformers 79

Figure 18 Performances M_{iF1} suivant les différentes stratégies d'AA pour chaque transformers après pré-clustering 83
Figure 19 Exemple d'interface utilisateur d'annotation avec Argilla.io. 90
Figure 20 Cycle classique d'AA sans méthode d'échantillonnage parallèle pour éviter le temps d'attente de l'annotateur. Cette figure met en évidence que lorsque le modèle travaille l'annotateur attend et lorsque le modèle attend l'annotateur travaille. La longueur des flèches "Annotating" et "Waiting" ne sont pas proportionnelles avec les durées réelles de ces actions. 94
Figure 21 Cycle d'AA avec une méthode d'échantillonnage alternative pour éviter le temps d'attente de l'annotateur. Nous précisons que la Figure n'est pas à l'échelle du temps. 95
Figure 22 Composition du lot d'entraînement à travers les cycles d'AA avec une méthode d'échantillonnage alternative pour éviter le temps d'attente de l'annotateur. 99
Figure 23 Performances M_{iF1} suivant les différentes stratégies d'AA pour chaque transformers et chaque méthode alternative d'échantillonage 105
Figure 24 Différentes allures de l'impact sur les performances du modèles en fonction du paramètre alpha. 106
Figure 25 Performances M_{iF1} suivant les différentes stratégies d'AA pour chaque transformers en faisant varier le paramètre alpha 107
Figure 26 Schéma d'un perceptron. 149
Figure 27 Schéma d'un perceptron à multiple couches. 149
Figure 28 Exemple d'un réseau neuronal convolutif [66]. 150
Figure 29 Schéma de deux vues "de coté" d'un réseau de neurones récurrents. 152
Figure 30 Schéma d'une cellume *long short-term memory*. 153
Figure 31 Exemple du procédé d'auto-attention. Le mécanisme d'attention a remarqué que les tokens "Un" et "ami" étaient pertinents pour contextualisé sur le token "malade". 155

Figure 32 Exemple du procédé d'auto-attention multi-tête. L'une des têtes d'attention se focalise sur "Un" et "ami", tandis que l'autre sur "mangé" et "huîtres", pour contextualiser malade. Cela parait pertinent puisque "malade" réfère à la fois à l'individu et à son état, chaque tête se concentrant sur un sens du token. 156
Figure 33 Exemple du fonctionnement d'un transformers sur la tâche de traduction automatique. 157

LISTE DES TABLEAUX

Table 1 Caractéristiques des jeux de données 68
Table 2 Pourcentage des jeux de données annotés, associés aux pourcentages de performances atteintes par les stratégies par rapport à une supervision complète. 76
Table 3 "M_{iF1}/Sim_batch", obtenu suivant chaque stratégies d'AA. Les résultats en rouge indiquent un score M_{iF1} inférieur à l'échantillonnage aléatoire, les résultats en bleu indiquent un score M_{iF1} supérieur à l'échantillonnage aléatoire. 78
Table 4 Corrélation de Pearson entre M_{iF1} et *Sim_batch* 80
Table 5 "M_{iF1} sans pré-clustering/M_{iF1} avec pré-clustering", obtenu suivant chaque stratégies d'AA. Les résultats en rouge indiquent un score M_{iF1} sans pré-clustering supérieur ou égale à avec pré-clustering , les résultats en bleu indiquent un score M_{iF1} sans pré-clustering inférieur à avec pré-clustering. 84
Table 6 Résultats expérimentaux (M_{iF1}) du paramètre Classic-AL et de trois méthodes d'échantillonnage parallèle à l'apprentissage actif (AL). Avec 'dB' pour distilBERT, 'dR' pour distilRoBERTa et 'Std-dev' pour l'écart-type. Les meilleures valeurs pour le modèle/la stratégie sont en gras. 102
Table 7 Moyenne en pourcentage de la différence de M_{iF1} par rapport au paramètre Classic-AL par méthode, modèle et ensemble de données (une valeur positive indique une amélioration), 'dB' pour distilBERT et 'dR' pour distilRoBERTa. Les meilleures valeurs sont en gras. 103

Table 8 Moyenne en pourcentage de la différence de M_{iF1} par rapport au paramètre Classic-AL par méthode, modèle et stratégie d'apprentissage actif (une valeur positive indique une amélioration). Les meilleures valeurs sont en gras. 103
Table 9 Moyenne du ratio moyen d'imbalance (MeanIR) par méthode et modèle (une valeur élevée indique un déséquilibre élevé des labels). Les meilleures valeurs sont en gras. 104
Table 10 Résultats expérimentaux (M_{iF1}) sur distilBERT en faisant varier le paramètre alpha. Les meilleures valeurs pour la stratégie/alpha sont en gras. 108

LISTINGS

ACRONYMES

AA apprentissage actif
IA intelligence artificielle
HITL *human-in-the-loop*

1 INTRODUCTION

1.1 CONTEXTE

L'apprentissage automatique, un pilier de l'intelligence artificielle (IA), explore l'idée que les machines peuvent apprendre grâce à de l'expérience. Ce domaine englobe une vaste gamme d'algorithmes et de modèles qui évoluent en plus ou moins grande autonomie, en tirant des leçons de données d'entraînement ou de leur propre activité. En se libérant de la nécessité d'une programmation explicite, l'apprentissage automatique permet aux machines d'accomplir diverses tâches avec une efficacité qui défie souvent les capacités humaines [158]. L'essor de l'apprentissage automatique a été profondément alimenté par l'accumulation de volume de données de plus en plus important. Dans des secteurs tels que la finance, la médecine et les médias, les données abondent et recèlent d'informations précieuses. Cependant, la croissance exponentielle des données en volume et en complexité a mis au défi les approches traditionnelles d'analyse de données. Face à cet enjeu, l'apprentissage automatique s'est imposé comme un moyen de distiller la signification de ces masses d'informations en extrayant les éléments essentiels. L'apprentissage automatique permet de former des modèles capables de prédire des tendances, de prendre des décisions et d'automatiser des processus complexes. L'apprentissage automatique n'a pas seulement profité de l'abondance de données, mais aussi de progrès théoriques significatifs. L'apprentissage profond, une sous-discipline de l'apprentissage automatique, implique l'utilisation de réseaux de neurones profonds permettant de capturer des relations subtiles dans les données, aboutissant à des avancées majeures dans de nombreux domaines, notamment en reconnaissance d'images, en traitement automatique du langage naturel et en génération de contenu. L'apprentissage par renforcement, une technique d'apprentissage automatique où un agent interagit avec un environnement, réalise des actions, et reçoit des récompenses en retour, a connu un succès remarquable dans les jeux et la robotique.

Une autre technique de l'apprentissage automatique, l'apprentissage supervisé, se distingue comme l'une des plus prédominantes et des plus pratiques. Ici, les modèles sont nourris d'exemples labélisés (ou étiquetés), où les données d'entrée sont associées à des sorties correspondantes. En apprenant de ces exemples, les modèles sont en mesure de généraliser leurs connaissances pour effectuer des prédictions sur de nouvelles données. Les performances de ces modèles

dépendent largement des quantités de données annotées. De plus, la relation entre les performances d'un modèle et la quantité de données d'entraînement ne suit pas une progression linéaire. En effet, les premières étapes d'ajout de données se traduisent par une amélioration significative des performances. Toutefois, à mesure que la quantité de données augmente, les gains associés deviennent moins marqués, conduisant éventuellement à un point de saturation où l'ajout de nouvelles données n'engendre que des améliorations marginales. Cependant, la capacité d'un modèle à tirer parti d'une quantité croissante de données émerge comme un critère distinctif entre différentes générations de modèles, caractérisées par des écarts significatifs de performances. Cela souligne le rôle essentiel de l'annotation des données dans le processus d'apprentissage supervisé [67]. L'acquisition de données annotées est un processus que l'on peut souvent ramener à deux étapes : se *procurer* des données brutes (non-annotées) et *annoter* ces données. La difficulté d'acquisition de données brutes en terme de quantité n'est pas toujours un enjeu important dans le monde industriel, surtout pour les géants de l'informatique [77]. L'annotation par contre, est un enjeu à la fois économique et qualitatif. Bien que les méthodes dites d'externalisation (*crowdsourcing*) permettent d'accéder à de nombreux annotateurs à relativement faible coût [92], dans de nombreux domaines, cette pratique est impossible puisque le procédé d'annotation nécessite une expertise particulière. Dans le domaine légal, médical ou tout simplement quand les données ne peuvent être externalisées pour des raisons de confidentialité ou de sécurité, le recours à des experts est alors nécessaire [248]. Le temps de ces experts est précieux, autrement dit, l'annotation peut devenir une étape onéreuse et limitante. C'est d'autant plus vrai si la tâche d'annotation est complexe. De plus, tous les experts ne sont pas égaux, la qualité d'annotation d'un expert peut directement impacter la performance du modèle final en introduisant des erreurs ou des biais dans les données d'entraînements.

Les fulgurantes avancées de l'IA ont engendré des préoccupations éthiques profondes. L'appréhension persistante selon laquelle les machines pourraient surpasser les capacités intellectuelles et le pouvoir des humains demeure ancrée dans l'imaginaire collectif. Cette inquiétude, amplifiée par la culture populaire et les médias qui dépeignent l'IA comme une menace dans des productions cinématographiques telles que "The Terminator" [1] ou la série "Black Mirror" [2], reste d'actualité. Un autre motif de méfiance découle du manque de transparence et de responsabilité de certains systèmes d'IA, entravant la compréhension et la remise en question des décisions prises par ces modèles. Récemment, les modèles de génération d'images (DALL-E [170], Sta-

1. https://w.wiki/88hm
2. https://w.wiki/88hk)

bleDiffusion [180], MidJourney [3]) et de texte (GPT [24], Bard [4], LLaMa [227]) ont gagné en popularité et ont fait l'objet d'une couverture médiatique importante. Les procédés de création de ces modèles, souvent gardés secrets, soulèvent des problématiques éthiques (biais des données, licenciement des équipes éthiques, détournement de technologies) et juridiques (données d'entraînement soumises au droit d'auteur, place de la propriété intellectuelle/artistique dans l'entraînement de ces modèles), tandis que la simple nature disruptive de ces nouveaux outils pour la société alimente les craintes.

Répondre à cette problématique de confiance de l'humain envers les modèles d'IA est l'un des objectifs des méthodes dites "d'humain dans la boucle" (*human-in-the-loop* (HITL)) [241]. Ces approches réintroduisent une dimension humaine cruciale dans le développement, l'entraînement et l'évaluation de ces modèles. Le concept sous-jacent aux méthodes HITL est d'engager activement des individus, qu'ils soient des experts dans le domaine ou des utilisateurs finaux. Outre le gain en interprétabilité et en transparence lié à l'adaptation de ces modèles à l'interaction humaine, ces méthodes permettent souvent aux modèles d'être mieux alignés aux besoins des utilisateurs et valeurs des concepteurs. Plutôt que de concevoir des modèles en vase clos, ces méthodes intègrent activement les rétroactions des parties prenantes humaines pour les orienter de manière plus précise. Dans l'arsenal des méthodes HITL, l'annotation joue un rôle clé. Il s'agit d'une des deux approches les plus couramment mises en œuvre, aux côtés de la prise en compte des retours utilisateurs. L'annotation implique l'ajout d'informations supplémentaires, telles que des métadonnées, des balises ou des étiquettes, aux données d'origine. Cette pratique permet aux modèles d'IA de s'entraîner avec une compréhension contextuelle améliorée, renforçant ainsi leur capacité à interpréter et à généraliser à partir des données annotées. Cette étape offre un contrôle précis et un regard humain sur les données d'entrée, contribuant ainsi à garantir la qualité et la pertinence des informations avec lesquelles les modèles travaillent. En outre, l'annotation est une démarche qui valorise l'expertise et le savoir-faire humain. Dans des domaines où la compréhension nuancée des données est essentielle, les compétences humaines sont irremplaçables. L'annotation devient donc une forme de reconnaissance de cette expertise, mettant en lumière l'importance continue de l'humain dans le développement de l'IA.

L'apprentissage actif (AA) est le sous-domaine d'apprentissage automatique qui se concentre sur cette valorisation de l'interaction humain-machine lors de l'annotation [200]. Contrairement à une annotation aléatoire des données, les stratégies d'AA sont conçues pour une sélection méthodique des ensembles de données à annoter, dans le but

3. https://www.midjourney.com/
4. https://bard.google.com/

de maximiser l'acquisition d'informations pertinentes pour le modèle lors de son prochain cycle d'entraînement. Ces stratégies sont un moyen de simuler une forme de curiosité du modèle, conceptuellement proche de notre façon d'apprendre. Outre le fait d'impliquer efficacement les humains dans la boucle, l'utilisation de l'AA se révèle particulièrement pertinente dans plusieurs scénarios [202]. Tout d'abord, lorsque les ressources pour l'annotation sont limitées et que seule une fraction des données brutes peut être annotée, l'AA permet de maximiser l'utilisation de ces ressources limitées en choisissant stratégiquement quelles données annoter. Ensuite, lorsque les données brutes présentent une grande spécificité ou diversité, l'annotation aléatoire ou uniforme ne suffit pas forcément à couvrir l'ensemble de subtilités présentes. L'AA garantit que les données annotées captureront mieux les particularités du jeu de donnée [270]. Enfin, dans certains cas, des réglementations peuvent exiger qu'un humain annote toutes les données d'entraînement, et dans de telles situations, l'AA peut être utilisé pour sélectionner l'ordre dans lequel les données sont annotées, maximisant ainsi l'efficacité du processus.

1.2 PROBLÉMATIQUE

La classification de textes multi-labels est un problème intéressant et difficile à résoudre dans le domaine du traitement du langage naturel. Cette tâche implique l'attribution de plusieurs catégories ou labels à un document textuel donné. En effet, la classification de textes multi-labels se distingue de la classification mono-label (binaire) par sa complexité accrue. Cette tâche est pertinente dans diverses applications du monde réel, telles que l'indexation de contenu, l'analyse des sentiments et la catégorisation des textes, où un texte peut appartenir à plusieurs catégories ou exprimer plusieurs sentiments. Elle implique de gérer les interdépendances entre les labels, car la présence ou l'absence d'un label peut influencer la présence d'autres labels dans un même document. De plus, elle doit composer avec le déséquilibre potentiel des labels, où certains labels peuvent être plus fréquents ou plus pertinents que d'autres, ce qui ajoute un niveau de sophistication à la tâche. Il est important de noter que la classification multi-labels se différencie également de la classification multi-classes. Dans la classification multi-classes, chaque document est associé à une seule classe parmi plusieurs possibles, tandis que dans la classification multi-labels, un document peut être associé à plusieurs labels simultanément. L'application de l'AA dans le contexte de la classification de textes multi-labels se révèle particulièrement judicieuse en raison de la complexité inhérente du processus d'annotation. En effet, annoter exhaustivement les textes pour tous les labels peut devenir fastidieux pour les annotateurs, en particulier lorsque l'ensemble des

labels est grand. De plus, l'annotation de certains labels peut exiger une analyse approfondie du texte, ce qui rallonge le processus.

L'association des transformers aux tâches de classification multi-labels et d'apprentissage actif suscite un intérêt particulier. Cette convergence présente des aspects conceptuellement intéressants à plusieurs égards. Tout d'abord, les transformers, en tant qu'architectures massives de réseaux neuronaux, peuvent profiter de l'apprentissage actif pour améliorer leur performance de manière frugale. En restreignant le besoin de très nombreuses données d'entraînement, l'apprentissage actif devient un processus optimal pour peaufiner les modèles. Alors que les recherches conjointes sur l'AA et les transformers se limitent généralement aux tâches multi-classes et binaires, notre attention se porte spécifiquement sur la classification multi-labels, une dimension souvent négligée mais riche en potentiel d'exploration.

Nous tenons à souligner l'importance de l'étude de l'apprentissage actif dans le contexte de cette thèse, qui se caractérise par une synergie entre les sphères académique et industrielle. Dans le milieu académique, où les jeux de données constituent souvent une base constante, l'évaluation se concentre principalement sur le développement de nouvelles approches ainsi que sur leur efficacité. Dans ce cadre, l'apprentissage actif offre une méthode efficace pour optimiser l'utilisation de ces jeux de données limités, permettant de maximiser l'acquisition d'informations pertinentes avec des ressources restreintes. D'un autre côté, dans le contexte industriel, l'amélioration des modèles dépend souvent de l'annotation de nouvelles données. Cette dynamique se traduit par une évolution constante des données elles-mêmes, soumises à des changements fréquents. Dans de telles situations, réentraîner un modèle avec des données actualisées ou pertinentes pour un nouveau domaine peut s'avérer plus efficace que d'adapter continuellement un modèle existant à des données en perpétuelle mutation. Il en découle une nécessité impérieuse de développer des stratégies d'annotation appropriées dans l'industrie. Ces stratégies visent à répondre à l'exigence croissante d'accéder en permanence à de nouvelles données annotées, essentielles pour maintenir et améliorer les performances des modèles dans un environnement industriel en constante évolution.

Cette thèse, consacrée à l'application de l'apprentissage actif aux transformers pour la classification multi-label, explore un terrain à la fois académique et industriel, cherchant à répondre à diverses problématiques :

— Au sein de l'environnement académique, cette recherche aborde un défi de taille. Elle souligne un défi continu : l'application de stratégies d'apprentissage actif aux transformers, qui, bien que prometteuses en théorie, affichent des performances variables dans la pratique. Cette inadéquation entre les attentes et les résultats observés révèlent dans un premier temps un besoin

d'analyse afin de trouver les sources de cette incohérence, dans un second temps un besoin pressant de rendre l'application de l'apprentissage actif aux transformers plus fiable et cohérente.

— D'un point de vue plutôt industriel, l'implémentation de l'apprentissage actif suscite d'autres questions cruciales. Les contraintes liées à son déploiement dans des environnements industriels ajoutent une dimension pragmatique à cette étude. Premièrement, la taille des modèles de transformers, souvent considérable, peut être en décalage avec les exigences d'interactivité et d'efficacité de l'apprentissage actif. Deuxièmement, pour qu'une entreprise investisse dans l'implémentation de l'apprentissage actif, elle doit avoir une certitude quant à la rentabilité de cette démarche. Il est essentiel en amont du projet d'annotation de pouvoir prédire quelle sera la meilleure stratégie d'apprentissage actif à implémenter et d'être certain que son implémentation aboutira à un bénéfice concret.

1.3 CONTRIBUTIONS

Impact sur les transformers de six stratégies d'apprentissage actifs multi-labels basées sur l'incertitude. Cette contribution porte sur l'étude de l'application de stratégies d'apprentissage actif multi-labels aux architectures d'apprentissage de type transformers. À travers une série d'expériences, nous avons confirmé que quatre de ces stratégies d'apprentissage actif, à savoir *Max Loss* [130], *Mean Max Loss* [130], *Label Cardinality Inconsistency* [131] et *Catefiry Vector Inconsistency and Ranking of Scores* [176] n'offrent pas de bénéfices notables par rapport à un échantillonnage aléatoire de données d'entraînement. Cependant, notre étude a également mis en lumière que deux stratégies d'apprentissage actif, notamment *Minimum Confidence No weighting* [63] et *Max Margin Uncertainty sampling* [131] améliorent significativement les performances des modèles. En parallèle de ces expérimentations, nous avons tenté de mieux comprendre pourquoi certaines stratégies étaient plus efficaces que d'autres et à identifier les pistes d'amélioration possibles en accord avec nos résultats. Nous avons constaté que ces stratégies ont tendance à sélectionner des instances présentant une grande similitude entre elles, engendrant ainsi une certaine redondance. Toutefois, en introduisant une forme de pré-clustering, dont l'objectif est de diversifier les instances choisies, nous observons que si les résultats évoluent, ils ne s'améliorent pas toujours. Cela s'explique par le fait que, en éliminant la possibilité d'annoter deux instances similaires, nous écartons également des scénarios cruciaux où deux instances ont une proximité sémantique tout en étant associées à des labels différents. Cette première contribution académique [10], apporte des éclairages pertinents pour orienter de manière stratégique le recours aux méthodes d'apprentissage actif

dans le contexte des transformers appliqués à la classification multi-labels.

Implémentation d'un outil d'annotation de données pour des transformers suivant des stratégies d'apprentissage actif. Parallèlement à ces travaux, lors de la mise en place d'un outil d'annotation basé sur l'apprentissage actif en conditions réelles, nous avons identifié un défi spécifique. En effet, entre deux cycles d'apprentissage actif, il existe un temps d'attente incompressible. Cela signifie que si, aucune deuxième source de données à annoter n'est mise en place, les annotateurs se retrouvent à attendre de manière significative lors des séances d'annotation, ce qui peut être inefficace et coûteux.

Rééquilibrage de la distribution des labels en éliminant le temps d'attente inhérent de l'apprentissage actif multi-labels appliqués aux transformers. Cette contribution adresse la problématique des temps d'attente de l'annotateur durant les cycles classique d'AA. Nous avons abordé cette problématique pratique en évaluant différentes méthodes parallèles à l'apprentissage actif. Nous avons proposé notre propre méthode, caractérisée par sa capacité à rééquilibrer la distribution des labels annotés améliorant ainsi considérablement les performances finales peu importe la stratégie d'apprentissage actif choisie. Cette deuxième contibution académique [9] représente ainsi, à la fois une réponse concrète aux problématiques de performances soulevées par nos précédents travaux [10], tout en proposant une solution pratique pour optimiser le processus d'annotation dans le contexte de l'apprentissage actif et valoriser la totalité du temps des annotateurs.

1.4 ORGANISATION DE LA THÈSE

La suite de la thèse est organisée de la manière suivante. Dans la partie 1, nous mettons en place un état de l'art. Le chapitre 2 est consacré à l'état de l'art dans le domaine de l'apprentissage actif et de son application à la classification multi-labels. Dans le chapitre 3, nous nous penchons sur l'apprentissage actif profond, explorant les aspects liés aux modèles neuronaux. Dans la partie 2, nous détaillons nos contributions. Dans le chapitre 4, nous effectuons une analyse approfondie de l'application de stratégies d'apprentissage actif aux transformers dans le cadre de la classification multi-labels. Nous mettons en lumière les avantages de ces méthodes, mais également les obstacles à leur adoption dans des projets industriels. De plus, nous explorons la pertinence des solutions classiques pour surmonter ces obstacles. Le chapitre 5 aborde une problématique originale, à savoir comment valoriser les temps d'attente incompressibles des annotateurs lors des cycles d'apprentissage actif. Nos contributions conservent en perspectives les contraintes du monde industrielle et sont basées sur des cadrés théoriques éprouvés. De plus, elles sont validées expérimentalement sur des collections de test standard du

domaine. Enfin, dans la conclusion de la thèse, nous synthétisons l'ensemble des contributions, discutons de leurs implications et ouvrons des perspectives pour de futures recherches.

Première partie

SYNTHÈSE DES TRAVAUX DE L'ÉTAT DE L'ART

2

APPRENTISSAGE ACTIF MULTI-LABELS

L'apprentissage actif (AA) est une branche de l'apprentissage automatique qui a gagné de l'intérêt au cours des dernières décennies en raison de sa capacité à optimiser la collecte de données et à améliorer l'efficacité des modèles prédictifs [270]. Cette approche repose sur le principe fondamental que l'algorithme d'apprentissage peut choisir activement les exemples d'entraînement les plus intéressants pour améliorer ses performances au lieu de dépendre d'une collecte aléatoire ou exhaustive des données.

L'AA a trouvé rapidement des applications dans de nombreux domaines, notamment dans la classification d'images [226], dans la robotique [186], dans la recherche d'information [223] ou encore dans le traitement automatique des langues [151]. Nous nous concentrons dans cette thèse sur les tâches de classification de textes et plus précisément sur la tâche de classification multi-labels.

2.1 APPRENTISSAGE ACTIF

L'AA a émergé en tant que domaine distinct dans les années 1990. Les premières bases conceptuelles ont étés posées par des chercheurs pionniers [42, 129] qui ont exploré la notion de "stratégie de requête" permettant aux algorithmes d'apprentissage automatique de sélectionner sur quelles données il serait opportun de s'entraîner et donc quelles données doivent être annotées. L'AA est en opposition avec l'apprentissage passif, c'est-à-dire l'apprentissage sur l'ensemble des données à disposition qui doivent alors être annotées de façon exhaustive ou aléatoire.

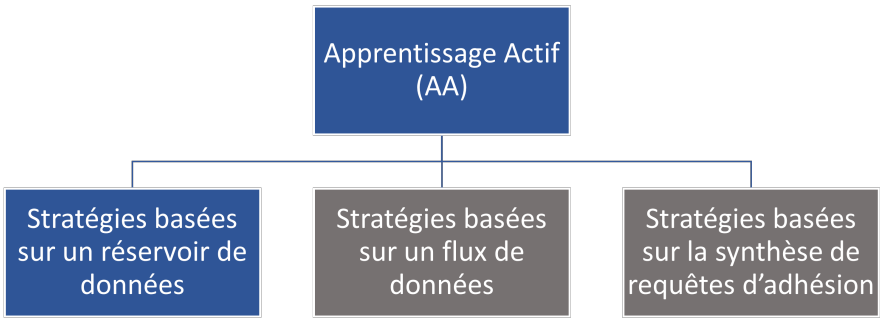


FIGURE 1 – Premier niveau de catégorisation hiérarchique des stratégies d'AA. Les catégories en gris seront moins explorées dans cette thèse.

De nos jours, de nombreux travaux ont développé ces concepts et il est important de pouvoir catégoriser les différentes stratégies de requêtes d'AA proposées [124, 199]. Le premier niveau de catégorisation qui peut être effectué sépare les stratégies dites basées sur un réservoir de données, sur un flux de données et sur la synthèse de requête d'adhésion (cf Figure 1).

— L'**AA basé sur la synthèse de requêtes d'adhésion**, appelée *membership query synthesis based active learning* en anglais, consiste en la génération d'instance près des frontières de décisions du modèle apprenant [8]. Ce scénario d'apprentissage actif, bien qu'efficace pour des cas d'utilisation tels que la génération d'images de pathologies rares, n'est pas approprié dans d'autres contextes. En particulier, dans le traitement du langage naturel, l'annotateur humain peut rencontrer des difficultés à interpréter une instance générée de manière synthétique. Par exemple, dans des travaux sur la reconnaissance de caractères manuscrits, le modèle apprenant génère des symboles non compréhensible, hybride entre différents caractères. Pour le traitement du langage naturel, les instances générées correspondraient sûrement à du charabia [242]. Cependant, des travaux plus récents ont réussi à utiliser avec succès des auto-encodeurs afin de générer des instances textuelles de qualité pour cette famille d'AA [196].

— L'**AA basé sur un flux de données** appelé *stream-based active learning* en anglais, se concentre sur le scénario où les données ne sont pas générées mais sont plutôt acquises au fur et à mesure de la vie du modèle [276]. La prise de décision sur l'annotation ou le rejet d'une instance doit alors être réalisée en temps réel, instance par instance. Dans ce scénario, l'un des principaux défis est la prise en compte des dérives de concepts au sein des données au cours du flux [278].

— Enfin, l'**AA basé sur un réservoir de données** appelé *pool-based active learning* en anglais s'intéresse au cas où un grand nombre de données non-annotées peuvent être obtenues. Dans ce scénario, un ensemble de données est collecté et stocké dans un réservoir, généralement appelé "pool" [128]. L'algorithme d'AA commence par évaluer l'intérêt de chaque instance composant ce réservoir de données puis sélectionne de manière itérative des exemples à annoter parmi ces données préalablement collectées. Entre chaque itération, le modèle se met à jour et l'intérêt de chaque instance est recalculé. Ce cycle continue jusqu'à ce qu'un critère de performance soit atteint ou jusqu'à l'épuisement d'un budget d'annotation. Comme nous nous focaliserons sur cette catégorie d'AA dans la suite de la thèse nous détaillons son cycle dans la Figure 2.

La distinction fondamentale, entre l'AA basé sur un réservoir de données, et celui basé sur un flux, réside dans la manière dont les

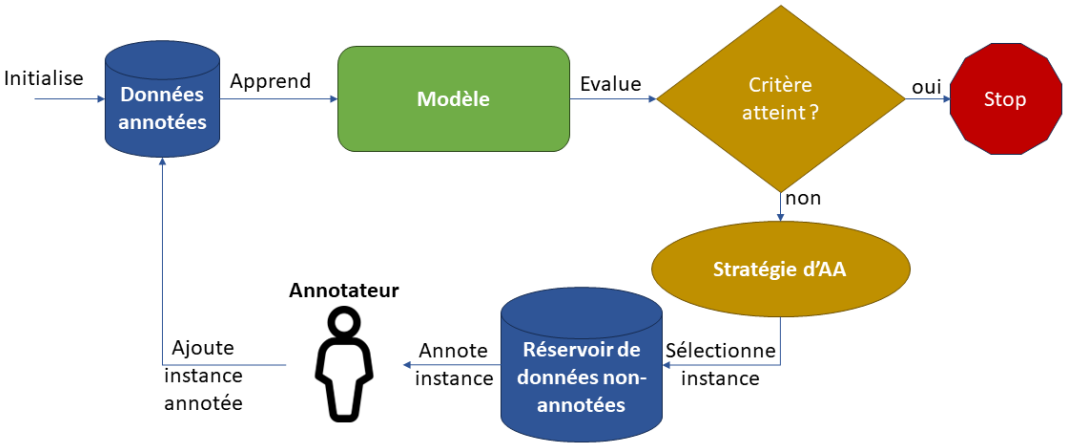


FIGURE 2 – Schéma du cycle d'AA basé sur un réservoir de données.

instances sont évaluées et les décisions sont prises. Dans le cas de l'AA sur un flux, les instances sont évaluées séquentiellement, avec des décisions prises individuellement pour chaque instance au fur et à mesure de leur arrivée dans le flux de données. En revanche, pour l'AA sur un réservoir, l'évaluation des instances se fait de manière globale, avec une prise de décision sous forme de classement pour l'ensemble des instances disponibles.

Au sein des stratégies d'AA basé sur un réservoir, une nouvelle distinction peut être réalisée entre les stratégies basées sur l'informativité et les stratégies basées sur la représentativité (cf Figure 3). En effet, le postulat de l'AA étant que certaines données entraînent mieux le modèle à un moment donné que d'autres conduit à la question : comment déterminer l'intérêt d'une donnée ?

— Dans l'**AA basé sur l'informativité**, l'objectif principal est de sélectionner les instances qui comblent le plus les lacunes du modèle et apportent donc le plus d'information. L'idée sous-jacente est donc d'explorer les zones de l'espace de données où le modèle est le plus en difficulté. L'hypothèse suivie, est qu'en se concentrant sur les instances qu'il classifie mal, le modèle réduira ses faiblesses et donc s'améliorera [63].

— A l'opposé, l'**AA basé sur la représentativité** se concentre sur la sélection d'instances qui représentent au mieux la distribution globale des données. L'objectif principal est de s'assurer que l'ensemble de données annotées reflète fidèlement la variabilité de l'ensemble de données non-annotées de départ. Ces stratégies maximisent donc la couverture de l'espace de données. L'idée sous-jacente ici, est de garantir que l'ensemble annoté est équilibré, tout en incluant des classes minoritaires provenant de régions moins denses de l'espace des données. L'hypothèse suivie ici est qu'en apprenant au fur et à mesure des fractions re-

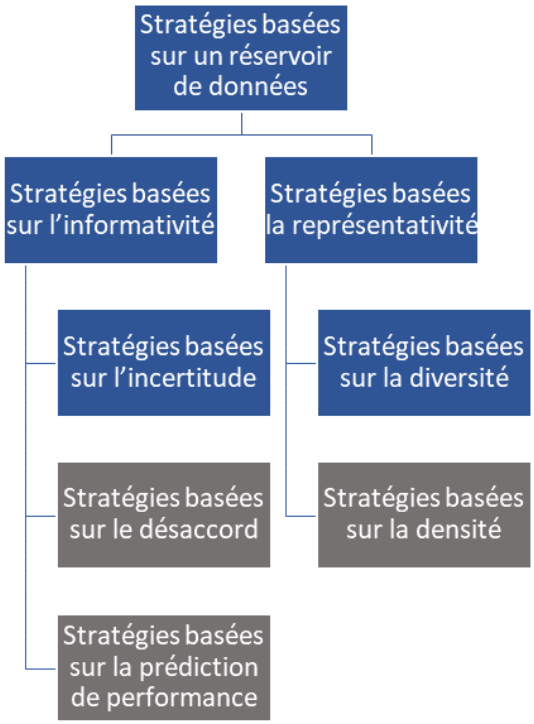


FIGURE 3 – Deuxième et troisième niveau de catégorisation hiérarchique des stratégies d'AA. Les catégories en gris seront moins explorées dans cette thèse.

présentatives de l'ensemble des données, le modèle sera de plus en plus polyvalent et donc de plus en plus performant [249].

Il est intéressant de faire un parallèle avec l'apprentissage chez les humains. Ces catégories de stratégies peuvent s'apparenter à une différence de stratégies que deux élèves humains adopteraient pour obtenir la meilleure moyenne générale. Dans le cas de l'élève choisissant l'informativité, celui-ci révisera chaque jour les leçons qu'il comprend le moins bien. Si l'élève a des difficultés sur une leçon, plusieurs jours peuvent être passés dessus. Par contre, l'élève ne passe pas de temps à réviser des leçons qu'il maîtrise bien. Dans le cas de l'élève choisissant la représentativité, l'élève révisera chaque jour un ensemble de leçons balayant le programme. A priori, il n'y a pas de stratégie meilleure qu'une autre, la performance obtenue dépendra des facultés de départ des élèves et des particularités des leçons. En faisant le parallèle dans le sens inverse, il n'y a donc pas de catégorie de stratégies supérieure entre informativité et représentativité, les performances obtenues dépendront des modèles et des données [270].

Enfin, un troisième niveau de catégorisation peut être réalisé dans les stratégies basées sur l'informativité et la représentativité (cf Figure 3). Pour l'informativité, les stratégies peuvent se décliner en stratégies basées sur l'incertitude, sur le désaccord ou sur la prédiction de performance. Pour la représentativité, les stratégies peuvent se décliner en stratégies basées sur la diversité et sur la densité. Ces catégories

sont détaillées de façon plus approfondie dans les sous-chapitres suivants.

2.1.1 *Notations*

Soit X^a l'ensemble des instances textuelles annotées et X^u l'ensemble des instances textuelles non-annotées. Soit $X = X^a \cup X^u$ l'ensemble de toutes les données à notre disposition. X est un ensemble de taille n composé des instances textuelles $x_1, ..., x_n$ et soit l notre espace des labels de taille q avec $l = l^1, ..., l^q$. Pour une instance donnée x_i nous représentons sa distribution probabiliste de labels par $y_i = [y_i^1, ..., y_i^q] \in [0, 1]$ où plus y_i^j est proche de 1, plus le modèle est confiant que x_i est labellisé comme j et où plus y_i^j est proche de 0, plus le modèle est confiant que x_i n'est pas labellisé comme j. Nous notons $\hat{x}$ l'instance sélectionnée par la stratégie d'AA. Nous prenons donc pour la suite du manuscrit l'habitude de placer en indice les valeurs correspondantes aux instances et en exposant les valeurs correspondantes aux labels. y_i^j correspondra par exemple à la valeur de l'annotation du j-ème label de la i-ème instance.

2.1.2 *Stratégies basées sur l'informativité*

Ces stratégies privilégient les données qui apportent le plus d'informations au modèle, en ciblant les instances pour lesquelles le modèle actuel peine dans sa classification. L'objectif est d'améliorer la performance du modèle en se concentrant sur les zones de l'espace des données où il y a le plus à apprendre. Dans cette section, $\hat{x}$ représentera donc l'instance la plus informative du réservoir de données.

2.1.2.1 *Stratégies basées sur l'incertitude*

Dans le contexte de l'apprentissage automatique, l'incertitude peut référer à l'incertitude aléatoire, *aleatoric* ou l'incertitude épistémique, *epistemic* [105]. Dans le cas de l'AA, l'incertitude que l'on cherche à mesurer, et sur laquelle les stratégies suivantes se focalisent, est l'incertitude épistémique [164, 165]. En effet, l'incertitude aléatoire réfère à l'incertitude inhérente, à un phénomène, ou à un processus. Il s'agit de l'incertitude irréductible qui, dans notre cas, est souvent liée à la variabilité naturelle des données textuelles (les variabilités syntaxiques, les ambiguïtés lexicales, les variabilités culturelles, ...). Au contraire, l'incertitude épistémique est l'incertitude liée à un manque de connaissance, c'est-à-dire à l'ignorance d'un modèle, et qui peut donc, en principe, être réduite par l'acquisition d'informations supplémentaires.

L'AA basé sur l'incertitude fait partie des approches les plus populaires à l'AA. Le concept est facile à comprendre, les stratégies

sélectionnent les instances sur lesquelles le modèle est le plus incertain dans ses classifications. Par exemple, lorsque l'on emploie un modèle probabiliste pour de la classification binaire (entre 0 et 1), les instances les plus incertaines sont celles dont la probabilité estimée d'appartenance à la classe positive est la plus proche de 0.5. Lorsque l'on emploie un modèle probabiliste pour la tâche multi-classe, cette approche peut être généralisée à une stratégie portant le nom de moindre confiance ou *least confidence* [43] :

$$\hat{x} = \underset{x_i}{\operatorname{argmax}} \left[1 - \max_j(y_i^j) \right], \tag{1}$$

avec max qui retourne le plus haut score de probabilité d'appartenance à une classe. Selon cette stratégie, les instances les plus incertaines sont donc celles dont les prédictions sont les moins confiantes, c'est-à-dire les instances où la probabilité de la classe prédite est la plus basse.

La plus grande faiblesse de la stratégie de moindre confiance est qu'elle ne prend en compte que le score de probabilité de la classe la plus probable. Dans un problème à 6 classes, l'instance x_α avec $y_\alpha = [0.5, 0.1, 0.1, 0.1, 0.1, 0.1]$ à le même score d'incertitude que l'instance x_β avec $y_\beta = [0, 0, 0, 0.1, 0.4, 0.5]$. Pourtant comme l'écart de probabilité entre la classe la plus probable et la seconde classe la plus probable est plus faible avec x_β, on peut dire que l'incertitude du modèle est plus grande pour x_β que pour x_α. Dans cette situation, l'échantillonnage par marge ou *margin sampling* permet de différencier ces instances [189, 190] :

$$\hat{x} = \underset{x_i}{\operatorname{argmin}} \left[\max_j(y_i^j) - \underset{j}{\operatorname{second\_max}}(y_i^j) \right], \tag{2}$$

avec $second\_max$ qui retourne le second plus haut score de probabilité d'appartenance à une classe. Selon cette stratégie, plus la marge entre la classe la plus probable et la seconde classe la plus probable est faible, plus le modèle est incertain dans ces prédictions et donc plus l'instance a un score d'incertitude associé élevé.

La stratégie d'AA la plus populaire, basée sur l'entropie [204] est encore plus généralisable que l'échantillonnage par marge :

$$\hat{x} = \underset{x_i}{\operatorname{argmax}} \left[-\sum_j y_i^j * \log(y_i^j) \right], \tag{3}$$

La mesure d'entropie représente la quantité d'information nécessaire pour encoder une distribution de probabilité. Plus l'entropie de la distribution probabiliste des prédictions d'une instance est grande, plus on considère que le modèle est incertain sur cette instance. On notera que pour la classification binaire, l'échantillonnage basé sur

l'entropie est équivalente aux stratégies par marge et de moindre confiance.

Plus récemment, une autre façon de mesurer l'incertitude des résultats a été explorée, consistant en l'évaluation de la divergence des prédictions d'un modèle dans la région locale d'une instance. En d'autres termes, cela revient à mesurer la variation des prédictions du modèle sur des instances voisines de l'instance considérée. Une instance proche d'un seuil de décision est alors une instance pour laquelle les prédictions du modèle sont fortement différentes des prédictions sur les instances voisines. Pour explorer ces divergences locales, certains explorent les k-voisins les plus proches [150] tandis que d'autres sélectionnent les instances les plus sensibles à des perturbations induites [267].

2.1.2.2 *Stratégies basées sur le désaccord*

Le principe des stratégies d'AA basées sur des comités ou des ensembles de modèles repose sur le fait de maintenir en parallèle plusieurs sources de classification et d'annoter les instances sur lesquelles les sources divergent dans leur classification. Ces stratégies font parties des stratégies basées sur le désaccord, *disagreement-based strategies*. La première des stratégies basées sur le désaccord mise en place, est aussi la plus connue, il s'agit de la stratégie de requête par comité [203], *Query-By-Committee*. Un comité de différents membres, où un membre est un modèle de classification, est obtenu suite à une initialisation sur des sous espaces différents du jeu de données déjà annotées. La première hypothèse sur laquelle repose cette stratégie, est d'avoir à disposition, au départ du projet d'annotation, un nombre suffisant de données d'entraînements déjà annotées pour initialiser plusieurs modèles sans qu'il n'y ait de superposition entre les différents jeux d'initialisation. L'instance maximisant le désaccord (dont nous détaillons le calcul ci-dessous) entre les différents modèles du comité est ensuite sélectionnée pour annotation, puis chaque modèle est entraîné avec cette nouvelle instance. Afin de maximiser les performances de cette approche, des travaux complémentaires [153] ont proposé de focaliser différents jeux d'initialisation sur des parties différentes de l'espace du jeu d'entraînement déjà annoté. L'objectif est ainsi de maximiser le désaccord entre les membres du comités. En effet, si les modèles de classification sont initialisés avec des jeux de données similaires, le désaccord au sein du comité sera faible. D'autres travaux quant à eux [160], ont montré que l'approche était toujours pertinente si on réduisait la taille du comité à deux membres tout en faisant en sorte d'initialiser ces modèles avec des données les plus différentes possibles.

Plutôt que de former et de maintenir plusieurs modèles distincts, des stratégies basées sur le *Monte Carlo dropout* permettent de générer plusieurs échantillons stochastiques à partir d'un modèle de base

en utilisant la technique du *dropout*. En évaluant ces échantillons, on peut estimer le désaccord entre les prédictions, puis sélectionner les exemples où ce désaccord est le plus élevé pour annotation [207, 210] Le *dropout* est une technique de régularisation utilisée lors de l'entraînement d'un réseau de neurones. Pendant l'entraînement, certains neurones du réseau sont désactivés de manière aléatoire (c'est-à-dire "abandonnés") à chaque itération. Cela empêche le modèle de devenir trop dépendant de certains chemins neuronaux et améliore sa généralisation [215]. Bien que cette technique convienne donc particulièrement bien aux modèles neuronaux, elle peut aussi être étendue à d'autres types de modèles de classification en adoptant une approche bayesienne sur leurs paramètres [93].

Le calcul du désaccord dans ces différentes stratégies varie principalement autour de trois approches :

— Le **taux de variation** [110], est une mesure du désaccord qui se concentre sur la proportion de membres du comité qui ne sont pas d'accord sur la prédiction majoritaire. Il est calculé en comptant le nombre de membres du comité dont la prédiction diffère de la prédiction majoritaire, puis en divisant ce nombre par le nombre total de membres du comité. Cette mesure est la plus simple, indiquant s'il y a un désaccord mais ne prenant pas en compte la force de ce désaccord.

— La **Divergence de Kullback-Leibler** [151], est une mesure de la divergence entre deux distributions de probabilité. Dans le contexte du désaccord, elle est utilisée pour mesurer la différence entre la distribution des prédictions des membres du comité et une distribution de référence. Cette mesure prend en compte la force du désaccord mais nécessite une distribution de référence. Une autre faiblesse de cette mesure est qu'elle peut être sensible aux valeurs nulles dans les distributions, donnant des résultats moins stables si les distributions sont très dispersées, par exemple lorsque les classes sont nombreuses ou lorsque leur présence est très déséquilibrée.

— L'**entropie de vote** [59], *vote entropy*, est une mesure du désaccord qui se base sur la distribution des votes (prédictions) des membres du comité. Elle est calculée en utilisant la notion d'entropie de l'information, qui mesure l'incertitude dans la distribution des votes. Plus l'entropie est élevée, plus il y a de désaccord entre les membres du comité. Cette mesure avantage l'instance ayant reçu la plus grande diversité de prédictions et ne nécessite pas de distribution de probabilité de référence.

Les stratégies basées sur le désaccord nécessitant l'entraînement et le maintient de plusieurs modèles ou du moins la réalisation de multiples inférences des variations d'un même modèle, cette famille de stratégies est souvent réservée à des modèles qui sont peu coûteux à utiliser en terme de temps et de ressources informatiques.

2.1.2.3 *Stratégies basées sur la prédiction de performance*

D'autres stratégies reposent sur l'utilisation de la prédiction de la performance d'un modèle en tant qu'indicateur d'informativité. L'objectif est alors de sélectionner pour annotation, l'instance qui minimise le plus les erreurs futures du modèle [181]. Cependant, cette approche peut s'avérer coûteuse en termes de ressources informatiques, car un nouveau modèle doit être entraîné pour chaque instance candidate. En effet, il faut non seulement estimer l'erreur attendue future pour chaque instance candidate, mais un nouveau modèle doit être ré-entraîné pour chaque annotation possible, ce qui atteint une complexité en $\mathcal{O}(n * 2^q)$.

Afin de surmonter cette limitation, d'autres méthodes utilisent un second modèle qui évalue les erreurs futures du premier modèle, et cible de façon plus précise les instances intéressantes. Ces seconds modèles peuvent être entraînés à sélectionner les instances réduisant le plus les erreurs futures du premier modèle à partir d'apprentissage par renforcement [65] ou encore d'apprentissage par imitation [137]. Pour que ces approches fonctionnent, il faut, dans un premier temps, acquérir des données annotées pour entraîner les modèles d'évaluation d'erreur, soit à partir de données issues d'une tâche différente [14], soit en effectuant un premier effort d'annotation destiné à cela.

Pour remédier à cela, d'autres stratégies entraînent elles aussi un second modèle, mais cette fois-ci en même temps que le modèle initial, ne requérant donc pas l'acquisition de données annotées supplémentaires. Ces seconds modèles calculent une fonction de perte afin d'évaluer les erreurs de prédictions du modèle principal en les comparant aux annotations "vraies" d'un jeu de données de validation [26, 206].

Les stratégies basées sur la prédiction de performance ont pour faiblesse de nécessiter des ressources de départ importantes (ressources informatiques importantes, acquisition d'un jeu de données déjà annotées sur une tâche proche ou encore un jeu de validation annoté). Étant donné que l'AA est souvent mis en place dans des contextes contraints en ressources, cette famille de stratégie est réservée aux modèles de classification les plus simples et aux jeux de données à faible dimension.

2.1.3 *Stratégies basées sur la représentativité*

Deux des faiblesses particulières à l'AA basé sur l'informativité sont la sélection d'instances aberrantes [116] et le biais d'échantillonage [169], où les stratégies se focalisent sur une même sous-partie de l'espace des données, que le modèle n'arrive jamais à classifier. S'affranchissant de ces problèmes, les stratégies d'AA basées sur la représentativité visent à sélectionner des instances d'apprentissage qui

sont les plus représentatives de l'ensemble des données non-annotées. L'idée est donc de choisir des instances capturant le mieux la distribution des données. Certaines approches privilégient de ne pas oublier des espaces spécifiques de la distribution des données tandis que d'autres se concentrent sur le fait de sélectionner les instances qui partagent des caractéristiques avec un grand nombre d'autres instances.

2.1.3.1 *Stratégies basées sur la densité*

Dans le but de reconstruire un sous-espace le plus fidèle possible du jeu de données non-annotées dans son entièreté, les stratégies basées sur la densité sélectionnent l'instance la plus représentative du jeu de données non-annotées. Ces stratégies permettent d'éviter la sélection d'instances aberrantes, *outliers*, en sélectionnant les instances au sein de parties de l'espace des données à haute densité [201]. Pour mesurer la densité, des techniques courantes de recherche d'information telles que les n-grams ou les comptes de mots peuvent être employés [272].

Une autre approche pour évaluer la représentativité d'une instance est de mesurer la distance (ou similarité) de celle-ci avec toutes les instances récemment annotées [151, 201] et de sélectionner l'instance minimisant le plus ses distances. La similarité cosinus, la divergence de Kullback-Leibler ou la similarité gaussienne peuvent être utilisées comme mesure de distance [68]. Comme cette approche peut vite devenir coûteuse en terme de complexité, on peut limiter le calcul des distances aux k-voisins les plus proches d'une instance donnée [274].

Poussant l'idée de voisinage plus loin, les techniques basées sur des agrégats, *clusters* [163, 250], regroupent les instances entre elles et déterminent que les instances les plus représentatives appartiennent au sous-espace composé des instances centroïdes de chaque agrégat.

Les stratégies basées sur la densité ont notamment pour avantage que de nombreux calculs peuvent être réalisés en amont de la mise en place du procédé d'AA, ce qui permet de fluidifier l'interactivité avec les annotateurs humains dans un second temps.

2.1.3.2 *Stratégies discriminantes*

Les stratégies d'AA discriminantes sélectionnent les instances les plus différentes des instances déjà annotées afin de représenter l'espace du jeu de données non-annotées. Comme pour les stratégies basées sur la densité, des mesures basées sur des caractéristiques simples des instances peuvent être utilisées comme les n-grams ou les mots hors du vocabulaire [60], de même que des mesures de distances entre les instances [259]. D'autres stratégies, elles, discriminent

le jeu de données annotées du jeu de données non-annotées, en entraînant les modèles de façon contradictoire, *adversarial* [73, 211].

Ces stratégies s'opposent par leur fonctionnement au biais d'échantillonage des stratégies basées sur l'informativité où plusieurs instances très similaires peuvent être choisies pour être annotées. Par contre, elles peuvent encourager la sélection d'instances aberrantes.

2.1.3.3 *Stratégies basées sur la diversité*

Lorsque l'annotation des instances ne se fait pas instance par instance, mais par lot d'instances comme détaillé dans la partie 2.1.4, la représentativité des instances à annoter n'est plus seulement importante par rapport à l'ensemble des instances non-annotées mais au sein même du lot d'instances sélectionné. En effet, la présence d'instances aberrantes ou encore une trop grande similarité dans des lots d'instances peut conduire à différents biais d'entraînement comme le sur-ajustage [87]. Afin de garantir que chaque lot d'instances soit représentatif, des stratégies discriminantes ou basées sur la densité peuvent être appliquées à l'échelle du lot d'instances à sélectionner. Ces lots d'instances peuvent, par exemple, être assemblés de façon itérative, la première étape initialisant le lot avec une instance, suivie d'étapes utilisant un critère discriminant pour s'assurer que les instances ajoutées soient suffisamment différentes jusqu'à la complétion du lot d'instances à annoter [23]. La méthode de *coreset* [2], réduisant de grand espaces de données à un faible échantillon représentatif, se prête bien à cette approche [198]. D'autres approches ajoutent comme critère supplémentaire de sélection, lors de la complétion du lot d'instances, que chaque instance provienne d'une région de l'espace des données différentes. Comme dans les stratégies basées sur la densité, des agrégats d'instances sont alors calculés, notamment grâce à un partitionnement par k-moyennes par exemple [257, 273]. Ces méthodes étaient particulièrement efficaces lorsque l'AA était réalisé sur des comités de machines de vecteurs à supports, par rapport à des complétions de lot d'instances par échantillonage aléatoire ou par sélection des N instances les plus informatives [199].

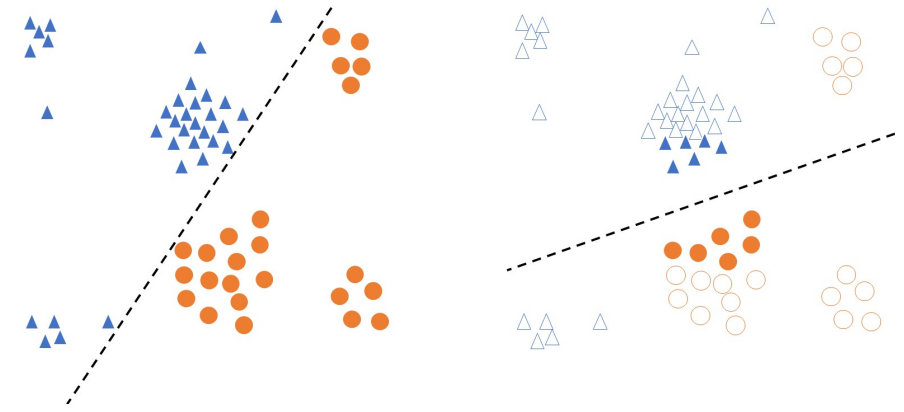
2.1.4 *Apprentissage actif par lot de données*

Traditionnellement, l'AA repose sur la sélection séquentielle d'exemples, un par un. Toutefois, cette approche peut être inadéquate lorsque l'entraînement d'un modèle est long et coûteux, notamment dans le cas de méthodes basées sur des comités de modèles [117] ou de réseaux neuronaux profonds [11]. Afin de limiter la fréquence de réentraînement des modèles, il est alors intéressant que plusieurs instances soient sélectionnées pour annotation [29]. L'AA par lot de données est souvent lié à des modèles apprenant par lot, *batch learning*, pratique couramment utilisée en apprentissage automatique. L'un

des avantages majeurs de cette approche réside dans son efficacité computationnelle. Plutôt que de s'entraîner sur les instances une par une, les exemples sont regroupés en lots de taille fixe, ce qui permet d'accélérer considérablement le processus d'entraînement. De plus, l'AA par lot de données trouve des applications dans des scénarios où plusieurs annotateurs travaillent simultanément sur un ensemble de données partagé. Dans de tels scénarios collaboratifs, la coordination et la répartition judicieuse des tâches d'annotation sont essentielles pour éviter des doublons ou des redondances dans les annotations [199]. L'AA par lot permet alors de sélectionner intelligemment les instances à annoter, en s'assurant que chaque annotateur contribue de manière efficace en se concentrant sur des instances provenant de parties distinctes de l'espace des données, contribuant ainsi à une utilisation optimale des ressources humaines [277].

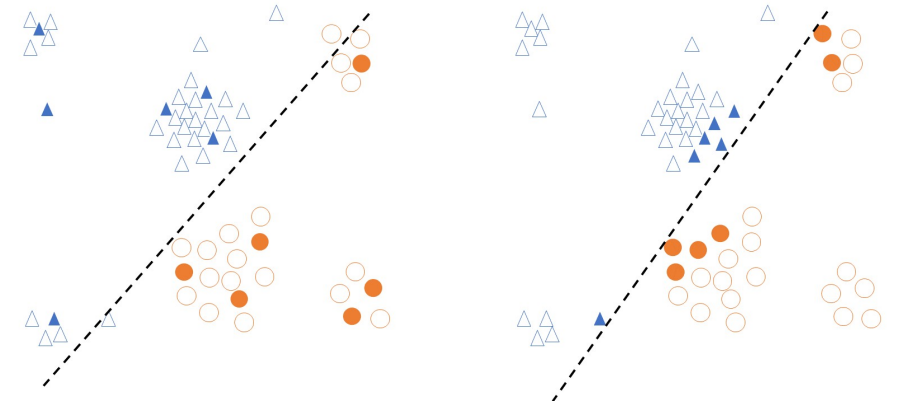
Le cœur de l'AA par lot de données réside dans la création des meilleurs lots de données à annoter possibles. Le choix de la taille du lot annoté est alors d'une importance capitale. Un échantillon plus large réduit le besoin de ré-entraînement fréquent des modèles et des itérations entre les annotateurs humains et les machines, ce qui peut générer des économies substantielles de temps et de ressources. En revanche, un échantillon plus restreint permet de se concentrer sur des exemples spécifiques, optimisant ainsi la précision de l'estimation de l'incertitude du modèle. Le choix de la taille de l'échantillon dépend étroitement du budget d'annotation disponible et de l'importance accordée à chaque annotation. Ces décisions doivent être prises avec soin afin d'optimiser le processus d'AA dans ces contextes spécifiques [39].

Dans la majorité des cas, la stratégie d'AA par lot de données privilégie la sélection des K-meilleures instances en se basant sur les scores évalués par une stratégie d'AA. Cependant, il est important de noter que cette heuristique ne garantit pas toujours de meilleurs résultats par rapport à un échantillonnage aléatoire des données. De plus, certaines approches plus récentes, bien que plus sophistiquées, entraînent des coûts plus élevés en termes de complexité [11, 119]. Très récemment, une approche stochastique consistant à sélectionner les instances selon une distribution déterminée par les scores d'AA individuels, au lieu de se limiter à l'acquisition des K-meilleurs instances, a été proposée [120]. Cette dernière stratégie se profile comme une approche prometteuse qui pourrait bien remplacer la méthode classique des "K-meilleurs". Malgré un coût computationnel comparable à celui de l'approche classique qui sélectionne les K-meilleures instances, elle permet de constituer des lots d'instances conduisant à des performances d'entraînement comparables à l'état de l'art des méthodes d'AA par lot de données.



(a) Problème de classification binaire

(b) Instances sélectionnées par informativité



(c) Instances sélectionnées par représentativité

(d) Instances sélectionnées par combinaison informativité et représentativité

FIGURE 4 – Exemple de sélections d'instances par informativité ou par représentativité dans un problème de classification binaire

2.1.5 *Combinaisons d'approches*

La plupart des stratégies abordées dans les parties 2.1.2 (et respectivement 2.1.3) se concentrent presque exclusivement sur les aspects informatifs (et respectivement représentatifs) des instances à sélectionner pour annotation. Il y a une interrelation entre ces deux mesures : lorsque l'informativité d'une instance sélectionnée augmente, il est possible que sa représentativité diminue. Des principes issus de ces deux approches peuvent être combinés dans des méthodes dites hybrides pour explorer cet équilibre [81]. En ramenant la recherche d'instance à annoter à un problème d'optimisation entre exploitation et exploration, les instances les plus incertaines sont sélectionnées pour l'exploitation tandis que pour l'exploration, ce sont les instances les plus différentes des instances déjà annotées qui sont mises en valeur [253].

La figure 4 inspirée de [101], montre que pour résoudre un problème de classification binaire, les approches exclusivement basées sur l'informativité ou la représentativité peuvent être moins efficaces pour déterminer la bonne frontière de décision entre les deux classes. Par informativité, des espaces de données peuvent être ignorés tandis que par représentativité, la convergence vers une bonne classification peut être trop lente. L'exemple choisi montre que dans ce cas, une combinaison des deux approches permet de rapidement converger vers une frontière de décision dans la classification binaire très proche de la solution idéale.

La méthode la plus simple pour combiner deux stratégies, consiste à fusionner plusieurs critères en un seul score global [51]. Le score d'intérêt de chaque instance est obtenu en agrégeant les sous-scores pondérés de chaque critère par multiplication [33, 240] ou somme [201]. En simplifiant, nous ramenons ces approches aux formules suivantes :

$$score\_hybride = \beta * informatitvité + (1-\beta) * représentativité \quad (4)$$

ou

$$score\_hybride = \beta * informatitvité * (1-\beta) * représentativité \quad (5)$$

Plutôt que d'avoir une valeur de β fixée en début d'expérience, il est intéressant de faire varier cette valeur au cours du processus d'AA [246]. Ce concept d'AA "dynamique" se fonde sur l'hypothèse que, au début du processus d'AA, le modèle n'a pas encore été suffisamment entraîné sur un grand nombre d'instances. Par conséquent, ses prédictions ne sont pas encore suffisamment stabilisées pour obtenir des calculs d'informativité fiables. L'idée est de commencer le processus d'AA en sélectionnant les instances pour leur représentativité, puis de basculer sur une sélection par informativité en fin d'AA [49]. La bascule peut se faire au bout d'un nombre fixe d'instances annotées par représentativité ou d'un score objectif atteint mais aussi sous forme d'une transition graduelle d'une approche à l'autre, plus ou moins rapide suivant la courbe d'apprentissage du modèle [7].

Plutôt que de combiner l'informativité et la représentativité en un score hybride, leur combinaison peut être réalisée en deux étapes, *multi-step active learning*. Une première approche consiste à commencer par calculer des agrégats d'instances puis, par agrégats, sélectionner l'instance la plus informative [219]. Une seconde approche consiste à d'abord sélectionner des instances très informatives, puis de réaliser la répartition en agrégats afin de constituer un lot divers de données [157, 205, 249].

2.1.6 *L'humain au centre du processus*

L'un des défis majeurs dans le développement de modèles d'IA est le manque de confiance envers ces systèmes automatisés. Les modèles

d'IA, en particulier les modèles d'apprentissage profond, peuvent sembler opaques et déroutants pour les utilisateurs, ce qui entraîne un manque de confiance envers leurs décisions. Cependant, l'intégration de l'humain dans la boucle d'AA peut contribuer à attenuer ce problème de confiance [241, 247]. L'un des avantages essentiels de l'AA est l'intérêt accru pour les annotateurs. Contrairement à d'autres formes d'annotation automatisée, l'AA donne aux annotateurs un rôle actif et influant dans le processus. Leurs décisions ont un impact direct sur les prochaines instances à annoter. Cela crée un sentiment d'importance et de contribution à l'amélioration du modèle, ce qui peut être motivant pour les annotateurs et peut renforcer la confiance envers le modèle. L'humain est reconnu comme une ressource précieuse, et son expertise est mise en avant. Du point de vue des utilisateurs, savoir que de l'expertise humaine est directement responsable des réponses fournies par le modèle génère un sentiment de réassurance [191]. Pour tirer le meilleur parti de cette coopération, certaines bonnes pratiques d'inclusion dans l'AA peuvent être mises en avant. Il est essentiel de faciliter la tâche de classification pour les annotateurs. Cela peut se faire en proposant des tâches de classification label par label [68], en permettant des comparaisons entre plusieurs instances [254] ou en fournissant des textes plus courts et plus simples à annoter [107]. Diversifier la tâche est également un aspect crucial, en incluant des textes provenant de différents domaines et contextes. Une interface utilisateur (UI) agréable et efficace est essentielle pour faciliter la coopération entre l'humain et la machine. Pour améliorer davantage l'expérience utilisateur (UX), il serait par exemple envisageable d'explorer des concepts de gamification dans le processus d'annotation [147]. Le gain engendré par une bonne UX peut être d'une importance similaire au gain lié au choix d'une bonne stratégie d'AA [50]. Enfin, des séances d'annotations courtes permettent de maintenir l'engagement des annotateurs et de maximiser leur productivité.

L'humain joue un rôle central dans l'AA, en tant qu'annotateur responsable de labeliser les données. La qualité, l'accord sur les annotations et le confort des annotateurs sont cruciaux pour le succès d'un projet d'AA [88]. La qualité des annotations générées par les annotateurs est un pilier essentiel de la réussite de l'AA. Leur rôle est de transformer des données brutes en informations précieuses, fournissant ainsi la base sur laquelle les modèles d'apprentissage automatique s'appuient pour évoluer. Il est à noter que la qualité des annotations peut fluctuer grandement d'un annotateur à l'autre. Cette variabilité peut découler de multiples facteurs, y compris des différences dans l'interprétation des consignes, des biais personnels, et même des erreurs de saisie de données. Cette diversité dans la qualité des annotations peut considérablement affecter les performances des modèles. D'autant plus que le bruit généré par des erreurs humaines

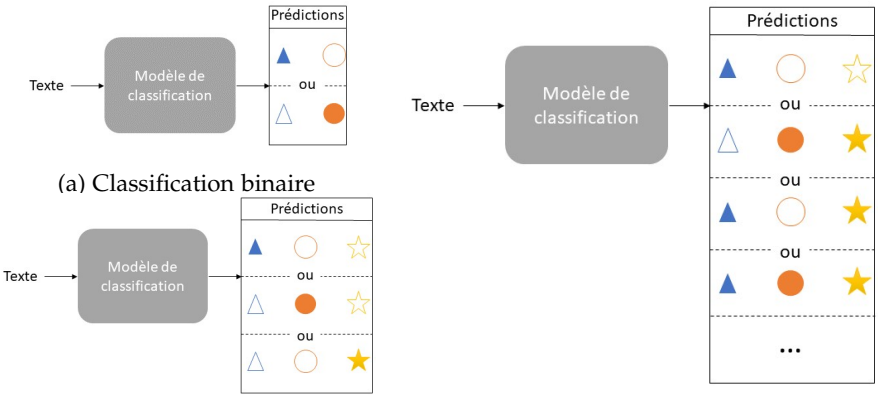
n'est souvent pas aléatoire [159], ce qui peut biaiser profondément les modèles au cours de leurs entraînements. L'accord inter-annotateur, c'est-à-dire la cohérence des annotations entre différents annotateurs, est également un aspect essentiel de l'AA. Il permet de mesurer la fiabilité des annotations et de déterminer dans quelle mesure les annotateurs s'accordent sur l'attribution des labels. Les niveaux d'accord entre annotateurs influencent directement la qualité des données d'entraînement, et par conséquent, les performances du modèle. Une faible concordance entre annotateurs peut indiquer des problèmes de consignes peu claires, de préjugés, ou encore des difficultés inhérentes à la tâche d'annotation [113]. Le confort des annotateurs est également un facteur clé. Cela comprend le temps d'attente entre les cycles d'annotation, ainsi que la complexité et la charge cognitive des tâches d'annotation. L'efficacité de l'AA dépend en grande partie de la rapidité du cycle d'AA. Les annotateurs doivent attendre la mise à jour du modèle après chaque cycle d'AA, et tout temps d'attente excessif peut entraîner un ralentissement du processus et encouragé une dispersion de l'attention, impactant négativement la qualité d'annotation. Par ailleurs, si les tâches d'annotation sont trop complexes ou difficiles, cela peut augmenter la charge de travail et la fatigue des annotateurs qui, à leur tour, peuvent influencer négativement la qualité des annotations [159].

Lorsque plusieurs annotateurs interagissent avec le même processus d'AA, de nouveaux paramètres sont à prendre en compte. Le processus doit en effet tenir compte du fait que la vitesse d'annotation a de forte chance de varier d'un annotateur à un autre. De plus, il est courant qu'un annotateur soit plus lent au début de la tâche, lorsqu'il s'acclimate aux exigences, ainsi qu'à la fin de la tâche, lorsque la fatigue s'installe [200]. Ces variations individuelles dans le rythme d'annotation peuvent influencer la dynamique globale du processus d'AA et nécessitent une gestion adéquate pour maintenir l'efficacité et la cohérence du système. Dans le contexte de l'AA basé sur des métriques de coût, l'intégration de l'expertise humaine devient d'autant plus cruciale. En effet, ces stratégies d'AA visent à équilibrer judicieusement le coût, notamment en termes de temps d'annotation humain, avec les avantages que ces annotations peuvent apporter au modèle. Dans cette optique, l'humain est considéré non seulement comme un annotateur, mais également comme une ressource précieuse dont il faut tenir compte à plusieurs égards. Les métriques de coût incluent souvent des estimations du temps probable nécessaire à un annotateur pour étiqueter des exemples particuliers, en fonction de leur complexité ou de leur spécificité. Cette approche cherche à optimiser l'utilisation du temps et de l'expertise humaine, tout en maximisant l'efficacité globale du modèle d'apprentissage automatique [83].

2.2 CLASSIFICATION MULTI-LABELS

La classification multi-label est une tâche essentielle en traitement automatique du langage (TAL) et dans de nombreuses applications du monde réel. Elle se distingue par son objectif d'attribuer plusieurs étiquettes ou labels à une instance textuelle donnée. Cette distinction la rend différente de deux autres tâches de classification courantes : la classification binaire (ou mono-label) et la classification multi-classes. La classification binaire est la plus simple de toutes. Elle consiste à attribuer une seule étiquette ou catégorie à une instance textuelle. Par exemple, dans une application de détection de spam, un e-mail peut être classifié comme "spam" ou "non-spam", et un seul label est associé à chaque e-mail [216]. La classification multi-classes est un peu plus complexe, car elle consiste à attribuer une instance à l'une des nombreuses catégories mutuellement exclusives. Par exemple, dans un système de classification de documents, un article peut être catégorisé comme "sport", "politique" ou "économie", mais il ne peut appartenir qu'à une seule de ces catégories [154]. La classification multi-labels, quant à elle, diffère de ces deux approches en permettant qu'une instance textuelle soit associée à plusieurs labels simultanément. Par exemple, dans le domaine de l'analyse de sentiments, une revue de film pourrait être étiquetée avec "comédie", "français" et "évaluation positive" en même temps, reflétant la nature complexe des opinions humaines. De plus, contrairement à la classification multi-classes où les catégories sont mutuellement exclusives, la classification multi-labels permet une superposition de labels. Cela signifie qu'une instance textuelle peut appartenir à un ensemble variable de labels, ce qui la rend particulièrement adaptée à des situations où les frontières entre les catégories sont floues, ou, où un texte peut être lié à une multitude de concepts [20]. La classification multi-labels, englobant la classification multi-classes en tant que cas particulier, offre une représentation plus riche et nuancée des informations textuelles.

Dans la classification multi-labels, l'annotation consiste à attribuer à chaque classe ou catégorie pertinente, soit un label positif pour indiquer que la classe est présente, soit un label négatif pour signifier que la classe est absente. Généralement, un seuil de décision est déterminé : lorsque le score de prédiction du modèle d'appartenance à une classe est au-dessus (respectivement en-dessous) de ce seuil, alors l'instance obtient un label positif (respectivement négatif) sur cette classe. Cette approche d'annotation permet de représenter de manière précise et nuancée les relations entre une instance donnée et plusieurs classes. L'attribution de labels positifs et négatifs à différentes classes permet aux modèles de classifier ces instances de manière flexible en reflétant leur complexité intrinsèque. La présence de labels négatifs est essentielle car elle informe le modèle que certaines



(a) Classification binaire

(b) Classification multi-classes

(c) Classification multi-labels

FIGURE 5 – Schéma explicatif des différents types de classification.

classes ne sont pas applicables à une instance donnée, améliorant ainsi la précision des prédictions. Dans la Figure 5, l'annotation d'un label négatif est associée à une forme vide alors que celle d'un label positif est associée à une forme pleine.

2.2.1 *Catégorisation d'approches pour solutionner la problème de classification multi-labels*

Deux familles d'approches principales existent : la transformation du problème et l'adaptation d'algorithmes [224].

— **Méthodes de Transformation du Problème.** Cette catégorie d'algorithmes aborde l'apprentissage multi-étiquettes en le convertissant en d'autres scénarios d'apprentissage bien établis. Les méthodes de Pertinence Binaire (*Binary Relevance*) [21] et de Chaînes de Classifieurs (*Classifier Chains*) [171], regroupent des approches du premier ordre reposant sur la décomposition de problème multi-labels en plusieurs sous-problèmes de classification binaire. Les approches du second-ordre transforment le problème multi-labels vers une tâche de classement relatifs de labels [69]. Enfin les approches d'ordre-supérieur transforment le problème multi-labels en problèmes multi-classes [231]. Par exemple, Label Powerset transforme chaque combinaison de multiples labels comme une nouvelle classe dans un problème multi-classes [229].

— **Méthodes d'Adaptation d'Algorithmes.** La catégorie d'adaptation implique la modification d'algorithmes de base de classification simple pour traiter directement les données à multi-labels. Ces méthodes sont celles qui ajustent des modèles de classification classiques aux données multi-labels. Les modèles adaptés de cette façon sont par exemple : les arbres de décisions [40], les SVM [57] ou les réseaux neuronaux [265].

En d'autres termes, ces deux approches s'opposent. Dans la transformation du problème, l'objectif est d'adapter les données aux algorithmes, tandis que dans les méthodes d'adaptation d'algorithmes, l'objectif est d'ajuster les algorithmes aux données [266].

2.2.2 *Particularités de la tâche de classification multi-labels*

Afin de prendre en compte ces particularités, diverses problématiques sont liées, comme la gestion de la corrélation entre les labels, la prise en compte des labels manquants ou le choix de la fonction d'activation.

2.2.2.1 *Fonction d'activation multi-labels*

Dans le contexte de la classification multi-labels, où chaque instance de données peut être associée à plusieurs étiquettes simultanément, la fonction sigmoïde est très généralement utilisée pour modéliser ces relations de manière probabiliste étant notamment l'option par défaut dans les librairies de classification telles que *sci-kit learn* [1] [168] ou *keras* [2] [38]. En effet, la fonction sigmoïde représente la fonction de répartition de la loi logistique, elle est couramment utilisée pour les problèmes de classification binaire. Elle place tout nombre réel dans l'intervalle (0, 1). La fonction sigmoïde la plus commune est la fonction logistique.

Les avantages liés à son utilisation sont nombreux :

— **Interprétation probabiliste.** Les fonctions d'activation sigmoïdes offrent une interprétation probabiliste naturelle de l'appartenance à une classe. La sortie de chaque modèle/neurone/branche représente la probabilité qu'une instance soit labélisée positivement pour une classe particulière. Avec autant de fonctions d'augmentation que de sorties du modèles, plusieurs neurones peuvent être actifs simultanément rendant la fonction sigmoïde particulièrement adaptée à la classification multi-labels.

— **Compatibilité avec la Descente de Gradient.** Les fonctions sigmoïdes sont différentiables, ce qui les rend adaptées à l'entraînement de réseaux neuronaux à l'aide d'algorithmes d'optimisation basés sur le gradient, tels que la descente de gradient stochastique (SGD). Les gradients des fonctions sigmoïdes peuvent être facilement calculés, facilitant des mises à jour efficaces des poids pendant l'entraînement.

— **Non-linéarité.** Les fonctions d'activation introduisent de la non-linéarité, permettant ainsi la capture de relations complexes et la mise en évidence de frontières de décision dans les données. Cette caractéristique est cruciale pour résoudre les pro-

1. https://scikit-learn.org/
2. https://keras.io/

blèmes de classification multi-labels, où l'appartenance à une classe peut dépendre de combinaisons non triviales de caractéristiques.

— **Mise à l'échelle.** La fonction sigmoïde met en échelle sa sortie entre 0 et 1, ce qui est avantageux pour la classification multi-labels. La sortie peut être seuillée à un niveau approprié (par exemple, 0.5) pour déterminer l'appartenance à une classe, offrant ainsi une manière simple d'interpréter les résultats.

La fonction softmax, utilisée dans les problèmes multi-classes, prend en entrée un vecteur de scores, où chaque score est associé à une classe potentielle et normalise ces scores de manière à ce que leur somme soit égale à 1, permettant ainsi d'interpréter les sorties comme des probabilités conditionnelles. Les classes sont donc mutuellement exclusives, chaque classe recevant une part des probabilités totales. En revanche, les fonctions sigmoïdes produisent des sorties binaires, ce qui signifie que pour un problème de classification multi-labels, autant de fonctions sigmoïdes doivent être utilisées que de labels. Les classes sont donc indépendantes, chaque classe ayant une probabilité d'appartenance distincte.

Cependant les fonctions sigmoïdes ne sont pas exemptes de limites. En effet, en présence de valeurs d'entrée fortement positives ou négatives, les fonctions activations sigmoïdes se saturent, la sortie de la fonction devenant très proche de 0 ou 1. Des valeurs d'entrées extrêmes déséquilibrent, au sein des classes représentées, les classes sous-représentées et sur-représentées. Dans cet état, quand le gradient du neurone est proche de zéro, cela entraîne un apprentissage très lent [90]. Cela peut rendre difficile, pour un réseau, de s'adapter à des données comportant des valeurs extrêmes [16]. De plus, les fonctions sigmoïdes ne sont pas calibrées ce qui peut fortement affecter l'explicabilité des modèles associés [123]. La calibration se réfère à la correspondance entre les prédictions d'un modèle et la probabilité réelle qu'un événement se produise.

Bien que d'autres fonctions d'activation comme ReLU [1] ou tanh [114] solutionnent certaines de ces limites, les fonctions sigmoïdes sont, dans la grande majorités des cas, favorisées dans les travaux multi-labels [162].

2.2.2.2 *Déséquilibre entre labels*

Le déséquilibre des labels constitue un défi majeur dans le domaine de la classification multi-labels. Ce déséquilibre peut se manifester de deux manières : d'abord, par une distribution inégale des classes, où certaines classes sont prédominantes, et ensuite, par un déséquilibre dans le rapport entre les labels positifs et négatifs [16]. Ces disparités posent des défis substantiels pour la performance des modèles. En effet, une gestion adéquate de ces déséquilibres est essentielle afin de garantir que les modèles puissent prendre en compte toutes les

classes de manière équitable, améliorant ainsi leur capacité de généralisation et de prise de décision précise. Ces approches peuvent être catégorisés en quatre familles [222] :

— **Adaptation de classifieur.** Ces approches abordent le déséquilibre entre les labels en adaptant les algorithmes d'apprentissage du modèle de classification. Généralement, cette adaptation se fait en associant des poids différents aux classes pendant l'entraînement en fonction de leur rareté, afin de corriger les biais causés par la sous-représentation de certains labels [36]. Ces poids peuvent, par exemple, être pris en compte dans la fonction de perte, élément clé de l'apprentissage, de manière à tenir compte de la rareté des labels [178]. Ces poids peuvent aussi intervenir dans la structure même des réseaux neuronaux afin de, par exemple, augmenter les valeurs de sorties des chemins neuronaux associés aux labels rares [263].

— **Approches par ensembles.** Certaines méthodes d'apprentissage, par ensemble de classifieurs qui combinent les labels d'origines dans des sous-espaces de labels, ont tendance à aggraver le déséquilibre des labels [231]. Afin de remédier à cela, ces approches doivent être modifiées pour prendre en compte la rareté des labels annotés. Une telle modification consiste en la transformation d'algorithme d'AA en remplaçant l'objectif de sélection d'une instance par un objectif de sélection d'un sous-espace des labels. L'objectif à chaque round d'annotation devient donc d'annoter le sous-espace des labels qui contient les labels les plus rares [238].

— **Méthodes de ré-échantillonnage.** Ces approches de pré-traitement consistent à produire une version du jeux d'entraînement possédant une distribution de labels plus équilibrée que les données d'entraînement de départ. Cette équilibrage peut être réalisé de deux manières principales : en réduisant le nombre d'instances associées aux labels les plus fréquents, ce que l'on appelle le sous-échantillonnage [135], ou en augmentant le nombre d'instances associées aux labels minoritaires, ce qui est connu sous le nom de sur-échantillonnage [183]. Le sous-échantillonnage consiste simplement à ne pas sélectionner les instances liées aux labels majoritaires, tandis que le sur-échantillonnage fonctionne en générant de nouvelles instances avec des labels rares. Les deux approches ne sont pas exclusives et peuvent être combinées en sous-échantillonnant les classes sur-représentées et sur-échantillonnant les classes rares [218].

— **Méthodes par coût.** Ces approches attribuent un coût d'annotation à chaque instance, avec pour objectif la minimisation des coûts totaux. Une stratégie, pour intégrer cette approche dans le contexte du déséquilibre des classes présent dans les problèmes multi-labels, consiste à ajuster à la baisse (ou à la hausse) le

coût d'une instance en fonction de la rareté (ou de l'abondance) de ses étiquettes [139]. L'approche la plus populaire consiste à modifier les seuils de décision de chaque labels dans le but d'acquérir plus d'instances annotées avec des labels rares [6].

La plupart des méthodes présentées ci-dessus se concentrent sur le déséquilibre entre les classes des labels. Cependant, une autre forme de déséquilibre existe, à savoir la disparité entre la quantité de labels positifs et négatifs. En effet, dans la plupart des problèmes de classification multi-labels, la proportion de labels positifs est nettement inférieure à celle des labels négatifs [262]. Cela crée un biais en faveur de l'amélioration de la classification pour l'absence d'appartenance à une classe plutôt que pour l'appartenance à une classe. Afin de tenir compte de ce déséquilibre, des modifications sont apportées aux fonctions d'apprentissage, telles que les fonctions de perte, pour accroître l'influence des labels positifs sur l'entraînement des modèles [16].

2.2.2.3 *Corrélation entre labels*

La prise en compte des corrélations, ou dépendances, entre les labels au sein des problèmes de classification multi-labels s'avère cruciale pour renforcer la précision des modèles. En effet, dans certaines configurations de jeux de données et leurs espaces de labels associés, les relations entre ces labels renferment une richesse d'informations. Ce phénomène se manifeste notamment lorsque des hiérarchies entre les labels sont présentes. Ainsi, lorsqu'un label spécifique est attribué à une instance, il est quasiment inévitable que des labels-parents soient également associés à cette même instance. Cela peut aussi se vérifier dans des cas où des labels partagent un sous-domaine similaire dans la tâche de classification. À titre d'exemple, dans un système de classification des avis clients, il est raisonnable de penser que les labels concernant les restaurants et ceux relatifs aux boutiques soient en parti distincts. Par exemple, il est bien plus probable qu'une instance étiquetée "taille des portions" soit également associée au label "sélection de vins" plutôt qu'aux labels "gamme des tailles" ou "marques de renom". Une approche visuelle utile pour appréhender la corrélation entre les labels réside dans l'utilisation de matrices de corrélation, qui rendent clairement visibles les relations entre les paires de labels.

Dans la Figure 6, les corrélations entre les labels sont clairement visibles, ce qui permet même d'inférer leurs domaines associés. Les labels regroupés dans le coin supérieur gauche présentent une forte corrélation entre eux, ce qui suggère qu'ils sont associés au même domaine, en l'occurence celui des restaurants. De même, les labels situés dans le coin inférieur droit semblent être davantage liés entre eux et correspondent au domaine des boutiques. Les labels "prix" et "livraison rapide" affichent des corrélations partielles avec les deux parties de la matrice, suggérant que les concepts associés à ces labels appartiennent potentiellement aux deux domaines simultanément.

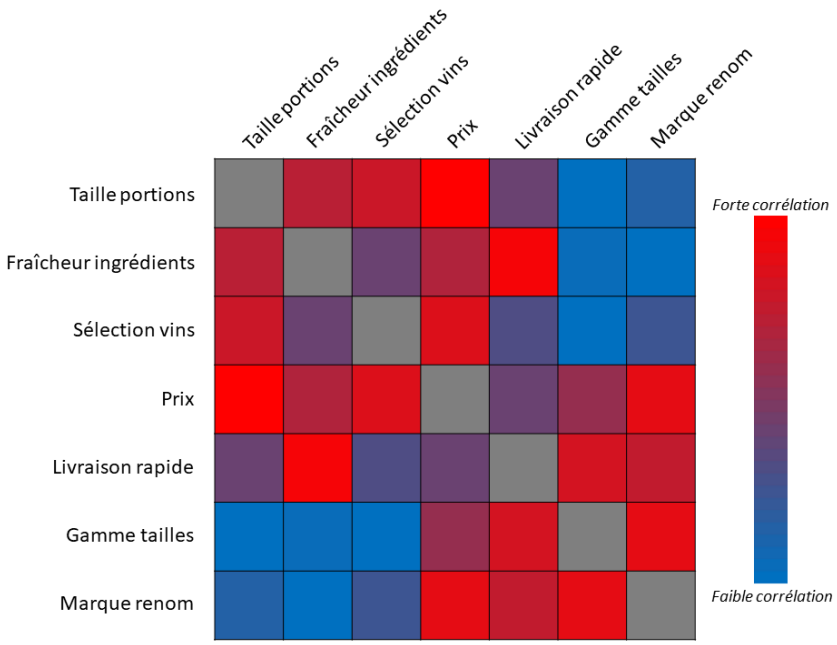


FIGURE 6 – Exemple d'une matrice de corrélation

Les méthodes de classification multi-labels basées sur la corrélation entre les labels sont divisées en trois catégories : les méthodes du premier ordre, du deuxième ordre et d'ordre supérieur. Chacune de ces catégories aborde le problème de la classification multi-labels en exploitant différemment les relations entre les labels [266].

— **Méthodes du premier ordre.** Chaque label est traité individuellement et la corrélation entre labels est ignorée. Le problème de classification multi-labels est décomposé en autant de sous-problèmes de classification binaires indépendants qu'il y a de labels [21]. La simplicité conceptuelle et la rapidité d'exécution de ces méthodes se font au détriment de la prise en compte des interdépendances entre les labels, seules les particularités associées à chaque label sont prises en compte [264].

— **Méthodes du second ordre.** Les labels sont traités par paires. Les relations par paires peuvent être inductives, exclusives, indépendantes [72, 95, 96] ou encore une comparaison de pertinence d'un label par rapport à l'autre pour une instance donnée [69, 275]. Cependant, il existe certaines applications du monde réel où les corrélations entre les labels vont au-delà de l'hypothèse du deuxième ordre.

— **Méthodes d'ordre supérieur.** Les labels sont traités au moyen de diverses combinaisons, allant au-delà de simples paires. Les relations prises en compte sont plus complexes comme les corrélations entre des sous-ensembles de labels [142, 172, 231] ou encore l'influence de l'ensemble des labels sur chaque label [37, 74, 109, 133]. Ces méthodes ont pour principal désavantage leur

FIGURE 7 – Légende des symboles de la Figure 5 prenant en compte l'absence de labels

difficulté à supporter un passage à l'échelle dû à l'augmentation du besoin en termes de ressources computationnelles.

En fin de compte, le choix de la méthode à adopter dépendra principalement de deux critères majeurs : les caractéristiques du jeu de données et de l'espace de labels associé (hiérarchie des labels, forte corrélation entre les labels ou non, ...) ainsi que des ressources computationnelles disponibles. Il est en effet important de trouver un équilibre entre la capacité de modélisation offerte par les méthodes d'ordre supérieur et la faisabilité en termes de calcul.

2.2.2.4 *Labels manquants*

Dans le contexte de l'annotation multi-labels, il est souvent ardu d'attribuer de manière exhaustive tous les labels associés à une instance, en particulier lorsque l'espace des labels est vaste. Dans de telles situations, seuls certains labels sont annotés et utilisés pour l'entraînement des modèles [217]. Dans la Figure 5, une illustration visuelle de l'absence d'un label dans la classification multi-labels serait l'équivalent de l'absence de l'une ou de plusieurs des formes qui sont normalement associées à une instance, voire nécessiterait la mise en place d'une troisième catégorie de remplissage de symbole comme dans la Figure 7. Certaines architectures de modèles ne sont pas nécessairement affectées par cette problématique. Par exemple, l'approche de décomposition en sous-problèmes de classification binaire, parfois utilisée dans les problèmes de classification multi-labels, offre une solution simple pour traiter les labels manquants. En abordant chaque label comme une tâche distincte, lorsque qu'un label est absent pour une instance donnée, la branche correspondante de l'architecture est tout simplement ignorée lors de l'étape d'entraînement. Cela maintient la simplicité de la structure du modèle tout en permettant une gestion fluide des labels manquants [261]. Pour relever ce défi avec des architectures ne gérant pas nativement cette absence de labels, les techniques de classification multi-labels avec des labels

manquants, *Multi-Label Learning with Missing Labels (MLML)*, peuvent être regroupées en trois catégories distinctes [138] :

— La première catégorie d'approches, qui considère les labels manquants comme des annotations négatives, offre une solution élégante pour contourner les difficultés associées à la gestion de labels manquants. Elle permet d'utiliser des modèles de classification multi-labels standard sans nécessiter de modifications majeures. Cependant, cette approche comporte un inconvénient essentiel : elle traite les labels manquants comme s'ils avaient une signification négative, ce qui peut potentiellement entraîner des biais dans l'entraînement du modèle. En d'autres termes, elle ne capture pas la différence fondamentale entre une annotation négative et l'absence totale d'annotation [260].

— La deuxième catégorie d'approches, qui prend en compte la différence entre une annotation positive, une annotation négative et l'absence totale d'annotation, est plus nuancée. Ces méthodes reconnaissent que l'information contenue dans une annotation négative diffère de celle qui provient de l'absence totale d'annotation. Pour cela, elles exigent souvent des ajustements au niveau de l'architecture du modèle ou l'utilisation de modèles spécifiquement conçus pour traiter correctement les labels manquants. Cependant, ces solutions offrent une plus grande précision en raison de leur capacité à différencier les diverses formes d'annotations [245]. L'une des approches les plus agnostiques en termes de modèle, au sein de cette catégorie, consiste à modifier la fonction de perte, étant donné que de nombreux modèles d'apprentissage s'appuient sur ce type de fonction [106]. Par exemple, il est possible de personnaliser la très répandue fonction d'entropie croisée binaire, *binary cross-entropy*, afin de prendre en compte l'absence de labels au sein de diverses architectures [53].

— La troisième catégorie d'approches propose une solution différente en remplaçant les labels manquants. Contrairement à la deuxième catégorie, ces méthodes permettent de traiter le problème des labels manquants sans avoir besoin de modifier l'architecture classique de la classification multi-labels. Les approches les plus populaires de cette catégorie se fondent sur des techniques de corrélation entre labels (cf Partie 2.2.2.3), pour remplacer les labels manquants par leur annotation la plus probable, en prenant en considération les relations entre les labels déjà obtenues [97].

La création de modèles capables de s'adapter à des jeux de données partiellement annotés ne se limite pas aux situations où l'annotation exhaustive est difficile. Ces modèles jouent un rôle central dans les stratégies d'AA axées sur la sélection des paires instance-label [78]. Plutôt que de se concentrer sur l'annotation complète d'instances in-

dividuelles, ces approches visent à une sélection plus fine : choisir une instance spécifique et un label à annoter. Cette approche présente des avantages significatifs. Garantissant notamment que chaque annotation apportée est véritablement cruciale pour le modèle, améliorant ainsi l'efficacité globale du processus d'annotation. En d'autres termes, chaque label ajouté est soigneusement sélectionné pour maximiser sa valeur pour le modèle. Cependant, il existe un revers à cette médaille. Cette approche transfère une partie de l'activité de l'annotateur d'une manière qui peut ne pas être optimale. Étant donné que l'annotation se concentre sur des paires instance-label, le temps consacré à chaque instance est réduit, ce qui signifie que les annotateurs passent moins de temps sur l'annotation proprement dite et davantage sur la lecture d'instances.

2.2.3 *Stratégies d'apprentissage multi-labels étudiées*

La tâche de classification multi-labels consiste à assigner les labels appropriés à des instances textuelles. Contrairement à la classification multi-classes, plusieurs labels peuvent être associés à une même instance. Pour chacune de nos expériences, notre espace de label est prédéfini et n'évolue pas au cours de la tâche. L'objectif de l'AA est de sélectionner les meilleures instances possibles à annoter pour l'entraînement. Cette sélection peut se faire selon différentes *stratégies*. Nos stratégies sont basées sur l'estimation de l'*incertitude* du modèle sur chaque instance, c'est-à-dire la confiance du modèle dans la prédiction des labels associés à cette instance. Ces stratégies reposent sur l'hypothèse qu'en s'entraînant sur des exemples difficiles (où le modèle hésite), le modèle va gagner en performance.

Comme suggéré dans [193], nous concentrons notre étude sur des stratégies d'AA basées sur l'incertitude et plus précisément sur six stratégies d'AA multi-labels. Nous utilisons les notations introduites dans la Partie 2.1.1 afin de définir ces stratégies plus en détail :
Max Loss (ML) sélectionne les instances pour lesquelles la fonction de perte est la plus élevée [130] :

$$\underset{x_i}{\operatorname{argmax}}\Big[\sum_{j=1}^{q} \max\{1 - m_j * f_j(x_i), 0\}\Big] \tag{6}$$

où $m_j = 1$ si $j = u$, $m_j = -1$ sinon, u correspondant au label l^u associé avec la plus grande probabilité à une instance donnée et où $f_j(x_i)$ est défini par :

$$f_j(x_i) = 2 * y_i^j - 1 \tag{7}$$

Mean Max Loss (MML) sélectionne les instances pour lesquelles la fonction de perte moyenne est la plus élevée [130] :

$$\underset{x_i}{\operatorname{argmax}} \frac{1}{q}\Big[\sum_{k=1}^{q}\sum_{j=1}^{q} \max\{1 - o_{kj} * f_j(x_i), 0\}\Big] \tag{8}$$

où $o_{kj} = 1$ si $j = k$, $o_{kj} = -1$ sinon et $f_j(x_i)$ est défini dans (7).

A l'origine conçues pour des modèles SVM dans le contexte de la classification multi-labels d'images [130], les stratégies Max Loss (ML) et Mean Max Loss (MML) se distinguent par leur simplicité conceptuelle et de leur facilité d'application. Ainsi ces stratégies offrent une opportunité précieuse pour évaluer une implémentation basique de l'incertitude appliquée aux transformers.

Minimum Confidence Min/Max No weighting (CMN) sélectionne les instances pour lesquelles la confiance du modèle est la plus basse [63] :

$$\underset{x_i}{\operatorname{argmin}}\Big(\min_{j=1}^{q} f_j(x_i)\Big) \tag{9}$$

avec $f_j(x_i)$ défini dans (7).

La stratégie CMN a été spécifiquement élaborée pour aborder la catégorisation de documents textuels à étiquettes multiples, en se basant sur une décomposition binaire du problème et l'exploitation de modèles reposant sur le *boosting*, une technique d'ensemble bien connue [62]. Chacune des initiales de CMN représente une sélection particulière de méthode dans l'une des trois dimensions distinctes du problème multi-labels [63] :

— **C** est associé à la dimension *Evidence* et à la sélection de la méthode *Minimum Confidence* impliquant le choix de se baser sur les scores de confiance faibles plutôt que de privilégier les scores élevés de confiance.

— **M** est associé à la dimension *Class* et à la sélection de la méthode *Min/Max* impliquant le choix de se concentrer sur les instances où les scores de confiances sont extrêmes pour une classe (score de confiance *Min*imum quand associé à *Minimum Confidence*) plutôt que de réaliser une moyenne des scores de confiance sur toutes les classes.

— **N** est associé à la dimension *Weight* et à la sélection de la méthode *No Weighting* impliquant le choix de traiter toutes les classes de façon similaire plutôt que de réaliser une pondération pour se focaliser sur les scores associés aux classes sur lesquelles le modèle performe moins bien.

Cette synergie de méthodes dans les trois dimensions s'est révélée être la plus performante en termes de résultats obtenus dans les travaux présentant cette stratégie et comparant différentes approches [63].

Max Margin Uncertainty sampling (MMU) sélectionne les instances qui ont la plus petite marge de séparation entre les groupes prédits de labels positifs et négatifs [131] :

$$\underset{x_i}{\operatorname{argmax}} \frac{1}{\min pos(x_i) - \max neg(x_i)} \tag{10}$$

où $pos(x_i) = [pos_1(x_i), ..., pos_q(x_i)]$ et $neg(x_i) = [neg_1(x_i), ..., neg_q(x_i)]$, avec :

$$pos_j(x_i) = \begin{cases} f_j(x_i) & \text{si } f_j(x_i) > 0 \\ +\infty & \text{sinon} \end{cases} \quad \text{et} \tag{11}$$

$$neg_j(x_i) = \begin{cases} f_j(x_i) & \text{si } f_j(x_i) < 0 \\ -\infty & \text{sinon} \end{cases} \tag{12}$$

avec $f_j(x_i)$ défini dans (7).

La stratégie MMU a été élaborée pour des modèles SVM et évaluée sur des tâches de classification multi-labels d'images et de textes. Le principe suivi dans cette stratégie est que pour une instance sur laquelle le modèle est certain, il est certain des labels positifs et négatifs associés [82]. Ainsi, l'instance la plus incertaine est celle ayant la marge de séparation la plus petite entre la valeur de prédiction du label positif le plus bas et la valeur de prédiction du label négatif le plus haut.

Label Cardinality Inconsistency (LCI) sélectionne les instances qui maximisent la distance euclidienne entre le nombre de labels positifs prédits et la cardinalité des labels du jeu de données annotées [131] :

$$\underset{x_i}{\operatorname{argmax}} \sqrt{\left(\left(\sum_{j=1}^{q} y_i^j\right) - L\right)^2} \tag{13}$$

avec L le nombre moyen de labels (cardinalité) sur les instances déjà annotées.

Comme la stratégie précédente, LCI a été élaborée pour des modèles SVM et évaluée sur des tâches de classification multi-labels d'images et de textes. Bien que présentée dans son papier comme mesure d'incertitude, elle correspondrait mieux à une mesure de prédiction de performance d'après la catégorisation réalisée dans la Partie 2.1.2.3.

Category Vector Inconsistency and Ranking of Scores (CVIRS) sélectionne les instances en combinant deux mesures. La première mesure, est basée sur une agrégation de rang des marges de différence

entre la prédiction de positivité ou de négativité d'un label par le classifieur [176].

Pour chaque label de chaque instance on calcule une marge de différence entre la probabilité que ce label soit positif ou négatif pour cette instance :

$$m(i,j) = y_i^j - (1 - y_i^j) \tag{14}$$

Pour chaque label, on ordonne la liste d'instances non-annotées du score *m(i,j)* le plus petit (instance la plus incertaine puisque la marge est faible) au plus grand. Ainsi on obtient q listes ordonnées où chaque instance à potentiellement un rang différent dans chaque liste. Le problème consistant à déterminer l'instance qui a globalement le rang le plus haut dans ces listes est un problème d'agrégation de rang classique. La méthode de Borda est utilisée pour le résoudre dans cette stratégie [54]. Cette méthode est dite positionnelle puisqu'elle associe un score de position à chaque instance dans chaque liste afin de calculer son rang global.

La seconde mesure est basée sur l'incohérence des ensembles de labels prédits par rapport aux labels de l'ensemble des instances déjà annotées. Cette mesure correspond à une mesure de prédiction de performance d'après la catégorisation réalisée dans la Partie 2.1.2.3. Cette incohérence est égale à la moyenne des distances entropiques [79] entre le vecteur de prédiction de l'instance non-annotée et les vecteurs de labels des instances déjà annotées. Plus la structure des labels prédits d'une instance diffère des structures de labels déjà annotés, plus cette instance est considérée comme intéressante à annoter.

Ces deux mesures sont ensuite combinées pour chaque instance par multiplication afin d'obtenir le score d'intérêt d'une instance.

Ces stratégies sont dites *myopes*, dans le sens où elles évaluent l'incertitude instance par instance. Bien qu'il existe des stratégies prenant en compte la composition des lots [78], les travaux appliquant des stratégies d'AA aux transformers favorisent principalement des stratégies *myopes*, de part leur rapidité de calculs. Comme dans [176], nous adaptons ces stratégies à l'apprentissage par lots de données simplement : au lieu de sélectionner uniquement l'instance pour laquelle le modèle est le plus incertain, nous sélectionnons les instances les plus incertaines pour remplir notre lot d'entraînement.

2.3 APPRENTISSAGE ACTIF SUR LA TÂCHE DE CLASSIFICATION MULTI-LABELS (AAML)

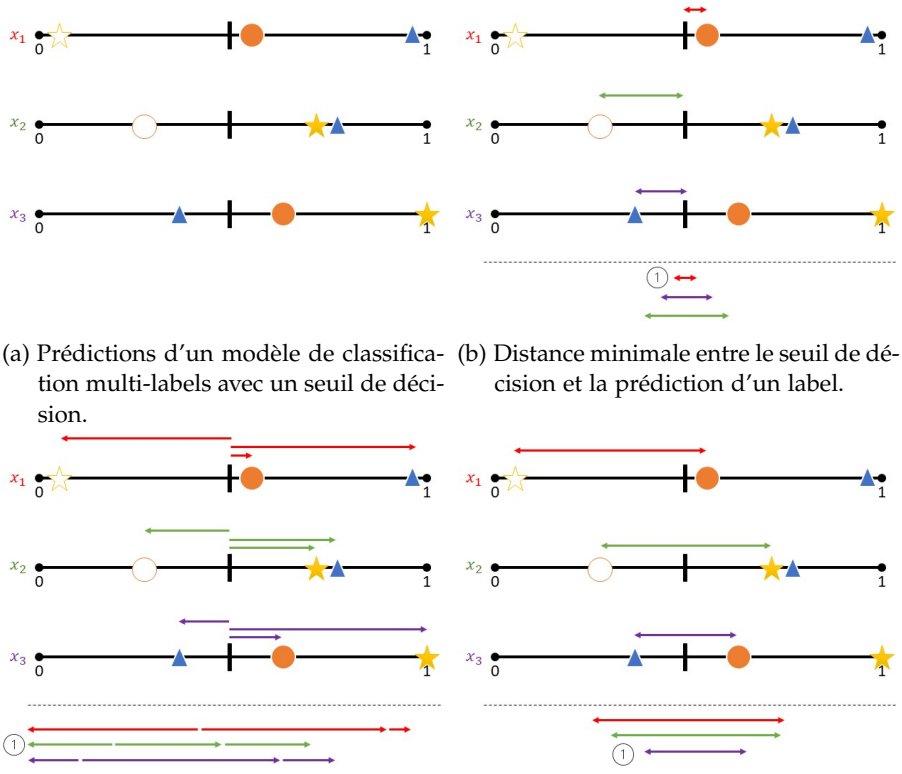
L'un des premiers aspects à considérer dans la convergence de l'apprentissage actif et de la classification multi-étiquettes concerne la problématique de déséquilibre. En effet, l'AA peut involontairement biaiser le modèle en faveur des labels fréquemment observés et

donc aggraver le problème de déséquilibre présent dans les données multi-labels en sélectionnant principalement une classe d'instances [13]. Des travaux inspirants ont émergé dans le domaine de la classification multi-classes couplée à l'apprentissage actif, où, si des seuils de déséquilibre sont atteints pendant le cycle d'apprentissage actif standard, celui-ci est interrompu et des étapes de rééquilibrage sont exécutées [3]. Les mêmes auteurs se sont également penchés sur la résolution directe de ce déséquilibre tout au long du cycle d'apprentissage actif en ne sélectionnant que les scores d'instances provenant des classes minoritaires [4]. Dans la Figure 3, nous avons observé que les stratégies d'AA peuvent être classées en fonction de leur orientation vers l'informativité ou la représentativité, ainsi que par leur méthode de sélection d'instances, qui peut se faire soit de manière individuelle, soit sous forme de lot. Ces catégorisations sont également applicables pour classer les stratégies d'AA dédiées à la tâche de classification multi-labels (AAML). Bien que certaines catégorisations soient communes aux tâches de classification binaire et multi-classe, les différences entre ces tâches se manifestent dans la façon dont ces catégorisations sont mises en œuvre. De plus, pour la classification multi-labels, une catégorisation spécifique peut être ajoutée, distinguant les stratégies qui sélectionnent des instances à annoter de manière exhaustive de celles qui choisissent des paires instance-étiquette à annoter. Malgré l'aperçu des différentes approches d'AAML présentées dans ce chapitre, il convient de souligner que les stratégies d'AAML évaluées dans le cadre de cette thèse sont exposées en détail dans la Partie 2.2.3.

2.3.1 *AAML : informativité vs représentativité*

La tâche de classification multi-labels est couramment formalisée comme suit : chaque classe peut être étiquetée avec un label parmi deux valeurs, généralement $0, 1$, bien que dans certaines études, on puisse également les voir égales à $-1, 1$. Les scores de prédiction des modèles de classification se situent donc dans l'intervalle $[a, b]$. Selon le modèle, plus le score de prédiction se rapproche de a (respectivement de b), plus la probabilité que le label associé à la classe en question soit égal à a (respectivement à b) est élevée.

Cependant, étant donné que le modèle doit classifier, chaque classe doit être associée à un label. Par conséquent, il existe une valeur seuil dans l'intervalle $[a, b]$, en dessous (respectivement au-dessus) de laquelle le label associé est a (respectivement b). Cette valeur seuil est appelée le seuil de décision et est généralement établie au milieu de l'intervalle $[a, b]$, c'est-à-dire à $\frac{a+b}{2}$. En pratique, en fonction des particularités des ensembles de données, il peut être pertinent de déplacer ce seuil de décision [61].



(a) Prédictions d'un modèle de classification multi-labels avec un seuil de décision.

(b) Distance minimale entre le seuil de décision et la prédiction d'un label.

(c) Distance moyenne entre le seuil de décision et la prédiction des labels.

(d) Distance minimale entre les deux prédictions les plus proches du seuil de décision.

FIGURE 8 – Exemple de diverses mesures d'incertitude dans l'AAML. Le ① correspond à l'instance sélectionnée par apprentissage actif.

Dans l'AAML, l'informativité et plus précisément l'incertitude des modèles est souvent calculée à l'aide du seuil de décision. En effet, une des manières de conceptualiser l'incertitude du modèle sur une instance, est de regarder la distance entre les prédictions de labels du modèle sur celle-ci et le seuil de décision. Divers exemples de stratégies sont illustrés dans la Figure 8.

Dans la sous-figure 8b, une incertitude plus élevée est associée aux instances dont la prédiction pour l'un de leurs labels est la plus proche de leur seuil de décision [63]. Cette approche s'avère particulièrement efficace avec les modèles dont les prédictions sont fortement concentrées vers les extrémités de l'intervalle, notamment grâce à l'utilisation d'une fonction logistique.

Dans la sous-figure 8c, une incertitude plus élevée correspond aux instances pour lesquelles la distance moyenne entre les prédictions de labels et le seuil de décision est la plus grande. Cependant, cette approche peut passer à côté d'instances avec des niveaux de confiance très faibles pour certains labels si ces faibles niveaux de confiance sont compensés par d'autres labels.

Dans la sous-figure 8d, pour une instance donnée, on sélectionne de chaque côté du seuil de décision la prédiction la plus proche du seuil. L'incertitude de l'instance est alors liée à la distance entre ces deux prédictions, comme démontré dans une étude antérieure [131]. Cependant, il convient de noter que ce calcul de l'incertitude ne repose que sur deux labels, ce qui peut se révéler peu efficace lorsque l'espace des labels est étendu ou complexe.

En ce qui concerne la représentativité dans l'AAML, l'aspect distinct par rapport à l'AA pour les tâches de classification binaire ou multi-classes réside dans l'espace des labels [99]. Par exemple, l'approche peut consister à comparer la distribution des labels prédits avec la distribution des instances déjà annotées [176]. Une autre approche consiste à utiliser des mesures de corrélation entre labels pour encourager l'annotation de labels rares [256]

La manière de combiner l'informativité et la représentativité demeure en grande partie similaire aux méthodes employées dans l'AA dédié à d'autres tâches, comme cela a été présenté dans la Partie 2.1.5.

2.3.2 *AAML : instance individuelle vs lot de données*

La plus plupart des stratégies d'AA sont dites myopes [176], c'est-à-dire qu'elles sélectionnent les instances pour être annotées séquentiellement, une par une. Cette famille d'approches s'opposent à celles par lot de données qui consistent à la composition d'un groupe de plusieurs instances qui seront annotées en même temps.

L'utilisation de stratégies d'AAML myope est accompagnée du risque la redondance d'information lors de l'annotation. Plus spécifiquement, dans le cas des stratégies basées sur l'incertitude, si le modèle éprouve des difficultés persistantes à classifier certaines instances, l'algorithme de sélection peut se concentrer sur des instances similaires entre elles. Cependant, l'annotation de ces instances similaires ne permet ni aux modèles de s'améliorer significativement, ni d'apporter des informations nouvelles. Par conséquent, il est possible que les stratégies d'AAML entraînent des modèles moins performants que ce que l'on obtiendrait en utilisant un échantillonnage aléatoire des instances à annoter [202]. Pourtant, il est important de noter que ces stratégies demeurent les plus couramment employées, principalement pour leur rapidité de calcul, ainsi qu'en raison de leur simplicité conceptuelle, qui facilite leur intégration dans la plupart des architectures de classification.

En effet, la principale faiblesse des stratégies d'AAML par composition de lot de données est qu'elles s'accompagnent de calculs combinatoires conduisant à un surcoût computationnel [177]. Autrement, ces stratégies surpassent généralement les performances des approches myopes, notamment dû au fait qu'elles facilitent la com-

binaison des approches par informativité et représentativité, constituant des lots de données plus optimaux [78].

Ces deux approches partagent certaines faiblesses avec les tâches de classification binaire et multi-classes, notamment le fait que les stratégies visant à construire un lot de données optimal entraînent souvent un coût computationnel élevé. De plus, les stratégies basées sur des instances individuelles peuvent induire une redondance lors de l'annotation, en sélectionnant des instances similaires entre elles.

2.3.3 *AAML : instance vs paire instance-label*

L'enjeu pour entraîner un modèle avec de l'AA consiste à l'élaboration d'une stratégie pour sélectionner une instance à annoter plutôt qu'une autre [199]. La plupart des travaux sur l'AAML suivent ce principe, faisant ensuite une requête à un oracle pour obtenir tous les labels pour une instance sélectionnée [176]. Cela peut conduire à une redondance lorsque, par exemple, deux labels sont fortement corrélés. Dans ce cas, l'annotation du deuxième label ne fournit pas beaucoup d'informations supplémentaires si l'instance a déjà été annotée avec le premier label.

D'autres approches prennent donc le parti de faire une sélection plus fine et de donner à annoter une paire instance-label. Cela évite efficacement la redondance d'information mais peut rendre plus difficile l'apprentissage des corrélations entre labels pour le modèle [99, 100]. Ces approches permettent d'obtenir des performances similaires aux stratégies par sélection d'instance tout en réduisant le nombre d'annotation de label à effectuer [71] (même si le nombre d'instance à lire par l'annotateur peut être supérieur).

Ces méthodes gagnent un intérêt lorsqu'elles sont combinées à des approches d'apprentissage par lot de données. En effet, lors de la composition de lot de pair instance-label les intéractions entre labels comme les déséquilibre ou les corrélations peuvent être prises en compte [78].

Une alternative proche des méthodes de sélection de paire instance-label consiste à présenter à un annotateur deux labels potentiels associés à une instance et où la tâche de l'annotateur consiste à choisir [98].

2.4 DISCUSSION ET CONCLUSION

Dans ce chapitre, nous avons réalisé un tour des travaux existants dans deux domaines de l'apprentissage automatique et leur intersection : l'apprentissage actif (AA) et la classification multi-labels. En premier lieu, nous avons abordé l'AA, son cadre ainsi que les notations associées. Nous avons aussi présenté un classement de ces stratégies en fonction de leur orientation vers l'informativité ou la re-

présentativité. Nous avons examiné de manière plus précise diverses sous-catégories d'approches, notamment celles basées sur l'incertitude des modèles, le désaccord entre les modèles, et la prédiction de performance, ainsi que les stratégies fondées sur la densité, la discrimination, et la diversité. De plus, nous avons discuté de comment ces différentes approches peuvent être combinées entre elles. Enfin, nous avons discuté des raisons pratiques donnant de l'intérêt à l'AA par lot de données, tout en mettant en évidence que la place de l'humain dans l'AA est centrale. Il est essentiel de noter que ces différentes approches de l'AA ont chacune leurs avantages et leurs inconvénients. Le choix de sélectionner ou de combiner ces approches dépend des caractéristiques propres à l'ensemble de données, au modèle de classification, ainsi qu'à l'optimisation des gains et des coûts apportés par chaque stratégie.

Ensuite, nous nous sommes concentrés sur la classification multi-labels. Nous avons exploré les diverses catégories d'approches visant à résoudre ce problème. Nous avons mis en évidence ses particularités, telles que le déséquilibre entre les labels, la corrélation entre les labels et les labels manquants. Nous avons également discuté des forces et faiblesses de la fonction d'activation multi-labels la plus souvent utilisée, la fonction logistique. Cette partie nous a permis de mettre en évidence la complexité inhérente à cette tâche, laquelle se révèle plus exigeante en comparaison avec la classification binaire ou multi-classes. Il est important de souligner que la prise en compte croissante des diverses facettes complexes de l'espace des labels peut entraîner des coûts significatifs. Lorsque l'on met en place des solutions pour la classification multi-labels, il est donc impératif de réaliser des choix tenant compte à la fois des gains potentiels et des coûts associés.

Enfin, nous avons exploré l'intersection de ces deux domaines en examinant l'application de l'AA à la classification multi-labels (AAML). En effet, les approches qui émergent à la convergence de ces deux domaines peuvent être catégorisées en tenant compte des catégories issues de chacun d'eux. La convergence de ces deux domaines combine leurs défis spécifiques, créant ainsi des défis propres à cette intersection, que nous avons explorés en détail. L'étude de l'AAML revêt donc un intérêt particulier, notamment du fait de l'ambivalence de l'objectif : trouver des solutions frugales pour aborder des problématiques complexes.

En conclusion, ce chapitre a posé les bases indispensables pour comprendre les concepts et les méthodes liés à l'AA et à la classification multi-labels. En explorant l'état de l'art des méthodes visant à améliorer les performances des modèles tout en réduisant les coûts d'annotation, nous commençons à entrevoir plusieurs perspectives de recherche prometteuses, certaines desquelles seront approfondies dans les contributions de cette thèse.

3

APPRENTISSAGE PROFOND ACTIF

L'Apprentissage Actif Profond (AAP) est le domaine issu de la convergence de l'Apprentissage Actif (AA) et de l'Apprentissage Profond (AP), *deep learning*. Dans le Chapitre 2 nous avons abordé en détail des sujets autour de l'AA. Dans ce chapitre nous nous concentrerons notamment sur les spécificités des modèles d'AP et de ce que l'intégration de l'AA avec ce type de modèle implique.

Tout d'abord, il est important de noter que l'intérêt de l'utilisation de l'AA et AP a grandi en raison du développement rapide de l'accès à l'information, qui nous a plongés dans une ère caractérisée par une abondance d'informations et une quantité massive de données disponibles. Les modèles d'AP sont connus pour leurs appétits insatiables en matière de données. En effet, leur entraînement nécessite une quantité considérable de données étiquetées pour optimiser un grand nombre de paramètres et extraire des caractéristiques de haute qualité. Cependant, l'acquisition d'un grand nombre de jeux de données annotés de haute qualité nécessite des ressources considérables en termes de main-d'œuvre, rendant cette approche peu réalisable dans des domaines exigeant un haut niveau d'expertise, tels que la reconnaissance vocale, l'extraction d'informations ou l'analyse d'images médicales, entre autres. Par conséquent, l'ajout de stratégies d'AA qui permettent de réduire les coûts liés à l'annotation en sélectionnant la meilleure petite proportion d'échantillons parmi les données non étiquetées pour l'apprentissage, devient intéressant.

Ainsi, il est naturel d'explorer comment l'AA peut être utilisé pour minimiser les coûts liés à l'annotation d'échantillons, tout en optimisant les performances des modèles d'AP, donnant naissance à l'AAP. Cette approche représente une solution viable pour maximiser les performances du modèle d'AP dans un environnement où les coûts d'annotation sont limités.

3.1 APPRENTISSAGE ACTIF AVEC DES RÉSEAUX NEURONAUX

Les réseaux de neurones profonds (DNN pour *deep neural networks*), représentent une avancée majeure dans le domaine de l'apprentissage automatique. Ces réseaux tentent de simuler la structure du cerveau humain pour construire des modèles d'apprentissage. Ces modèles bénéficient d'une capacité d'apprentissage de représentation dans une architecture sur-paramétrée et sont devenus des outils importants dans diverses tâches d'apprentissage automatique. Les DNN

peuvent notamment traiter des ensembles de données d'entraînement volumineux et fournir de bonnes performances.

Malgré la popularité actuelle de modèles issus des DNN, leur utilisation n'a pas toujours été populaire. Le point de départ de l'AP est considéré comme datant de 1943 [152]. De nombreux travaux précurseurs de l'AP tel que nous le connaissons aujourd'hui, ont vu le jour dans les années 90. On peut notamment mentionner la création de la technique de retropropagation [182], ainsi que les réseaux de neurones récurrents (RNN pour *recurrent neural network*) [112] ou les réseaux de neurones convolutifs (CNN pour *convolutional neural network*) [126]. Cependant, l'exploration de ces travaux fut contrainte par les ressources informatiques limitées de l'époque.

Ces modèles sont revenus sur le devant de la scène autour des années 2010, prenant une place de plus en plus prédominante dans les travaux de l'apprentissage automatique [127]. Outre les volumes importants de données, ce sont également des innovations matérielles comme les développements de GPU (Graphical Processing Units) et de leur mise en parallèle qui ont permis de réduire les temps d'entraînements de ces modèles et donc de faciliter leur utilisation [17]. La popularité de ces méthodes fut en grande partie issue des excellents résultats qu'elles obtinrent dans diverses compétitions [121].

Il est intéressant de noter que, s'inspirant du cerveau humain, les DNN et plus généralement l'AP, sont actuellement associés au concept d'IA. L'intérêt médiatique et scientifique fut au cours de l'histoire récente suivi d'un processus d'intérêt cyclique scientifiquement bien défini [46]. En substance, de belles perspectives technologiques entraînent de fortes promesses qui peuvent ensuite mener à une déception puis à un désintéressement. Un exemple de telle promesse est par exemple, l'avènement proche d'une Intelligence Artificielle Générale. En effet, bien que l'IA soit actuellement à un sommet de popularité, il est déjà arrivé dans le passé d'avoir un "hiver de l'IA" en raison de promesses impossibles à tenir [64]. Les modèles d'AP dont l'amélioration constante de performance est actuellement due en grande partie à l'accumulation de volume de données de plus en plus imposants et de plus en plus de puissance de calculs pourrait par exemple un jour atteindre une limite entraînant un effet important de désillusion pour le grand publique et les acteurs du domaine.

Une brève définitions des réseaux neuronaux est disponible en annexe A.1.

3.1.1 *Catégorisation et stratégies d'apprentissage actif avec des réseaux neuronaux*

L'AAP présente des défis spécifiques, notamment en ce qui concerne l'estimation de l'incertitude dans les réseaux de neurones. Les modèles de réseaux de neurones ne fournissent pas toujours une indi-

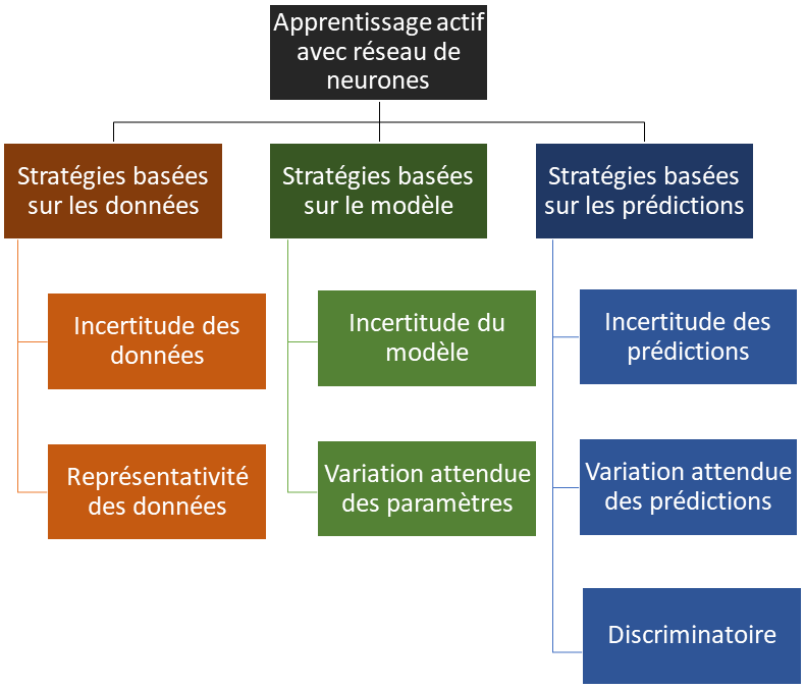


FIGURE 9 – Catégorisation des stratégies d'AA par phase de prise de décision.

cation claire et surtout fiable de leur incertitude, ce qui rend la sélection des exemples incertains complexe [80]. Effectivement, les réseaux neuronaux ont tendance à manifester un biais en faveur d'une grande confiance dans leurs prédictions [198]. Plusieurs méthodes ont été développées pour atténuer ce problème, notamment l'utilisation de l'apprentissage bayésien [210], la désactivation de certains neurones au cours de l'entraînement, *dropout* [70, 215] ou l'utilisation de réseaux de neurones probabilistes [125].

Une approche différente pour catégoriser les stratégies d'AA de celle décrite dans la Partie 2.1, consiste à les associer à la phase du processus à partir de laquelle elles tirent principalement leur prise de décision [192]. En effet, le processus d'apprentissage automatique peut être généralisé en trois étapes interconnectées : la manipulation des données, la construction du modèle et les prédictions générées par ce modèle. Comme le montre la Figure 9, si les stratégies prennent leurs décisions principalement en fonction des éléments issus d'une de ces étapes, nous les classons en conséquence.

3.1.1.1 *Stratégies basées sur les données*

L'AA a pour objectif de réduire le nombre de données annotées nécessaires pour obtenir une performance donnée après entraînement. Les stratégies basées sur les données sont souvent celles les moins efficaces pour cela. En effet, elles se concentrent uniquement sur les caractéristiques brutes des données en entrée, ainsi que sur les la-

bels des exemples déjà annotés. On peut les diviser en trois sous-catégories :

- **Incertitude des données** : Ces stratégies utilisent des informations liées à l'incertitude des données, telles que la distribution des données, la distribution des labels et la corrélation entre les labels [252]. Elles visent à identifier les exemples pour lesquels le modèle présente une incertitude élevée, ce qui peut indiquer la nécessité d'une annotation supplémentaire.
- **Représentativité des données** : Les stratégies de représentativité cherchent à comprimer géométriquement un ensemble de points de données en utilisant un nombre réduit d'exemples représentatifs. L'idée est de sélectionner des instances qui peuvent représenter efficacement les propriétés de l'ensemble de données. La compression est généralement réalisée par agrégation, *clustering* [155, 156] ou par construction d'ensemble de données [198].
- **Augmentation de données** : Les stratégies d'augmentation de données ont pour principe de générer des données synthétiques à annoter, augmentant le volume de données non-annotées disponibles. Au lieu de chercher des échantillons informatifs au sein des données non-annotées, ces méthodes dites de synthèse de requêtes d'adhésion, *Membership query synthesis* génèrent directement des instances informatives qui doivent être annotées par un oracle [103, 268]. La génération de lots d'instances synthétiques, même non-annotées, est également un moyen efficace d'améliorer l'entraînement, car la plupart des architectures d'AP nécessitent des ensembles d'entraînement volumineux [228]. Cependant, il est important de noter que la génération d'instances non-annotées peut parfois conduire à des instances étranges ou difficiles à interpréter par un humain dans certains domaines. Une manière d'augmenter le volume de données d'entraînements sans augmenter la charge d'annotation, ni rendre difficile la compréhension des instances, est de réaliser des annotations automatiques (pseudo-annotations) sur les instances où le modèle est très confiant [237].

3.1.1.2 *Stratégies basées sur le modèle*

Les stratégies basées sur le modèle disposent de connaissances à la fois sur les données et sur le modèle lui-même. Elles sélectionnent des exemples en fonction des mesures ou des indicateurs fournis par le modèle pour chaque instance. On peut les diviser en deux sous-catégories :

- **Incertitude du modèle** : Les stratégies basées sur l'incertitude du modèle cherchent à identifier les instances pour lesquelles le modèle est le moins sûr de ses prédictions. La désactivation de certains neurones au cours de l'entraînement sur une instance

peut, par exemple, permettre d'évaluer l'incertitude d'un modèle sur cette instance [70]. Ces stratégies s'appuient sur l'idée familière que les instances sur lesquelles le modèle est incertain ont le plus grand potentiel d'apprentissage.

— **Variation attendue des paramètres** : Les stratégies basées sur le changement attendu des paramètres prennent en compte l'impact que l'annotation d'une instance particulière aurait sur les poids du modèle. Elles se concentrent sur les modifications des paramètres du modèle après l'ajout d'une instance à l'ensemble d'entraînement [11]. Une instance est sélectionnée si son annotation est susceptible d'entraîner le plus de changements significatifs dans les paramètres du modèle [269]. Le calcul de ces variations est souvent réalisé avec l'information de Fisher qui permet de mesurer quelles instances sont les plus informatives pour estimer un paramètre donné [213]. Ces approches sont utiles pour identifier les exemples qui peuvent influencer le modèle de manière significative, contribuant ainsi à l'amélioration de sa performance.

3.1.1.3 *Stratégies basées sur les prédictions*

Les stratégies d'apprentissage actif basées sur les prédictions se concentrent sur les sorties du modèle, en évaluant les prédictions de ce dernier pour sélectionner les instances les plus informatives à annoter. Cette catégorie peut être subdivisée en trois sous-catégories distinctes :

— **Incertitude des prédictions** : Les stratégies basées sur l'incertitude des prédictions cherchent à identifier les instances pour lesquelles le modèle est le moins sûr de ses prédictions. L'incertitude est souvent mesurée en examinant la variabilité des probabilités associées aux différentes classes de sortie. Plus l'incertitude est élevée pour une instance donnée, plus elle est susceptible de contenir des informations importantes et de mériter une annotation supplémentaire [28, 111, 146]. Les stratégies de cette catégorie se rapprochent de celles vues dans la Section 2.1.2.1.

— **Variation attendue des prédictions** : Les stratégies basées sur le changement attendu des prédictions sélectionnent les instances dont l'annotation est susceptible d'avoir le plus grand impact sur les performances de sorties du modèle. Elles mesurent la variation attendue des prédictions du modèle après l'annotation d'une instance particulière. Les métriques de changement attendu des prédictions peuvent être basées sur des critères, tels que l'évolution de métriques de performances ou la fonction de perte du modèle attendue [148, 255].

— **Discriminatoire** : Les stratégies discriminatoires ont pour objectif de repérer les instances dont les prédictions sortent du

lot. Des méthodes calculant les marges autour des frontières de décisions permettent de détecter ces instances "anormales" et de les sélectionner [52]. Une autre manière de discriminer est d'identifier les exemples non labélisés qui sont les plus éloignés de l'ensemble déjà annoté existant, sous l'hypothèse que ces exemples apportent des informations importantes pour l'amélioration du modèle d'apprentissage automatique. Pour cela, le modèle tente de prédire si une instance appartient à l'ensemble déjà annoté de données ou à l'ensemble non-annoté de données. L'instance qui présente la plus grande certitude quant à son appartenance à l'ensemble de données non annotées est sélectionnée [73].

3.2 APPRENTISSAGE ACTIF AVEC DES TRANSFORMERS

L'apprentissage actif profond constitue un défi majeur dans le contexte de l'efficacité d'apprentissage avec des ensembles de données limités ; une situation qui contraste avec la tendance des réseaux de neurones à exiger des volumes massifs de données pour obtenir des performances optimales. Les réseaux de neurones, en raison de leur complexité, ont une propension à surajuster sur des petits ensembles de données, entraînant ainsi une baisse de leur capacité de généralisation. Afin de pallier cette limitation, des stratégies telles que le pré-entraînement et le transfert d'apprentissage sont fréquemment employées pour compenser le manque d'annotations. Il est intéressant de noter l'émergence des transformers dans ce contexte. Ces modèles sont pré-entraînés sur des corpus volumineux, capturant ainsi des représentations riches et abstraites et nécessitant un volume comparativement petit de données, pour s'entraîner et être performant pour une application donnée.

Au cours des dernières années, nous avons assisté à une accélération spectaculaire de l'intérêt pour les méthodes d'AP, largement stimulé par l'avènement des transformers [234] et des modèles de grande envergure. L'émergence des transformers, en particulier, a révolutionné le traitement des séquences, propulsant des modèles tels que BERT [48] et GPT-3 [24] au sommet des performances en traitement du langage naturel. Cette tendance s'est accompagnée de premiers travaux de recherche visant à exploiter la puissance de l'AA en tandem avec les capacités des transformers.

L'avènement de modèles tels que GPT-3 de OpenAI, avec son accès public via ChatGPT [140], a joué un rôle significatif dans la démocratisation de ces approches. La facilité d'utilisation et la popularité croissante de ces modèles parmi le grand public ont contribué à leur adoption généralisée. ChatGPT, en particulier, en tant qu'outil grand public, a ouvert de nouvelles perspectives pour l'exploration et l'ap-

plication de ces modèles à diverses tâches, allant de la génération de texte à la compréhension contextuelle avancée.

Parallèlement à la montée en puissance des méthodes d'AP et des transformers, une tendance notable s'est développée vers la personnalisation de ces modèles pour répondre aux besoins spécifiques de chaque individu ou entreprise. En effet, les volumes de données et les puissances de calcul nécessaires à l'entraînement de ce type de modèle sont actuellement accessibles uniquement aux grandes entreprises privées ou à des consortiums d'acteurs publics. A terme, l'objectif de ces modèles et de permettre aux utilisateurs de les ajuster en fonction de leur domaine d'expertise et de leurs cas d'utilisation particuliers. Cette idée de personnalisation va de pair avec les objectifs fondamentaux de l'AA, qui vise à sélectionner un faible volume de données pertinentes pour entraîner un modèle de manière efficace.

Une brève définitions des architectures transformers est disponible en annexe A.2.

3.2.1 *Application de l'apprentissage actif avec des transformers*

L'apprentissage actif (AA) a longtemps été une stratégie privilégiée pour améliorer l'efficacité des modèles d'apprentissage automatique classiques tels que les machines à vecteurs de support (SVM) ou les réseaux de neurones à convolution (CNN) [34, 78, 124, 161, 176, 192]. Cependant, avec l'avènement des transformers [234], de nouvelles architectures d'apprentissage profond révolutionnaires, l'application de l'AA à ces modèles suscite un intérêt croissant.

Certains travaux préliminaires ont examiné l'efficacité de l'AA sur les transformers, en particulier dans le contexte de la classification binaire [55, 145]. La synergie entre les deux approches est intéressante à étudier puisque appliquer l'AA à des *transformers* permet notamment de réduire le coût d'annotation durant le *fine-tuning* (spécialisation d'un modèle sur une nouvelle tâche) en sélectionnant le meilleur ensemble de données à annoter. Ces études ont notamment montré que l'AA contribuait à réduire les biais pendant les premières étapes de l'entraînement des transformers. Cependant, d'autres résultats préliminaires suggèrent que les stratégies d'AA conçues pour des modèles antérieurs, peuvent ne pas être directement applicables aux transformers, introduisant même de l'instabilité dans le processus d'entraînement [45].

L'utilisation de l'AA avec des transformers présente des défis uniques, notamment en raison du grand nombre de paramètres de ces modèles, ce qui entraîne une augmentation significative du temps de calcul lors de chaque étape d'entraînement. Les études se sont concentrées principalement sur les stratégies d'AA basées sur l'incertitude des approches particulièrement adaptées aux transformers [145]. Cependant, une analyse approfondie [193], a révélé que les stratégies

qui ont montré leur efficacité avec des modèles antérieurs ne sont pas toujours adaptées aux transformers. Cette constatation souligne l'importance cruciale d'une évaluation continue des stratégies d'AA existantes pour les adapter judicieusement à ces nouvelles architectures dynamiques.

Une étude récente [75], s'est concentrée sur l'impact des fonctions d'activation alternatives à softmax dans le contexte des transformers, mettant en lumière leur potentiel significatif de gains de performance, notamment en adoptant des fonctions d'activation non biaisées envers les données aberrantes. Cette exploration s'inscrit dans une quête continue d'optimisation des modèles transformers pour des tâches de classification multi-classes.

Par ailleurs, une étude novatrice a exploré la question de la transférabilité des ensembles de données acquises de manière active dans le contexte de la classification de texte avec des modèles transformers [108]. L'objectif principal était d'évaluer dans quelle mesure les avantages de l'apprentissage actif, initialement démontrés pour le fine-tuning de modèles spécifiques, pouvaient être généralisés à d'autres modèles. Les résultats mettent en avant le rôle prédominant du choix de la méthode d'apprentissage actif dans la similarité des séquences d'acquisition, par rapport au choix du modèle. En outre, les résultats indiquent qu'une approche combinant des fonctions d'acquisition basées sur l'incertitude et la diversité, semble favoriser la transférabilité, soulignant la nécessité de stratégies d'AA adaptatives et flexibles pour optimiser l'AA dans le contexte des transformers.

3.2.2 *Comparaison avec des modèles de classification antérieurs*

Les travaux préliminaires à cette thèse ont révélé des différences substantielles dans les performances des stratégies d'AA lorsqu'elles sont appliquées à des modèles de classification tels que les SVM, par rapport aux transformers. Ce constat a été récemment confirmé dans un contexte multi-classes, où des fonctions populaires et efficaces, telle que l'entropie, qui ont traditionnellement bien fonctionné, ont montré des performances mitigées pour les transformers [195].

Une comparaison rapide entre les deux modèles, les SVM et les transformers, révèle des particularités importantes qui influent sur l'efficacité des stratégies d'AA. Les SVM, fondées sur des techniques d'optimisation convexe, sont classiquement utilisées pour des tâches de classification binaire, et leur extension à des problèmes multi-classes ou multi-labels se fait souvent par la méthode de *binary relevance*. Cette approche consiste à traiter chaque classe comme une tâche binaire distincte, négligeant potentiellement les relations complexes entre les classes. Une autre manière de réaliser une classification multi-classes avec des SVM consiste à employer l'algorithme *Random Forest* [22] qui combine les prédictions de plusieurs arbres de

décision afin de produire une classification robuste et précise. Cette approche approche ayant cependant pour limite qu'un grand nombre d'arbres généré peut rendre l'algorithme trop lent et inefficace pour des prédictions en temps réel, ce qui est particulièrement problématique dans le contexte de l'AA.

D'un autre côté, les transformers, ont montré une capacité exceptionnelle à capturer des dépendances à long terme et des structures hiérarchiques dans les données. Cependant, cette capacité à gérer des informations complexes peut également introduire des défis lors de l'application d'AA. Les mesures traditionnelles d'incertitude peuvent ne pas bien refléter l'incertitude inhérente à des modèles aussi complexes, nécessitant ainsi des adaptations spécifiques [235].

En ce qui concerne les méthodes d'AA elles-mêmes, les heuristiques sur lesquelles elles s'appuient sont censées s'appliquer aux transformers. Cependant, étant donné qu'elles ont été développées pour d'autres modèles, leur application directe aux transformers peut ne pas exploiter pleinement leurs capacités [45].

3.2.3 *Application à la tâche de classification multi-label*

Pour réaliser les tâches de classification multi-labels avec un transformers, l'approche classique est de rajouter un réseau neuronal après un encodeur comme on peut le voir dans la Figure 10. Dans les transformers, le vecteur caché de classification [CLS] est une représentation agrégée d'une séquence entière (par exemple une phrase) générée par le modèle à partir de l'ensemble des tokens d'une séquence d'entrée. L'utilisation du [CLS] est courante dans les tâches de classification, où ce vecteur encapsule l'information globale de la séquence.

Pour effectuer la classification multi-labels, le vecteur caché [CLS] est utilisé comme entrée pour un réseau neuronal entièrement connecté. Ce réseau comprend généralement une ou plusieurs couches denses, la première couche ayant la taille du vecteur [CLS] et la dernière couche ayant un nombre de neurones égal au nombre de classes dans la tâche de classification multi-labels [209]. Chaque neurone de la couche de sortie est associé à une classe spécifique et est activé par une fonction d'activation sigmoïde.

La fonction de perte utilisée est généralement l'entropie croisée binaire. Elle mesure la divergence entre les probabilités prédites et les étiquettes réelles pour chaque classe. Cette fonction de perte est bien adaptée aux tâches de classification multi-labels. Le modèle global, comprenant le transformer et le réseau neuronal entièrement connecté, est entraîné sur un ensemble de données annoté avec des labels. L'objectif est d'ajuster les poids du modèle pour minimiser la fonction de perte.

Une source de travaux intéressante sur la convergence de la classification mutli-labels et les transformers est la classification multi-labels

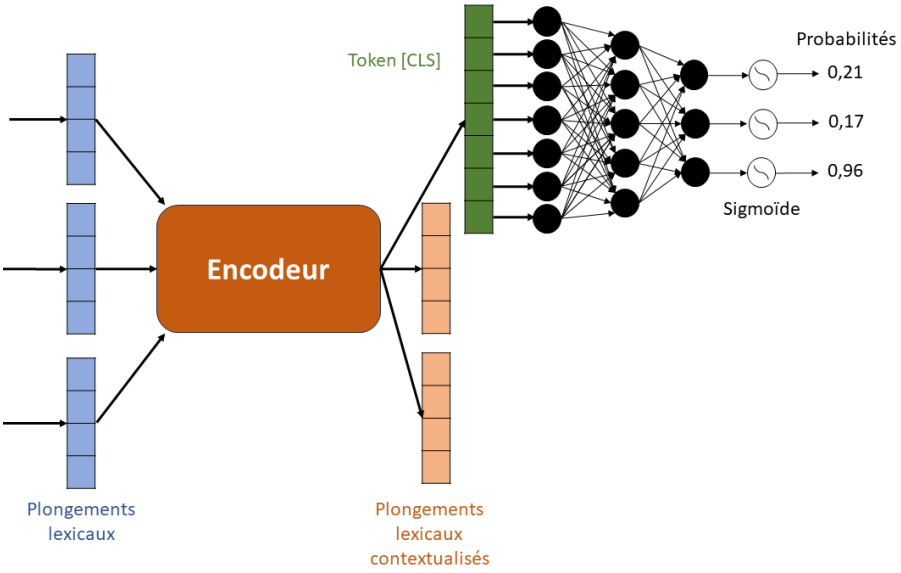


FIGURE 10 – Architecture du modèle encodeur + réseau neuronal pour classification multi-labels.

extrême (XMLC). Cette tâche, qui implique la prédiction simultanée d'un grand nombre de labels pour une seule instance, présente des défis uniques en raison de sa complexité et de la diversité des relations entre les labels. Les transformers, avec leur architecture d'attention, se sont avérés efficaces pour capturer ces dépendances complexes et ont souvent surpassé les approches traditionnelles [32]. Une étude récente s'est penchée sur l'intégration de l'AA avec les transformers pour la tâche de XMLC [243]. Cette étude met en évidence les difficultés de la convergence de ces différentes problématiques. En effet, l'étude souligne qu'aucune des stratégies d'AA étudiées n'a permis d'améliorer de façon substantielle les modèles par rapport à un échantillonage aléatoire.

3.3 MITIGER L'IMPACT DE LA TAILLE DES MODÈLES

L'avènement de modèles d'apprentissage profond de plus en plus massifs, caractérisés par une expansion considérable du nombre de paramètres, soulève des défis importants en termes de coût d'entraînement, de coût d'utilisation et de demande en ressources computationnelles. Cette tendance presque exponentielle qu'on peut voir dans la Figure 11 à l'expansion des modèles découle de la recherche constante d'une meilleure représentation des données et de performances améliorées dans diverses tâches d'apprentissage automatique [271]. Bien que la taille effective de GPT-4, sorti en 2023, soit actuellement confidentielle [166], les estimations et les fuites potentielles

d'informations suggèrent qu'elle tourne autour de 2 trillions de paramètres. [1] [2]

Le coût d'entraînement de ces modèles, en particulier sur des jeux de données volumineux, s'est considérablement accru, nécessitant des infrastructures matérielles avancées, souvent équipées de GPU puissants, pour accélérer les calculs complexes associés à l'optimisation des paramètres. De même, le coût d'utilisation de ces modèles en production, que ce soit en termes d'énergie, de mémoire, de stockage ou de temps de calcul, peut être prohibitif, surtout lorsque la rapidité est une exigence.

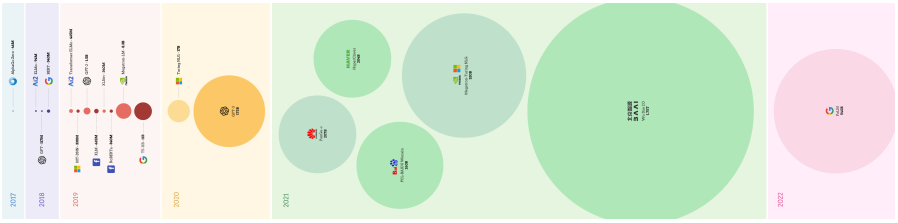


FIGURE 11 – Visualisation de la croissance en nombre de paramètres des modèles de langues. [3]

L'expansion dimensionnelle des modèles en apprentissage profond [115], génère une pression considérable en terme de ressources, affectant ainsi non seulement les grandes entreprises, mais aussi les petites et moyennes entreprises. Même dans le cas des entreprises de taille moyenne, voire grande, ainsi que des particuliers, cette évolution crée des défis significatifs, surtout pour les cas d'usage exigeant une rapidité d'exécution ou évoluant dans des environnements aux moyens limités. Il est essentiel de noter que seuls les géants du numérique, disposant de vastes ressources en termes de matériel, de données et d'expertise, sont en mesure de faire face à ces pressions de manière significative, instaurant ainsi une forme de dominance dans le domaine [244]. Les préoccupations concernant les coûts environnementaux associés à l'entraînement et au déploiement de ces modèles sont également en augmentation [197]. Devant ces contraintes, des approches novatrices et adaptatives sont indispensables pour atténuer les répercussions de la taille croissante des modèles et rendre les avantages de l'apprentissage profond accessibles à une gamme plus large d'acteurs.

3.3.1 *Utilisation de la distillation*

La distillation de modèle, également connue sous le nom de compression de modèle, est une technique en apprentissage automatique visant à réduire la taille d'un modèle tout en préservant autant que

1. https://cobusgreyling.medium.com/what-are-realistic-gpt-4-size-expectations-73f00c39b832
2. https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked/
3. https://venngage.com/blog/ai-growth/

possible ses capacités de prédiction [25, 89]. Cette approche est notamment populaire dans le déploiement des modèles sur des appareils avec des ressources limitées, tels que les appareils mobiles ou les objets connectés, où la taille du modèle et la consommation de mémoire sont des préoccupations majeures [19].

Le processus de distillation de modèle implique généralement deux phases distinctes : la phase d'entraînement du modèle enseignant (*teacher model*) et la phase de transfert des connaissances au modèle étudiant (*student model*). Initialement, un modèle complexe et performant, le modèle enseignant, est entraîné sur un ensemble de données volumineux et représentatif. Ce modèle est capable de capturer des relations complexes au sein des données d'entraînement et de générer des prédictions précises. Ensuite, la connaissance du modèle enseignant est transférée à un modèle plus petit et plus léger, le modèle étudiant, qui est destiné à être déployé sur des dispositifs avec des contraintes de ressources. Ce transfert de connaissances se fait en utilisant les sorties du modèle enseignant comme cibles pour l'entraînement du modèle étudiant, plutôt que l'ensemble de données d'entraînement de départ. En ajustant les poids du modèle étudiant pour minimiser la divergence entre ses prédictions et celles du modèle enseignant, le modèle étudiant apprend à généraliser les relations importantes sans conserver la complexité inutile du modèle enseignant [76].

La distillation de modèle présente plusieurs avantages. Tout d'abord, elle permet de réduire considérablement la taille des modèles, facilitant ainsi leur déploiement sur des appareils avec des capacités de stockage limitées. De plus, elle peut accélérer les inférences, car les modèles plus petits nécessitent moins de calculs pour générer des prédictions [220].

DistilBERT, développé par Hugging Face [187], est un exemple concret de la distillation de modèle appliquée au modèle de langage BERT (Bidirectional Encoder Representations from Transformers) [48]. Ce dernier est structuré de manière à conserver l'essentiel des capacités de BERT, mais avec une architecture simplifiée. En effet, DistilBERT est 40% plus petit, 60% plus rapide et retient 97% des capacités de BERT.

Lorsqu'on intègre la distillation dans un cadre d'apprentissage actif, le modèle étudiant distillé, caractérisé par sa rapidité et son efficacité, peut être déployé pour effectuer des prédictions sur des données non annotées. Le modèle étudiant distillé peut ainsi être utilisé pour identifier les points d'incertitude ou les zones les plus complexes du domaine, guidant ainsi le processus d'annotation vers les données les plus stratégiques. Une fois que ces données sélectionnées ont été annotées, elles peuvent être incorporées dans l'entraînement du modèle enseignant initial, plus complexe. Cette démarche hybride permet de tirer parti des avantages de la distillation pour l'apprentissage actif

tout en préservant la capacité du modèle enseignant à capturer des relations plus complexes, maximisant ainsi les performances globales du système [232].

3.3.2 *Temps d'attente durant cycle d'apprentissage actif*

L'apprentissage actif s'inscrit dans un processus itératif et cyclique, illustré dans la Figure 2. Ces cycles peuvent être impactés par des temps d'attente liés à différentes phases. Les deux principales périodes d'attente surviennent lors de l'inférence des modèles précédents le calcul de l'intérêt des instances par AA et lors de la mise à jour des modèles avec les données sélectionnées et annotées. Il s'agit de périodes d'attente, car par défaut, ce sont des moments d'inactivité pour les annotateurs qui doivent attendre la fin de ces processus avant de recevoir de nouvelles instances à annoter.

En plus de la frustration engendrée par l'attente entre chaque annotation, l'apprentissage actif est utilisé dans des contextes où l'accès aux ressources est limité, que ce soit en termes de disponibilité humaine ou de capacités informatiques. Par conséquent, un objectif sous-jacent est de maximiser l'exploitation efficace des ressources disponibles. Dans les projets d'annotations, les annotations sont la plupart du temps budgétisées de deux manières : par annotation ou par heure. Dans ce dernier cas, il est alors impératif d'avoir un flux constant de données à annoter pour maximiser son budget d'annotation [159]. En réduisant la fréquence des mises à jour du modèle et des inférences, l'apprentissage actif par lot de données s'est avéré être une solution atténuant cette problématique [200]. L'introduction d'une source d'instances parallèle au cycle d'apprentissage actif traditionnel peut éliminer totalement ces temps d'attente, par exemple en utilisant les scores d'informativité issus des cycles précédents d'apprentissage actif, même si cela s'accompagne d'une diminution des performances des modèles entraînés [84].

Dans les contextes d'AA contemporains, où les modèles et les ensembles de données non annotées sont de plus en plus volumineux, le temps d'attente est moins induit par les mises à jour du modèle (entraînement sur un lot de données) que par l'inférence sur ces vastes ensembles [270]. Le sous-échantillonage peut être utilisé comme approche de réduction des temps d'inférence, permettant au modèle de n'effectuer des inférences que sur une fraction des instances non annotées, cette fraction pouvant varier entre chaque cycle d'apprentissage actif [232]. Une alternative pour diminuer le temps d'inférence consiste à orienter les calculs d'apprentissage actif en se basant sur des caractéristiques préalablement calculées pour les instances, combinées à une pré-agrégation d'instances[12]. Cette approche transfère une part importante de la charge de calcul, et donc d'attente, en amont du processus d'apprentissage actif.

3.4 PARADIGMES VOISINS DE L'APPRENTISSAGE ACTIF

Comme discuté précédemment, l'apprentissage actif émerge comme un domaine crucial de l'apprentissage automatique notamment à l'heure actuelle avec le besoin toujours plus grand en données annotées. Toutefois, il est essentiel de reconnaître l'existence de paradigmes d'apprentissage voisins, partageant des caractéristiques et des objectifs parfois similaires. Cette section vise à préciser ces paradigmes connexes, à savoir l'apprentissage en continu, l'apprentissage en ligne, l'apprentissage tout-au-long de sa vie et l'apprentissage suivant un curriculum. En français, les connotations associées au terme "actif" peuvent parfois induire en erreur quant à la nature réelle de l'AA. En clarifiant les distinctions entre ces approches, nous cherchons à éliminer toute confusion potentielle avec l'apprentissage actif. De plus, en explorant ces paradigmes, nous mettons en évidence des techniques transférables et des synergies potentielles.

3.4.1 *Apprentissage en continu, tout-au-long de sa vie et en ligne*

L'apprentissage continu ou apprentissage tout-au-long de sa vie, *lifelong learning*, repose sur l'idée fondamentale d'apprendre de manière continue au fil du temps [18, 35, 167, 185]. Contrairement à l'hypothèse courante en apprentissage automatique, selon laquelle les données sont a priori disponibles, l'apprentissage continu reconnaît la nature dynamique des données avec ses dimensions de volume, de variété et de vitesse [236]. L'apprentissage en continu se distingue par son engagement à s'adapter constamment à l'arrivée de nouvelles données voire même de nouvelles tâches, sans nécessiter de ré-entraînement complet du modèle [15, 212]. L'intégration de nouvelles connaissances, tout en préservant les anciennes, avec l'objectif final d'une meilleure généralisation dans le temps, n'est pas un processus direct [132, 144]. Ce domaine se concentre sur le développement de techniques d'apprentissage capables de traiter efficacement un flux potentiellement illimité de données en constante évolution, tout en respectant des ressources computationnelles et de mémoire définies [143].

L'apprentissage en ligne est un paradigme où les données d'entraînement sont traitées un exemple à la fois de façon séquentielle [91]. Lorsqu'un nouveau point de données est introduit, le modèle existant est rapidement ajusté pour produire le meilleur modèle jusqu'à présent [184]. Cette approche est souvent adoptée lorsque l'entraînement sur l'ensemble complet des données est inabordable sur le plan computationnel ou impraticable en raison de contraintes matérielles. Les méthodes d'apprentissage en ligne se distinguent par leur efficacité en termes de mémoire et de temps d'exécution, en réponse aux exigences de latence dans des scénarios du monde réel.

Bien que l'apprentissage en ligne soit une forme d'apprentissage continu, les deux paradigmes diffèrent dans leurs objectifs [136]. L'apprentissage en ligne vise à optimiser les performances sur une tâche d'apprentissage donnée en apprenant de manière plus efficace lorsque de nouvelles données deviennent disponibles. En revanche, l'apprentissage continu aspire à apprendre à partir de séquences de lots/-tâches différents, préservant les connaissances acquises pour faciliter l'apprentissage des tâches futures. De plus, l'apprentissage continu et en ligne se distinguent de l'AA par leur accent sur l'évolution constante des données et la nécessité de gérer des contraintes réelles telles que les ressources limitées et les changements fréquents dans les données. En d'autres termes, l'apprentissage en continu et en ligne se concentrent sur l'adaptation constante, tandis que l'AA se préoccupe spécifiquement de la sélection minutieuse des données d'entraînement.

3.4.2 *Apprentissage suivant un curriculum*

L'apprentissage par curriculum est un processus d'entraînement qui propose une séquence de tâches ou de données à un algorithme d'apprentissage, dans le but de le rendre capable d'aborder, en fin de compte, une tâche généralement plus difficile. Contrairement à l'AA, où l'accent est mis sur la sélection dynamique d'échantillons d'entraînement, l'apprentissage par curriculum s'articule autour de la structuration progressive des tâches ou des données [239].

Dans l'apprentissage par curriculum , les tâches sont soigneusement choisies et organisées de manière à faciliter l'apprentissage de la tâche finale de manière plus efficiente. Cette structuration tient compte des différentes difficultés et dépendances fonctionnelles entre les tâches, créant ainsi une progression naturelle de l'apprentissage [214]. L'objectif principal est de permettre à l'algorithme d'acquérir des compétences graduellement, en commençant par des tâches simples et en évoluant vers des tâches plus complexes.

L'apprentissage par curriculum se distingue de l'AA sur plusieurs aspects. Tout d'abord, dans l'apprentissage par curriculum, les tâches sont choisies délibérément et organisées de manière à faciliter l'apprentissage progressif, tandis que dans l'AA, les échantillons d'entraînement sont sélectionnés dynamiquement en fonction de leur impact sur le modèle. Des travaux ont tout de même exploré l'intersection des deux paradigmes, en faisant en sorte que les stratégies d'AA prennent en compte une évolution de la difficulté des instances pour le modèle [107, 221].

3.4.3 *Apprentissage actif dans l'éducation*

L'AA, dans le contexte de l'éducation, constitue une approche pédagogique novatrice visant à optimiser l'acquisition des connaissances par les apprenants. Contrairement à l'AA dans le domaine de l'apprentissage automatique, qui se concentre sur la sélection intelligente des exemples d'entraînement pour améliorer les performances d'un modèle, l'AA en éducation se focalise sur l'engagement actif des étudiants dans le processus d'apprentissage. Cette approche encourage la participation active, la réflexion, et la résolution de problèmes au lieu d'une simple réception passive d'informations [118].

Dans le cadre éducatif, l'AA implique souvent des méthodes interactives telles que les discussions en classe, les projets collaboratifs, les études de cas, et d'autres activités engageantes. L'objectif est de favoriser la réflexion critique, la créativité, et la compréhension approfondie des concepts plutôt que de simplement mémoriser des informations [104]. Contrairement à l'AA en apprentissage automatique, qui est en partie automatisée, l'AA en éducation repose quasi-exclusivement sur l'interaction humaine, le dialogue, et la construction conjointe du savoir [85].

En résumé, bien que les termes "apprentissage actif" soient utilisés dans des contextes différents, leurs objectifs et méthodes diffèrent considérablement. L'AA en éducation se concentre sur l'engagement actif des apprenants, tandis que dans le domaine de l'apprentissage automatique, il se réfère à une stratégie de sélection d'exemples d'entraînement pour améliorer les performances d'un modèle.

3.5 DISCUSSION ET CONCLUSION

Dans ce chapitre, nous avons dressé un panorama des modèles neuronaux et de leurs associations avec les stratégies d'apprentissage actif. En revisitant l'évolution des modèles neuronaux et de l'IA en général, nous avons souligné l'intérêt cyclique qu'on lui porta au fil du temps. Intérêt qui est actuellement stimulé par d'excellentes performances, la démocratisation d'outils, ainsi que par des promesses et des aspirations ambitieuses telles que l'émergence d'une Intelligence Artificielle Générale. Nous avons débuté en détaillant l'architecture des modèles neuronaux pour une meilleure compréhension de leur fonctionnement, puis nous avons adopté une catégorisation adaptée aux réseaux neuronaux pour présenter différentes stratégies d'apprentissage actif qui leur sont associées.

Ensuite, nous nous sommes penchés sur les modèles de langues pré-entraînés, en particulier les transformers. Après avoir examiné leur architecture et leurs spécificités, nous avons abordé les défis liés à leur utilisation conjointe avec l'apprentissage actif. Nous avons effectué un état de l'art sur l'intersection entre les transformers et

l'apprentissage actif, en mettant l'accent sur la tâche de classification multi-labels.

Nous avons ensuite souligné une tendance importante dans le développement des modèles neuronaux : la propension à devenir toujours plus grands. La taille croissante des modèles en termes de paramètres et la taille des ensembles d'entraînement sont souvent des moteurs de progrès, mais cela se traduit par des modèles de plus en plus volumineux et coûteux, impactant les aspects financiers et environnementaux de leur utilisation. Nous avons également discuté de la concentration de cette expertise chez les géants du numérique en raison de ces échelles colossales. À travers des approches telles que la distillation ou l'optimisation de l'utilisation de ces modèles imposants, nous avons exploré des solutions intéressantes pour rendre ces outils puissants accessibles au grand public, qu'il s'agisse de particuliers ou d'entreprises ordinaires.

Enfin, nous avons examiné les paradigmes voisins de l'apprentissage actif pour dissiper toute ambiguïté et clarifier les distinctions entre l'apprentissage actif et d'autres approches.

En conclusion, ce chapitre a établi les fondements de diverses architectures neuronales et a retracé leur évolution. Nous avons examiné la synergie entre l'apprentissage actif et les modèles neuronaux, avec une attention particulière portée aux modèles transformers. En soulignant l'intersection entre les transformers, l'apprentissage actif et la classification multi-labels, nous avons identifié un terrain propice à l'exploration. Cette exploration est approfondie dans les chapitres suivants de cette thèse, où nous présenterons nos contributions significatives.

Deuxième partie

CONTRIBUTIONS

4

APPLICATION DE L'APPRENTISSAGE MULTI-LABELS ACTIF BASÉ SUR L'INCERTITUDE AUX TRANSFORMERS

Une partie du contenu de ce chapitre a été publiée lors de la conférence CORIA-TALN 2023 [10]. Ce chapitre constitue une exploration approfondie des stratégies d'AA dans le contexte des modèles transformers, avec un accent particulier sur la tâche de classification multi-labels. Les transformers, en raison de leur architecture novatrice et de leur succès dans diverses tâches de traitement du langage naturel, ont suscité un intérêt croissant dans l'intégration de stratégies d'AA. Toutefois, le défi majeur réside dans l'adaptation des stratégies d'apprentissage actif traditionnelles à ces modèles, compte tenu de la complexité accrue et des besoins de calcul substantiels associés à leur entraînement. Ce chapitre se propose d'analyser en détail les implications, les avantages et les limitations de l'application de l'apprentissage actif basé sur l'incertitude aux transformers dans le contexte spécifique de la classification multi-labels. En examinant les résultats de diverses expériences et en confrontant diverses stratégies populaires, ce chapitre vise à éclairer la voie vers une utilisation plus efficace et ciblée de l'AA dans le domaine des transformers appliqués à des tâches multi-labels complexes.

4.1 CONTEXTE ET MOTIVATIONS

L'enjeu pour entraîner un modèle avec de l'AA consiste en l'élaboration d'une stratégie pour sélectionner une instance plutôt qu'une autre [199]. Une fois la stratégie appliquée et l'instance sélectionnée, la plupart des travaux sur l'AA dans la classification multi-labels font ensuite une requête à un oracle afin d'obtenir tous les labels associés à cette instance [176]. Les études récentes sur l'application de l'AA aux transformers, notamment sur des tâches de classification binaire [55, 145], ont démontré que cette approche contribuait à atténuer les biais en début d'entraînement. Nuançant ce constat, d'autres résultats préliminaires [45] suggèrent que les stratégies conçues pour des modèles antérieurs rencontrent des difficultés avec les transformers, introduisant de l'instabilité dans le processus d'entraînement.

Les stratégies d'AA basées sur les ensembles [122, 208] et les gradients [27] s'adaptent mal au grand nombre de paramètres des transformers [193]. Comme suggéré par [145], pour l'AA dans le contexte des transformers, nous nous concentrons sur l'étude des stratégies basées sur l'incertitude [41, 63, 129-131, 176].

Bien que l'importance de l'AA dans le contexte des transformers soit reconnue [56], des questions cruciales demeurent. Notamment, sur l'existence ou non d'une variabilité de la performance des stratégies d'AA d'un transformer à l'autre. Face aux besoins importants en données de ces modèles et aux coûts significatifs associés à l'acquisition de données annotées, l'alliance entre l'AA et les transformers offre une opportunité de perfectionner ces modèles de manière efficace et économique.

4.1.1 *Travaux connexes*

Alors que certaines études [55, 145] ont mis en évidence l'intérêt de l'AA en mettant en évidence sa capacité à atténuer efficacement les biais pendant les premières phases d'entraînement du modèle, d'autres résultats préliminaires rapportés par D'Arcy et Downey [45] suggèrent l'inverse et que l'application de l'AA sur les transformers peut conduire à une instabilité dans le processus d'entraînement. Ce désaccord met en évidence l'une des principales faiblesses actuelles de l'AA dans le contexte des transformers : le manque de preuves démontrant un bénéfice substantiel de sa mise en œuvre [175].

En nous inspirant des observations faites par Lu et MacNamee [145], notre étude se focalise sur l'application de stratégies basées sur l'incertitude pour l'AA dans le contexte des transformers. Cependant, il est essentiel de souligner que l'efficacité de ces stratégies peut varier considérablement en fonction du type d'architecture utilisée. Par exemple, les stratégies basées sur l'incertitude peuvent présenter des faiblesses lorsqu'elles sont appliquées aux modèles de type CNN, car ces derniers sont particulièrement sensibles aux biais potentiels induits par la sélection d'instances très similaires entre elles à un cycle donné d'AA [198]. Afin d'exclure la possibilité que les performances médiocres de certaines stratégies soient uniquement attribuables à ce biais, nous intégrerons une étape de pré-clustering dans certaines de nos expériences. Cette approche vise à garantir une diversité dans les instances sélectionnées et à mieux évaluer l'efficacité des stratégies d'AA dans notre contexte d'étude.

L'étude approfondie menée par Schröder, Niekler et Potthast [193] sur l'utilisation des stratégies d'AA basées sur l'incertitude dans le contexte des transformers révèle une diversité d'efficacité par rapport aux architectures précédentes telles que les SVM ou les CNN. En effet, les stratégies optimales pour les transformers dans le cadre de la classification multi-classes peuvent différer de celles utilisées pour d'autres types d'architectures. En outre, cette étude met en évidence l'un des principaux avantages de l'AA : même en utilisant seulement une fraction des instances annotées, il permet d'atteindre des performances similaires à celles obtenues avec une supervision totale du modèle. Toutefois, il est important de noter que cette étude ne se

concentre pas spécifiquement sur la classification multi-labels, bien que cette dernière nous intéresse et soit cruciale dans le domaine de l'AA, comme le souligne Liu et al. [138].

Comme montré dans [243], certaines stratégies d'AA multi-labels [73, 176, 258], dans le contexte des transformers, ne semblent pas apporter d'améliorations et performent même moins bien qu'un échantillonage aléatoire des données d'entraînement. Ces travaux proposent une approche basée sur le ratio de *subword* dans une phrase, suivant l'hypothèse que le nombre de *subword* est proportionnel à la rareté des mots présents et donc à l'intérêt que cette phrase aurait pour le classifieur. Bien que cette stratégie ne performe généralement pas mieux que les stratégies d'AA classiques envisagées par les auteurs, elle nécessitent beaucoup moins de puissance de calcul pour fonctionner. Ainsi d'un point de vue de ratio "coût d'utilisation / performance", cette approche permet de mettre en évidence que les stratégies d'AA ne seraient pas très efficaces sur les transformers.

A notre connaissance, il n'y a pas encore d'explications quant aux difficultés de l'AA pour performer dans les contexte des transformers et nous tenterons de combler ce manquement tout en étendant notre étude autour de six stratégies différentes basées sur l'incertitude. Ainsi, ce chapitre se donne pour mission d'explorer ces questions, de déterminer pourquoi certaines stratégies surpassent d'autres, d'identifier des approches constamment performantes par rapport à un échantillonnage aléatoire, de dévoiler les causes potentielles des sous-performances, et enfin, de proposer des solutions pour optimiser le *fine-tuning* des transformers par le biais de l'AA.

4.2 DÉFINITION DU PROBLÈME ET CONTEXTE EXPÉRIMENTAL

Nous adaptons des approches d'AA basées sur l'incertitude à des transformers, en utilisant six stratégies d'AA basées sur l'incertitude. L'objectif principal est d'évaluer l'impact de l'AA exploitant des transformers, tout au long de l'apprentissage et d'analyser les performances obtenues. Une première question guide notre démarche : "L'apprentissage actif apporte-t-il réellement des avantages à l'entraînement des transformers ?" Nous mettons en œuvre un protocole expérimental rigoureux, suivant les tendances des courbes d'apprentissage pour détecter des différences tout au long du processus. En examinant les performances finales, notre étude vise à déterminer quelles stratégies d'AA émergent comme des valeurs sûres, lesquelles démontrent une fiabilité moyenne, et celles qui sous-performent.

4.2.1 *Méthodologie*

Notre objectif est d'appliquer et d'évaluer différentes stratégies d'AA sur différents modèles de type transformers. Ces stratégies d'AA ba-

sées sur l'incertitude ont été initialement élaborées pour des modèles traditionnels tels que les SVM et les CNN. Cependant, avec l'avènement des transformers, il est crucial d'analyser leur adaptation à ces nouveaux paradigmes. L'aspect novateur des transformers soulève des questions quant à la transférabilité de ces stratégies préexistantes. Nous visons donc à examiner comment ces stratégies se comportent face aux spécificités des transformers, en tenant compte de la diversité des modèles et des jeux de données utilisés.

Cette étude permettra de déterminer si les stratégies d'AA, initialement conçues pour des modèles traditionnels, conservent leur efficacité dans le contexte des transformers. En observant comment ces stratégies évoluent et s'adaptent à ces nouveaux modèles, nous espérons identifier les forces et les faiblesses de chaque approche, ainsi que les éventuels obstacles à leur mise en œuvre.

Une autre dimension de notre étude consiste à comparer ces stratégies d'AA à un échantillonnage aléatoire, souvent considéré comme la méthode de référence en l'absence de connaissances préalables [200]. Nous cherchons à déterminer si certaines stratégies se distinguent de manière significative par rapport à cette approche de base, et si ces différences sont cohérentes à travers différents modèles et jeux de données. Une telle analyse nous permettra de dégager des recommandations pratiques quant au choix des stratégies d'AA les plus adaptées aux transformers.

L'AA a pour promesse d'améliorer la qualité d'un projet d'annotation. Cette amélioration de la qualité se manifeste généralement par des performances supérieures du modèle entraîné avec les données sélectionnées par l'AA par rapport à un modèle entraîné à partir d'un échantillonnage aléatoire. Cette promesse revêt une importance particulière, surtout puisqu'elle intervient à l'étape de l'annotation, l'étape souvent initial d'un projet d'entraînement. L'AA est souvent envisagé dans des contextes où l'optimisation des ressources est primordiale pour atteindre des performances optimales, et donc possiblement dans des environnements industriels. Dans de tels cas, il est essentiel de justifier l'effort lié à l'intégration de l'AA grâce à une certitude élevée de gain significatif de performance. Cependant, avec l'émergence de techniques d'entraînement semi-supervisé [58], où les données peuvent être utilisées sans annotation préalable, il est d'autant plus impératif de vérifier si la promesse de l'AA reste valide dans le contexte des transformers. En effet, si l'intégration de l'AA pendant le processus d'annotation ne se traduit pas par une amélioration significative et relativement certaine des performances du modèle final, l'adoption de ces approches devient beaucoup moins justifiée. Ainsi, il est crucial de mener des études approfondies pour évaluer l'efficacité de l'AA dans ce nouveau contexte, afin de garantir des progrès significatifs lorsque appliqué aux transformers.

La tâche de classification multi-labels consiste à assigner les labels appropriés à des instances textuelles. Contrairement à la classification multi-classes, plusieurs labels peuvent être associés à une même instance. Pour chacune de nos expériences, notre espace de label est prédéfini et n'évolue pas au fur et à mesure. L'objectif de l'AA est de sélectionner les meilleures instances possibles à annoter pour l'entraînement. Cette sélection peut se faire selon différentes *stratégies*. Nos stratégies sont basées sur l'estimation de l'*incertitude* du modèle sur chaque instance, c'est-à-dire la confiance du modèle dans la prédiction des labels associés à cette instance. Ces stratégies sont basées autour de l'hypothèse qu'en s'entraînant sur des exemples difficiles (où le modèle hésite), le modèle va gagner en performance.

4.2.2 *Mise en œuvre*

Dans nos expérimentations, l'AA multi-labels suit le processus suivant : tout d'abord, nos modèles sont initialisés en les entraînant avec 25 instances sélectionnées aléatoirement. Ensuite, pour chaque stratégie, nous effectuons 50 itérations d'AA où pour chaque instance non-annotée, nous calculons un score qui indique l'incertitude du modèle sur ses prédictions associées puis sélectionnons les 25 instances les plus incertaines pour être ensuite annotées par un oracle, ce qui permet de collecter un total de 1250 instances annotées par stratégie. Après chaque itération, nous entraînons à nouveau le modèle avec le nouveau lot d'instances annotées. Ces valeurs sont sélectionnées afin d'être comparable à des travaux proches [195].

Nous comparons six des stratégies les plus populaires d'AA multi-labels détaillées dans la partie 2.2.3[63, 130, 131, 176] sur deux transformers, distilBERT et distilRoBERTa [48, 141, 188], les versions distillés de deux transformers à l'usage très répandu. Nos expériences sont menées sur quatre jeux de données multi-labels et nous reportons nos résultats via une métrique largement utilisée afin de les généraliser. Ces expérimentations ont été réalisées grâce à la librairie *small-text* [1] [194].

Les résultats sont une moyenne calculée sur cinq exécutions de chaque expérience, l'écart-type est indiqué sur les graphes.

4.2.3 *Jeux de données*

Pour les modèles antérieurs aux transformers, l'ensemble de jeux de données référence dans la classification multi-labels était Mulan [230]. Les stratégies d'AA que nous étudions ont souvent été évaluées sur cet ensemble. Cependant, comme les instances sont données sous forme de caractéristiques numériques et que le texte original brut

1. https://small-text.readthedocs.io/en/latest/

TABLE 1 – Caractéristiques des jeux de données

Nom	Labels	Entraînement	Test	Cardinalité	Densité
Jigsaw_toxic	6	159,571	63,978	0.222	0.037
Go_emotions	27	43,410	5,427	0.848	0.031
EUR_Lex	100	55,000	5,000	4.526	0.036
UNFAIR-ToS	8	5,532	1,607	0.124	0.016

n'est pas fourni, ce jeu de données n'est pas approprié pour l'apprentissage des transformers.

Le Tableau 1 montre les caractéristiques des quatre jeux de données utilisés. La cardinalité est le nombre moyen de labels par instance. La densité est la cardinalité divisée par le nombre total d'instances.

Nous avons choisi ces jeux de données afin de réaliser nos expériences sur des textes présentant une variation du niveau de langue employé (de l'insulte au texte légal, en passant par des commentaires de réseaux sociaux) ainsi qu'une variation du nombre de labels associés à chaque instance (de 6 à 100).

La cardinalité correspond au nombre moyen de labels par instance. La densité correspond à la cardinalité divisée par le nombre total d'instances. EUR_Lex est le jeu de données le plus complexe, son espace de labels est plus grand et sa cardinalité cinq fois plus élevée que pour les autres.

4.2.3.1 *Jigsaw toxic comment classification (Jigsaw_Toxic)*

Jigsaw_Toxic est le jeu de données associé à une compétition Kaggle [2] dont le but est de détecter et classifier six différents types de toxicité que l'on peut trouver en ligne. Les instances sont issues de commentaires sur des pages wikipédia. Les différents labels, se référant à différents types de toxicité, sont souvent corrélés (par exemple, toutes les instances de "toxicité sévère" sont aussi labellisées "toxicité"). Près de 90% des instances du jeu de données ne présentent aucune forme de toxicité et ne sont donc associées à aucun label. Comme on peut le voir dans la Figure 12, la répartition des labels est déséquilibrée.

Une instance textuelle typique de ce jeu de données est par exemple : *Hi ! I am back again ! Last warning ! Stop undoing my edits or die !* associée aux catégories "toxique" et "menace".

4.2.3.2 *Go_Emotions*

Go_Emotions est un jeu de données composé de commentaires Reddit [3] labelisés sur 27 catégories d'émotions comme "colère" ou

2. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
3. https://www.reddit.com/

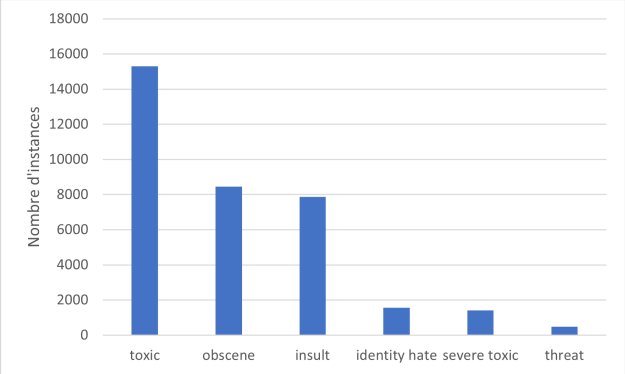


FIGURE 12 – Nombre d'instances par label dans le jeu de données Jigsaw_Toxic.

"curieux" [47]. Un peu plus de 30% des instances sont "neutres", c'est-à-dire non associées avec un label. Comme on peut le voir dans la Figure 13 la répartition entre labels est déséquilibrée.

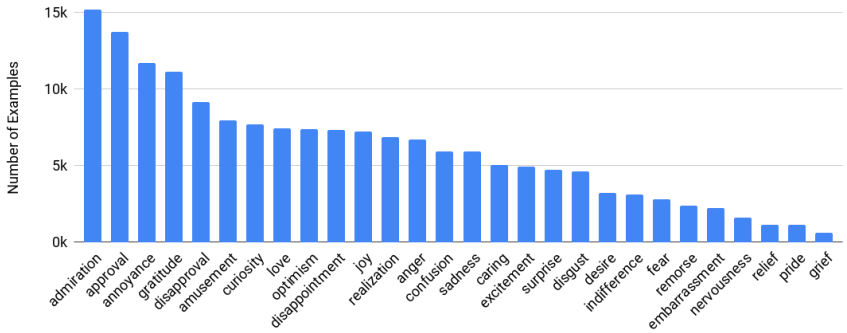


FIGURE 13 – Nombre d'instances par label dans le jeu de données Go_Emotions. [4]

Une instance textuelle typique de ce jeu de données est par exemple : *I have considered that, and if is possible, I will see where it leads me, thanks.* associée aux catégories "gratitude" et "optimisme".

4.2.3.3 *EUR_Lex57K (EUR_Lex)*

EUR_Lex est un jeu de données composé de textes légaux [30] issus du site du même nom [5]. Ce jeu de données est souvent utilisé dans les travaux sur la classification multi-labels extrême grâce à son nombre important de labels possibles (supérieur à 4000). Les labels correspondent à des descripteurs de la hiérarchie EUROVOC [6]. Le déséquilibre des labels est très grand sur ce jeu de données, comme on peut le voir dans la Figure 14.

4. https://blog.research.google/2021/10/goemotions-dataset-for-fine-grained.html
5. https://eur-lex.europa.eu
6. https ://op.europa.eu/en/web/eu-vocabularies

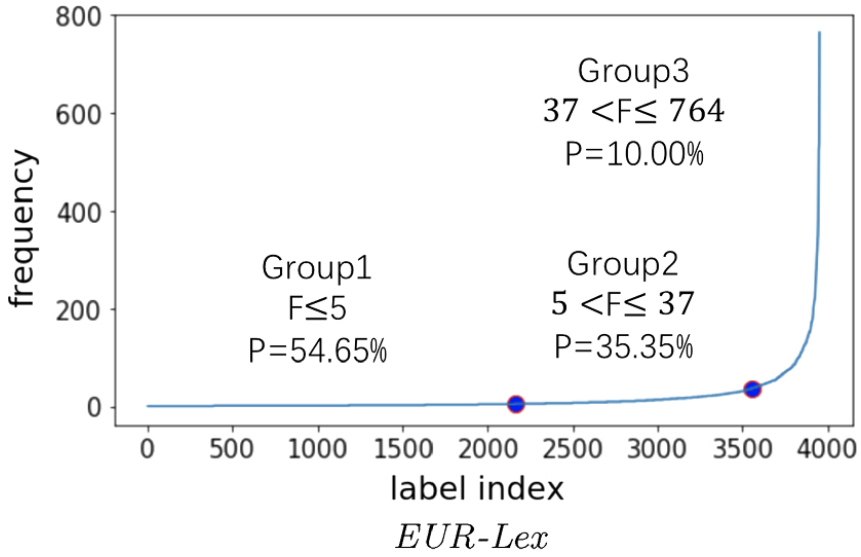


FIGURE 14 – Distribution de la fréquence des libellés EUR_Lex [102]. Tous les labels sont répartis en trois groupes (dans chaque groupe, F indique la fréquence du libellé, P est la proportion que les libellés dans le groupe représentent par rapport à l'ensemble complet des libellés).

Une instance textuelle typique de ce jeu de données est beaucoup plus longue, par exemple : *Council Decision 2003/484/CFSP of 27 June 2003 implementing Common Position 2003/280/CFSP in support of the effective implementation of the mandate of the International Criminal Tribunal of the former Yugoslavia (ICTY) THE COUNCIL OF THE EUROPEAN UNION, Having regard to Council Common Position 2003/280/CFSP of 16 April 2003 in support of the effective implementation of the mandate of the ICTY(1), and in particular Article 2 thereof, in conjunction with Article 23(2) of the Treaty on European Union, Whereas : (1) By Common Position 2003/280/CFSP the Council adopted measures to prevent the entry into, or transit through, the territories of Member States of individuals who are engaged in activities which help persons at large continue to evade justice for crimes for which the ICTY has indicted them. (2) Following recommendations from the office of the High Representative for Bosnia and Herzegovina, further individuals should be targeted by those measures, HAS DECIDED AS FOLLOWS : Article 1 The list of persons set out in the Annex to Common Position 2003/280/CFSP is hereby replaced by the list set out in the Annex to this Decision. Article 2 This Decision shall take effect on the date of its adoption. Article 3 This Decision shall be published in the Official Journal of the European Union. Done at Brussels, 27 June 2003.* La cardinalité étant en général assez élevée sur ce jeu de données, cette instance est, par exemple, associée à 5 labels.

4.2.3.4 *UNFAIR - Terms of Services (UNFAIR-ToS)*

UNFAIR-ToS est un jeu de données composé de textes annotés avec huit types de termes contractuels injustes [134], c'est-à-dire, des termes qui violent potentiellement les droits des consommateurs selon la loi de consommation européenne. Comme on peut le voir dans la Figure 15, il existe un déséquilibre entre les labels de UNFAIR-ToS mais qui est proportionnellement moins important que sur les trois autres jeux de données.

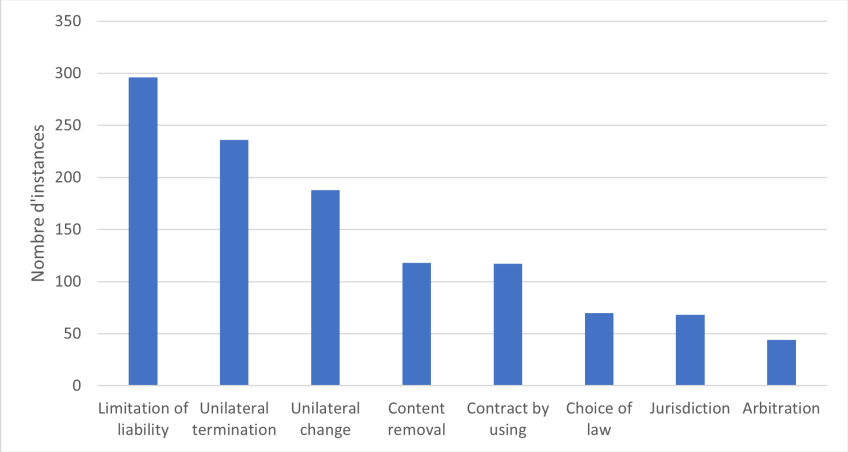


FIGURE 15 – Nombre d'instances par label dans le jeu de données UNFAIR-ToS.

Une instance textuelle typique de ce jeu de données présentes des tournures légales subtiles : *we may make changes to this agreement and to the services from time to time.* associée au label "changement unilatéral".

Dans nos travaux, nous avons utilisé les versions de EUR_Lex et UNFAIR-ToS fournies dans *Legal General Language Understanding Evaluation* (LexGLUE) [31].

Dans nos expériences, 10% du jeu d'entraînement est utilisé pour la validation et les résultats sont obtenus sur le jeu de test. Nous avons utilisé les versions de EUR_Lex et UNFAIR-ToS disponibles dans LexGLUE [31].

4.2.4 *Références et modèles*

4.2.4.1 *Oracle*

La simulation d'un oracle humain, annotant les instances non-annotées sélectionnées par les différentes stratégies, est réalisée par l'utilisation des jeux de données multi-labels annotés. A chaque itération d'entraînement, la stratégie d'AA sélectionne les meilleurs instances à partir du jeu d'entraînement sans avoir accès aux labels correspondants. Ces instances et leurs labels associés composent le prochain lot d'entraînement.

4.2.4.2 *Modèles et références*

Comme dans [193], les deux transformers utilisés dans cette étude sont basés sur BERT [48] et RoBERTa[141], deux transformers très populaire. Étant donné que [232] montre que dans les processus d'AA les versions distillées de ces modèles obtiennent des performances similaires à celles obtenues par les modèles originaux, tout en étant moins gourmands en ressources informatiques, nous utilisons également les versions distillées de ces modèles [188]. Au-dessus des deux modèles, nous ajoutons une couche de neurones dense de la taille de la sortie du modèle ainsi qu'une couche sigmoïde afin de réaliser la classification multi-labels.

Random (RD) est une référence commune dans l'AA, où les instances à annoter sont échantillonnées de manière aléatoire à partir du jeu de données non-annotées. Afin d'évaluer les performances finales atteintes par les différentes stratégies, nous utilisons comme référence le modèle entraîné sur l'ensemble du jeu de données en supervision totale. Nous nommons cette valeur de référence **Full-Supervision (FULL)**.

Afin de comparer l'utilisation d'algorithmes d'AA sur les transformers, nous entraînons une architecture de SVM et plus précisement une *Linear Support Vector Classification*[7]. Le choix de cette architecture n'est pas anodin puisqu'elle est proche de celles utilisées dans les travaux présentant les stratégies que nous étudions. Nous utilisons des caractéristiques obtenues grâce à l'utilisation d'un vectoriseur TF-IDF[8] sur nos instances de texte brut, avec un paramètre max_features fixé à 50 000. Comme proposé dans [251], un noyau linéaire et un paramètre de pénalité fixé à 1.0 ont été utilisés.

4.2.4.3 *Paramètres et détails d'implémentation*

DistilBERT est composé de 6 couches, des unités cachées de taille 768 et de 66 millions de paramètres. DistilRoBERTa est structuré d'une façon comparable, à l'exception de son nombre de paramètres, qui est de 82 millions. La taille maximum des tokens d'entrées pour les deux modèles est fixée à 128, le nombre d'époques à 15 et la taille des lots d'entraînement à 25. Pour optimiser les paramètres des modèles, nous avons choisi AdamW avec un taux d'apprentissage de 2e-5. Les expériences ont été réalisées sur une Nvidia GTX1080Ti (32 GB). Les mêmes hyper-paramètres ont été utilisés par les deux modèles sur les quatre jeux de données. Nous suivons [94] pour notre processus de *fine-tuning*.

7. https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC
8. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

4.2.5 *Évaluation*

Pour mesurer la performance, nous utilisons une métrique communément utilisée dans la classification multi-labels [229]. Nous utilisons les notations de la Partie 2.1.1, en ajoutant : pour un label donné l^j nous notons les vrais positifs (vp^j), les faux positifs (fp^j), les faux négatifs (fn^j) et définissons la F1-mesure comme :

$$F1(vp^j, fp^j, fn^j) = \frac{vp^j}{vp^j + \frac{1}{2}(fp^j + fn^j)} \tag{15}$$

Nous utilisons une F1-mesure avec une micro-moyenne, c'est-à-dire que nous faisons la somme de tous les vrais positifs, faux positifs et faux négatifs pour tous les labels puis nous calculons la F1-mesure (plus la valeur est haute plus la performance est bonne) :

$$M_{iF1} = F_1(\sum_{j=1}^{q} vp^j, \sum_{j=1}^{q} fp^j, \sum_{j=1}^{q} fn^j) \tag{16}$$

Pour mieux comprendre comment certains biais de sélection peuvent avoir un impact sur les performances d'une stratégie d'AA, nous avons examiné plusieurs caractéristiques des instances : la taille et la cardinalité, ainsi que certaines caractéristiques des lots : la présence d'aberrations et la similarité des instances. Lorsque l'on adapte des stratégies d'AA *myopic* à de l'apprentissage par lot, il y a un risque d'avoir au sein d'un même lot des instances vectrices d'informations redondantes [78]. Cette redondance d'information au sein des lots étant proche conceptuellement de la similarité des instances au sein d'un lot, nous nous concentrons sur cette caractéristique.

Afin de mieux répondre à la problématique de redondances des instances similaires, nous proposons d'évaluer la similarité entre les instances sélectionnées en calculant `Sim_batch`, selon la méthode suivante (détaillée dans l'algorithme 1) : pour commencer nous calculons les plongements lexicaux de chaque instance dans le lot grâce à SentenceTransformer [173][9]. Ensuite, nous faisons la moyenne pour chaque lot de la similarité cosinus par paire entre les plongements lexicaux présents dans le lot. Enfin, nous calculons `Sim_batch`, la moyenne du score de similarité sur tous les lots.

4.2.6 *Résultats et analyses*

La Figure 16 montre les courbes d'apprentissage en M_{iF1} de nos deux transformers selon chaque stratégie d'AA sur les différents jeux

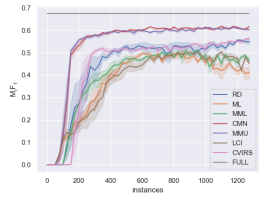
9. nous avons utilisé 'sentence-transformers/nli-distilroberta-base-v2' sur distilRoBERTa et 'sentence-transformers/nli-distilbert-base' sur distilBERT

Algorithme 1 Calcul du score Sim_batch

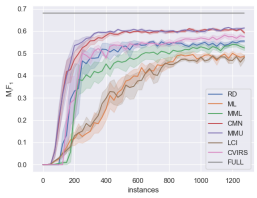
Soit : 'jeu_annote' la liste des lots de données annotées accumulées au fur et à mesure de l'exécution des stratégies d'apprentissage actif
moyenne_lot ← [ ]
pour chaque lot dans jeu_annote **:**
pl ← [ ]
pour chaque instance dans lot **:**
pl.ajouter(SentenceTransformer(instance))
fin pour chaque
moyenne_lot.ajouter($\frac{\sum_{x \in paires} similarite\_cosinus(x)}{\frac{|pl|*(|pl|-1)}{2}}$) , avec 'paires' la liste des paires uniques d'éléments $\in$ pl
fin pour chaque
Sim_batch $= \frac{\sum_{x \in moyenne\_lot} x}{|moyenne\_lot|}$
retourner Sim_batch

de données. Ces résultats indiquent que les stratégies d'AA peuvent améliorer de façon significative les performances atteintes, s'approchant d'une supervision complète à un coût moindre d'annotations. En effet, pour atteindre les performances de l'échantillonnage aléatoire après sélection de 1250 instances, MMU nécessite seulement 175 instances pour distilBERT (225 pour distilRoBERTa) et CMN 175 instances pour distilBERT (275 pour disilRoBERTa). L'écart-type des performances au cours des différentes exécutions est faible pour les stratégies MML, CMN, MMU et CVIRS sur les différents jeux de données, présentant un écart non-significatif (p-value>0.05). En effet, la "graine aléatoire" ne détermine que la composition du lot de données d'initialisation dans toutes les stratégies, à l'exception de RD dans laquelle elle joue un rôle plus important. Nous observons dans ces graphes que les modèles de SVM performent relativement bien par rapport aux transformers ce qui en fait une bonne référence.

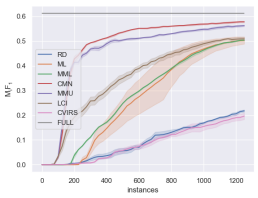
Dans le Tableau 2 nous indiquons le pourcentage du jeu de données que représentent les 1250 instances annotées et le pourcentage des performances de FULL qu'atteignent CMN et MMU, les deux stratégies d'AA les plus performantes, comparativement à l'échantillonnage aléatoire RD. Par exemple pour Jigsaw_Toxic, nous voyons qu'avec moins de 1% du jeu de données (0,78%), nous atteignons plus de 90% des performances de la supervision complète pour CMN (91,54% pour distilBERT et 90,29% pour distilRoBERTa), contre seulement 81% de ces performances avec RD (81,45% pour distilBERT et 81,62% pour distilRoBERTa). De plus, nous voyons sur la Figure 16a que CMN et MMU atteignent ces performances autour de seulement 400 données sélectionnées. Ces résultats mettent en évidence les gains potentiels liés à l'application de stratégies d'AA performantes sur les transformers.



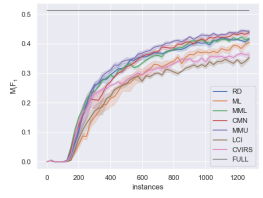
(a) distilBERT sur Jigsaw_toxic



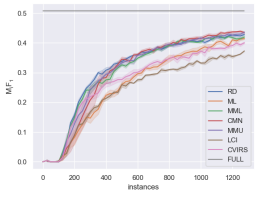
(b) distilRoBERTa sur Jigsaw_toxic



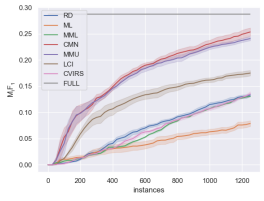
(c) SVM sur Jigsaw_toxic



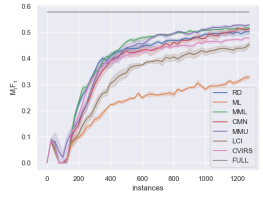
(d) distilBERT sur Go_emotions



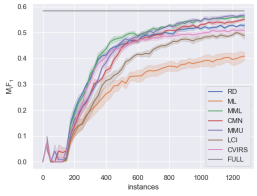
(e) distilRoBERTa sur Go_emotions



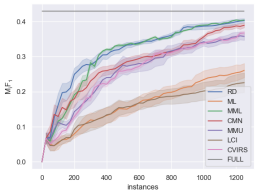
(f) SVM sur Go_emotions



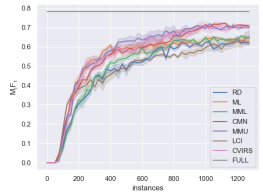
(g) distilBERT sur EUR_Lex



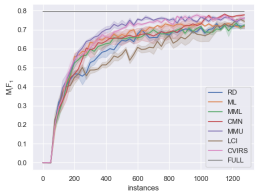
(h) distilRoBERTa sur EUR_Lex



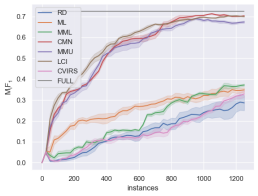
(i) SVM sur EUR_Lex



(j) distilBERT sur UNFAIR-ToS



(k) distilRoBERTa sur UNFAIR-ToS



(l) SVM sur UNFAIR-ToS

FIGURE 16 – Performances M_{iF1} suivant les différentes stratégies d'AA pour chaque transformers

TABLE 2 – Pourcentage des jeux de données annotés, associés aux pourcentages de performances atteintes par les stratégies par rapport à une supervision complète.

Jeux de données	Modèles	% jeu de données	% performances M_{iF1}		
			RD	CMN	MMU
Jigsaw	distilBERT	0.78	81.45	91.54	89.47
	distilRoBERTa	0.78	81.62	87.21	90.29
goemotions	distilBERT	2.88	81.25	85.55	86.13
	distilRoBERTa	2.88	82.84	85.80	85.80
eurlex	distilBERT	2.27	86.28	88.00	90.9
	distilRoBERTa	2.27	91.52	95.16	98.1
unfairtos	distilBERT	22.60	79.28	90.54	89.51
	distilRoBERTa	22.60	90.19	97.86	95.85

Le Tableau 3 compare les performances obtenues après entraînement sur 1250 instances annotées pour chaque combinaison de modèle, jeu de données et stratégies. Afin de mieux illustrer la relation entre les performances du modèle et la similarité des instances sélectionnées, nous avons ajouté le score Sim_batch à côté de la métrique de performance (M_{iF1}/Sim_batch).

L'un des résultats que nous pouvons tirer du Tableau 3 est qu'à la fin de l'entraînement, la performance (en terme de M_{iF1}) des transformers dépend de la stratégie d'AA. Cela suggère que l'ordre des instances sur lequel les transformers s'entraînent a bien une importance. Pour les deux transformers, CMN et MMU surpassent constamment l'échantillonnage aléatoire et toutes les autres stratégies AA, se rapprochant le plus de FULL sur tous les jeux de données.

Une seconde observation est que contrairement aux transformers, pour les SVM, la plupart des stratégies d'AA performent mieux que l'échantillonnage aléatoire. Cela confirme que les stratégies d'AA ne fonctionnent pas toutes aussi bien pour les transformers que pour les SVM où l'on peut être plus certain d'avoir un gain de performance en faisant de l'AA.

Une troisième observation que nous pouvons faire, est qu'à chaque fois qu'une stratégie d'AA performe moins bien que l'échantillonnage aléatoire, le score Sim_batch est significativement plus élevé que pour les stratégies performantes ou que l'échantillonnage aléatoire. Dans le Tableau 4, nous remarquons une corrélation linéaire (corrélation de Pearson) entre M_{iF1} et Sim_batch. En effet, il y a une corrélation statistiquement significative (p-value<0.05) entre les performances des stratégies d'AA et le score *Sim_batch*. Plus les instances sélectionnées sont similaires, plus la performance obtenue est

basse, à l'exception du jeu de donnée UNFAIR-ToS, pour lequel la corrélation est inversée.

En sélectionnant des instances similaires, certaines stratégies entraînent le modèle sur des informations redondantes, ce qui explique la corrélation obtenue sur les trois premiers jeux de données. La Figure 17, montre l'évolution du score *Sim_batch* au fil de l'apprentissage. Nous remarquons une grande augmentation de score Sim_batch en début d'expérience pour de nombreuses stratégies. Lorsque nous regardons en détail les lots associés à ces fortes augmentations, nous constatons la présence de nombreuses paires de données similaires et même pratiquement identiques. Lorsque ces paires de données très similaires intègrent les lots d'entraînement, cela entraîne un effet domino avec une proportion de paires similaires au sein des lots de plus en plus importante. En poussant plus loin cette analyse, nous remarquons également ce qui différencie les stratégies performantes des autres : lorsque cet effet domino s'enclenche, CMN et MMU parviennent à en sortir plus rapidement que les autres stratégies. En effet, les courbes de Sim_batch pour CMN et MMU ont l'allure d'un pic tandis que les courbes des autres stratégies ont plutôt l'allure d'un plateau.

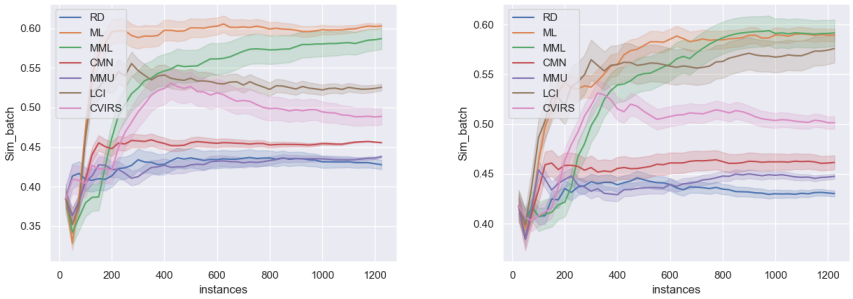
Pour expliquer la corrélation inversée sur UNFAIR-ToS, un jeu de données composé de phrases annotées de huit différents types de termes contractuels injustes, nous estimons que la similarité des instances joue un rôle clé pour comparer des contrats proches et identifier les parties injustes. Cela peut aussi expliquer pourquoi UNFAIR-ToS est le seul jeu de données pour lequel toutes les stratégies d'AA performent mieux que l'échantillonnage aléatoire.

4.2.7 *Bilan*

Ces expériences évaluent l'impact sur les transformers de six stratégies d'apprentissage actifs basées sur l'incertitude. Nous avons montré que deux de ces stratégies, CMN et MMU, fournissent un gain de performance constant et substantiel par rapport à l'échantillonnage aléatoire, soulignant l'utilité de l'AA appliquée aux transformers. Ces deux stratégies seraient notamment de bonnes références pour des prochains travaux sur l'AA multi-labels dans le contexte des transformers. Les quatre autres stratégies fournissent des résultats équivalents ou inférieurs à l'échantillonnage aléatoire. Nous avons investigué les raisons possibles de ces différences de performances. Nous avons d'abord trouvé que CMN et MMU sélectionnent en moyenne des lots d'instances avec une diversité textuelle plus grande que ceux sélectionnés par les stratégies en sous-performance. En poussant notre analyse, nous avons ensuite mis en évidence qu'en début d'expérience, certaines stratégies d'AA sélectionnent des lots de données avec de plus en plus de paires d'instances similaires : CMN et MMU

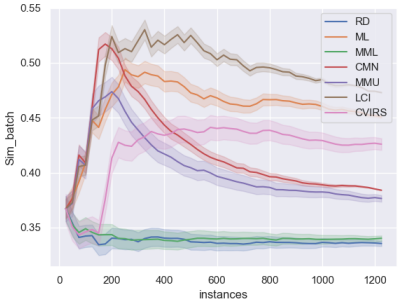
TABLE 3 – "M_{iF1}/Sim_batch", obtenu suivant chaque stratégies d'AA. Les résultats en rouge indiquent un score M_{iF1} inférieur à l'échantillonnage aléatoire, les résultats en bleu indiquent un score M_{iF1} supérieur à l'échantillonnage aléatoire.

Jeux de données	Modèles	RD	ML	MML	CMN	MMU	LCI	CVIRS	FULL
Jigsaw_toxic	SVM	0.223/0.122	0.505/0.318	0.506/0.292	0.580/0.156	0.564/0.149	0.510/0.252	0.455/0.173	0.612/-
	distilBERT	0.549/0.426	0.412/0.602	0.467/0.588	0.617/0.455	0.603/0.439	0.453/0.526	0.564/0.487	0.674/-
	distilRoBERTa	0.555/0.430	0.486/0.590	0.524/0.592	0.593/0.461	0.614/0.448	0.482 /0.578	0.575/0.498	0.680/-
Go_emotions	SVM	0.135/0.053	0.082/0.077	0.131/0.054	0.254/0.078	0.242/0.082	0.176/0.166	0.213/0.117	0.287/-
	distilBERT	0.416/0.336	0.404/0.456	0.417/0.341	0.438/0.383	0.441/0.377	0.353/0.474	0.362/0.426	0.512/-
	distilRoBERTa	0.420/0.339	0.414/0.453	0.428/0.341	0.435/0.369	0.435/0.368	0.373/0.463	0.400/0.419	0.507/-
EUR_Lex	SVM	0.403/0.666	0.261/0.738	0.410/0.663	0.390/0.666	0.359/0.663	0.228/0.783	0.213/0.725	0.429/-
	distilBERT	0.503/0.741	0.329/0.788	0.517/0.740	0.513/0.736	0.530/0.738	0.454/0.752	0.479/0.758	0.583/-
	distilRoBERTa	0.529/0.736	0.409/0.781	0.560/0.739	0.550/0.746	0.567/0.742	0.488/0.765	0.509/0.756	0.578/-
UNFAIR-ToS	SVM	0.296/0.210	0.360/0.245	0.376/0.280	0.702/0.313	0.670/0.314	0.700/0.305	0.535/0.223	0.724/-
	distilBERT	0.620/0.454	0.622/0.490	0.647/0.491	0.708/0.522	0.700/0.504	0.650/0.497	0.708/0.523	0.782/-
	distilRoBERTa	0.717/0.455	0.718/0.512	0.724/0.492	0.778/0.553	0.762/0.535	0.744/0.539	0.760/0.511	0.795/-

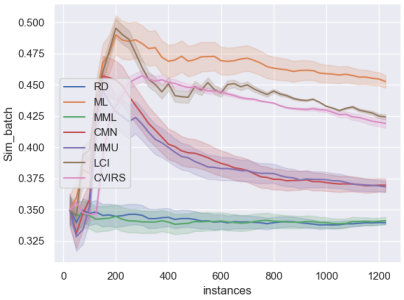


(a) distilBERT sur Jigsaw_toxic

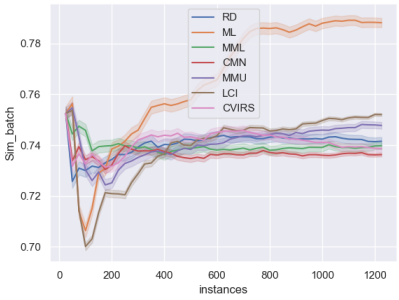
(b) distilRoBERTa sur Jigsaw_toxic



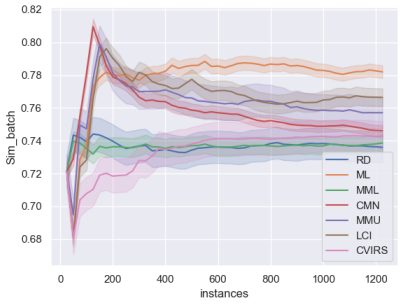
(c) distilBERT sur Go_emotions



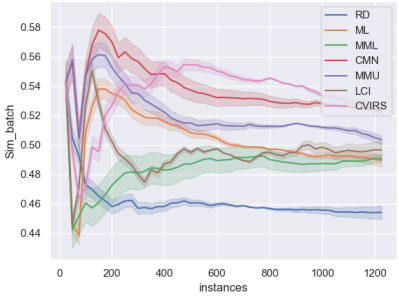
(d) distilRoBERTa sur Go_emotions



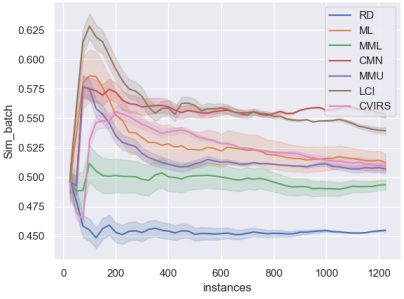
(e) distilBERT sur EUR_Lex



(f) distilRoBERTa sur EUR_Lex



(g) distilBERT sur UNFAIR-ToS



(h) distilRoBERTa sur UNFAIR-ToS

FIGURE 17 – *Sim_batch* suivant les différentes stratégies d'AA pour chaque transformers

TABLE 4 – Corrélation de Pearson entre M_{iF1} et *Sim_batch*

Jeux de données	Modèles	Pearson	p-value
Jigsaw_toxic	distilBERT	-0.877	0.0096
	distilRoBERTa	-0.843	0.0170
Go_emotions	distilBERT	-0.781	0.0381
	distilRoBERTa	-0.769	0.0431
EUR_Lex	distilBERT	-0.970	0.0003
	distilRoBERTa	-0.930	0.0024
UNFAIR-ToS	distilBERT	0.863	0.0123
	distilRoBERTa	0.776	0.0419

se différenciant par le fait qu'elles arrivent mieux à re-sélectionner des lots avec de moins en moins de paires d'instances similaires dans les itérations d'apprentissage suivantes.

Fort ce ces constatations, nous émettons l'hypothèse que réduire le taux de similarité des instances de chaque lot d'apprentissage permettra d'améliorer la performance de certaines stratégies. Afin d'explorer cette hypothèse nous avons donc mis en place d'autres expériences, forçant les instances sélectionnées au cours de l'AA à ne pas être redondantes les unes avec les autres.

4.3 RÉDUCTION DE LA REDONDANCE DES INSTANCES SÉLECTIONNÉES

Afin de réduire la similarité des instances sélectionnées nous prooposons d'inclure un pré-clustering en amont de l'application des stratégies AA basées sur l'incertitude. L'idée fondamentale derrière cette méthodologie réside dans le regroupement des instances similaires au sein de clusters, puis de représenter ces clusters par une unique instance réduisant ainsi la probabilité de sélection répétée d'instances très similaires par les stratégies d'AA. L'idée est qu'en évitant la redondance dans le choix des instances, ceci permettrait aux stratégies d'AA d'opérer de manière plus efficace, en se concentrant sur une diversité plus large d'instances et en évitant la concentration excessive sur des sous-ensembles d'informations similaires. Cette approche peut être particulièrement pertinente dans le contexte des modèles basés sur les transformers, puisque s'agissant de pré-traitement, la charge de calcul a lieu en amont du déploiement des processus d'AA et ne ralentit donc pas l'interaction entre l'annotateur et le modèle.

4.3.1 *Architecture pré-clustering*

Dans cette partie, on procède à un pré-clustering hiérarchique en se basant sur des scores de similarité obtenus par la distance cosinus entre les plongements lexicaux des instances. Cette approche consiste à regrouper les instances dont le score de similarité dépasse le seuil de 0.9 au sein de clusters. Pour ce faire, une première passe est effectuée sur l'ensemble des instances textuelles afin d'évaluer leur similarité avec toutes les autres instances, regroupant les instances si leur score de similarité est supérieur à une valeur seuil. Une deuxième passe est ensuite réalisée de manière à ce que chaque instance n'appartienne qu'à un cluster, celui pour lequel sa similarité est la plus élevée. Enfin, la dernière étape de ce pré-traitement vise à ne conserver qu'une seule instance par cluster, en choisissant l'instance centroïde de ce dernier. Le centroïde est un point qui minimise la somme des distances euclidiennes entre lui-même et tous les autres points du cluster. Ce processus de pré-clustering hiérarchique permet ainsi de réduire la redondance au sein des instances, en ne conservant que des représentants de chaque groupe sémantique distinct.

Afin d'expliquer cette valeur de seuil et de mieux visualiser ce qu'un seuil de similarité à 0.9 signifie voici quelques paires d'instances ci-dessous associées à leur score de similarité :

— score de 0.907 : *I have, and now that you mention it, I think that's what triggered my nostalgia. / I remember, and now that it's come to mind, I realize that's what stirred up my nostalgic feelings.*
— score de 0.991 : *I have, and now that you mention it, I think that's what triggered my nostalgia. / I have, and speaking of it, I guess that's what caused my longing for the past.*
— score de 0.893 : *I have, and now that you mention it, I think that's what triggered my nostalgia. / I have, and now that you mention it, I think that's what makes me love you.*

4.3.2 *Résultats et analyses*

Nous répétons la même expérience d'entraînement que dans la partie 4.2.6 en rajoutant l'étape de pré-clustering. La Figure 18 montre les courbes de performances après pré-clustering. En comparant ces graphes avec ceux de la Figure 16, on ne révèle pas de différences significatives en termes d'amélioration notable. Malgré l'application du pré-clustering visant à réduire la redondance entre les instances, il apparaît que certaines stratégies d'apprentissage actif continuent de présenter des performances inférieures à celles d'un simple échantillonnage aléatoire. De plus, pour plusieurs stratégies, on observe même une dégradation des performances, caractérisée par des courbes qui mettent davantage de temps avant de connaître une croissance significative. Cette observation suggère que le pré-clustering ne garantit

pas nécessairement une amélioration uniforme et systématique des résultats pour toutes les stratégies d'AA.

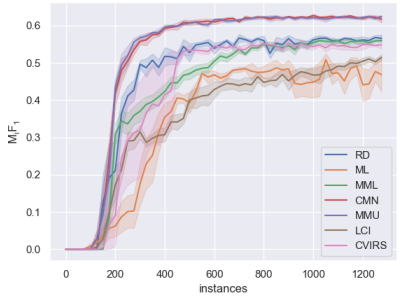
Les résultats présentés dans le tableau 5, comparant les performances avant et après le pré-clustering, révèlent une amélioration dans seulement 9 des 16 cas étudiés. Cette observation a incité la réalisation de tests statistiques qui ont démontré l'absence de différences statistiquement significatives entre les résultats obtenus avant et après le pré-clustering. Constatant que le pré-clustering ne modifie pas significativement les performances observées, une explication possible serait que la redondance, parfois présente dans les instances proches sémantiquement mais labellisées différemment, peut être bénéfique. En effet, ces instances offrent au modèle la possibilité d'apprendre des subtilités et de renforcer des caractéristiques distinctives de chaque label. Par conséquent, la perte de ces instances redondantes utiles semble compenser et annuler les gains de performances potentiels liés à la réduction de la redondance des instances qui ne fournissent pas d'informations complémentaires au modèle.

En conclusion, notre analyse suggère que l'ajout de diversité par pré-clustering, et plus spécifiquement la suppression de la redondance des instances, ne constitue pas nécessairement une solution directe pour améliorer les performances des stratégies d'apprentissage actif ML, MML, LCI et CVIRS dans le cadre d'une application aux transformers.

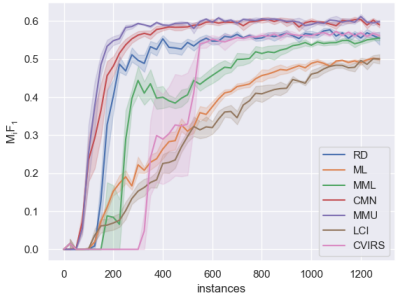
4.4 DISCUSSION ET CONCLUSION

Ce chapitre met en évidence des résultats variés quant à l'efficacité des stratégies d'AA appliquées aux transformers. Bien que certaines stratégies (CMN et MMU) aient démontré des gains significatifs et constants dans l'amélioration des performances des transformers, cette tendance n'est pas universelle et toutes les stratégies ne produisent pas les mêmes résultats. Un constat intéressant émerge de l'observation selon laquelle certaines stratégies, efficaces pour les SVM, ne semblent pas offrir les mêmes avantages aux transformers, renforçant ainsi la nécessité d'une évaluation spécifique à ces architectures.

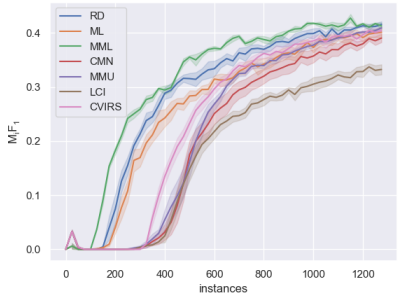
Dans notre quête d'explications, une analyse approfondie a révélé que la redondance des instances sélectionnées par les stratégies moins performantes est plus prononcée que celle des stratégies performantes. Cette observation a conduit à une série d'expériences visant à évaluer l'impact de la réduction de la redondance des instances sur les performances des différentes stratégies. Les résultats de ces expériences n'ont toutefois pas montré de gains statistiquement significatifs, suggérant que la perte d'instances redondantes utiles peut compenser les bénéfices de la réduction de la redondance.



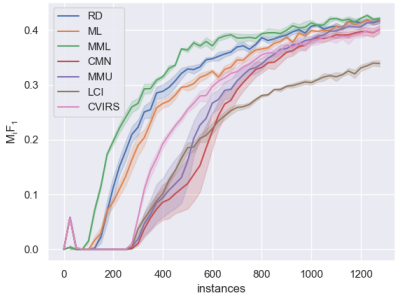
(a) distilBERT sur Jigsaw_toxic



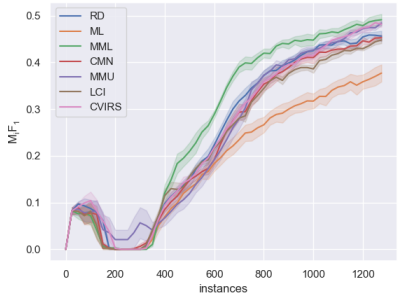
(b) distilRoBERTa sur Jigsaw_toxic



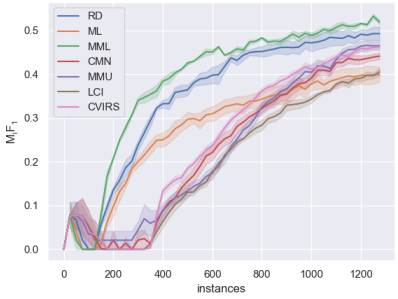
(c) distilBERT sur Go_emotions



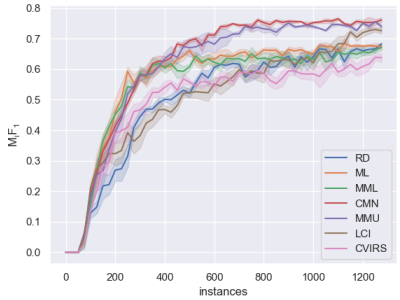
(d) distilRoBERTa sur Go_emotions



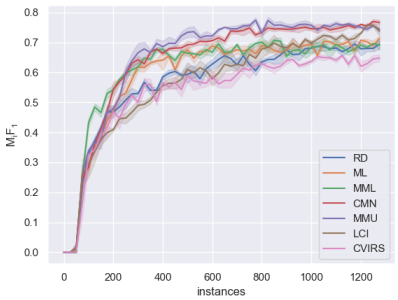
(e) distilBERT sur EUR_Lex



(f) distilRoBERTa sur EUR_Lex



(g) distilBERT sur UNFAIR-ToS



(h) distilRoBERTa sur UNFAIR-ToS

FIGURE 18 – Performances M_{iF1} suivant les différentes stratégies d'AA pour chaque transformers après pré-clustering

TABLE 5 – "M_{iF1} sans pré-clustering/M_{iF1} avec pré-clustering", obtenu suivant chaque stratégies d'AA. Les résultats en rouge indiquent un score M_{iF1} sans pré-clustering supérieur ou égale à avec pré-clustering , les résultats en bleu indiquent un score M_{iF1} sans pré-clustering inférieur à avec pré-clustering.

Jeux de données	Modèles	RD	ML	MML	CMN	MMU	LCI	CVIRS
Jigsaw_toxic	distilBERT	0.549/0.566	0.412/0.469	0.467/0.560	0.617/0.617	0.603/0.624	0.453/0.515	0.564/0.548
	distilRoBERTa	0.555/0.555	0.486/0.500	0.524/0.556	0.593/0.590	0.614/0.598	0.482 /0.500	0.575/0.561
Go_emotions	distilBERT	0.416/0.416	0.404/0.401	0.417/0.413	0.438/0.390	0.441/0.409	0.353/0.332	0.362/0.401
	distilRoBERTa	0.420/0.418	0.414/0.418	0.428/0.421	0.435/0.402	0.435/0.412	0.373/0.339	0.400/0.402
EUR_Lex	distilBERT	0.503/0.456	0.329/0.377	0.517/0.491	0.513/0.453	0.530/0.484	0.454/0.448	0.479/0.485
	distilRoBERTa	0.529/0.493	0.409/0.400	0.560/0.520	0.550/0.441	0.567/0.465	0.488/0.405	0.509/0.462
UNFAIR-ToS	distilBERT	0.620/0.683	0.622/0.674	0.647/0.671	0.708/0.761	0.700/0.740	0.650/0.726	0.708/0.638
	distilRoBERTa	0.717/0.694	0.718/0.713	0.724/0.690	0.778/0.766	0.762/0.744	0.744/0.737	0.760/0.648

Une perspective majeure de ce chapitre est que l'implémentation de l'AA avec des transformers ne garantit pas systématiquement un avantage de performance par rapport à un échantillonnage aléatoire. Ainsi, le choix judicieux d'une stratégie d'AA devient un facteur critique pour le succès d'un projet d'annotation, soulignant la nécessité d'une approche personnalisée et attentive à la spécificité des transformers.

5

VALORISATION DU TEMPS D'ATTENTE DES ANNOTATEURS DURANT UN CYCLE D'APPRENTISSAGE ACTIF

Une partie du contenu de ce chapitre a été publiée lors de la conférence LREC-COLING 2024. Dans la pratique, un aspect primordial de la conception de systèmes d'AA à prendre en compte est la rentabilisation du temps d'accès aux annotateurs. Le temps des annotateurs est souvent une ressource précieuse, notamment lorsque l'annotation requiert une forme d'expertise de leur part. Lors de la phase d'annotation, la quantité de temps d'attente des annotateurs représente un paramètre déterminant, impactant directement la productivité et l'efficacité du processus dans son entièreté. En effet, seul durant le temps d'annotation effectif (le temps que passe l'annotateur à utiliser l'interface d'annotation moins son temps d'attente) de nouvelles instances sont annotées. Nous souhaitons explorer des stratégies innovantes visant à valoriser le temps d'attente des annotateurs au cours des cycles d'apprentissage actif afin qu'il n'impacte plus négativement la réussite d'un projet d'annotation. En analysant des mécanismes existants, en identifiant leurs éventuelles lacunes et en proposant des améliorations adaptées, ce chapitre contribue à l'optimisation globale du processus d'annotation, offrant ainsi des pistes précieuses pour améliorer la rentabilité et l'efficacité des systèmes d'AA.

5.1 CONTEXTE ET MOTIVATIONS

Dans des environnements industriels, lorsqu'on vise à intégrer l'AA dans les flux de travail, des défis tant théoriques que pratiques émergent. La recherche se penche principalement sur des préoccupations théoriques telles que la conception de la meilleure stratégie pour évaluer l'informativité et choisir une instance à annoter par rapport à une autre [174, 199, 270], négligeant les aspects stratégiques et ceux de l'humain dans la boucle pourtant présents dans ce type de solution [159]. Cependant, des pistes de recherche intéressantes se trouvent également dans les défis pratiques de l'AA. Par exemple, un obstacle majeur dans la mise en œuvre de l'AA en pratique est le fait qu'il est difficile d'anticiper l'efficacité d'une stratégie donnée sur un ensemble de données inconnu en amont de son utilisation, la performance de la stratégie étant liée à l'ensemble de données [243]. De plus, notre mise en oeuvre de l'AA détaillée dans la Section 5.1.2 a révélé que, lorsque l'AA est utilisé en conjonction avec des transformers, le temps entre deux cycles d'AA peut devenir relativement élevé. En

fonction des capacités matérielles et de la taille des ensembles de données non étiquetées, ce temps d'attente peut réduire ou même annuler les avantages de la mise en œuvre de l'AA. Nous sommes alors confrontés à un dilemme : *soit augmenter les coûts matériels pour réduire ce temps d'attente, soit payer le temps d'attente des annotateurs*. En AA, nous ciblons des scénarios où les ressources sont limitées, où un accès limité aux annotateurs peut être lié à des contraintes de ressources informatiques. Le coût de l'apprentissage automatique réside aujourd'hui à la fois dans l'annotation, et dans les ressources informatiques nécessaires à l'entraînement du modèle, rendant l'idée de réduire le temps d'entraînement/inférence par le biais d'ordinateurs plus puissants impraticable. Notre approche contourne ce dilemme en mettant en œuvre une méthode alternative pour annoter en parallèle à l'AA afin de maintenir l'engagement des annotateurs pendant que les scores d'AA sont calculés, évitant ainsi les temps d'attente et la nécessité de plus de puissance de calcul.

Nous présentons une nouvelle approche qui peut non seulement être utilisée en conjonction avec l'AA et élimine efficacement les temps d'attente des annotateurs, mais améliore également les performances du modèle entraîné. De plus, elle réduit l'importance du choix de la stratégie car les performances du modèle sont proches, indépendamment de la stratégie d'AA choisie. Enfin, elle aborde également un autre défi inhérent aux tâches de classification multi-labels : la distribution inégale des labels.

5.1.1 *Travaux connexes*

La classification multi-label pose un ensemble unique de défis dans le domaine de l'AA en raison du déséquilibre inhérent souvent observé au sein des ensembles de données multi-labels [222]. Le problème de déséquilibre devient double : non seulement certains labels sont plus fréquents que d'autres dans l'ensemble de données, mais la distribution des instances positives et négatives pour chaque label peut également varier de manière significative [16]. Cela signifie que l'AA peut involontairement biaiser le modèle vers des labels fréquents ou aggraver le problème de déséquilibre en sélectionnant principalement une classe d'instances [13]. Des travaux inspirants peuvent être trouvés dans le domaine de la classification multi-classe associée à l'AA, où, si des seuils de déséquilibre sont atteints pendant le cycle standard d'AA, celui-ci est mis en pause et des étapes de rééquilibrage sont exécutées [3]. Les mêmes auteurs ont également exploré la résolution de ce déséquilibre directement tout au long du cycle d'AA en ne sélectionnant que les scores d'instances des classes minoritaires [4].

Les ressources d'annotation peuvent être budgétisées sur une base par annotation ou par heure, et dans ce dernier cas, il devient impé-

ratif de garantir un approvisionnement continu en instances à annoter sans causer de temps d'attente inutile [159]. Cette motivation a d'abord conduit au développement de l'AA en mode batch [202], qui fournit des instances par lots, réduisant la fréquence des mises à jour du modèle. La sélection d'instances d'AA par lots est bien adaptée à la manière dont les transformers sont entraînés. Cependant, CITOVSKY et al. [39] ont démontré que l'annotation d'instances redondantes est un risque de l'AA en mode batch. De plus, même en mode batch, il reste un temps d'attente pour l'annotateur pendant la mise à jour du modèle entre les lots annotés.

Dans les scénarios contemporains où les modèles et les ensembles de données non étiquetés deviennent plus volumineux, le temps d'attente n'est plus principalement induit par les mises à jour du modèle mais par l'inférence du modèle sur l'ensemble étendu non étiqueté [270]. Pour atténuer ce temps d'attente induit, TSVIGUN et al. [233] se concentrent sur le sous-échantillonnage, où seule une partie de l'ensemble non étiqueté est inférée à chaque cycle d'AA. Le travail de ASHRAFI ASLI et al. [12] explore les stratégies d'AA avec des fonctionnalités précalculées pour le pré-clustering des instances. Bien que prometteur, nous avons l'intention d'explorer davantage ces approches dans le but de non seulement atténuer, mais d'éliminer complètement les temps d'attente pour les annotateurs.

Étonnamment, très peu de recherches [270] ont été menées pour déterminer quelles méthodes d'échantillonnage peuvent coexister harmonieusement avec l'AA pour maximiser l'ensemble du temps d'annotation, plutôt que seulement la portion dédiée à l'AA. Par conséquent, nous explorons plusieurs méthodes alternatives conjointement à différentes stratégies d'AA et montrons que notre flux de travail réaliste réduit certains des biais de l'AA mis en évidence dans des études précédentes, tels que l'échantillonnage d'instances redondantes [39]. HAERTEL et al. [84] abordent ce problème en introduisant un processus parallèle à l'AA qui propose des instances à annoter en fonction de leur informativité des cycles d'AA précédents, au prix d'une précision réduite.

Cette dernière approche est particulièrement prometteuse, car elle introduit le fait d'utiliser une seconde source de sélection d'instances à annoter en complément du processus d'AA, offrant ainsi la possibilité d'éliminer le temps d'attente perçu par l'annotateur. Pendant les périodes d'attente entre les cycles d'AA, l'annotateur peut se consacrer à l'annotation des instances sélectionnées par cette source alternative. Nous nous inspirons donc de ces travaux ainsi que des travaux sur le déséquilibre dans la tâche multi-labels en se concentrant sur un objectif double : réduire le temps d'attente des annotateurs en fournissant une alternative à la sélection d'instances par AA, tout en contribuant à équilibrer le jeu d'instances multi-label sélectionné.

5.1.2 *Mise en oeuvre d'un outil d'annotation en entreprise*

Nous avons implémenté un outil d'annotation dans une entreprise en informatique spécialisée dans le traitement des textes. Cette mise en pratique, outre l'intérêt direct d'accéder à des données annotées de qualités, a permis de nous confronter aux défis pratiques de l'AA en condition industrielle et notamment au temps d'attente des cycles d'AA.

Notre objectif lors de la réalisation de ce projet d'annotation était de mieux comprendre et d'évaluer l'applicabilité des techniques d'AA dans des environnements industriels. En nous concentrant sur l'intégration de stratégies d'AA dans un cadre industriel au sein d'un projet d'annotation, nous cherchions à obtenir une vision plus concrète et pratique de ces approches. Notre démarche visait à identifier les défis spécifiques rencontrés lors de l'utilisation de l'AA dans des scénarios réels, ainsi que les opportunités et les limites de son intégration.

Ce projet constitue une étude préliminaire pour notre recherche, et ses conclusions ont influencé la recherche menée dans ce chapitre de la thèse.

A des fins d'annotations, nous avions accès à environ 1h30 du temps de 8 annotateurs différents ainsi que plusieurs dizaines de milliers d'instances. Les instances textuelles à annoter provenaient de 4 sources de données différentes, chacune comprenant des données en anglais ou en français. La tâche à réaliser était proche de notre tâche multi-labels puisqu'il s'agissait de réaliser une analyse de sentiment à 5 niveaux. Les instances étaient typiques d'un avis posté sur internet sur un service ou un lieu et l'objectif de classification était multi-classes : -2 pour avis très négatif, -1 pour négatif, 0 pour neutre, 1 pour positif, 2 pour très positif. A des fins de calcul d'accord inter-annotateur, nous avons organisé les annotateurs par paires afin qu'ils annotent des données issues de la même source de donnée.

Nous avons choisi d'utiliser Argilla.io [1] pour les annotations en raison de la qualité exceptionnelle de son interface utilisateur, comme illustré dans l'image 19. La facilité de déploiement de l'outil à travers l'utilisation conjointe de Google Colaboratory [2] et Hugging Face Space [3], nous a permis de créer des interfaces en ligne facilement administrables. L'une des contraintes était que l'outil était très récent et ne disposait pas d'une intégration native d'AA. Nous avons donc intégré cette fonctionnalité nous-mêmes.

Suivant les conseils d'un guide pratique de l'apprentissage actif [159], nous avons d'abord organisé une séance de calibrage, c'est-à-dire une séance d'annotation collective avec les huit annotateurs afin d'éclaircir les doutes existants sur le processus d'annotation. Du-

1. https ://docs.argilla.io/
2. https ://colab.google/
3. https ://huggingface.co/spaces

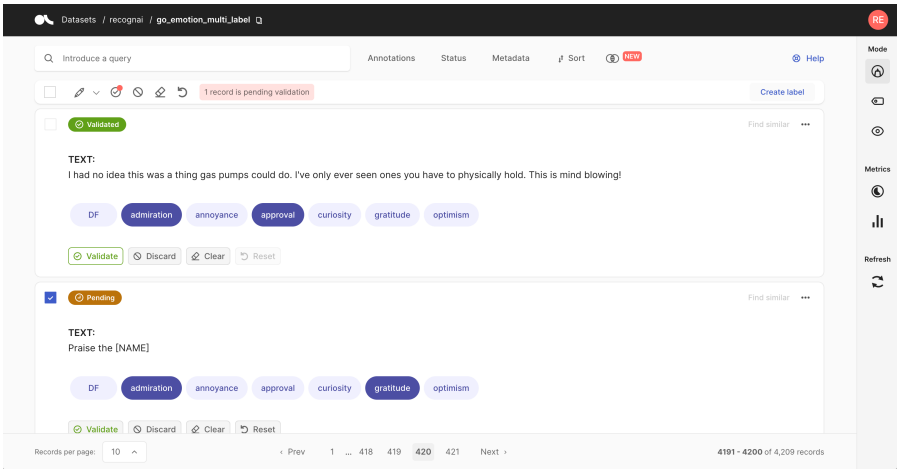


FIGURE 19 – Exemple d'interface utilisateur d'annotation avec Argilla.io.

rant cette séance nous avons expliqué le fonctionnement de l'outil et montré comment effectuer les distinctions entre chaque classe à l'aide d'exemples concrets. En expliquant ainsi la tâche, nous cherchions à réduire la subjectivité des annotateurs dans la classification entre les catégories "très négative"/"négative" et "positif"/"très positif".

Cette première séance a été suivie par deux séances de travail par paires de 45 minutes. Les paires d'annotateurs étaient en communication, ce qui leur permettaient de débattre des instances difficiles à classifier. Cette approche collaborative a contribué à renforcer la cohérence dans l'annotation en favorisant la discussion et la clarification des critères d'évaluation entre les annotateurs.

Le principal défi auquel nous avons été confrontés était l'ajout de temps d'attente entre chaque cycle d'AA. Initialement, sans AA, dans l'outil d'annotation, l'entièreté du jeu de données à annoter était ajouté, ce qui rendait plusieurs pages d'annotations disponibles. L'annotateur pouvait alors annoter des instances textuelles jusqu'à ce que son temps d'annotation soit écoulé. Cependant, l'ajout de l'AA ajouta la contrainte de ne proposer qu'un lot de données pour annotation à la fois, se traduisant par la disponibilité d'une seule page d'instances textuelles à annoter. Il était donc important de s'assurer que lorsqu'un annotateur arrivait à la fin de sa page d'instances à annoter, une nouvelle page soit immédiatement disponible.

Le processus est orchestré de la manière suivante : un premier lot d'instances est sélectionné de manière aléatoire parmi les instances à annoter. Ce premier lot est identique pour la paire d'annotateurs. Les annotations obtenues sont ensuite fournies à notre "brique applicative" d'AA, où notre modèle se met à jour pour proposer une nouvelle page d'instances à annoter. Pendant l'exécution de ce processus, une page de "temporisation" est également fournie à l'annotateur, avec des instances sélectionnées de manière aléatoire. À la fin de la page de temporisation, la page d'instances issue de l'AA devient

disponible. Les instances annotées dans la page de temporisation et dans la page issue de l'AA sont ensuite renvoyées à la brique d'AA, et en attendant son retour, une nouvelle page de temporisation est proposée. Ce processus se répète jusqu'à ce que l'annotateur épuise son temps d'annotation. En fin de session d'annotation, les annotations produites sont manuellement extraites et sauvegardées par l'administrateur du projet d'annotation.

Un autre changement induit par l'inclusion de l'AA est la divergence entre les données annotées par les deux membres de chaque paire d'annotateurs. En raison de la nature de l'AA, chaque annotateur est engagé dans un processus interactif d'AA distinct, ce qui signifie que les accords entre utilisateurs ne peuvent être établis que sur les données annotées en commun par chaque annotateur de la paire.

Du point de vue de l'annotateur, le processus se déroulait comme suit : tout d'abord, une étape d'identification permettait à l'annotateur de rejoindre sa session utilisateur. Une session utilisateur est composée d'un utilisateur et de ses projets d'annotation. Dans notre cas, un projet d'annotation est constitué des instances déjà annotées et des instances restant à annoter. L'annotateur devait ensuite cliquer sur son projet d'annotation, puis sélectionner comme paramètres de n'afficher que les instances non annotées et que chaque page contienne un maximum de 50 instances. Dans ce mode, une seule page d'instances restait à annoter.

Ensuite, l'annotateur pouvait commencer à annoter. Une fois arrivé à la fin de sa page, il n'avait qu'à actualiser la page, et une nouvelle série d'instances à annoter lui était immédiatement proposée. Du point de vue de l'annotateur, il n'y avait aucun indice du processus sous-jacent derrière la proposition des instances, et rien n'indiquait si les instances qui lui étaient proposées étaient sélectionnées de manière aléatoire ou par l'AA.

En raison du caractère multilingue de la classification nous avions choisi comme modèle le transformers distillé *LEALLA* [149]. Nous avions choisi comme stratégie d'AA multi-classes "Breaking Ties" [225] en nous basant sur les conclusions des travaux de [195]. Après des tests réalisés en amont nous étions arrêtés sur des lots de 50 instances de données. En effet, les calculs d'AA étaient réalisés sur plus de 5000 instances par des cartes graphiques V100 en moins de 3 minutes pour le premier cycle et moins de 2 minutes pour les suivants. Au vu de la longueur moyenne des textes annotés, notre annotateur le plus rapide arrivait à annoter de 20 à 25 instances en 3 minutes. Nous avons donc doublé ce chiffre afin d'être certain qu'un annotateur n'arrive pas à la fin de sa page d'annotation avant que nous ayons eu le temps de lui proposer la prochaine.

En parallèle, nous avons également évalué le coût matériel de nos séances d'annotation. Nous avons utilisé 8 sessions de 1h30 de temps

V100 sur Google Colab, ce qui équivaut à environ 8 fois 9 unités de calcul, soit 71 unités de calcul. En arrondissant, nous pouvons considérer cela comme 75 unités de calcul, en tenant compte des quelques tests réalisés en amont du déploiement de la solution. Sachant que 500 unités de calcul correspondent à 50 euros, nos 75 unités de calcul représentent donc 7,50 euros.

En considérant le fait que nos annotateurs ont chacun produit entre 300 et 700 annotations pendant cette période, pour un total d'environ 4000 annotations, nous obtenons un coût matériel d'environ 0,002 euros par instance annotée.

5.1.3 *Retour d'expérience : mise en pratique de l'AA*

Les retours des annotateurs concernant l'interface utilisateur ont été extrêmement positifs. La fluidité et la qualité graphique de l'outil ont été particulièrement appréciées. La présentation des textes, leur longueur et leur niveau de difficulté ont également été jugés satisfaisants. Certains annotateurs ont soulevé la question de l'ambiguïté de la tâche de classification, surtout lorsque la même instance comportait à la fois une partie positive et une partie négative. Étant donné que la tâche d'annotation s'est déroulée au cours de deux séances distinctes, les annotateurs ont également noté que, en se rapprochant de la fin de l'heure d'annotation, la tâche devenait parfois rébarbative et plus complexe à accomplir correctement. La durée relativement "courte" des séances, d'environ 45 minutes, a donc été appréciée.

En ce qui concerne la qualité des données annotées, elle a contribué de manière significative à l'amélioration des performances d'un classifieur destiné à un client de l'entreprise.

L'accord inter-utilisateur a permis, quant à lui, de déterminer que les utilisateurs étaient globalement d'accord sur leurs annotations ce qui n'est guère étonnant au vu de la méthodologie employée : séance de calibrage en amont des séances d'annotations et possibilité de débattre avec l'autre annotateur lorsqu'il y a une confrontation à des difficultés de classification d'une instance.

L'accord inter-utilisateur a permis de constater que les annotateurs étaient généralement en accord sur leurs annotations, ce qui n'est pas surprenant compte tenu de la méthodologie employée. En effet, une séance de calibrage a été réalisée en amont des séances d'annotations, et les annotateurs avaient la possibilité de débattre avec l'autre annotateur en cas de difficultés de classification d'une instance.

En ce qui concerne le recoupement des instances annotées entre chaque paire d'annotateurs, on observe que sur la totalité des instances annotées, environ un tiers (le chiffre variant selon les sources de données) ont été annotées deux fois. Puisque l'AA a pour objectif de déterminer quelles instances sont les plus intéressantes à annoter, il n'est pas étonnant qu'au cours de deux processus d'AA distinct les

mêmes instances soient souvent sélectionnées. Cette double annotation contribue à renforcer la fiabilité des annotations en fournissant une mesure de consensus entre les annotateurs.

En tant que réalisateur et administrateur de ce projet, un premier point à retenir est que la vitesse d'annotation d'un annotateur à un autre pouvait grandement varier. En effet, avec seulement 8 annotateurs, entre notre annotateur le plus lent et le plus rapide il y avait presque un facteur x2.5. Cette différence de vitesse d'annotation est due à la rapidité de lecture et aussi à la rapidité de la prise de décision qui devient un facteur important à prendre en compte lorsqu'on utilise de l'AA dans le calcul du temps maximal d'inactivité des annotateurs à palier.

Un point essentiel que nous avons tiré de la mise en pratique de notre recherche est que dans un projet d'annotation avec AA concret, impliquant des ressources réalistes, le temps d'attente d'un annotateur – c'est-à-dire le temps passé hors de l'annotation d'instances sélectionnées par AA – peut représenter jusqu'à la moitié du temps total de l'annotateur. Cette constatation est d'autant plus significative compte tenu du fait, confirmée par cette expérience, que l'accès aux annotateurs est limité et que leur temps est précieux. "Perdre" ainsi la moitié du temps d'accès aux annotateurs n'est pas viable d'un point de vue de la rentabilité de la mise en place de l'AA. En effet, la mise en œuvre de l'AA entraîne des coûts de développement, et si l'on ajoute le fait que seulement la moitié du temps des annotateurs est effectivement consacrée à l'annotation, l'AA ne contribue plus de manière significative, mais coûte davantage.

Durant notre mise en oeuvre de l'AA, nous avons partiellement pallié à cette situation en fournissant des instances sélectionnées aléatoirement pendant les temps d'attente. Cependant, nous avons trouvé cette situation frustrante et contre-productive, notamment en ce qui concerne la perception de l'AA. En effet, constater qu'au final, moins de la moitié des instances annotées ont été sélectionnées de manière intelligente peut entacher l'image et l'intérêt perçue de l'apprentissage actif. Par conséquent, nous nous sommes intéressés par la suite à la manière de valoriser le temps d'attente de l'annotateur pour sélectionner intelligemment des instances, même si ce n'est pas par l'utilisation des techniques basées sur l'AA.

Le critère de réussite du classifieur, s'entraînant sur les données annotées, reposait non seulement sur le niveau de performance, mais aussi sur le caractère équilibré de ses performances par classe, y compris pour les classes les plus rares. C'est pourquoi nous avons choisi d'explorer, comment pendant ce temps d'attente, sélectionner des instances qui permettraient de réduire les écarts de performances entre les différentes classes, pour mieux s'ajuster aux problématiques de l'AA pour les tâches industrielles.

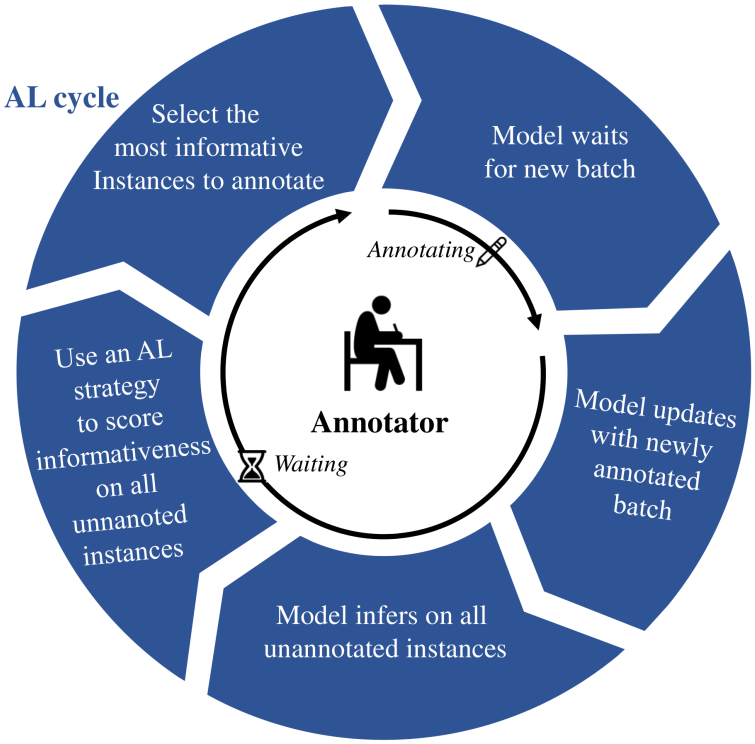


FIGURE 20 – Cycle classique d'AA sans méthode d'échantillonnage parallèle pour éviter le temps d'attente de l'annotateur. Cette figure met en évidence que lorsque le modèle travaille l'annotateur attend et lorsque le modèle attend l'annotateur travaille. La longueur des flèches "Annotating" et "Waiting" ne sont pas proportionnelles avec les durées réelles de ces actions.

5.2 DÉFINITION DU PROBLÈME

En AA, nous ciblons des scénarios où les ressources sont limitées, et un accès restreint aux annotateurs est souvent lié à des contraintes de ressources informatiques. Avec les modèles transformers, les temps d'attente inhérents proviennent des prédictions du modèle sur l'ensemble de données non annotées, évoluant avec la taille de l'ensemble de données. En comparaison, les calculs AA basés sur l'incertitude et les mises à jour du modèle sont relativement rapides.

En effet, à partir de nos expériences pratiques menées avec une variante simplifiée de BERT [188], un ensemble de données de 100 000 instances de texte non annotées, et une Nvidia V100 (32 Go), nous avons remarqué que la durée moyenne entre deux cycles d'apprentissage actif est d'environ 5 minutes. Cette durée comprend environ 25 secondes pour les mises à jour du modèle et environ 15 secondes pour les calculs d'apprentissage actif, pouvant varier d'environ 10 secondes en fonction de la stratégie choisie.

Bien que nous puissions réduire les temps d'attente en utilisant une puissance de calcul accrue, il n'est pas réaliste de les éliminer complè-

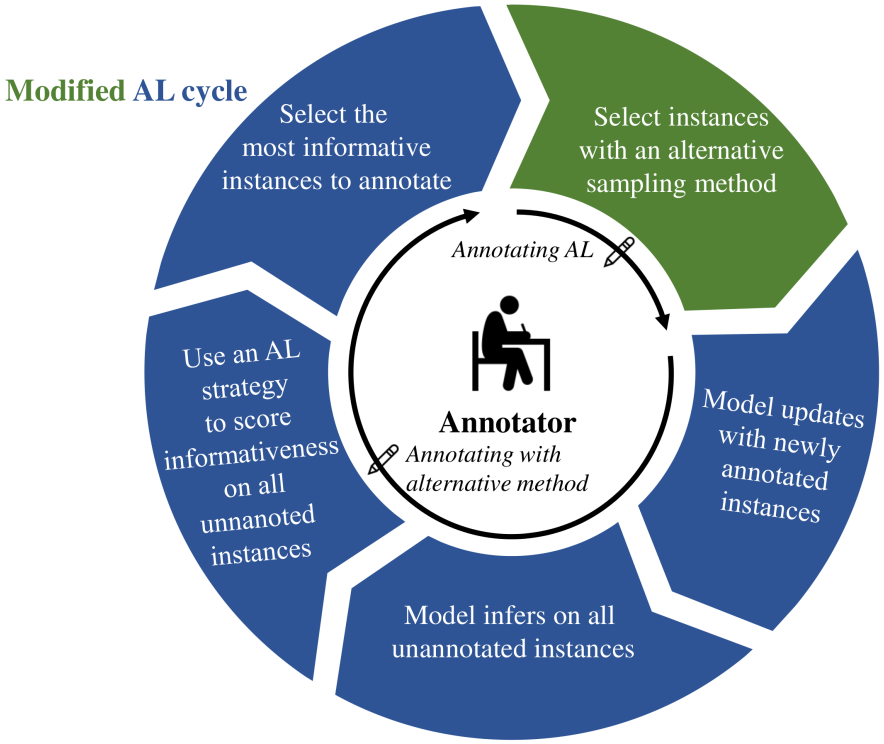


FIGURE 21 – Cycle d'AA avec une méthode d'échantillonnage alternative pour éviter le temps d'attente de l'annotateur. Nous précisons que la Figure n'est pas à l'échelle du temps.

tement de cette manière. Ainsi, dans le cycle AA régulier (illustré dans la Figure 20), nous supposons que les annotateurs devront toujours attendre une période significative.

Pour optimiser ce temps d'attente et le budget d'annotation, nous suggérons d'utiliser des méthodes de sélection des données à annoter qui ne nécessitent pas un modèle à jour incorporant les dernières données annotées. En utilisant des méthodes d'annotation alternatives, nous obtenons ainsi un cycle AA modifié (illustré dans la Figure 21), où les annotateurs disposent constamment de données à annoter, éliminant ainsi le temps d'attente.

5.3 CONTEXTE EXPÉRIMENTAL

Pour garantir la continuité de nos expérimentations et maintenir une comparabilité avec celles que nous avons précédemment menées, nous conservons les mêmes jeux de données, stratégies d'apprentissage actif, détails d'implémentation, modèles et métrique de performance que ceux exposés dans la Section 4.2.

5.3.1 *Mise en oeuvre*

Nous mettons en place le paramètre alpha (α) qui est égal à la proportion des lots d'entraînements qui provient d'une méthode d'échantillonage alternative. Diverses valeurs d'alpha sont explorées dans nos expériences (0.25, 0.5 et 0.75) comme on peut le voir dans la Section 5.4.2.1. Cependant, d'après nos investigations empiriques, si l'objectif est des mises à jour régulières du modèle et, par conséquent, un apprentissage actif plus précis, seulement environ la moitié du temps d'annotation peut être allouée à l'apprentissage actif. Dans notre travail, en plus du lot initial d'entraînement (une initialisation aléatoire initiale du modèle), nous fixons donc $\alpha = 0.5$ pour nos expériences et tous les lots de données utilisés pour l'entraînement du modèle sont ainsi composés de manière égale de données provenant de notre méthode alternative et de données provenant de l'apprentissage actif.

Dans nos expériences, l'apprentissage actif multi-labels se déroule selon le processus suivant : d'abord, nos modèles sont initialisés en les entraînant avec 25 instances sélectionnées de manière aléatoire. Suivant [193], nous réalisons ensuite 50 itérations d'apprentissage actif où chaque lot est composé de 25 instances, soit un total de 1250 instances annotées collectées. Pour atteindre notre objectif de $\alpha = 0.5$, à chaque itération, le lot est composé de 12 instances sélectionnées par une méthode alternative et de 13 instances choisies par une stratégie d'apprentissage actif. Après chaque itération, nous entraînons davantage le modèle avec le lot d'instances nouvellement annotées.

5.3.1.1 *Référence*

À des fins de comparaison, nous utiliserons les résultats issus du cadre idéaliste fréquemment utilisé dans les travaux sur l'apprentissage actif, où le temps de mise à jour de l'apprentissage actif et le temps d'inférence du modèle sont considérés comme négligeables. Ce scénario (avec $\alpha = 0$), appelé **Classic-AL**, entraîne en pratique une perte de temps d'annotation (avec un impact négatif sur les coûts de l'annotation) et ne répond pas aux besoins de notre problématique. Cependant, il sera comparé aux résultats de nos trois paramétrages où nous avons pris en compte les temps d'attente en ajoutant différentes méthodes d'échantillonnage alternatives.

5.3.1.2 *Evaluation*

Pour mesurer le déséquilibre des labels, nous utilisons le Ratio Moyen de Déséquilibre (MeanIR) [222] (une valeur plus élevée indique un déséquilibre plus élevé dans la distribution des labels).

$$MeanIR = \frac{1}{q} \sum_{j=1}^{q} IRLbl(l^j) \tag{17}$$

Avec IRLbl représentant le Ratio de Déséquilibre par Label, tel que :

$$\mathrm{IRLbl}(l^j) = \frac{\max_{\lambda \in l} \sum_{i=1}^{n} h(\lambda, y_i)}{\sum_{i=1}^{n} h(l^j, y_i)} \tag{18}$$

, avec h une variable telle que :

$$h(\lambda, y_i) = \begin{cases} 1 \text{ si } \lambda \in y_i \\ 0 \text{ sinon} \end{cases} \tag{19}$$

5.4 OPTIMISATION DU TEMPS D'ATTENTE DE L'ANNOTATEUR

L'intégration d'une source d'échantillonnage en parallèle de l'AA vient compléter le processus de manière significative. Elle permet notamment de pallier l'une des faiblesses inhérentes à l'AA, à savoir le temps d'attente présent à chaque cycle d'AA, composé du temps de mise à jour du modèle, du temps de calcul des stratégies, et surtout du temps d'inférence du modèle sur l'ensemble des données annotées.

Plusieurs approches peuvent être envisagées pour sélectionner des données à annoter, de manière plus ou moins complémentaire avec l'AA. Le développement d'une méthode d'échantillonnage alternative, conçue pour compléter les stratégies d'AA, offre la possibilité de compenser d'autres problématiques liées à l'AA, notamment celles associées à l'incertitude. Une telle approche peut permettre de réduire les problématiques liées à la redondance des instances, tout en préservant la possibilité de comparer des instances sémantiquement similaires. Cette approche peut aussi remplir l'objectif d'assurer une amélioration, peu importe la stratégie d'AA utilisée en conjonction avec la source d'échantillonnage alternatif, comparativement à un échantillonnage purement aléatoire. Enfin, cette approche pourrait même renforcer les performances finales atteintes par les modèles après l'entraînement, accentuant ainsi l'intérêt de mettre en place une sélection intelligente des données.

5.4.1 *Lot d'entraînement hybride : apprentissage actif et échantillonage alternatif*

Ces méthodes alternatives doivent rapidement sélectionner des instances de données pour annotation et les fournir à l'annotation avant que les annotateurs n'aient terminé l'annotation de l'ensemble fournie par l'AA. Dans notre cadre expérimental, nous supposons que les annotateurs travaillent à une vitesse constante et que le temps d'attente entre les cycles d'AA est constant.

Les données annotées par des méthodes alternatives sont intégrées dans les cycles d'AA. À chaque mise à jour d'AA, il y a une proportion α d'instances composant le lot d'entraînement qui provient de la méthode d'échantillonnage alternative et une proportion $1 - \alpha$ qui a été sélectionnée selon une stratégie d'AA. La Figure 22 permet de schématiser comment la composition "hybride" des lots d'entraînements se réalise au fil du temps d'annotation.

Lorsque $\alpha = 0$, cela représente le scénario idéal et irréaliste dépeint dans la plupart des travaux sur l'apprentissage actif, où le temps d'attente n'est pas pris en compte, et toutes les données annotées proviennent de la sélection de la stratégie d'apprentissage actif. Lorsque $\alpha = 1$, cela représente un cas où nous ne réalisons pas d'apprentissage actif et s'écarte ainsi de notre cas d'utilisation.

Ainsi, nous évaluons deux méthodes alternatives d'échantillonage, à savoir, **Aléatoire** et **Stale** [84], tout en introduisant notre méthode **Eq_label**, conçue pour équilibrer les labels annotées afin d'améliorer les performances.

5.4.1.1 *Apprentissage actif et annotation aléatoire*

L'échantillonnage aléatoire (**Aléatoire**) est une référence fréquemment utilisée en apprentissage actif, où les instances nécessitant une annotation sont choisies au hasard parmi l'ensemble des données non étiquetées. Dans notre cadre, nous pouvons facilement appliquer cette méthode de sélection, en choisissant les données à annoter de manière aléatoire pendant que l'annotateur attend entre les cycles d'apprentissage actif. Cette approche vaut la peine d'être étudiée car des travaux récents ont démontré que l'échantillonnage aléatoire est une référence robuste lorsque l'apprentissage actif est appliqué aux modèles transformers [44].

En effet, certaines stratégies d'apprentissage actif présentent des biais de sélection, ce qui entraîne une absence de diversité parmi les données choisies. Cela peut conduire à des redondances et même à un biais du modèle parmi les données sélectionnées, malgré l'accent inhérent de l'apprentissage actif sur la sélection des données les plus informatives pour l'annotation. En utilisant l'échantillonnage aléatoire, nous choisissons délibérément des données variées et dissimilaires, fournissant une contre-mesure contre la redondance et le biais potentiel. De plus, cette méthode entraîne un surcoût computationnel minimal.

5.4.1.2 *Apprentissage actif et scores d'incertitudes périmés*

Une approche alternative pour sélectionner des données à annoter consiste à utiliser les scores d'AA du cycle précédent (**Stale**), utilisant essentiellement des scores d'informativité "périmés" [84]. Par conséquent, un lot d'entraînement est composé de deux moitiés : l'une

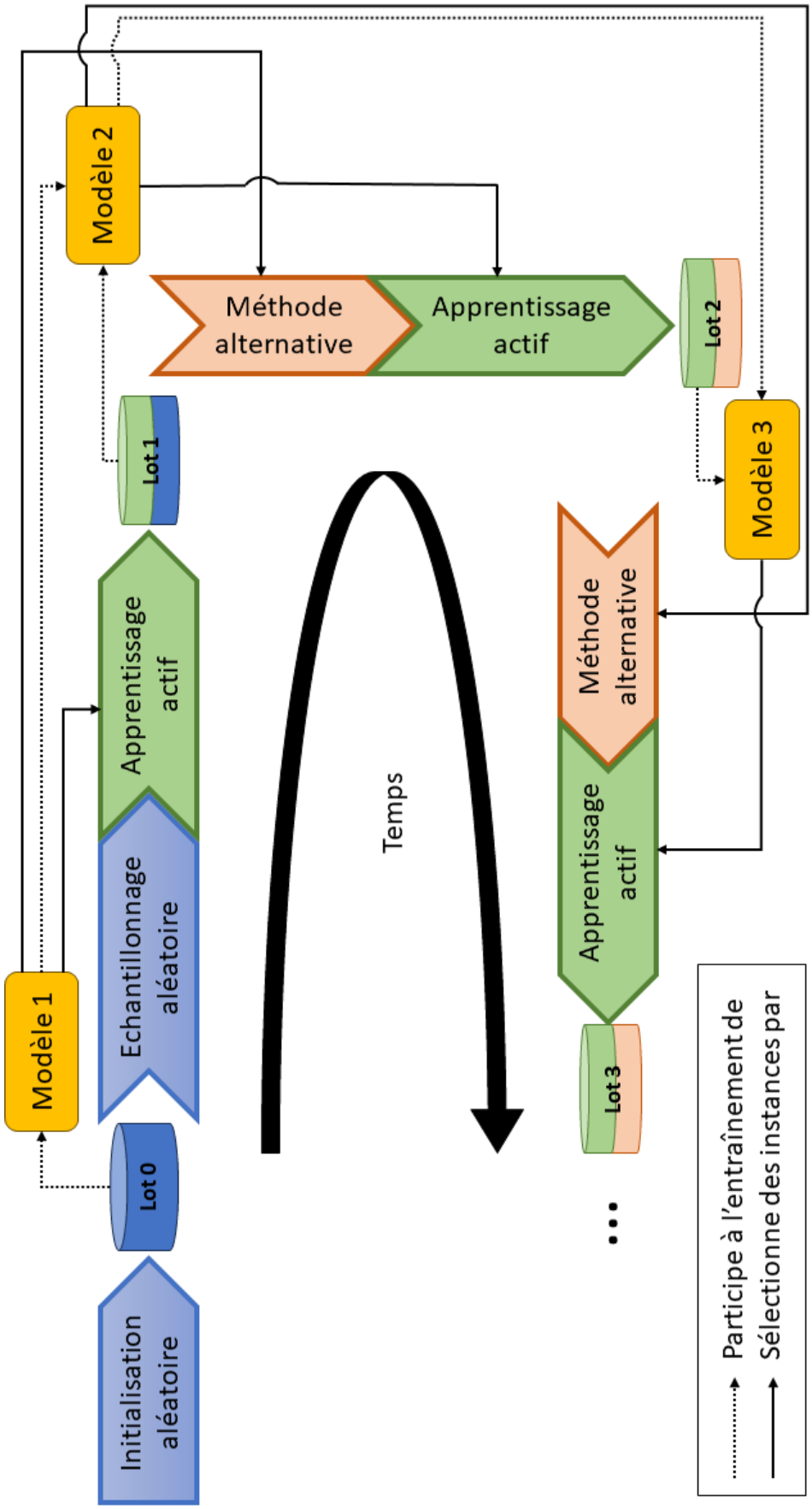


FIGURE 22 – Composition du lot d'entraînement à travers les cycles d'AA avec une méthode d'échantillonnage alternative pour éviter le temps d'attente de l'annotateur.

dérivée des calculs d'AA basés sur les prédictions du modèle avant la mise à jour, et l'autre dérivée de la sélection d'AA suivant la mise à jour du modèle. Cela implique de sélectionner des données moins bien classées par rapport à leurs scores d'informativité. De plus, cette méthode alternative, appelée "Stale", comporte également le risque d'amplifier les lacunes de certaines stratégies d'AA, exacerbant éventuellement leurs biais associés tels que la redondance d'information ou le déséquilibre des labels dans les lots sélectionnés.

5.4.1.3 *Apprentissage actif et équilibrage des labels annotés*

Dans cette méthode alternative de sélection (**Eq_label**) des données à annoter, nous utilisons des scores obsolètes d'informativité, plus précisément des scores d'incertitude. Cependant, contrairement à la méthode précédente (vue dans la Section 5.4.1.2), nous ne sélectionnons pas les données sur lesquelles le modèle est le plus incertain ; au lieu de cela, nous choisissons des données sur lesquelles le modèle est le plus certain. La prémisse sous-jacente à cette idée découle de l'observation qu'entre chaque mise à jour du modèle pendant les cycles d'apprentissage actif, les instances sur lesquelles le modèle est le plus incertain changent significativement, tandis que les instances sur lesquelles le modèle est confiant dans ses prédictions restent relativement constantes. En d'autres termes, la péremption de l'incertitude est bien plus importante que celle de la certitude.

En pondérant les scores d'incertitude avec la probabilité que l'instance soit étiquetée avec un label rare (détaillée dans la Section 5.4.1.3), nous obtenons une annotation plus uniforme sur différents labels. La réduction de la disparité des labels améliore non seulement les performances globales, mais peut également être un critère souhaitable dans les projets visant des performances égales sur tous les labels.

Pour atténuer le déséquilibre des labels, nous nous appuyons sur les probabilités de présence des labels, et inhéremment, les instances sur lesquelles le modèle est incertain sont celles où ces probabilités sont les moins informatives. De plus, il est raisonnable de supposer que des lots contenant à la fois des instances certaines et incertaines encourageront naturellement la diversité, car elles sont probablement différentes, améliorant ainsi la précision de l'apprentissage.

Nous souhaitons obtenir un score d'intérêt pour une instance afin de déterminer quelles instances sont susceptibles d'avoir les labels les plus intéressants. Une instance est considérée comme intéressante si son étiquetage contribue à rééquilibrer les labels que nous avons annotés, ou en d'autres termes, si cette instance est étiquetée avec des labels rares.

Du point de vue de la notation, nous désignons nos instances de texte par $x_1, ..., x_n$ et notre espace de labels par $l = l^1, ..., l^q$. Pour une instance donnée x_i, nous représentons sa distribution de labels prédits de manière similaire à une probabilité par $\hat{y_i} = [\hat{y_i^1}, ..., \hat{y_i^q}], \hat{y_i^j} \in$

$[0, 1]$, où plus $\hat{y}_i^j$ est proche de 1, plus le modèle est confiant que x_i est labellisé comme l^j, et où plus $\hat{y}_i^j$ est proche de 0, plus le modèle est confiant que x_i n'est pas labellisé comme l^j.

Notre méthode **Eq_label** correspond au produit de deux facteurs. Le premier facteur représente la confiance du modèle dans ses prédictions de labels pour une instance, et le deuxième facteur est un score associé aux labels que le modèle attribue à cette instance. Plus la rareté des labels est élevée, plus ce score est élevé.

Soit le premier facteur du produit final être le score de certitude c_i (inverse des scores d'incertitude calculés par la stratégie d'AA basée sur l'incertitude vue dans la Section 2.2.3) associé à l'instance x_i.

Le deuxième facteur du produit final est calculé à travers une série d'étapes de calcul. Soit $L(t) = [L(t)^1, \ldots, L(t)^q]$ avec $L(t)^j$ étant la somme des instances annotées positivement avec le label j au temps t.

Et soit $\omega(t) = [\omega(t)^1, \ldots, \omega(t)^q]$ le vecteur de pondération intermédiaire, où $\omega(t)^j = (\max_j(L(t)) + L(t)^j)/(2 \cdot L(t)^j)$ avec $L(t)^j \neq 0$.

Les cas où $L(t)^j = 0$ sont calculés à une étape ultérieure pour être égaux à $2 \cdot \max(\omega_t^j)$. À ce stade, plus un label est rare, plus son $\omega(t)^j$ associé est élevé.

Pour stabiliser notre méthode et rendre la comparaison plus fiable entre différentes instances, nous normalisons les scores obtenus. Soit $\omega'(t) = [\omega'(t)^1, \ldots, \omega'(t)^q]$ le vecteur de pondération normalisé, avec $\omega'(t) = \mathit{softmax}(\omega(t))$.

Enfin, nous obtenons le deuxième facteur du produit final, le score d'équilibrage des labels $e(t)_i$ associé à l'instance x_i au temps t,comme suit : $e(t)_i = \sum_{j=1}^{q} \hat{y}_i^j \cdot \omega'(t)^j$.

Ce score d'équilibrage contribue au score d'intérêt de l'instance (les scores plus élevés sont sélectionnés pour être annotés), qui est donné par $score(t)_i = c_i \cdot e(t)_i$.

5.4.2 *Résultats et Analyses*

Dans notre étude, nous établissons un point de comparaison robuste dans des conditions classiques, où l'annotateur alloue tout le temps disponible pour annoter les instances sélectionnées par l'AA. Nous notons qu'un point de comparaison plus équitable, bien que plus faible, consisterait à évaluer les performances du modèle en utilisant les stratégies d'AA avec seulement la moitié des annotations. Cela tiendrait compte du fait qu'en l'absence d'une méthode alternative d'échantillonnage, la moitié du temps effectif des annotateurs est perdue.

Le tableau 6 présente les résultats détaillés de notre analyse expérimentale. Tout d'abord, nous observons que Random et Eq_label (notre méthode) surpassent Stale et le paramètre Classic-AL dans de

TABLE 6 – Résultats expérimentaux (M_{iF1}) du paramètre Classic-AL et de trois méthodes d'échantillonnage parallèle à l'apprentissage actif (AL). Avec 'dB' pour distilBERT, 'dR' pour distilRoBERTa et 'Std-dev' pour l'écart-type. Les meilleures valeurs pour le modèle/la stratégie sont en gras.

	Méthode →	Classic-AL		Random		Stale		Eq_label	
	Modèle →	dB	dR	dB	dR	dB	dR	dB	dR
Jeux de données ↓	**Stratégie ↓**								
Jigsaw_toxic	ML	0.412	0.486	**0.57**	**0.577**	0.529	0.522	0.567	0.548
	MML	0.467	0.524	**0.572**	**0.59**	0.51	0.582	0.547	0.581
	CMN	0.617	0.593	0.603	0.6	0.605	**0.619**	**0.631**	0.618
	MMU	0.603	0.614	**0.623**	0.609	0.609	0.613	0.61	**0.615**
	LCI	0.453	0.482	**0.556**	**0.587**	0.496	0.499	0.551	0.546
	CVIRS	0.564	0.575	0.551	0.58	**0.591**	**0.598**	0.537	0.565
	Std dev	0.086	0.056	**0.028**	**0.012**	0.051	0.05	0.038	0.031
Go_emotions	ML	0.404	0.414	0.411	**0.432**	0.386	0.383	**0.412**	0.415
	MML	0.414	0.428	0.404	0.421	0.412	0.427	**0.416**	**0.431**
	CMN	0.438	0.435	0.423	0.416	0.408	0.433	**0.448**	**0.443**
	MMU	0.441	0.435	0.419	0.405	0.418	0.413	**0.455**	**0.441**
	LCI	0.353	0.373	0.383	**0.402**	0.363	0.37	**0.422**	0.389
	CVIRS	0.362	0.4	0.408	**0.441**	**0.41**	0.395	0.405	0.408
	Std dev	0.037	0.024	**0.014**	**0.015**	0.021	0.025	0.02	0.021
EUR_Lex	ML	0.329	0.409	0.495	0.467	0.401	0.42	**0.502**	**0.531**
	MML	0.517	**0.56**	0.52	0.53	0.517	0.513	**0.543**	0.553
	CMN	0.513	0.55	0.464	0.526	0.511	0.522	**0.521**	**0.556**
	MMU	0.53	**0.567**	0.539	0.489	0.518	0.526	**0.545**	0.565
	LCI	0.454	0.488	**0.521**	0.5	0.47	0.478	0.515	**0.525**
	CVIRS	0.479	0.509	**0.565**	0.535	0.562	0.5	0.534	**0.537**
	Std dev	0.075	0.06	0.035	0.027	0.056	0.04	**0.016**	**0.026**
UNFAIR-ToS	ML	0.622	0.718	0.694	0.726	0.682	0.688	**0.703**	**0.757**
	MML	0.647	0.724	0.696	0.71	0.651	0.704	**0.717**	**0.758**
	CMN	0.708	**0.778**	0.756	0.777	0.749	0.689	**0.767**	0.758
	MMU	0.7	0.762	**0.741**	**0.773**	**0.741**	0.763	0.738	0.762
	LCI	0.65	0.744	0.732	**0.753**	0.619	0.713	**0.765**	0.738
	CVIRS	0.708	0.76	0.723	0.746	**0.746**	0.703	0.721	**0.765**
	Std dev	0.037	0.023	0.027	0.026	0.056	0.028	**0.026**	**0.009**

nombreux cas. En effet, dans 48 cas, Random surpasse Classic-AL 32 fois, Eq_label surpasse Classic-AL 41 fois, et Random surpasse Stale 34 fois, tandis qu'Eq_label surpasse Stale 39 fois. Ensuite, nous observons que les tendances de performance associées à Stale ne sont pas très distinctes de celles du paramètre Classic-AL. Ce résultat suggère que le déplacement d'un demi-lot de données annotées selon la méthode Stale, par rapport au paramètre Classic-AL, n'affecte pas de manière significative la progression de l'annotation selon une stratégie d'apprentissage actif spécifique. Selon les critères statistiques conventionnels, la différence entre Stale et le paramètre Classic-AL n'est pas considérée comme statistiquement significative (valeur de p > 0,05 avec un test t apparié par ensemble de données et modèle). Au contraire, Random et Eq_label s'écartent significativement du para-

mètre Classic-AL de manière extrêmement statistiquement significative (valeur de p < 0,001 avec un test t apparié par ensemble de données et modèle). Plus précisément, en réalisant des t-tests par paires d'ensemble de données et modèle, nous avons calculé des valeurs de p à deux queues de 0,0008 et 0,0001 pour Random et Eq_label, respectivement, en comparaison avec le paramètre Classic-AL. De plus, nous avons calculé une valeur de p à deux queues de 0,3248 pour Stale par rapport au paramètre Classic-AL.

TABLE 7 – Moyenne en pourcentage de la différence de M_{iF1} par rapport au paramètre Classic-AL par méthode, modèle et ensemble de données (une valeur positive indique une amélioration), 'dB' pour distilBERT et 'dR' pour distilRoBERTa. Les meilleures valeurs sont en gras.

	Jeux de données	Jigsaw_toxic		Go_emotions		EUR_Lex		UNFAIR-ToS	
	Modèle	dB	dR	dB	dR	dB	dR	dB	dR
Méthode	Random	**+11.521**	**+6.139**	+1.493	-1.489	+9.993	+0.681	+7.608	-0.022
	Stale	+7.189	+4.856	-0.622	-2.575	+5.563	-4.022	+3.792	-5.038
	Eq_label	+10.494	+6.078	**+6.053**	**+1.690**	**+11.977**	**+5.968**	**+9.318**	**+1.159**

Les améliorations globales de performance observées avec la méthode Random proviennent probablement de la diversification du lot, tandis que pour notre méthode Eq_label, ces améliorations résultent non seulement de la diversification du texte au sein du lot, mais aussi d'une représentation plus équilibrée des différentes labels.

De plus, il est à noter dans les lignes d'écart type que les méthodes Random et Eq_label présentent toutes deux une réduction des différences de performance entre les différentes stratégies d'apprentissage actif. Ce résultat est attendu, car maintenant seule la moitié des données annotées pendant les expériences proviennent de ces stratégies. Néanmoins, c'est un résultat intéressant, car l'un des défis majeurs lors de la mise en œuvre de l'apprentissage actif est de choisir une stratégie d'apprentissage actif à l'avance. Suivant notre cadre, ce choix devient moins critique, simplifiant la mise en œuvre pratique de l'apprentissage actif.

TABLE 8 – Moyenne en pourcentage de la différence de M_{iF1} par rapport au paramètre Classic-AL par méthode, modèle et stratégie d'apprentissage actif (une valeur positive indique une amélioration). Les meilleures valeurs sont en gras.

	Méthode	Random		Stale		Eq_label	
	Modèle	distilBERT	distilRoBERTa	distilBERT	distilRoBERTa	distilBERT	distilRoBERTa
Stratégie	ML	+22.807	+8.633	+13.073	-0.691	**+23.599**	**+11.051**
	MML	+7.188	+0.671	+2.200	-0.447	**+8.704**	**+3.891**
	CMN	-1.318	-1.570	-0.132	-3.947	**+3.998**	**+0.806**
	MMU	+2.111	-4.289	+0.528	-2.649	**+3.254**	**+0.210**
	LCI	+14.764	**+7.427**	+1.990	-1.294	**+17.958**	+5.319
	CVIRS	+6.342	**+2.585**	**+9.276**	-2.139	+3.975	+1.381

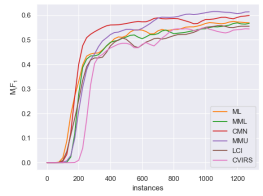
TABLE 9 – Moyenne du ratio moyen d'imbalance (MeanIR) par méthode et modèle (une valeur élevée indique un déséquilibre élevé des labels). Les meilleures valeurs sont en gras.

	Méthode	Classic-AL	Random	Stale	Eq_label
Jeux de données	**Modèle**				
Jigsaw_toxic	distilBERT	6.681	6.846	5.776	**4.886**
	distilRoBERTa	6.782	6.869	6.548	**5.874**
Go_emotions	distilBERT	12.945	13.035	15.065	**10.017**
	distilRoBERTa	14.238	12.719	15.111	**10.690**
EUR_Lex	distilBERT	54.560	37.422	53.409	**34.826**
	distilRoBERTa	37.745	34.667	38.288	**30.048**
UNFAIR-ToS	distilBERT	5.945	5.018	5.781	**2.795**
	distilRoBERTa	5.960	5.972	7.286	**2.894**

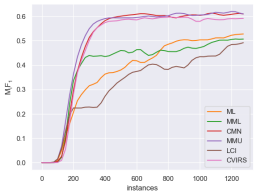
Les résultats de la méthode Random sont généralement bons, surtout lorsqu'on considère sa facilité de mise en œuvre et de compréhension. Random est donc une option solide à considérer lors de la mise en place d'une méthode d'échantillonnage alternative pour éliminer le temps d'attente. Cependant, cette méthode semble dégrader les performances dans certains cas, tels que la stratégie CMN ou l'ensemble de données Go_emotions. Comparés à Random et Eq_label, les résultats de la méthode Stale sont inférieurs, faisant de cette option la moins intéressante à mettre en œuvre pour éliminer les temps d'attente.

Les résultats de la méthode Eq_label sont généralement les plus prometteurs dans nos expériences, améliorant toutes les stratégies et tous les ensembles de données, souvent plus que Random. Il est intéressant de se pencher sur un autre aspect de cette méthode, qui est la distribution des labels (voir Table 9). Encore une fois, il n'est pas surprenant de n'observer aucune différence significative dans la distribution moyenne des labels entre le paramètre Classic-AL et Stale. Des expériences préliminaires ont montré que dans certains cas, l'apprentissage actif aggrave le déséquilibre des labels présents dans l'ensemble de données d'origine. Par conséquent, comme prévu, nous observons une amélioration significative entre Classic-AL/Stale et Random, où le MeanIR moyen pour ce dernier est notablement 10% plus bas pour distilRoBERTa et 20% plus bas pour distilBERT. Pour Eq_label, une distribution encore plus égalitaire des labels annotés peut être observée, car en moyenne, le MeanIR pour Eq_label est de 25% inférieur pour distilRoBERTa et 35% inférieur pour distilBERT par rapport à Classic-AL/Stale.

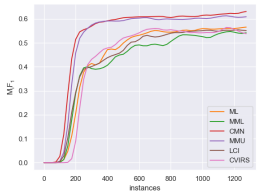
Les résultats pour chaque ensemble de données dans le Tableau 7 montrent qu'Eq_label surpasse les autres méthodes, obtenant des performances supérieures sur trois des quatre ensembles de données et se rapprochant de Random sur le quatrième, Jigsaw_toxic. Les ten-



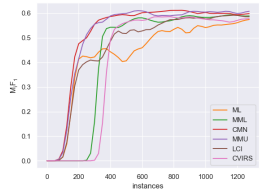
(a) distilBERT sur Jigsaw_toxic avec la méthode alternative **Random**



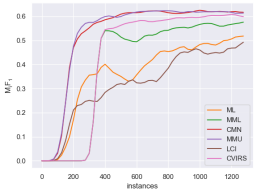
(b) distilBERT sur Jigsaw_toxic avec la méthode alternative **Stale**



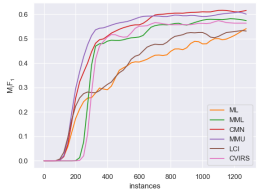
(c) distilBERT sur Jigsaw_toxic avec la méthode alternative **Eq_label**



(d) distilRoBERTa sur Jigsaw_toxic avec la méthode alternative **Random**



(e) distilRoBERTa sur Jigsaw_toxic avec la méthode alternative **Stale**



(f) distilRoBERTa sur Jigsaw_toxic avec la méthode alternative **Eq_label**

FIGURE 23 – Performances M_{iF1} suivant les différentes stratégies d'AA pour chaque transformers et chaque méthode alternative d'échantillonage

dances représentées dans le Tableau 8 correspondent à nos attentes, indiquant que les stratégies bénéficiant le plus de la mise en œuvre d'une méthode d'annotation alternative sont celles qui ont initialement montré des performances plus faibles, telles que ML et LCI. De plus, on peut observer que notre méthode Eq_label améliore les performances de manière similaire, voire dépasse Random. De plus, contrairement aux deux autres méthodes, l'inclusion de notre approche conduit à des améliorations dans toutes les stratégies.

Pour visualiser les nuances des courbes d'apprentissage, nous avons pris dans la Figure 23 l'exemple du jeu de données Jigsaw, où les performances des trois méthodes alternatives surpassent fréquemment celles du modèle d'apprentissage actif classique, et aucune des trois méthodes ne se distingue nettement par rapport aux deux autres.

Lorsqu'on analyse attentivement les courbes de performances comparées à la Figure 16, on observe des tendances significatives. Notamment, pour les méthodes Random et Eq_label, on constate une réduction notable des écarts de performances entre les différentes stratégies, créant des courbes plus étroitement groupées. Dans le cas de distilBert, il est particulièrement intéressant de noter que, pour les stratégies Stale et Eq_label, CMN et MMU affichent des performances nettement supérieures aux autres, reproduisant ainsi la tendance observée dans la Figure 16. En revanche, Random présente des écarts plus modestes entre les différentes stratégies. Cette observa-

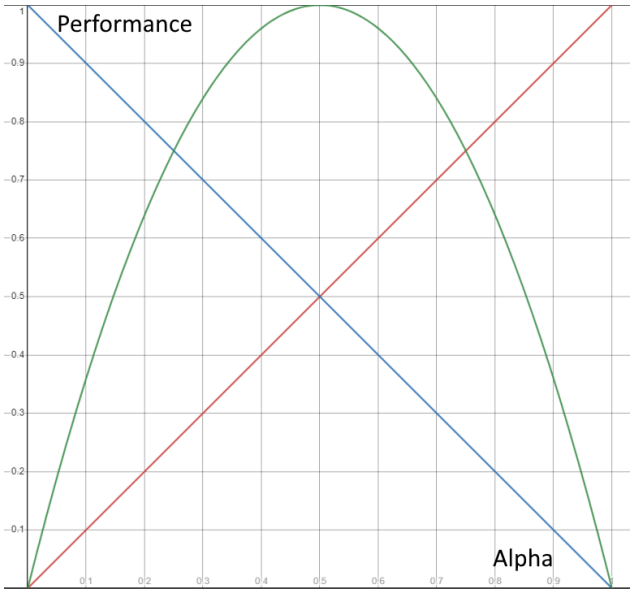


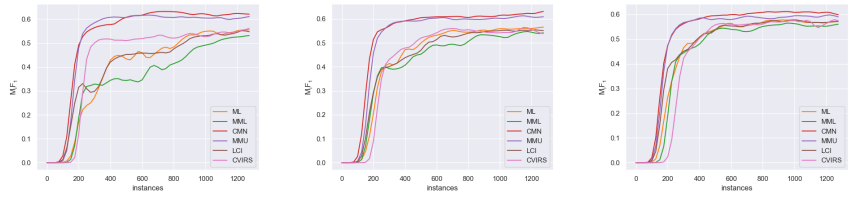
FIGURE 24 – Différentes allures de l'impact sur les performances du modèles en fonction du paramètre alpha.

tion est encore plus marquée pour le modèle distilRoBERTa, comme le confirment les performances finales répertoriées dans le Tableau 7.

Une autre distinction notable entre les courbes de performance de Stale et celles de la Figure 16, qui représentent le scénario Classic-AL, réside dans le fait que les plateaux de performances sont atteints avec un nombre d'instances plus élevé lorsque la méthode Stale est utilisée. En effet, en moyenne 450 instances sont nécessaires avec Stale contrairement aux 300 instances en moyenne nécessaires dans le scénario Classic-AL. Cette observation renforce l'idée que, selon cette méthode, les scores d'informativité utilisés sont moins pertinents, conduisant ainsi à un entraînement des modèles plus lents.

5.4.2.1 *Variations d'alpha*

Pour le modèle distilBERT, l'exploration des variations d'alpha au sein de la stratégie eq_label a fourni des perspectives intéressantes. En ajustant le paramètre alpha, qui représente la proportion d'instances provenant de la méthode alternative d'échantillonnage, nous avons exploré trois scénarios différents. Le premier où un quart des instances seulement provient de la méthode d'échantillonage alternative eq_label. Le deuxième, familier, où la moitié des instances provient de eq_label. Le troisième où les trois quart des instances proviennent de eq_label. Bien que nous ayons justifié l'usage du second scénario par le fait qu'il se rapprochait des conditions observées lors de notre mise en pratique de l'AA dans des conditions réelles, il est intéressant de voir comment la variation de ce paramètre influe sur les performances des stratégies.



(a) distilBERT sur Jigsaw_toxic pour alpha = 0.25

(b) distilBERT sur Jigsaw_toxic pour alpha = 0.5

(c) distilBERT sur Jigsaw_toxic pour alpha = 0.75

FIGURE 25 – Performances M_{iF1} suivant les différentes stratégies d'AA pour chaque transformers en faisant varier le paramètre alpha

L'exploration des variations d'alpha, dont l'impact sur les performances peuvent varier suivant des tendances illustrées dans la Figure 24, ouvre la voie à plusieurs perspectives prometteuses qui peuvent éclairer la mise en place d'un protocole d'annotation optimal (lorsque le choix d'alpha peut être effectué). Une corrélation positive entre la proportion d'exemples provenant de l'AA basée sur l'incertitude et les performances du modèle suggérerait de favoriser le nombre d'instances sélectionnées par AA. En revanche, une corrélation négative indiquerait la nécessité de prioriser les instances permettant un équilibrage des labels, soulignant ainsi l'importance de la méthode alternative d'échantillonnage. L'identification d'un point d'équilibre optimal entre l'AA et la méthode alternative serait particulièrement intéressante, démontrant la complémentarité de ces deux approches. Cette approche équilibrée pourrait représenter une stratégie gagnante, tirant parti des avantages respectifs de l'AA basée sur l'incertitude et de la méthode alternative d'échantillonnage pour maximiser l'efficacité du processus d'annotation active.

Dans la Figure 25 on peut voir les différentes courbes de performances sur le modèle distilBert suivant les différentes valeurs d'alpha. L'une des premières choses que l'on peut remarquer c'est que l'écart de performances entre les différentes stratégies est de moins en moins important lorsque la valeur d'alpha augmente, ce qui n'est pas surprenant étant donné que la part d'instances d'entraînements provenant d'une sélection par ces stratégies diminue au fur et à mesure que ce paramètre augmente. De plus, le point autour duquel se rassemblent ces stratégies semble assez proche de la moyenne des différentes stratégies, ce qui a pour conséquence fâcheuse de diminuer les performances des stratégies les plus performantes.

Afin d'étendre notre investigation à l'ensemble des jeux de données et de réaliser une évaluation des résultats plus précises nous affichons les performances finales de ces modèles en fin d'entraînement dans le Tableau 10.

Notre démarche d'analyse des résultats a débuté par la réalisation de t-tests statistiques par paires, mettant en évidence une différence statistiquement significative ($p < 0{,}05$ avec un test t apparié par en-

TABLE 10 – Résultats expérimentaux (M_{iF1}) sur distilBERT en faisant varier le paramètre alpha. Les meilleures valeurs pour la stratégie/alpha sont en gras.

Jeux de données ↓	**Alpha →** / **Stratégie ↓**	Classic-AL	0.25	0.5	0.75
Jigsaw_toxic	ML	0,412	0,555	0,567	**0,574**
	MML	0,467	0,531	0,547	**0,564**
	CMN	0,617	0,62	**0,631**	0,599
	MMU	0,603	**0,614**	0,610	0,586
	LCI	0,453	0,545	0,551	**0,572**
	CVIRS	0,564	**0,566**	0,537	**0,566**
	Moyenne	0,519	0,572	0,574	**0,577**
Go_emotions	ML	0,404	0,377	**0,412**	0,406
	MML	0,414	0,393	**0,416**	0,403
	CMN	0,438	0,409	**0,448**	0,403
	MMU	0,441	0,388	**0,455**	0,404
	LCI	0,353	0,344	**0,422**	0,389
	CVIRS	0,362	0,386	**0,405**	0,387
	Moyenne	0,402	0,383	**0,426**	0,399
EUR_Lex	ML	0,329	0,499	0,502	**0,516**
	MML	0,517	0,529	**0,543**	0,533
	CMN	0,513	0,517	**0,521**	0,515
	MMU	0,53	**0,545**	**0,545**	0,52
	LCI	0,454	0,519	0,515	**0,524**
	CVIRS	0,479	**0,545**	0,534	0,535
	Moyenne	0,470	0,526	**0,527**	0,524
UNFAIR-ToS	ML	0,622	**0,708**	0,703	0,694
	MML	0,647	0,709	**0,717**	0,707
	CMN	0,708	0,753	**0,767**	0,751
	MMU	0,7	0,732	0,738	**0,743**
	LCI	0,65	0,725	**0,765**	0,734
	CVIRS	0,708	0,708	**0,721**	0,697
	Moyenne	0,673	0,723	**0,735**	0,721

semble de données et valeur du paramètre alpha) entre les performances issues des différentes variations du paramètre alpha.

Il est intéressant de noter que, dans l'ensemble des résultats, les moyennes de performances associées aux différentes variations du paramètre alpha sont relativement proches les unes des autres. Il est particulièrement pertinent de souligner que, dans trois cas sur quatre, la moyenne pour la valeur de 0.5 est supérieure aux autres, suggérant que cette valeur pourrait constituer une cible optimale pour alpha. Ces performances semblent être impactées par la valeur d'alpha de manière similaire à la tendance observée sur la courbe verte de la Figure 24.

Le Tableau 10 présente en détail les résultats expérimentaux (M_{iF1}) sur distilBERT, en faisant varier le paramètre alpha. Les meilleures

valeurs pour la stratégie/alpha sont mises en gras pour chaque jeu de données. En observant ces résultats, on peut déduire que, en moyenne, la valeur de 0,5 pour alpha semble être une sélection judicieuse, avec des performances globalement équilibrées et compétitives pour plusieurs jeux de données.

5.5 DISCUSSION ET CONCLUSION

Dans ce chapitre, nous avons entamé notre exploration en détaillant une problématique bien réelle inhérente à la mise en œuvre de l'apprentissage actif (AA) : les temps d'attente pour les annotateurs, une réalité inhérente au déroulement des cycles d'AA. Ce constat, appuyé par des observations et des intuitions découlant de la conception d'un outil d'annotation avec AA dans un contexte industriel, a ouvert la voie à une extension de notre recherche en se penchant sur une caractéristique peu explorée de l'AA.

Saisissant l'opportunité d'approfondir notre compréhension de cette dimension sous-estimée de l'AA, nous avons développé une approche novatrice visant à résoudre deux problèmes simultanément : d'une part, pallier aux temps d'attente des annotateurs en leur proposant des instances à annoter issues d'une source alternative à l'AA, et d'autre part, garantir que cette source de données soit complémentaire à l'AA. Cette stratégie vise ainsi à optimiser l'efficacité globale du processus d'annotation, tout en offrant une perspective nouvelle sur la synergie entre l'apprentissage actif et d'autres méthodes d'échantillonnage.

Il est important de noter que notre approche, qui aborde le problème du temps d'attente de l'annotateur dans l'AA, a montré que l'intégration d'une méthode d'annotation parallèle peut améliorer significativement la production globale du processus d'annotation via l'AA. L'intégration de notre approche reste donc pertinente dans un projet où il n'y aurait pas de temps d'attente.

Notre approche, combinant trois concepts clés - prioriser la certitude sur l'incertitude en ce qui concerne la vétusté, améliorer les performances grâce au rééquilibrage de la distribution des labels et utiliser la certitude de la prédiction du modèle pour l'association des labels - améliore de manière constante les résultats sur tous les ensembles de données, modèles et stratégies d'AA.

D'un point de vue pratique, notre méthode élimine avec succès le temps d'attente qui peut exister lors d'applications classiques d'AA. Elle simplifie également le processus de décision au début d'un projet d'AA en rendant le choix d'une stratégie d'AA moins cruciale pour son succès. Bien que généralement moins performante, Random reste une option à considérer pour sa facilité de mise en œuvre.

Nous avons également poussé notre exploration en examinant différentes proportions d'instances provenant de l'AA et de notre mé-

thode complémentaire dans la constitution des lots de données d'entraînements. Ces expériences révèlent qu'un équilibre à parts égales entre les deux sources permet d'exploiter au mieux les avantages offerts par chacune d'entre elles dans la sélection des instances à annoter. Cette constatation souligne l'importance d'une approche hybride, où l'AA et notre méthode alternative se complètent mutuellement, renforçant ainsi la robustesse et l'efficacité du processus global d'annotation.

Troisième partie

CONCLUSION

6

CONCLUSION

En guise de conclusion, cette étude s'inscrit dans le contexte de l'apprentissage actif (AA) appliqué à des données multi-labels, en exploitant les avancées significatives des modèles transformers. L'utilisation de ces derniers, notamment des versions distillées de ces modèles, comme distilBERT et distilRoBERTa, a ouvert de nouvelles perspectives pour améliorer l'efficacité des tâches de classification. Notre exploration s'est concentrée sur la classification multi-labels, une tâche complexe en matière d'annotation, offrant un terrain propice à l'application de l'AA.

Le défi principal résidait dans l'adaptation des stratégies d'AA, majoritairement conçues avant l'avènement des transformers, à ces nouveaux modèles. L'AA, en tant que processus visant à optimiser l'annotation des données, implique des coûts d'implémentation significatifs lors de sa mise en œuvre. Son intérêt réside dans la promesse que son intégration conduira à une amélioration tangible. Cependant, pour les transformers, cette promesse est incertaine. En tant que modèles imposants avec des coûts temporels et matériels conséquents, les transformers combinés à l'interaction interactive avec des annotateurs humains, posent des défis particuliers.

Notre étude s'est concentrée sur les stratégies d'AA basées sur l'incertitude, non seulement en raison de leur explicabilité accrue (un critère intéressant avec les transformers, où l'explicabilité est un enjeu important), mais aussi en raison de la rapidité de calcul inhérente à ces stratégies. Ainsi, notre recherche a répondu à deux besoins essentiels : évaluer les stratégies d'AA basées sur l'incertitude qui fonctionnent bien en conjonction avec les transformers, tout en cherchant à valoriser ce qui est généralement une contrainte pratique de l'AA. En définitive, cette étude apporte des éclairages significatifs sur l'optimisation de l'AA dans le contexte des modèles transformers, contribuant ainsi à une meilleure compréhension des synergies potentielles entre ces deux domaines.

Les premières expérimentations menées dans le cadre de cette étude, explorant l'utilisation conjointe de l'AA avec des modèles de classification multi-labels basés sur les transformers, ont révélé la pertinence de deux stratégies basées sur l'incertitude parmi les six évaluées, à savoir CMN et MMU. Cependant, cette étude souligne également l'importance d'une phase initiale dans le projet visant à identifier la stratégie la mieux adaptée au contexte, en prenant en compte les spécificités des données et du modèle. En effet, la qualité du modèle

final dépend fortement de la stratégie d'apprentissage actif choisie et intégrée dans le projet d'annotation.

Par ailleurs, nous avons démontré que les performances relativement faibles des quatre stratégies d'AA (qui sont inférieures à un échantillonnage aléatoire) ne sont pas uniquement attribuables à des problèmes de redondance inhérents à l'AA basée sur l'incertitude ou à des difficultés de diversité liées à l'adaptation de ces stratégies au traitement par lots de données. Cette réalité découle plutôt de la reconnaissance que les besoins d'entraînement des modèles transformers sont variés, et que certaines instances redondantes jouent un rôle crucial dans la capture de subtilités et d'ambiguïtés.

Ainsi, ces premières expérimentations mettent en lumière la nécessité d'adopter une approche pro-active dès le début du projet, afin d'ajuster judicieusement les stratégies d'annotation en fonction des particularités des modèles et des caractéristiques du jeu de données. Cela garantit non seulement des performances optimales, mais assure également une utilisation efficiente des ressources d'annotation, soulignant ainsi l'importance de la prise en compte précoce et précise des dynamiques entre l'AA et les modèles transformers pour une mise en œuvre réussie.

Seulement la nécessité d'une phase initiale dans tout projet d'annotation avec de l'AA semble être un obstacle à la généralisation de son utilisation. Pendant la conception et le déploiement d'un outil d'annotation intégrant de l'AA, nous avons identifié un autre obstacle lié au temps d'attente de l'annotateur pendant un cycle d'AA actif. L'AA repose sur une interaction humain-machine, et pendant les phases où le modèle se met à jour, infère des prédictions ou réalise des calculs de sélection, l'annotateur attend passivement. Bien que fournir des données sélectionnées aléatoirement puisse atténuer cette situation de manière acceptable, nous estimons que cela compromet la crédibilité de l'AA, qui est censé reposer sur une sélection intelligente d'instances. Cela devient particulièrement problématique lorsque la part d'instances à fournir pendant ces temps d'attente est significative, comme observé dans nos expérimentations, représentant près de la moitié des instances à annoter.

En réponse à ces défis, nous avons élaboré une approche novatrice visant à combler les temps d'attente en intégrant une source complémentaire d'échantillonnage d'instances. Cette approche sert à valoriser ces temps d'attente tout en résolvant une autre problématique inhérente à la tâche multi-labels : le déséquilibre entre les labels. Fondée sur la certitude du modèle, notre approche permet naturellement de diversifier les données d'entraînement, car les instances sélectionnées par l'AA le sont en fonction de leur incertitude. Cette démarche a permis d'optimiser la productivité des annotateurs en maximisant l'utilisation du temps effectif d'annotation, tout en améliorant les performances des modèles.

Les résultats obtenus, tant au niveau de la maximisation du temps d'annotation que des performances des modèles, démontrent l'efficacité de cette approche hybride. Le gain en performance est particulièrement notable pour les stratégies d'AA qui fonctionnaient moins bien, car avec notre architecture hybride, toutes les stratégies d'AA surpassent l'échantillonnage aléatoire. En saisissant les opportunités offertes par l'AA et en cherchant à surmonter ses limites pratiques, cette recherche a abouti à une architecture qui intègre l'AA en tenant compte de ses aspects pratiques et ne nécessitant pas de phase initiale, tout en garantissant des gains de performance, peu importe la stratégie employée.

Au cours de la réalisation de ces travaux, nous avons perçu différentes perspectives de recherches que nous trouvons prometteuses dans le renforcement des performances et des promesses de l'AA appliqué aux transformers sur la tâche multi-labels :

— **Réseau neuronal de classification.** Dans le cadre de nos expérimentations, nous avons opté pour l'utilisation d'une architecture de réseau neuronal relativement simple en sortie des modèles transformers pour la classification. Nous postulons que l'utilisation d'architectures plus complexes pourrait considérablement améliorer les performances absolues des modèles de classification multi-labels basés sur les transformers. Bien que notre intuition suggère que les résultats relatifs des différentes stratégies étudiées dans cette recherche ne seront que peu impactés par ce changement, il serait néanmoins pertinent de le confirmer par des expérimentations supplémentaires. En considérant la mise en pratique réelle de l'AA, il devient crucial de déterminer comment obtenir les meilleures performances en matière de classification. À cet égard, nous croyons qu'explorer les aspects liés à l'architecture du réseau de classification constitue la première piste à investiguer. En se penchant sur les détails du réseau, notamment en l'optimisant ou en le complexifiant, il est envisageable d'exploiter pleinement les capacités des modèles transformers et, par conséquent, d'accroître l'efficacité de l'AA dans le contexte de la classification multi-labels.

— **Fonctions d'activation.** Dans la majorité écrasante des travaux dédiés à la tâche de classification multi-labels, y compris les nôtres, la fonction d'activation sigmoïde est employée. Cette fonction est utilisée pour convertir les prédictions du modèle en probabilités, facilitant ainsi le processus de classification. Cette pratique s'étend également à l'AA appliqué à la classification multi-labels. Toutefois, au sein de l'AA, nous postulons que certaines propriétés de la fonction sigmoïde pourraient entraver son efficacité. En effet, les fonctions sigmoïdes ont la particularité de "niveler" les valeurs qui se situent près de leurs limites (dans notre cas, 0 et 1). Cela peut sembler bénéfique à première

vue, car cela renforce les écarts entre les valeurs proches du centre de la plage (dans notre cas, 0.5) et être une caractéristique à valoriser dans le contexte de l'AA basé sur l'incertitude. Les modèles transformers, et les réseaux neuronaux en général, ont tendance à manifester une confiance excessive dans leurs prédictions. Cela signifie que même les valeurs associées aux données les plus incertaines ont de fortes chances de se retrouver dans ces zones d'aplatissement induites par la sigmoïde. Dans ce contexte, il pourrait être intéressant d'explorer l'impact d'autres fonctions d'activation dans le cadre de l'AA multi-labels en utilisant des transformers. En évaluant différentes fonctions d'activation, nous pourrions mieux comprendre comment adapter ces choix pour améliorer la performance de l'AA et atténuer la propension des modèles à être trop confiants dans leurs prédictions.

— **Valoriser le temps de l'annotateur.** La considération de la rentabilité est intrinsèque à l'AA, et dans nos travaux, nous avons choisi d'approfondir cette notion en explorant la valorisation du temps de l'annotateur. Cette approche prend une importance particulière, étant donné que l'AA repose sur l'interaction entre l'humain et la machine. Valoriser le temps de l'annotateur peut non seulement rendre cette interaction plus intéressante pour la machine, mais aussi plus gratifiante pour l'humain. Dans nos expérimentations préliminaires, nous avons observé que la sélection d'instances plus simples à lire, c'est-à-dire plus courtes et moins complexes, avait un impact négligeable sur les performances, bien que légèrement négatif. Nous croyons que ce domaine mérite une exploration approfondie. En choisissant des instances plus faciles à annoter, il est possible de réduire le nombre d'erreurs d'annotation tout en accélérant le processus d'annotation, permettant ainsi d'annoter un plus grand nombre d'instances et rendant la tâche plus simple de renforcer l'adhésion de l'annotateur. Une étude impliquant la participation de plusieurs annotateurs et variant ces paramètres pourrait apporter des intuitions significatives sur la manière d'optimiser la rentabilité temporelle tout en maintenant des niveaux élevés de performances et de qualité d'annotation.

— **Exploration d'autres tâches.** Notre recherche a mis en lumière un protocole d'AA reposant sur une architecture hybride, que nous considérons comme une pratique prometteuse à généraliser pour étendre l'utilisation de l'AA. Il serait donc pertinent de concevoir des architectures similaires adaptées à d'autres tâches de classification, telles que la classification multi-classes, ou même pour d'autres types de données comme les images. Nous croyons que l'adaptation de la méthode d'échantillonnage alternative aux caractéristiques spécifiques de la nouvelle tâche

permettrait de bénéficier de manière significative à tout projet d'annotation. La confirmation expérimentale de cette hypothèse serait particulièrement enrichissante. Explorer la transposition de notre architecture hybride vers d'autres domaines permettrait d'évaluer son applicabilité générale et sa capacité à offrir des gains de performance, tout en s'adaptant aux particularités de chaque tâche. Cette démarche contribuerait non seulement à valider l'efficacité de notre approche dans des contextes variés, mais également à élargir son champ d'application et à identifier les ajustements nécessaires pour optimiser son utilisation dans divers scénarios. Ainsi, la généralisation de notre protocole d'AA pourrait représenter une avancée significative dans la mise en œuvre pratique de l'AA, offrant des perspectives d'optimisation dans un large éventail d'applications.

BIBLIOGRAPHIE

[1] Abien Fred Agarap. *Deep Learning using Rectified Linear Units (ReLU)*. arXiv :1803.08375 [cs, stat]. Fév. 2019.

[2] P. Agarwal, Sariel Har-Peled et Kasturi R. Varadarajan. « Geometric Approximation via Coresets ». In : 2007.

[3] Umang Aggarwal, Adrian Popescu et Celine Hudelot. « Active Learning for Imbalanced Datasets ». en. In : *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Snowmass Village, CO, USA : IEEE, mars 2020, p. 1417-1426. isbn : 978-1-72816-553-0.

[4] Umang Aggarwal, Adrian Popescu et Céline Hudelot. « Minority Class Oriented Active Learning for Imbalanced Datasets ». In : *2020 25th International Conference on Pattern Recognition (ICPR)*. arXiv :2202.00390 [cs]. Jan. 2021, p. 9920-9927.

[5] Jay Alammar. « The illustrated transformer ». In : *The Illustrated Transformer–Jay Alammar–Visualizing Machine Learning One Concept at a Time* 27 (2018).

[6] Reem Alotaibi et Peter Flach. « Multi-label thresholding for cost-sensitive classification ». In : *Neurocomputing* 436 (mai 2021), p. 232-247. issn : 0925-2312.

[7] Vamshi Ambati, Stephan Vogel et Jaime Carbonell. « Multi-Strategy Approaches to Active Learning for Statistical Machine Translation ». In : *Proceedings of Machine Translation Summit XIII : Papers*. Xiamen, China, sept. 2011.

[8] Dana Angluin. « Queries and Concept Learning ». en. In : *Machine Learning* 2.4 (avr. 1988), p. 319-342. issn : 1573-0565.

[9] Maxime Arens, Lucile Callebert, Jose G Moreno et Mohand Boughanem. *Rebalancing Label Distribution while Eliminating Inherent Waiting Time in Multi Label Active Learning applied to Transformers*. 2024.

[10] Maxime Arens, Charles Teissèdre, Lucile Callebert, Jose G Moreno et Mohand Boughanem. « Impact de l'apprentissage multi-labels actif appliqué aux transformers ». French. In : *Actes de CORIA-TALN 2023. Actes de la 18e Conférence en Recherche d'Information et Applications (CORIA)*. Sous la dir. d'Haïfa Zargayouna. Paris, France : ATALA, juin 2023, p. 2-17.

[11] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford et Alekh Agarwal. *Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds*. arXiv :1906.03671 [cs, stat]. Fév. 2020.

[12] Seyed Arad Ashrafi Asli, Behnam Sabeti, Zahra Majdabadi, Preni Golazizian, Reza Fahmi et Omid Momenzadeh. « Optimizing Annotation Effort Using Active Learning Strategies : A Sentiment Analysis Case Study in Persian ». English. In : *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France : European Language Resources Association, mai 2020, p. 2855-2861. ISBN : 979-10-95546-34-4.

[13] Josh Attenberg et Şeyda Ertekin. « Class imbalance and active learning ». In : *Imbalanced Learning : Foundations, Algorithms, and Applications* (2013), p. 101-149.

[14] Philip Bachman, Alessandro Sordoni et Adam Trischler. *Learning Algorithms for Active Learning*. arXiv :1708.00088 [cs]. Juill. 2017.

[15] Shawn Beaulieu, Lapo Frati, Thomas Miconi, Joel Lehman, Kenneth O Stanley, Jeff Clune et Nick Cheney. « Learning to Continually Learn ». en. In : *Santiago de Compostela* (2020), p. 10.

[16] Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter et Lihi Zelnik-Manor. « Asymmetric Loss For Multi-Label Classification ». In : *arXiv :2009.14119 [cs]* (juill. 2021). arXiv : 2009.14119.

[17] James Bergstra et al. « Theano : Deep Learning on GPUs with Python ». en. In : ().

[18] Magdalena Biesialska, Katarzyna Biesialska et Marta R. Costa-jussà. « Continual Lifelong Learning in Natural Language Processing : A Survey ». In : *Proceedings of the 28th International Conference on Computational Linguistics* (2020). arXiv : 2012.09823, p. 6523-6541.

[19] Ilai Bistritz, Ariana Mann et Nicholas Bambos. « Distributed Distillation for On-Device Learning ». In : *Advances in Neural Information Processing Systems*. T. 33. Curran Associates, Inc., 2020, p. 22593-22604.

[20] Jasmin Bogatinovski, Ljupčo Todorovski, Sašo Džeroski et Dragi Kocev. *Comprehensive Comparative Study of Multi-Label Classification Methods*. arXiv :2102.07113 [cs]. Fév. 2021.

[21] Matthew R. Boutell, Jiebo Luo, Xipeng Shen et Christopher M. Brown. « Learning multi-label scene classification ». In : *Pattern Recognition* 37.9 (sept. 2004), p. 1757-1771. ISSN : 0031-3203.

[22] Leo Breiman. « Random Forests ». en. In : *Machine Learning* 45.1 (oct. 2001), p. 5-32. ISSN : 1573-0565.

[23] Klaus BRINKER. « Incorporating diversity in active learning with support vector machines ». In : *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*. ICML'03. Washington, DC, USA : AAAI Press, 2003, p. 59-66. ISBN : 978-1-57735-189-4.

[24] Tom B. BROWN et al. *Language Models are Few-Shot Learners*. arXiv :2005.14165 [cs]. Juill. 2020.

[25] Cristian BUCILUǍ, Rich CARUANA et Alexandru NICULESCU-MIZIL. « Model compression ». In : *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '06. New York, NY, USA : Association for Computing Machinery, août 2006, p. 535-541. ISBN : 978-1-59593-339-3.

[26] Tingting CAI, Zhiyuan MA, Hong ZHENG et Yangming ZHOU. « NE–LP : Normalized entropy- and loss prediction-based sampling for active learning in Chinese word segmentation on EHRs ». In : *Neural Computing and Applications* 33 (oct. 2021).

[27] Wenbin CAI, Yexun ZHANG, Ya ZHANG, Siyuan ZHOU, Wenquan WANG, Zhuoxiang CHEN et Chris DING. « Active Learning for Classification with Maximum Model Change ». In : *ACM Transactions on Information Systems* 36.2 (2017), 15 :1-15 :28. ISSN : 1046-8188.

[28] Xiangyong CAO, Jing YAO, Zongben XU et Deyu MENG. « Hyperspectral Image Classification With Convolutional Neural Network and Active Learning ». In : *IEEE Transactions on Geoscience and Remote Sensing* 58.7 (juill. 2020). Conference Name : IEEE Transactions on Geoscience and Remote Sensing, p. 4604-4616. ISSN : 1558-0644.

[29] Thiago N. C. CARDOSO, Rodrigo M. SILVA, Sérgio CANUTO, Mirella M. MORO et Marcos A. GONÇALVES. « Ranked batch-mode active learning ». In : *Information Sciences* 379 (fév. 2017), p. 313-337. ISSN : 0020-0255.

[30] Ilias CHALKIDIS, Manos FERGADIOTIS et Ion ANDROUTSOPOULOS. « MultiEURLEX - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer ». In : *CoRR* abs/2109.00904 (2021). arXiv : 2109.00904.

[31] Ilias CHALKIDIS, Abhik JANA, Dirk HARTUNG, Michael BOMMARITO, Ion ANDROUTSOPOULOS, Daniel Martin KATZ et Nikolaos ALETRAS. « LexGLUE : A Benchmark Dataset for Legal Language Understanding in English ». In : *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Dubln, Ireland, 2022.

[32] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang et Inderjit Dhillon. « X-BERT : eXtreme Multi-label Text Classification using Bidirectional Encoder Representations from Transformers ». en. In : (), p. 12.

[33] Chenhua Chen, Alexis Palmer et Caroline Sporleder. « Enhancing Active Learning for Semantic Role Labeling via Compressed Dependency Trees ». In : *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand : Asian Federation of Natural Language Processing, nov. 2011, p. 183-191.

[34] Shuyue Chen, Ran Wang, Jian Lu et Xizhao Wang. « Stable matching-based two-way selection in multi-label active learning with imbalanced data ». In : *Information Sciences* 610 (2022), p. 281-299. issn : 0020-0255.

[35] Zhiyuan Chen et Bing Liu. *Lifelong Machine Learning*. en. Synthesis Lectures on Artificial Intelligence and Machine Learning. Cham : Springer International Publishing, 2018. isbn : 978-3-031-00453-7 978-3-031-01581-6.

[36] Ke Cheng, Shang Gao, Wenlu Dong, Xibei Yang, Qi Wang et Hualong Yu. « Boosting label weighted extreme learning machine for classifying multi-label imbalanced data ». In : *Neurocomputing* 403 (août 2020), p. 360-370. issn : 0925-2312.

[37] Weiwei Cheng et Eyke Hüllermeier. « Combining instance-based learning and logistic regression for multilabel classification ». en. In : *Machine Learning* 76.2 (sept. 2009), p. 211-225. issn : 1573-0565.

[38] François Chollet et al. *Keras*. https://keras.io. 2015.

[39] Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh et Sanjiv Kumar. *Batch Active Learning at Scale*. arXiv :2107.14263 [cs]. Juill. 2021.

[40] Amanda Clare et Ross D. King. « Knowledge Discovery in Multi-label Phenotype Data ». en. In : *Principles of Data Mining and Knowledge Discovery*. Sous la dir. de Luc De Raedt et Arno Siebes. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2001, p. 42-53. isbn : 978-3-540-44794-8.

[41] David A. Cohn, Zoubin Ghahramani et Michael I. Jordan. « Active Learning with Statistical Models ». In : *J. Artif. Int. Res.* 4.1 (1996), 129–145. issn : 1076-9757.

[42] David Cohn, Les Atlas et Richard Ladner. « Improving generalization with active learning ». en. In : *Machine Learning* 15.2 (mai 1994), p. 201-221. issn : 1573-0565.

[43] Aron Culotta et Andrew McCallum. « Reducing Labeling Effort for Structured Prediction Tasks ». In : *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*. AAAI'05. Pittsburgh, Pennsylvania : AAAI Press, 2005, 746–751. isbn : 157735236x.

[44] Mike D'Arcy et Doug Downey. « Limitations of Active Learning With Deep Transformer Language Models ». en. In : (sept. 2021).

[45] Mike D'Arcy et Doug Downey. *Limitations of Active Learning With Deep Transformer Language Models*. 2022.

[46] Ozgur Dedehayir et Martin Steinert. « The hype cycle model : A review and future directions ». In : *Technological Forecasting and Social Change* 108 (juill. 2016), p. 28-41. issn : 0040-1625.

[47] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade et Sujith Ravi. « GoEmotions : A Dataset of Fine-Grained Emotions ». In : *58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2020.

[48] Jacob Devlin, Ming-Wei Chang, Kenton Lee et Kristina Toutanova. « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding ». In : *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*. Sous la dir. de Jill Burstein, Christy Doran et Thamar Solorio. Minneapolis, Minnesota : Association for Computational Linguistics, juin 2019, p. 4171-4186.

[49] Pinar Donmez, Jaime G. Carbonell et Paul N. Bennett. « Dual Strategy Active Learning ». en. In : *Machine Learning : ECML 2007*. Sous la dir. de Joost N. Kok, Jacek Koronacki, Raomon Lopez de Mantaras, Stan Matwin, Dunja Mladenič et Andrzej Skowron. T. 4701. ISSN : 0302-9743, 1611-3349 Series Title : Lecture Notes in Computer Science. Berlin, Heidelberg : Springer Berlin Heidelberg, 2007, p. 116-127. isbn : 978-3-540-74957-8 978-3-540-74958-5.

[50] Gregory Druck, Burr Settles et Andrew McCallum. « Active Learning by Labeling Features ». In : *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore : Association for Computational Linguistics, août 2009, p. 81-90.

[51] Bo Du, Zengmao Wang, Lefei Zhang, Liangpei Zhang, Wei Liu, Jialie Shen et Dacheng Tao. *Exploring Representativeness and Informativeness for Active Learning*. arXiv :1904.06685 [cs, stat]. Avr. 2019.

[52] Melanie Ducoffe et Frederic Precioso. *Adversarial Active Learning for Deep Networks : a Margin Based Approach*. arXiv :1802.09841 [cs, stat]. Fév. 2018.

[53] Thibaut Durand, Nazanin Mehrasa et Greg Mori. « Learning a Deep ConvNet for Multi-Label Classification With Partial Labels ». In : 2019, p. 647-657.

[54] Cynthia Dwork, Ravi Kumar, Moni Naor et D. Sivakumar. « Rank aggregation methods for the Web ». In : *Proceedings of the 10th international conference on World Wide Web*. WWW '01. New York, NY, USA : Association for Computing Machinery, 2001, p. 613-622. isbn : 978-1-58113-348-6.

[55] Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz et Noam Slonim. « Active Learning for BERT : An Empirical Study ». In : *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online : Association for Computational Linguistics, nov. 2020, p. 7949-7962.

[56] Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz et Noam Slonim. « Active Learning for BERT : An Empirical Study ». In : *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online : Association for Computational Linguistics, nov. 2020, p. 7949-7962.

[57] André Elisseeff et Jason Weston. « A kernel method for multi-labelled classification ». In : *Advances in Neural Information Processing Systems*. T. 14. MIT Press, 2001.

[58] Jesper E. van Engelen et Holger H. Hoos. « A survey on semi-supervised learning ». en. In : *Machine Learning* 109.2 (fév. 2020), p. 373-440. issn : 1573-0565.

[59] Sean P. Engelson et Ido Dagan. « Minimizing Manual Annotation Cost in Supervised Training from Corpora ». In : *34th Annual Meeting of the Association for Computational Linguistics*. Santa Cruz, California, USA : Association for Computational Linguistics, juin 1996, p. 319-326.

[60] Alexander Erdmann, David Joseph Wrisley, Benjamin Allen, Christopher Brown, Sophie Cohen-Bodénès, Micha Elsner, Yukun Feng, Brian Joseph, Béatrice Joyeux-Prunel et Marie-Catherine de Marneffe. « Practical, Efficient, and Customizable Active Learning for Named Entity Recognition in the Digital Humanities ». In : *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and*

Short Papers). Minneapolis, Minnesota : Association for Computational Linguistics, juin 2019, p. 2223-2234.

[61] Carmen Esposito, Gregory A. Landrum, Nadine Schneider, Nikolaus Stiefl et Sereina Riniker. « GHOST : Adjusting the Decision Threshold to Handle Imbalanced Data in Machine Learning ». In : *Journal of Chemical Information and Modeling* 61.6 (juin 2021). Publisher : American Chemical Society, p. 2623-2640. ISSN : 1549-9596.

[62] Andrea Esuli, Tiziano Fagni et Fabrizio Sebastiani. « MP-Boost : A Multiple-Pivot Boosting Algorithm and Its Application to Text Categorization ». en. In : *String Processing and Information Retrieval*. Sous la dir. de Fabio Crestani, Paolo Ferragina et Mark Sanderson. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2006, p. 1-12. ISBN : 978-3-540-45775-6.

[63] Andrea Esuli et Fabrizio Sebastiani. « Active Learning Strategies for Multi-Label Text Classification ». In : *Advances in Information Retrieval*. Sous la dir. de Mohand Boughanem, Catherine Berrut, Josiane Mothe et Chantal Soule-Dupuy. Berlin, Heidelberg : Springer Berlin Heidelberg, 2009, p. 102-113. ISBN : 978-3-642-00958-7.

[64] European Commission. Joint Research Centre. *AI Watch, historical evolution of artificial intelligence : analysis of the three main paradigm shifts in AI.* en. LU : Publications Office, 2020.

[65] Meng Fang, Yuan Li et Trevor Cohn. *Learning how to Active Learn : A Deep Reinforcement Learning Approach*. arXiv :1708.02383 [cs]. Août 2017.

[66] Damien Fourure. « Réseaux de neurones convolutifs pour la segmentation sémantique et l'apprentissage d'invariants de couleur ». fr. Thèse de doct. Université de Lyon, déc. 2017.

[67] Teodor Fredriksson, David Issa Mattos, Jan Bosch et Helena Holmström Olsson. « Data Labeling : An Empirical Investigation into Industrial Challenges and Mitigation Strategies ». In : *Product-Focused Software Process Improvement*. Sous la dir. de Maurizio Morisio, Marco Torchiano et Andreas Jedlitschka. Cham : Springer International Publishing, 2020, p. 202-216. ISBN : 978-3-030-64148-1.

[68] Yifan Fu, Xingquan Zhu et Bin Li. « A survey on instance selection for active learning ». en. In : *Knowledge and Information Systems* 35.2 (mai 2013), p. 249-283. ISSN : 0219-1377, 0219-3116.

[69] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía et Klaus Brinker. « Multilabel classification via calibrated label ranking ». en. In : *Machine Learning* 73.2 (nov. 2008), p. 133-153. ISSN : 1573-0565.

[70] Yarin Gal et Zoubin Ghahramani. « Dropout as a Bayesian Approximation : Representing Model Uncertainty in Deep Learning ». en. In : *Proceedings of The 33rd International Conference on Machine Learning*. ISSN : 1938-7228. PMLR, juin 2016, p. 1050-1059.

[71] Nengneng Gao, Sheng-Jun Huang et Songcan Chen. « Multi-label active learning by model guided distribution matching ». en. In : *Frontiers of Computer Science* 10.5 (oct. 2016), p. 845-855. ISSN : 2095-2236.

[72] Nadia Ghamrawi et Andrew McCallum. « Collective multi-label classification ». In : *Proceedings of the 14th ACM international conference on Information and knowledge management*. CIKM '05. New York, NY, USA : Association for Computing Machinery, oct. 2005, p. 195-200. ISBN : 978-1-59593-140-5.

[73] Daniel Gissin et Shai Shalev-Shwartz. « Discriminative Active Learning ». In : *CoRR* abs/1907.06347 (2019). arXiv : 1907.06347.

[74] Shantanu Godbole et Sunita Sarawagi. « Discriminative Methods for Multi-labeled Classification ». en. In : *Advances in Knowledge Discovery and Data Mining*. Sous la dir. d'Honghua Dai, Ramakrishnan Srikant et Chengqi Zhang. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2004, p. 22-30. ISBN : 978-3-540-24775-3.

[75] Julius Gonsior, Christian Falkenberg, Silvio Magino, Anja Reusch, Claudio Hartmann, Maik Thiele et Wolfgang Lehner. « Comparing and Improving Active Learning Uncertainty Measures for Transformer Models ». en. In : *Advances in Databases and Information Systems*. Sous la dir. d'Alberto Abelló, Panos Vassiliadis, Oscar Romero et Robert Wrembel. Lecture Notes in Computer Science. Cham : Springer Nature Switzerland, 2023, p. 119-132. ISBN : 978-3-031-42914-9.

[76] Jianping Gou, Baosheng Yu, Stephen John Maybank et Dacheng Tao. « Knowledge Distillation : A Survey ». In : *International Journal of Computer Vision* 129.6 (juin 2021). arXiv : 2006.05525, p. 1789-1819. ISSN : 0920-5691, 1573-1405.

[77] Hongfei Gu. « Data, Big Tech, and the New Concept of Sovereignty ». en. In : *Journal of Chinese Political Science* (mai 2023). ISSN : 1874-6357.

[78] Xiaoqiang Gui, Xudong Lu et Guoxian Yu. « Cost-effective Batch-mode Multi-label Active Learning ». en. In : *Neurocomputing* 463 (nov. 2021), p. 355-367. ISSN : 0925-2312.

[79] Silviu GUIAŞU et Corina REISCHER. « Some remarks on entropic distance, entropic measure of connexion and Hamming distance ». fr. In : *RAIRO. Informatique théorique* 13.4 (1979), p. 395-407. ISSN : 2777-3337.

[80] Chuan GUO, Geoff PLEISS, Yu SUN et Kilian Q. WEINBERGER. « On Calibration of Modern Neural Networks ». In : *Proceedings of the 34th International Conference on Machine Learning*. Sous la dir. de Doina PRECUP et Yee Whye TEH. T. 70. Proceedings of Machine Learning Research. PMLR, 2017, p. 1321-1330.

[81] Yuhong GUO et Dale SCHUURMANS. « Discriminative Batch Mode Active Learning ». In : *Advances in Neural Information Processing Systems*. T. 20. Curran Associates, Inc., 2007.

[82] Yuhong GUO et Dale SCHUURMANS. « Adaptive Large Margin Training for Multilabel Classification ». en. In : *Proceedings of the AAAI Conference on Artificial Intelligence* 25.1 (août 2011). Number : 1, p. 374-379. ISSN : 2374-3468.

[83] Robbie A HAERTEL. « Practical Cost-Conscious Active Learning for Data Annotation in Annotator-Initiated Environments ». en. In : ().

[84] Robbie HAERTEL, Paul FELT, Eric K. RINGGER et Kevin SEPPI. « Parallel Active Learning : Eliminating Wait Time with Minimal Staleness ». In : *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*. Los Angeles, California : Association for Computational Linguistics, juin 2010, p. 33-41.

[85] Susanna HARTIKAINEN, Heta RINTALA, Laura PYLVÄS et Petri NOKELAINEN. « The Concept of Active Learning and the Measurement of Learning Outcomes : A Review of Research in Engineering Higher Education ». en. In : *Education Sciences* 9.4 (déc. 2019). Number : 4 Publisher : Multidisciplinary Digital Publishing Institute, p. 276. ISSN : 2227-7102.

[86] Simon HAYKIN. *Neural Networks : A Comprehensive Foundation*. 2nd. USA : Prentice Hall PTR, 1998. ISBN : 978-0-13-273350-2.

[87] Rishi HAZRA, Parag DUTTA, Shubham GUPTA, Mohammed Abdul QAATHIR et Ambedkar DUKKIPATI. « Active^2 Learning : Actively reducing redundancies in Active Learning methods for Sequence Tagging and Machine Translation ». In : *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*. Online : Association for Computational Linguistics, juin 2021, p. 1982-1995.

[88] Marek Herde, Denis Huseljic, Bernhard Sick et Adrian Calma. « A Survey on Cost Types, Interaction Schemes, and Annotator Performance Models in Selection Algorithms for Active Learning in Classification ». In : *IEEE Access* 9 (2021). Conference Name : IEEE Access, p. 166970-166989. ISSN : 2169-3536.

[89] Geoffrey Hinton, Oriol Vinyals et Jeff Dean. *Distilling the Knowledge in a Neural Network*. arXiv :1503.02531 [cs, stat]. Mars 2015.

[90] Sepp Hochreiter. « The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions ». In : *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 06.02 (avr. 1998). Publisher : World Scientific Publishing Co., p. 107-116. ISSN : 0218-4885.

[91] Steven C. H. Hoi, Doyen Sahoo, Jing Lu et Peilin Zhao. « Online Learning : A Comprehensive Survey ». In : *arXiv :1802.02871 [cs]* (oct. 2018). arXiv : 1802.02871.

[92] Mokter Hossain et Ilkka Kauranen. « Crowdsourcing : a comprehensive literature review ». In : *Strategic Outsourcing : An International Journal* 8.1 (jan. 2015). Publisher : Emerald Group Publishing Limited, p. 2-22. ISSN : 1753-8297.

[93] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani et Máté Lengyel. *Bayesian Active Learning for Classification and Preference Learning*. arXiv :1112.5745 [cs, stat]. Déc. 2011.

[94] Jeremy Howard et Sebastian Ruder. « Universal Language Model Fine-tuning for Text Classification ». In : *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*. Melbourne, Australia : Association for Computational Linguistics, juill. 2018, p. 328-339.

[95] Jun Huang, Guorong Li, Qingming Huang et Xindong Wu. « Learning Label-Specific Features and Class-Dependent Labels for Multi-Label Classification ». In : *IEEE Transactions on Knowledge and Data Engineering* 28.12 (déc. 2016). Conference Name : IEEE Transactions on Knowledge and Data Engineering, p. 3309-3323. ISSN : 1558-2191.

[96] Jun Huang, Guorong Li, Qingming Huang et Xindong Wu. « Joint Feature Selection and Classification for Multilabel Learning ». eng. In : *IEEE transactions on cybernetics* 48.3 (mars 2018), p. 876-889. ISSN : 2168-2275.

[97] Jun Huang, Feng Qin, Xiao Zheng, Zekai Cheng, Zhixiang Yuan, Weigang Zhang et Qingming Huang. « Improving multi-label classification with missing labels by learning label-specific features ». en. In : *Information Sciences* 492 (août 2019), p. 124-146. ISSN : 0020-0255.

[98] Sheng-Jun HUANG, Songcan CHEN et Zhi-Hua ZHOU. « Multi-Label Active Learning : Query Type Matters ». In : *IJCAI*. 2015.

[99] Sheng-Jun HUANG, Rong JIN et Zhi-Hua ZHOU. « Active Learning by Querying Informative and Representative Examples ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.10 (2014), p. 1936-1949.

[100] Sheng-Jun HUANG et Zhi-Hua ZHOU. « Active Query Driven by Uncertainty and Diversity for Incremental Multi-label Learning ». In : *2013 IEEE 13th International Conference on Data Mining*. 2013, p. 1079-1084.

[101] Sheng-jun HUANG, Rong JIN et Zhi-Hua ZHOU. « Active Learning by Querying Informative and Representative Examples ». In : *Advances in Neural Information Processing Systems*. T. 23. Curran Associates, Inc., 2010.

[102] Xin HUANG, Boli CHEN, Lin XIAO, Jian YU et Liping JING. « Label-Aware Document Representation via Hybrid Attention for Extreme Multi-Label Text Classification ». en. In : *Neural Processing Letters* 54.5 (oct. 2022), p. 3601-3617. ISSN : 1573-773X.

[103] Miriam HUIJSER et Jan C. Van GEMERT. « Active Decision Boundary Annotation with Deep Generative Models ». en. In : *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice : IEEE, oct. 2017, p. 5296-5305. ISBN : 978-1-5386-1032-9.

[104] Jung HYUN, Ruth EDIGER et Donghun LEE. « Students' Satisfaction on Their Learning Process in Active Learning and Traditional Classrooms ». en. In : *International Journal of Teaching and Learning in Higher Education* 29.1 (2017). Publisher : International Society for Exploring Teaching and Learning ERIC Number : EJ1135821, p. 108-118.

[105] Eyke HÜLLERMEIER et Willem WAEGEMAN. « Aleatoric and epistemic uncertainty in machine learning : an introduction to concepts and methods ». en. In : *Machine Learning* 110.3 (mars 2021), p. 457-506. ISSN : 1573-0565.

[106] Karim M IBRAHIM, Elena EPURE, Geoffroy PEETERS et Gael RICHARD. « Confidence-based Weighted Loss for Multi-label Classification with Missing Labels ». In : *The 2020 International Conference on Multimedia Retrieval (ICMR '20)*. Dublin, Ireland, juin 2020.

[107] Borna JAFARPOUR, Dawn SEPEHR et Nick POGREBNYAKOV. « Active Curriculum Learning ». In : *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*. Online : Association for Computational Linguistics, août 2021, p. 40-45.

[108] Fran Jelenić, Josip Jukić, Nina Drobac et Jan Šnajder. *On Dataset Transferability in Active Learning for Transformers*. arXiv :2305.09807 [cs]. Sept. 2023.

[109] Shuiwang Ji, Lei Tang, Shipeng Yu et Jieping Ye. « Extracting shared subspace for multi-label classification ». In : *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '08. New York, NY, USA : Association for Computing Machinery, 2008, p. 381-389. isbn : 978-1-60558-193-4.

[110] Elmer H. Johnson. « Elementary Applied Statistics : For Students in Behavioral Science. By Linton C. Freeman. New York : John Wiley & Sons, 1965. 298 pp. Tables and Figures. $6.95 ». In : *Social Forces* 44.3 (mars 1966), p. 455-456. issn : 0037-7732.

[111] Gyoungdon Joo et Chulyun Kim. « MIDAS : Model-Independent Training Data Selection Under Cost Constraints ». In : *IEEE Access* 6 (2018). Conference Name : IEEE Access, p. 74462-74474. issn : 2169-3536.

[112] Michael I. Jordan. « Chapter 25 - Serial Order : A Parallel Distributed Processing Approach ». In : *Advances in Psychology*. Sous la dir. de John W. Donahoe et Vivian Packard Dorsel. T. 121. Neural-Network Models of Cognition. North-Holland, jan. 1997, p. 471-495.

[113] Josip Jukić, Fran Jelenić, Miroslav Bićanić et Jan Snajder. « ALANNO : An Active Learning Annotation System for Mortals ». In : *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics : System Demonstrations*. Dubrovnik, Croatia : Association for Computational Linguistics, mai 2023, p. 228-235.

[114] B.L. Kalman et S.C. Kwasny. « Why tanh : choosing a sigmoidal function ». In : *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*. T. 4. Juin 1992, 578-581 vol.4.

[115] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu et Dario Amodei. *Scaling Laws for Neural Language Models*. arXiv :2001.08361 [cs, stat]. Jan. 2020.

[116] Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei et Christopher Manning. « Mind Your Outliers ! Investigating the Negative Impact of Outliers on Active Learning for Visual Question Answering ». 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)*. Online : Association for Computational Linguistics, août 2021, p. 7265-7281.

[117] Seho KEE, Enrique del CASTILLO et George RUNGER. « Query-by-committee improvement with diversity and density in batch active learning ». In : *Information Sciences* 454-455 (juill. 2018), p. 401-418. ISSN : 0020-0255.

[118] Arshia KHAN, Ona EGBUE, Brooke PALKIE et Janna MADDEN. « Active Learning : Engaging Students To Maximize Learning In An Online Course ». en. In : *Electronic Journal of e-Learning* 15.2 (mai 2017). Number : 2, pp107-115-pp107-115. ISSN : 1479-4403.

[119] Andreas KIRSCH, Joost van AMERSFOORT et Yarin GAL. *BatchBALD : Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning*. arXiv :1906.08158 [cs, stat]. Oct. 2019.

[120] Andreas KIRSCH, Sebastian FARQUHAR, Parmida ATIGHEHCHIAN, Andrew JESSON, Frederic BRANCHAUD-CHARRON et Yarin GAL. *Stochastic Batch Acquisition : A Simple Baseline for Deep Active Learning*. arXiv :2106.12059 [cs, stat]. Sept. 2023.

[121] Alex KRIZHEVSKY, Ilya SUTSKEVER et Geoffrey E HINTON. « ImageNet Classification with Deep Convolutional Neural Networks ». In : *Advances in Neural Information Processing Systems*. T. 25. Curran Associates, Inc., 2012.

[122] Anders KROGH et Jesper VEDELSBY. « Neural Network Ensembles, Cross Validation, and Active Learning ». In : *Advances in Neural Information Processing Systems*. Sous la dir. de G. TESAURO, D. TOURETZKY et T. LEEN. T. 7. MIT Press, 1994.

[123] Meelis KULL, Telmo M. Silva FILHO et Peter FLACH. « Beyond sigmoids : How to obtain well-calibrated probabilities from binary classifiers with beta calibration ». In : *Electronic Journal of Statistics* 11.2 (jan. 2017). Publisher : Institute of Mathematical Statistics and Bernoulli Society, p. 5052-5080. ISSN : 1935-7524, 1935-7524.

[124] Punit KUMAR et Atul GUPTA. « Active Learning Query Strategies for Classification, Regression, and Clustering : A Survey ». en. In : *Journal of Computer Science and Technology* 35.4 (juill. 2020), p. 913-945. ISSN : 1000-9000, 1860-4749.

[125] Balaji LAKSHMINARAYANAN, Alexander PRITZEL et Charles BLUNDELL. « Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles ». In : *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA : Curran Associates Inc., 2017, 6405–6416. ISBN : 9781510860964.

[126] Y. LECUN, B. BOSER, J. S. DENKER, D. HENDERSON, R. E. HOWARD, W. HUBBARD et L. D. JACKEL. « Backpropagation Applied to Handwritten Zip Code Recognition ». In : *Neural Computation*

1.4 (déc. 1989). Conference Name : Neural Computation, p. 541-551. ISSN : 0899-7667.

[127] Yann LeCun, Yoshua Bengio et Geoffrey Hinton. « Deep learning ». en. In : *Nature* 521.7553 (mai 2015), p. 436-444. ISSN : 0028-0836, 1476-4687.

[128] David D. Lewis et Jason Catlett. « Heterogeneous Uncertainty Sampling for Supervised Learning. » In : *ICML*. Sous la dir. de William W. Cohen et Haym Hirsh. Morgan Kaufmann, 1994, p. 148-156. ISBN : 1-55860-335-2.

[129] David D. Lewis et William A. Gale. « A Sequential Algorithm for Training Text Classifiers ». In : *SIGIR '94*. Sous la dir. de Bruce W. Croft et C. J. van Rijsbergen. London : Springer London, 1994, p. 3-12. ISBN : 978-1-4471-2099-5.

[130] X. Li, L. Wang et E. Sung. « Multilabel SVM active learning for image classification ». In : *2004 International Conference on Image Processing,2004. ICIP '04.* T. 4. 2004, 2207-2210 Vol. 4.

[131] Xin Li et Yuhong Guo. « Active Learning with Multi-Label SVM Classification ». In : *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI '13. Beijing, China : AAAI Press, 2013, 1479–1485. ISBN : 9781577356332.

[132] Zhizhong Li et Derek Hoiem. « Learning without Forgetting ». In : *arXiv :1606.09282 [cs, stat]* (fév. 2017). arXiv : 1606.09282.

[133] Yaojin Lin, Qinghua Hu, Jinghua Liu, Xingquan Zhu et Xindong Wu. « MULFE : Multi-Label Learning via Label-Specific Feature Space Ensemble ». en. In : *ACM Transactions on Knowledge Discovery from Data* 16.1 (fév. 2022), p. 1-24. ISSN : 1556-4681, 1556-472X.

[134] Marco Lippi, Przemyslaw Palka, Giuseppe Contissa, Francesca Lagioia, Hans-Wolfgang Micklitz, Giovanni Sartor et Paolo Torroni. « CLAUDETTE : an Automated Detector of Potentially Unfair Clauses in Online Terms of Service ». In : *CoRR* abs/1805.01217 (2018). arXiv : 1805.01217.

[135] Bin Liu, Konstantinos Blekas et Grigorios Tsoumakas. « Multi-label sampling based on local label imbalance ». In : *Pattern Recognition* 122 (fév. 2022), p. 108294. ISSN : 0031-3203.

[136] Bing Liu. « Learning on the Job : Online Lifelong and Continual Learning ». en. In : *Proceedings of the AAAI Conference on Artificial Intelligence* 34.09 (avr. 2020). Number : 09, p. 13544-13549. ISSN : 2374-3468.

[137] Ming Liu, Wray Buntine et Gholamreza Haffari. « Learning How to Actively Learn : A Deep Imitation Learning Approach ». In : *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*. Melbourne, Australia : Association for Computational Linguistics, juill. 2018, p. 1874-1883.

[138] W. Liu, H. Wang, X. Shen et I. Tsang. « The Emerging Trends of Multi-Label Learning ». In : *IEEE Transactions on Pattern Analysis & Machine Intelligence* 01 (2021), p. 1-1. issn : 1939-3539.

[139] Yang Liu, Qince Li, Kuanquan Wang, Jun Liu, Runnan He, Yongfeng Yuan et Henggui Zhang. « Automatic Multi-Label ECG Classification with Category Imbalance and Cost-Sensitive Thresholding ». en. In : *Biosensors* 11.11 (nov. 2021). Number : 11 Publisher : Multidisciplinary Digital Publishing Institute, p. 453. issn : 2079-6374.

[140] Yiheng Liu et al. « Summary of ChatGPT-Related Research and Perspective Towards the Future of Large Language Models ». In : *Meta-Radiology* 1.2 (sept. 2023). arXiv :2304.01852 [cs], p. 100017. issn : 29501628.

[141] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer et Veselin Stoyanov. *RoBERTa : A Robustly Optimized BERT Pretraining Approach*. 2019.

[142] Zhifeng Liu, Chuanjing Tang, Stanley Ebhohimhen Abhadiomhen, Xiang-Jun Shen et Yangyang Li. « Robust Label and Feature Space Co-Learning for Multi-Label Classification ». In : *IEEE Transactions on Knowledge and Data Engineering* 35.11 (nov. 2023). Conference Name : IEEE Transactions on Knowledge and Data Engineering, p. 11846-11859. issn : 1558-2191.

[143] Vincenzo Lomonaco. *Continual Learning for Production Systems*. en. Août 2019.

[144] Jesus Lovon-Melgarejo, Laure Soulier, Karen Pinel-Sauvagnat et Lynda Tamine. « Studying Catastrophic Forgetting in Neural Ranking Models ». In : *arXiv :2101.06984 [cs]* (jan. 2021). arXiv : 2101.06984.

[145] Jinghui Lu et Brian MacNamee. « Investigating the Effectiveness of Representations Based on Pretrained Transformer-based Language Models in Active Learning for Labelling Text Datasets ». In : *CoRR* abs/2004.13138 (2020). arXiv : 2004.13138.

[146] Xiaoming Lv, Fajie Duan, Jia-Jia Jiang, Xiao Fu et Lin Gan. « Deep Active Learning for Surface Defect Detection ». en. In : *Sensors* 20.6 (jan. 2020). Number : 6 Publisher : Multidisciplinary Digital Publishing Institute, p. 1650. issn : 1424-8220.

[147] Nattaya Mairittha, Tittaya Mairittha et Sozo Inoue. « Optimizing activity data collection with gamification points using uncertainty based active learning ». In : *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*. UbiComp/ISWC '19 Adjunct. New York, NY, USA : Association for Computing Machinery, sept. 2019, p. 761-767. ISBN : 978-1-4503-6869-8.

[148] Jaya Krishna Mandivarapu, Blake Camp et Rolando Estrada. « Deep Active Learning via Open-Set Recognition ». In : *Frontiers in Artificial Intelligence* 5 (2022). ISSN : 2624-8212.

[149] Zhuoyuan Mao et Tetsuji Nakagawa. « LEALLA : Learning Lightweight Language-agnostic Sentence Embeddings with Knowledge Distillation ». In : *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Sous la dir. d'Andreas Vlachos et Isabelle Augenstein. Dubrovnik, Croatia : Association for Computational Linguistics, mai 2023, p. 1886-1894.

[150] Katerina Margatina, Giorgos Vernikos, Loïc Barrault et Nikolaos Aletras. « Active Learning by Acquiring Contrastive Examples ». In : *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online et Punta Cana, Dominican Republic : Association for Computational Linguistics, nov. 2021, p. 650-663.

[151] Andrew McCallum et Kamal Nigam. « Employing EM and Pool-Based Active Learning for Text Classification ». In : *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML '98. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1998, 350–358. ISBN : 1558605568.

[152] Warren S. McCulloch et Walter Pitts. « A logical calculus of the ideas immanent in nervous activity ». en. In : *The bulletin of mathematical biophysics* 5.4 (déc. 1943), p. 115-133. ISSN : 1522-9602.

[153] Prem Melville et Raymond J. Mooney. « Diverse ensembles for active learning ». en. In : *Twenty-first international conference on Machine learning - ICML '04*. Banff, Alberta, Canada : ACM Press, 2004, p. 74.

[154] Peter Mills. *Solving for multi-class : a survey and synthesis*. arXiv :1809.05929 [cs, stat]. Jan. 2021.

[155] Fan Min, Fu-Lun Liu, Liu-Ying Wen et Zhi-Heng Zhang. « Tri-partition cost-sensitive active learning through kNN ». en. In : *Soft Computing* 23.5 (mars 2019), p. 1557-1572. ISSN : 1433-7479.

[156] Fan MIN, Shi-Ming ZHANG, Davide CIUCCI et Min WANG. « Three-way active learning through clustering selection ». en. In : *International Journal of Machine Learning and Cybernetics* 11.5 (mai 2020), p. 1033-1046. ISSN : 1868-8071, 1868-808X.

[157] Seyed Abolghasem MIRROSHANDEL et Alexis NASR. « Active Learning for Dependency Parsing Using Partially Annotated Sentences ». In : *Proceedings of the 12th International Conference on Parsing Technologies*. Dublin, Ireland : Association for Computational Linguistics, oct. 2011, p. 140-149.

[158] T.M. MITCHELL. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997. ISBN : 9780071154673.

[159] Robert MONARCH et Christopher D. MANNING. *Human-in-the-Loop Machine Learning : Active Learning and Annotation for Human-Centered AI*. en. Sherlter Island, NY. ISBN : 978-1-61729-674-1.

[160] Ion MUSLEA, Steven MINTON et Craig A. KNOBLOCK. « Selective Sampling with Redundant Views ». In : juill. 2000.

[161] Felipe Kenji NAKANO, Ricardo CERRI et Celine VENS. *Active Learning for Hierarchical Multi-Label Classification*. eng. 2020-07-17.

[162] Jinseok NAM, Jungi KIM, Eneldo Loza MENCÍA, Iryna GUREVYCH et Johannes FÜRNKRANZ. « Large-Scale Multi-label Text Classification - Revisiting Neural Networks ». In : *ArXiv* abs/1312.5419 (2014).

[163] Hieu T. NGUYEN et Arnold SMEULDERS. « Active Learning Using Pre-Clustering ». In : *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada : Association for Computing Machinery, 2004, p. 79. ISBN : 1581138385.

[164] Vu-Linh NGUYEN, Sébastien DESTERCKE et Eyke HÜLLERMEIER. *Epistemic Uncertainty Sampling*. arXiv :1909.00218 [cs, stat]. Août 2019.

[165] Vu-Linh NGUYEN, Mohammad Hossein SHAKER et Eyke HÜLLERMEIER. « How to measure uncertainty in uncertainty sampling for active learning ». en. In : *Machine Learning* 111.1 (jan. 2022), p. 89-122. ISSN : 1573-0565.

[166] OPENAI. *GPT-4 Technical Report*. arXiv :2303.08774 [cs]. Mars 2023.

[167] German I. PARISI, Ronald KEMKER, Jose L. PART, Christopher KANAN et Stefan WERMTER. « Continual Lifelong Learning with Neural Networks : A Review ». In : *Neural Networks* 113 (mai 2019). arXiv : 1802.07569, p. 54-71. ISSN : 08936080.

[168] Fabian Pedregosa et al. « Scikit-learn : Machine Learning in Python ». In : *Journal of Machine Learning Research* 12.85 (2011), p. 2825-2830. ISSN : 1533-7928.

[169] Ameya Prabhu, Charles Dognin et Maneesh Singh. « Sampling Bias in Deep Active Classification : An Empirical Study ». 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)*. Hong Kong, China : Association for Computational Linguistics, nov. 2019, p. 4058-4068.

[170] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen et Ilya Sutskever. *Zero-Shot Text-to-Image Generation*. arXiv :2102.12092 [cs]. Fév. 2021.

[171] Jesse Read, Albert Bifet, Geoff Holmes et Bernhard Pfahringer. « Scalable and efficient multi-label classification for evolving data streams ». en. In : *Machine Learning* 88.1 (juill. 2012), p. 243-272. ISSN : 1573-0565.

[172] Jesse Read, Bernhard Pfahringer et Geoff Holmes. « Multi-label Classification Using Ensembles of Pruned Sets ». en. In : *2008 Eighth IEEE International Conference on Data Mining*. Pisa, Italy : IEEE, déc. 2008, p. 995-1000. ISBN : 978-0-7695-3502-9.

[173] Nils Reimers et Iryna Gurevych. « Sentence-BERT : Sentence Embeddings using Siamese BERT-Networks ». In : *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, nov. 2019.

[174] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen et Xin Wang. « A Survey of Deep Active Learning ». In : *arXiv :2009.00236 [cs, stat]* (août 2020). arXiv : 2009.00236.

[175] Simiao Ren, Yang Deng, Willie J. Padilla et Jordan Malof. *Towards Robust Deep Active Learning for Scientific Computing*. arXiv :2201.12632 [cs] version : 2. Oct. 2022.

[176] Oscar Reyes, Carlos Morell et Sebastián Ventura. « Effective active learning strategy for multi-label learning ». In : *Neurocomputing* 273 (2018), p. 494-508. ISSN : 0925-2312.

[177] Oscar Reyes et Sebastián Ventura. « Evolutionary Strategy to Perform Batch-Mode Active Learning on Multi-Label Data ». In : *ACM Transactions on Intelligent Systems and Technology* 9.4 (jan. 2018), 46 :1-46 :26. ISSN : 2157-6904.

[178] Mohammad Reza Rezaei-Dastjerdehei, Amirmohammad Mijani et Emad Fatemizadeh. « Addressing Imbalance in Multi-Label Classification Using Weighted Cross Entropy Loss Function ». In : *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*. Nov. 2020, p. 333-338.

[179] Pedro L. Rodriguez et Arthur Spirling. « Word Embeddings : What Works, What Doesn't, and How to Tell the Difference for Applied Research ». In : *The Journal of Politics* 84.1 (jan. 2022). Publisher : The University of Chicago Press, p. 101-115. issn : 0022-3816.

[180] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser et Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. arXiv :2112.10752 [cs]. Avr. 2022.

[181] Nicholas Roy et Andrew McCallum. « Toward Optimal Active Learning through Sampling Estimation of Error Reduction ». In : *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2001, 441–448. isbn : 1558607781.

[182] David E. Rumelhart, Geoffrey E. Hinton et Ronald J. Williams. « Learning representations by back-propagating errors ». en. In : *Nature* 323.6088 (oct. 1986). Number : 6088 Publisher : Nature Publishing Group, p. 533-536. issn : 1476-4687.

[183] Payel Sadhukhan et Sarbani Palit. « Reverse-nearest neighborhood based oversampling for imbalanced, multi-label datasets ». In : *Pattern Recognition Letters* 125 (juill. 2019), p. 813-820. issn : 0167-8655.

[184] Doyen Sahoo, Quang Pham, Jing Lu et Steven C. H. Hoi. *Online Deep Learning : Learning Deep Neural Networks on the Fly*. arXiv :1711.03705 [cs]. Nov. 2017.

[185] Bukola Salami, Keijo Haataja et Pekka Toivanen. « State-of-the-Art Techniques in Artificial Intelligence for Continual Learning : A review ». en. In : (2021), p. 10.

[186] Marcos Salganicoff, Lyle H. Ungar et Ruzena Bajcsy. « Active Learning for Vision-Based Robot Grasping ». en. In : *Machine Learning* 23.2 (mai 1996), p. 251-278. issn : 1573-0565.

[187] Victor Sanh, Lysandre Debut, Julien Chaumond et Thomas Wolf. « DistilBERT, a distilled version of BERT : smaller, faster, cheaper and lighter ». In : (oct. 2019).

[188] Victor Sanh, Lysandre Debut, Julien Chaumond et Thomas Wolf. « DistilBERT, a distilled version of BERT : smaller, faster, cheaper and lighter ». In : *ArXiv* abs/1910.01108 (2019).

[189] Tobias Scheffer, Christian Decomain et Stefan Wrobel. « Active Hidden Markov Models for Information Extraction ». In : *Advances in Intelligent Data Analysis*. Sous la dir. de Frank Hoffmann, David J. Hand, Niall Adams, Douglas Fisher et Gabriela Guimaraes. Berlin, Heidelberg : Springer Berlin Heidelberg, 2001, p. 309-318. isbn : 978-3-540-44816-7.

[190] Andrew Schein et Lyle Ungar. « Active Learning for Logistic Regression : An Evaluation ». In : *Machine Learning* 68 (août 2007), p. 235-265.

[191] Felix Schoeller, Mark Miller, Roy Salomon et Karl J. Friston. « Trust as Extended Control : Human-Machine Interactions as Active Inference ». In : *Frontiers in Systems Neuroscience* 15 (2021). issn : 1662-5137.

[192] Christopher Schröder et Andreas Niekler. « A Survey of Active Learning for Text Classification using Deep Neural Networks ». In : *CoRR* abs/2008.07267 (2020). arXiv : 2008.07267.

[193] Christopher Schröder, Andreas Niekler et Martin Potthast. « Revisiting Uncertainty-based Query Strategies for Active Learning with Transformers ». In : *Findings of the Association for Computational Linguistics : ACL 2022*. Dublin, Ireland : Association for Computational Linguistics, mai 2022, p. 2194-2203.

[194] Christopher Schröder, Lydia Müller, Andreas Niekler et Martin Potthast. *Small-Text : Active Learning for Text Classification in Python*. 2021. arXiv : 2107.10314 [cs.LG].

[195] Christopher Schröder, Andreas Niekler et Martin Potthast. « Revisiting Uncertainty-based Query Strategies for Active Learning with Transformers ». In : *Findings of the Association for Computational Linguistics : ACL 2022*. Sous la dir. de Smaranda Muresan, Preslav Nakov et Aline Villavicencio. Dublin, Ireland : Association for Computational Linguistics, mai 2022, p. 2194-2203.

[196] Raphael Schumann et Ines Rehbein. « Active Learning via Membership Query Synthesis for Semi-Supervised Sentence Classification ». en. In : *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China : Association for Computational Linguistics, 2019, p. 472-481.

[197] Roy Schwartz, Jesse Dodge, Noah A. Smith et Oren Etzioni. « Green AI ». In : *arXiv :1907.10597 [cs, stat]* (août 2019). arXiv : 1907.10597.

[198] Ozan Sener et Silvio Savarese. *Active Learning for Convolutional Neural Networks : A Core-Set Approach*. arXiv :1708.00489 [cs, stat]. Juin 2018.

[199] Burr SETTLES. « Active learning literature survey ». In : *Technical Report TR-1648. University of Wisconsin-Madison. Department of Computer Sciences* (2009).

[200] Burr SETTLES, M. CRAVEN et Lewis A. FRIEDLAND. « Active Learning with Real Annotation Costs ». In : 2008.

[201] Burr SETTLES et Mark CRAVEN. « An Analysis of Active Learning Strategies for Sequence Labeling Tasks ». In : *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii : Association for Computational Linguistics, oct. 2008, p. 1070-1079.

[202] Burr SETTLES, Mark CRAVEN et Lewis FRIEDLAND. « Active Learning with Real Annotation Costs ». en. In : ().

[203] H. S. SEUNG, M. OPPER et H. SOMPOLINSKY. « Query by Committee ». In : *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA : Association for Computing Machinery, 1992, 287–294. ISBN : 089791497X.

[204] C. E. SHANNON. « A Mathematical Theory of Communication ». In : *Bell System Technical Journal* 27.3 (1948), p. 379-423. eprint : https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x.

[205] Dan SHEN, Jie ZHANG, Jian SU, Guodong ZHOU et Chew-Lim TAN. « Multi-Criteria-based Active Learning for Named Entity Recognition ». In : *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. Barcelona, Spain, juill. 2004, p. 589-596.

[206] Shirong SHEN, Zhen LI et Guilin QI. « Active Learning for Event Extraction with Memory-based Loss Prediction Model ». In : *ArXiv* abs/2112.03073 (2021).

[207] Yanyao SHEN, Hyokun YUN, Zachary LIPTON, Yakov KRONROD et Animashree ANANDKUMAR. « Deep Active Learning for Named Entity Recognition ». In : *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Vancouver, Canada : Association for Computational Linguistics, août 2017, p. 252-256.

[208] Chuan SHI, Xiangnan KONG, Philip S. YU et Bai WANG. « Multi-label Ensemble Learning ». In : *Machine Learning and Knowledge Discovery in Databases*. Sous la dir. de Dimitrios GUNOPULOS, Thomas HOFMANN, Donato MALERBA et Michalis VAZIRGIANNIS. Berlin, Heidelberg : Springer Berlin Heidelberg, 2011, p. 223-239. ISBN : 978-3-642-23808-6.

[209] Heereen SHIM, Stijn LUCA, Dietwig LOWET et Bart VANRUMSTE. « Data augmentation and semi-supervised learning for deep neural networks-based text classifier ». en. In : *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. Brno Czech Republic : ACM, mars 2020, p. 1119-1126. ISBN : 978-1-4503-6866-7.

[210] Aditya SIDDHANT et Zachary C. LIPTON. « Deep Bayesian Active Learning for Natural Language Processing : Results of a Large-Scale Empirical Study ». In : *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium : Association for Computational Linguistics, oct. 2018, p. 2904-2909.

[211] Samarth SINHA, Sayna EBRAHIMI et Trevor DARRELL. « Variational Adversarial Active Learning ». In : *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), p. 5971-5980.

[212] Shagun SODHANI, Sarath CHANDAR et Yoshua BENGIO. « Towards Training Recurrent Neural Networks for Lifelong Learning ». In : *arXiv :1811.07017 [cs, stat]* (sept. 2019). arXiv : 1811.07017.

[213] Jamshid SOURATI, Murat AKCAKAYA, Todd K LEEN, Deniz ERDOGMUS et Jennifer G DY. « Asymptotic Analysis of Objectives Based on Fisher Information in Active Learning ». en. In : ().

[214] Petru SOVIANY, Radu Tudor IONESCU, Paolo ROTA et Nicu SEBE. « Curriculum Learning : A Survey ». en. In : *International Journal of Computer Vision* 130.6 (juin 2022), p. 1526-1565. ISSN : 1573-1405.

[215] Nitish SRIVASTAVA, Geoffrey HINTON, Alex KRIZHEVSKY, Ilya SUTSKEVER et Ruslan SALAKHUTDINOV. « Dropout : A Simple Way to Prevent Neural Networks from Overfitting ». In : *Journal of Machine Learning Research* 15.56 (2014), p. 1929-1958. ISSN : 1533-7928.

[216] Saurabh SRIVASTAVA. « Machine Learning : A Review on Binary Classification ». In : *International Journal of Computer Applications* 160 (fév. 2017), p. 11-15.

[217] Yu-Yin SUN, Yin ZHANG et Zhi-Hua ZHOU. « Multi-Label Learning with Weak Label ». en. In : *Proceedings of the AAAI Conference on Artificial Intelligence* 24.1 (juill. 2010). Number : 1, p. 593-598. ISSN : 2374-3468.

[218] Adil Yaseen TAHA, Sabrina TIUN, Abdul Hadi ABD RAHMAN et Ali SABAH. « MULTILABEL OVER-SAMPLING AND UNDER-SAMPLING WITH CLASS ALIGNMENT FOR IMBALANCED MULTILABEL TEXT CLASSIFICATION ». en. In : *Journal of Information and Communication Technology* 20 (juin 2021). ISSN : 1675-414X.

[219] Min TANG, Xiaoqiang LUO et Salim ROUKOS. « Active Learning for Statistical Natural Language Parsing ». In : *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA : Association for Computational Linguistics, juill. 2002, p. 120-127.

[220] Raphael TANG, Yao LU, Linqing LIU, Lili MOU, Olga VECHTOMOVA et Jimmy LIN. « Distilling Task-Specific Knowledge from BERT into Simple Neural Networks ». In : *arXiv :1903.12136 [cs]* (mars 2019). arXiv : 1903.12136.

[221] Ying-Peng TANG et Sheng-Jun HUANG. « Self-Paced Active Learning : Query the Right Thing at the Right Time ». en. In : *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (juill. 2019). Number : 01, p. 5117-5124. ISSN : 2374-3468.

[222] Adane Nega TAREKEGN, Mario GIACOBINI et Krzysztof MICHALAK. « A review of methods for imbalanced multi-label classification ». en. In : *Pattern Recognition* 118 (oct. 2021), p. 107965. ISSN : 00313203.

[223] Cynthia A. THOMPSON, Mary Elaine CALIFF et Raymond J. MOONEY. « Active Learning for Natural Language Parsing and Information Extraction ». In : *Proceedings of the Sixteenth International Conference on Machine Learning*. ICML '99. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1999, p. 406-414. ISBN : 978-1-55860-612-8.

[224] Vaishali S. TIDAKE et Shirish S. SANE. « Multi-label Classification : a survey ». en. In : *International Journal of Engineering & Technology* 7.4.19 (nov. 2018). Number : 4.19, p. 1045-1054. ISSN : 2227-524X.

[225] Tong LUO, K. KRAMER, S. SAMSON, A. REMSEN, D.B. GOLDGOF, L.O. HALL et T. HOPKINS. « Active learning to recognize multiple types of plankton ». en. In : *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Cambridge, UK : IEEE, 2004, 478-481 Vol.3. ISBN : 978-0-7695-2128-2.

[226] Simon TONG et Daphne KOLLER. « Support Vector Machine Active Learning with Applications to Text Classification ». In : *J. Mach. Learn. Res.* 2 (2001), p. 45-66.

[227] Hugo TOUVRON et al. *LLaMA : Open and Efficient Foundation Language Models*. arXiv :2302.13971 [cs]. Fév. 2023.

[228] Toan TRAN, Thanh-Toan DO, Ian REID et Gustavo CARNEIRO. *Bayesian Generative Active Deep Learning*. arXiv :1904.11643 [cs, stat]. Avr. 2019.

[229] Grigorios Tsoumakas, Ioannis Katakis et Ioannis Vlahavas. « Mining Multi-label Data ». en. In : *Data Mining and Knowledge Discovery Handbook*. Sous la dir. d'Oded Maimon et Lior Rokach. Boston, MA : Springer US, 2010, p. 667-685. isbn : 978-0-387-09823-4.

[230] Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek et Ioannis Vlahavas. « Mulan : A Java Library for Multi-Label Learning ». In : *Journal of Machine Learning Research* 12 (2011), p. 2411-2414.

[231] Grigorios Tsoumakas et Ioannis Vlahavas. « Random k-Labelsets : An Ensemble Method for Multilabel Classification ». en. In : *Machine Learning : ECML 2007*. Sous la dir. de Joost N. Kok, Jacek Koronacki, Raomon Lopez de Mantaras, Stan Matwin, Dunja Mladenič et Andrzej Skowron. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2007, p. 406-417. isbn : 978-3-540-74958-5.

[232] Akim Tsvigun, Artem Shelmanov, Gleb Kuzmin, Leonid Sanochkin, Daniil Larionov, Gleb Gusev, Manvel Avetisian et Leonid Zhukov. « Towards Computationally Feasible Deep Active Learning ». In : *Findings of the Association for Computational Linguistics : NAACL 2022*. Seattle, United States : Association for Computational Linguistics, juill. 2022, p. 1198-1218.

[233] Akim Tsvigun, Artem Shelmanov, Gleb Kuzmin, Leonid Sanochkin, Daniil Larionov, Gleb Gusev, Manvel Avetisian et Leonid Zhukov. « Towards Computationally Feasible Deep Active Learning ». In : *Findings of the Association for Computational Linguistics : NAACL 2022*. Seattle, United States : Association for Computational Linguistics, juill. 2022, p. 1198-1218.

[234] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser et Illia Polosukhin. « Attention is All you Need ». In : *Advances in Neural Information Processing Systems*. Sous la dir. d'I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan et R. Garnett. T. 30. Curran Associates, Inc., 2017.

[235] Artem Vazhentsev et al. « Uncertainty Estimation of Transformer Predictions for Misclassification Detection ». In : *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*. Sous la dir. de Smaranda Muresan, Preslav Nakov et Aline Villavicencio. Dublin, Ireland : Association for Computational Linguistics, mai 2022, p. 8237-8252.

[236] Gido M. van de Ven et Andreas S. Tolias. « Three scenarios for continual learning ». In : *arXiv :1904.07734 [cs, stat]* (avr. 2019). arXiv : 1904.07734.

[237] Keze WANG, Dongyu ZHANG, Ya LI, Ruimao ZHANG et Liang LIN. « Cost-Effective Active Learning for Deep Image Classification ». In : *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (déc. 2017). arXiv :1701.03551 [cs], p. 2591-2600. ISSN : 1051-8215, 1558-2205.

[238] Ran WANG, Sam KWONG, Xu WANG et Yuheng JIA. « Active k-labelsets ensemble for multi-label classification ». In : *Pattern Recognition* 109 (jan. 2021), p. 107583. ISSN : 0031-3203.

[239] Xin WANG, Yudong CHEN et Wenwu ZHU. « A Survey on Curriculum Learning ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (sept. 2022). Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence, p. 4555-4576. ISSN : 1939-3539.

[240] Zheng WANG et Jieping YE. « Querying Discriminative and Representative Samples for Batch Mode Active Learning ». In : *ACM Transactions on Knowledge Discovery from Data* 9.3 (2015), 17 :1-17 :23. ISSN : 1556-4681.

[241] Zijie J. WANG, Dongjin CHOI, Shenyu XU et Diyi YANG. « Putting Humans in the Natural Language Processing Loop : A Survey ». In : *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*. Online : Association for Computational Linguistics, avr. 2021, p. 47-52.

[242] Eleanor WATSON, Thiago VIANA et Shujun ZHANG. « Augmented Behavioral Annotation Tools, with Application to Multimodal Datasets and Models : A Systematic Review ». en. In : *AI* 4.1 (jan. 2023). Number : 1 Publisher : MDPI, p. 128-171. ISSN : 2673-2688.

[243] Lukas WERTZ, Katsiaryna MIRYLENKA, Jonas KUHN et Jasmina BOGOJESKA. « Investigating Active Learning Sampling Strategies for Extreme Multi Label Text Classification ». In : *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France : European Language Resources Association, juin 2022, p. 4597-4605.

[244] David Gray WIDDER, Sarah WEST et Meredith WHITTAKER. *Open (For Business) : Big Tech, Concentrated Power, and the Political Economy of Open AI*. en. SSRN Scholarly Paper. Rochester, NY, août 2023.

[245] Baoyuan WU, Zhilei LIU, Shangfei WANG, Bao-Gang HU et Qiang JI. « Multi-label Learning with Missing Labels ». en. In : *2014 22nd International Conference on Pattern Recognition*. Stockholm, Sweden : IEEE, août 2014, p. 1964-1968. ISBN : 978-1-4799-5209-0.

[246] Fangzhao Wu, Yongfeng Huang et Jun Yan. « Active Sentiment Domain Adaptation ». In : *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*. Vancouver, Canada : Association for Computational Linguistics, juill. 2017, p. 1701-1711.

[247] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma et Liang He. « A Survey of Human-in-the-loop for Machine Learning ». In : *arXiv :2108.00941 [cs]* (août 2021). arXiv : 2108.00941.

[248] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma et Liang He. « A Survey of Human-in-the-loop for Machine Learning ». In : *ArXiv* abs/2108.00941 (2021).

[249] Zhao Xu, Kai Yu, Volker Tresp, Xiaowei Xu et Jizhi Wang. « Representative Sampling for Text Classification Using Support Vector Machines ». en. In : *Advances in Information Retrieval*. Sous la dir. de Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen et Fabrizio Sebastiani. T. 2633. Series Title : Lecture Notes in Computer Science. Berlin, Heidelberg : Springer Berlin Heidelberg, 2003, p. 393-407. isbn : 978-3-540-01274-0 978-3-540-36618-8.

[250] Zuobing Xu, Ram Akella et Yi Zhang. « Incorporating Diversity and Density in Active Learning for Relevance Feedback ». en. In : *Advances in Information Retrieval*. Sous la dir. de Giambattista Amati, Claudio Carpineto et Giovanni Romano. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2007, p. 246-257. isbn : 978-3-540-71496-5.

[251] Bishan Yang, Jian-Tao Sun, Tengjiao Wang et Zheng Chen. « Effective Multi-Label Active Learning for Text Classification ». In : *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France : Association for Computing Machinery, 2009, 917–926. isbn : 9781605584959.

[252] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang et Alexander G. Hauptmann. « Multi-Class Active Learning by Uncertainty Sampling with Diversity Maximization ». en. In : *International Journal of Computer Vision* 113.2 (juin 2015), p. 113-127. issn : 1573-1405.

[253] Changchang Yin, Buyue Qian, Shilei Cao, Xiaoyu Li, Jishang Wei, Qinghua Zheng et Ian Davidson. « Deep Similarity-Based Batch Mode Active Learning with Exploration-Exploitation ». In : *2017 IEEE International Conference on Data Mining (ICDM)*. ISSN : 2374-8486. Nov. 2017, p. 575-584.

[254] Gal YONA, Shay MORAN, Gal ELIDAN et Amir GLOBERSON. « Active learning with label comparisons ». en. In : *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. ISSN : 2640-3498. PMLR, août 2022, p. 2289-2298.

[255] Donggeun YOO et In So KWEON. « Learning Loss for Active Learning ». en. In : *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA : IEEE, juin 2019, p. 93-102. ISBN : 978-1-72813-293-8.

[256] Guoxian YU, Xia CHEN, Carlotta DOMENICONI, Jun WANG, Zhao LI, Zili ZHANG et Xiangliang ZHANG. « CMAL : Cost-Effective Multi-Label Active Learning by Querying Subexamples ». In : *IEEE Transactions on Knowledge and Data Engineering* 34.5 (mai 2022). Conference Name : IEEE Transactions on Knowledge and Data Engineering, p. 2091-2105. ISSN : 1558-2191.

[257] Yue YU, Lingkai KONG, Jieyu ZHANG, Rongzhi ZHANG et Chao ZHANG. « AcTune : Uncertainty-Based Active Self-Training for Active Fine-Tuning of Pretrained Language Models ». In : *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*. Seattle, United States : Association for Computational Linguistics, juill. 2022, p. 1422-1436.

[258] Michelle YUAN, Hsuan-Tien LIN et Jordan BOYD-GRABER. « Cold-start Active Learning through Self-supervised Language Modeling ». In : *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online : Association for Computational Linguistics, nov. 2020, p. 7935-7948.

[259] Xiangkai ZENG, Sarthak GARG, Rajen CHATTERJEE, Udhyakumar NALLASAMY et Matthias PAULIK. « Empirical Evaluation of Active Learning Techniques for Neural MT ». In : *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. Hong Kong, China : Association for Computational Linguistics, nov. 2019, p. 84-93.

[260] Chuang ZHANG, Chen GONG, Tengfei LIU, Xun LU, Weiqiang WANG et Jian YANG. « Online Positive and Unlabeled Learning ». en. In : t. 3. ISSN : 1045-0823. Juill. 2020, p. 2248-2254.

[261] Min-Ling ZHANG, Yu-Kun LI, Xu-Ying LIU et Xin GENG. « Binary relevance for multi-label learning : an overview ». en. In : *Frontiers of Computer Science* 12.2 (avr. 2018), p. 191-202. ISSN : 2095-2228, 2095-2236.

[262] Min-Ling ZHANG, Yu-Kun LI, Hao YANG et Xu-Ying LIU. « Towards Class-Imbalance Aware Multi-Label Learning ». en. In : *IEEE Transactions on Cybernetics* 52.6 (juin 2022), p. 4459-4471. ISSN : 2168-2267, 2168-2275.

[263] Min-Ling Zhang et Zhi-Jian Wang. « MIMLRBF : RBF neural networks for multi-instance multi-label learning ». In : *Neurocomputing*. Financial Engineering 72.16 (oct. 2009), p. 3951-3956. ISSN : 0925-2312.

[264] Min-Ling Zhang et Lei Wu. « Lift : Multi-Label Learning with Label-Specific Features ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.1 (jan. 2015). Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence, p. 107-120. ISSN : 1939-3539.

[265] Min-Ling Zhang et Zhi-Hua Zhou. « Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization ». In : *IEEE Transactions on Knowledge and Data Engineering* 18.10 (oct. 2006). Conference Name : IEEE Transactions on Knowledge and Data Engineering, p. 1338-1351. ISSN : 1558-2191.

[266] Min-Ling Zhang et Zhi-Hua Zhou. « A Review on Multi-Label Learning Algorithms ». en. In : *IEEE Transactions on Knowledge and Data Engineering* 26.8 (août 2014), p. 1819-1837. ISSN : 1041-4347.

[267] Shujian Zhang, Chengyue Gong, Xingchao Liu, Pengcheng He, Weizhu Chen et Mingyuan Zhou. « ALLSH : Active Learning Guided by Local Sensitivity and Hardness ». In : *Findings of the Association for Computational Linguistics : NAACL 2022*. Seattle, United States : Association for Computational Linguistics, juill. 2022, p. 1328-1342.

[268] Xiao-Yu Zhang, Haichao Shi, Xiaobin Zhu et Peng Li. « Active semi-supervised learning based on self-expressive correlation with generative adversarial networks ». In : *Neurocomputing*. Deep Learning for Intelligent Sensing, Decision-Making and Control 345 (juin 2019), p. 103-113. ISSN : 0925-2312.

[269] Ye Zhang, Matthew Lease et Byron C. Wallace. « Active Discriminative Text Representation Learning ». In : *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA : AAAI Press, 2017, 3386–3392.

[270] Zhisong Zhang, Emma Strubell et Eduard Hovy. *A Survey of Active Learning for Natural Language Processing*. arXiv :2210.10109 [cs]. Oct. 2022.

[271] Wayne Xin Zhao et al. *A Survey of Large Language Models*. arXiv :2303.18223 [cs]. Avr. 2023.

[272] Yunpeng Zhao, Mattia Prosperi, Tianchen Lyu, Yi Guo, Le Zhou et Jiang Bian. « Integrating Crowdsourcing and Active Learning for Classification of Work-Life Events from Tweets ». In : *Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices*. Sous la dir. d'Hamido Fujita, Phi-

lippe FOURNIER-VIGER, Moonis ALI et Jun SASAKI. Cham : Springer International Publishing, 2020, p. 333-344. ISBN : 978-3-030-55789-8.

[273] Fedor ZHDANOV. *Diverse mini-batch Active Learning*. arXiv :1901.05954 [cs, stat]. Jan. 2019.

[274] Jingbo ZHU, Huizhen WANG, Tianshun YAO et Benjamin K TSOU. « Active Learning with Sampling by Uncertainty and Density for Word Sense Disambiguation and Text Classification ». In : *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK : Coling 2008 Organizing Committee, août 2008, p. 1137-1144.

[275] Xiaoyan ZHU, Jiaxuan LI, Jingtao REN, Jiayin WANG et Guangtao WANG. « Dynamic ensemble learning for multi-label classification ». en. In : *Information Sciences* 623 (avr. 2023), p. 94-111. ISSN : 00200255.

[276] Xingquan ZHU, Peng ZHANG, Xiaodong LIN et Yong SHI. « Active Learning from Data Streams ». In : *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. 2007, p. 757-762.

[277] Honglei ZHUANG et Joel YOUNG. « Leveraging In-Batch Annotation Bias for Crowdsourced Active Learning ». In : *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. WSDM '15. New York, NY, USA : Association for Computing Machinery, 2015, p. 243-252. ISBN : 978-1-4503-3317-7.

[278] Indrė ŽLIOBAITĖ, Albert BIFET, Bernhard PFAHRINGER et Geoffrey HOLMES. « Active Learning With Drifting Streaming Data ». In : *IEEE Transactions on Neural Networks and Learning Systems* 25.1 (jan. 2014). Conference Name : IEEE Transactions on Neural Networks and Learning Systems, p. 27-39. ISSN : 2162-2388.

Quatrième partie

ANNEXES

A

ANNEXES

A.1 BRÈVES DÉFINITIONS DE RÉSEAUX NEURONAUX

Dans cette section, nous explorerons les architectures neuronales les plus couramment employées dans les travaux d'apprentissage actif sur les réseaux de neurones. L'idée fondamentale derrière les réseaux neuronaux est de simuler le fonctionnement des neurones biologiques en utilisant des unités de traitement interconnectées, appelées "neurones artificiels" ou "unités neuronales". Ces neurones sont organisés en couches, créant ainsi une structure en réseau. Chaque neurone est connecté à un ou plusieurs autres neurones, et ces connexions ont des poids qui déterminent l'importance des signaux transmis entre les neurones.

Un **perceptron** est un modèle mathématique simple qui prend en entrée un vecteur de données, effectue des opérations de calcul sur ces données, et produit une sortie. Sa structure imagée dans la figure 26 se compose de trois éléments principaux :

1. Les **entrées** : Les données d'entrée sont représentées sous forme de valeurs numériques. Ces entrées peuvent provenir de diverses sources, telles que des capteurs, des images ou d'une extraction de caractéristiques textuelles.
2. Les **poids** : Chaque entrée est associée à un poids qui lui est spécifique. Les poids sont des paramètres ajustables qui permettent au perceptron d'attribuer une importance différente à chaque entrée. Ils servent à moduler l'influence des entrées sur la sortie.
3. La **fonction d'activation** : La fonction d'activation est une fonction mathématique qui prend en compte la somme pondérée des entrées multipliées par leurs poids. Elle produit la sortie finale du perceptron. Cette fonction peut introduire une notion de non-linéarité dans le modèle. Plusieurs fonctions d'activation courantes sont utilisées, telles que la fonction sigmoïde, la fonction ReLU (Rectified Linear Unit), et la fonction tangente hyperbolique.

Le fonctionnement d'un perceptron peut être décomposé en quelques étapes simples : d'abord les données d'entrée sont multipliées par leurs poids respectifs ; les produits pondérés sont sommés entre eux (une constante représentant généralement un biais peut être ajoutée à ce moment) ; cette somme est ensuite passée à travers la fonction d'activation qui génère la sortie du perceptron.

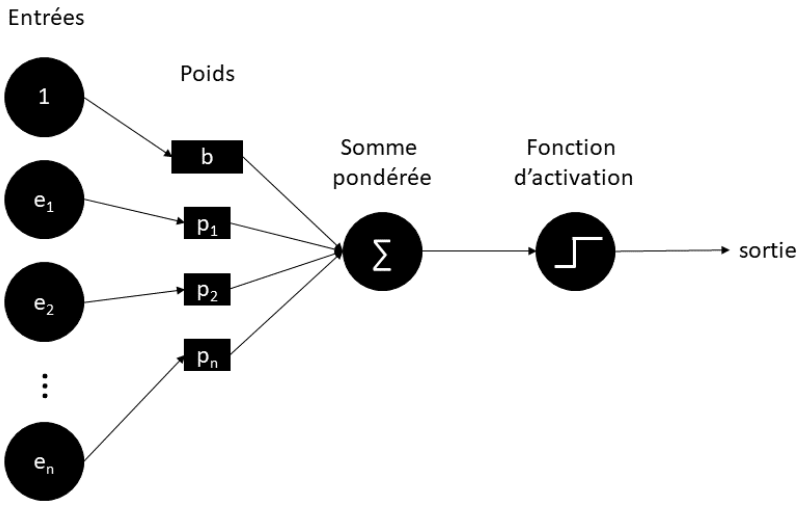


FIGURE 26 – Schéma d'un perceptron.

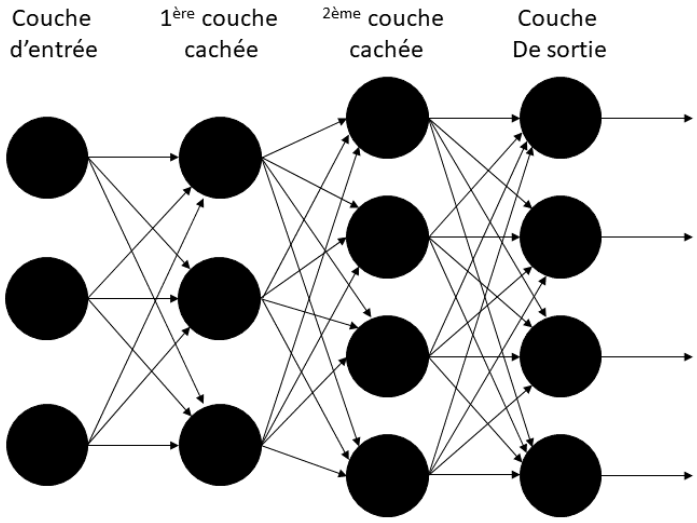


FIGURE 27 – Schéma d'un perceptron à multiple couches.

Les perceptrons sont souvent utilisés pour des tâches de classification binaire, où l'objectif est de séparer les données en deux catégories distinctes. Cependant, un seul perceptron est limité dans sa capacité à résoudre des problèmes complexes. Pour aborder des problèmes plus complexes, des réseaux de neurones multicouches sont construits en empilant de multiples perceptrons et en utilisant des fonctions d'activation non linéaires entre les couches.

Les **perceptrons à multiples couches**, *multilayer perceptron* [86], sont composé de plusieurs couches d'unités de traitement inter-connectées comme on peut le voir dans la Figure 27. Ils comprennent généralement trois types de couches :

1. **Couche d'entrée** : La première couche est la couche d'entrée, où les données sont introduites dans le réseau. Chaque unité dans cette couche correspond à une caractéristique ou une dimension

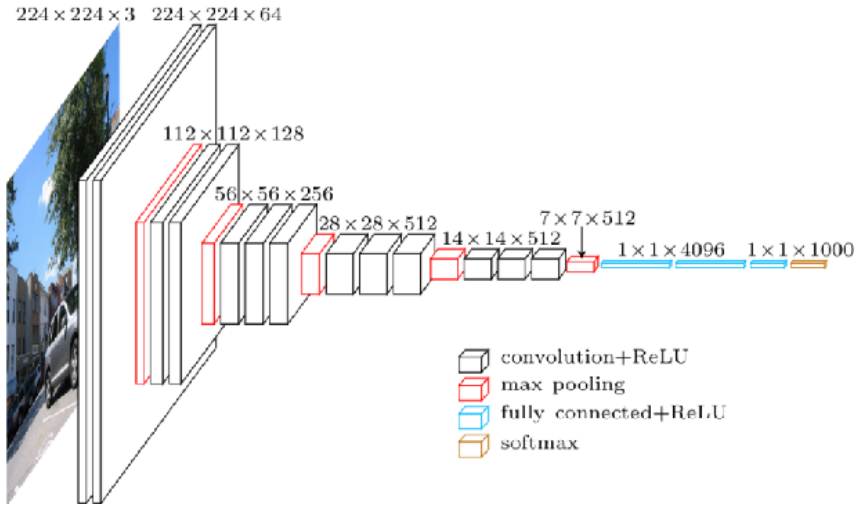


FIGURE 28 – Exemple d'un réseau neuronal convolutif [66].

des données d'entrée. Les valeurs de ces unités sont directement associées aux données fournies.

2. **Couches cachées** : Les couches cachées sont placées entre la couche d'entrée et la couche de sortie. Ces couches effectuent des opérations de calcul complexes en interne et transmettent les résultats aux couches suivantes. Le nombre de couches cachées et le nombre d'unités dans chaque couche cachée peuvent varier en fonction de l'architecture du réseau.
3. **Couche de sortie** : La dernière couche est la couche de sortie, où le réseau génère les prédictions ou les résultats. Le nombre d'unités dans cette couche dépend de la nature de la tâche, comme la classification, la régression ou d'autres types de prédictions.

Les données d'entrée sont propagées à travers le réseau, couche par couche, en effectuant des calculs à chaque couche. Chaque unité dans une couche cachée effectue une somme pondérée des valeurs de la couche précédente, puis applique une fonction d'activation non linéaire. Cette opération est répétée jusqu'à atteindre la couche de sortie. Une fois que les données ont atteint la couche de sortie, l'erreur entre les prédictions du réseau et les valeurs attendues est calculée. Cette erreur est utilisée pour évaluer la performance du réseau. L'erreur est propagée en sens inverse à travers le réseau pour ajuster les poids et les biais à chaque couche. Cette étape d'apprentissage permet au réseau de s'ajuster progressivement pour améliorer ses prédictions. Différentes techniques d'optimisation, telles que la descente de gradient, sont couramment utilisées pour répéter de manière itérative ce processus ou en d'autres termes, entraîner le modèle.

Un **réseau neuronal convolutif** (CNN) [126], est constitué d'un empilement de couches spéciales conçues pour effectuer deux opérations principales : la convolution et le sous-échantillonnage (ou *pooling*).

Ces couches peuvent être en trois dimensions ce qui les rend particulièrement efficaces pour le traitement de l'image (image en deux dimensions + la couleur). Ces couches sont généralement suivies de couches de neurones "classiques" entièrement connectées, comme on peut le voir dans la Figure 28. Les CNN comprennent donc généralement trois types de couches :

1. **Couches de convolution** : Les couches de convolution sont responsables de l'extraction des caractéristiques des données d'entrée. Elles utilisent des filtres (kernels) qui se déplacent sur l'image d'entrée pour effectuer des opérations de convolution. Chaque filtre identifie des motifs spécifiques dans l'image en effectuant des produits scalaires locaux. Les opérations de convolution sont effectuées de manière itérative, permettant au CNN de capturer des caractéristiques de plus en plus complexes à mesure que l'on avance dans les couches.
2. **Couches de sous-échantillonnage** : Après les couches de convolution, des couches de sous-échantillonnage réduisent la dimension spatiale des caractéristiques extraites, tout en préservant leur pertinence. Cette opération de sous-échantillonnage consiste généralement en une moyenne ou une prise du maximum sur des régions locales de l'image, réduisant ainsi le nombre de valeurs à traiter dans les couches suivantes.
3. **Couches connectées** : Les caractéristiques extraites par les couches précédentes sont aplaties et passées à travers des couches "classiques" entièrement connectées. Ces couches jouent un rôle clé dans la classification finale ou dans d'autres tâches spécifiques. Elles apprennent à associer les caractéristiques extraites aux labels de classe ou aux prédictions souhaitées.

Comme pour d'autres types de réseaux neuronaux, le CNN est entraîné à l'aide de données étiquetées par rétropropagation de l'erreur, où l'erreur entre les prédictions du modèle et les étiquettes réelles est minimisée. Les filtres et les paramètres du modèle sont ajustés au fur et à mesure de l'entraînement pour améliorer les performances. Les CNN sont particulièrement efficaces dans le traitement d'image.

Un **réseau de neurones récurrents** (RNN) [182], est un modèle d'apprentissage automatique spécialement conçu pour traiter des données séquentielles ou temporelles. Contrairement aux réseaux de neurones classiques, les RNN possèdent une structure interne qui leur permet de prendre en compte les dépendances temporelles et contextuelles dans les données. Il est composé d'unités récurrentes, également appelées cellules, disposées en une séquence d'étapes temporelles. Chaque unité reçoit des données d'entrée, effectue des calculs en utilisant ses propres paramètres, et produit une sortie à l'étape courante. De plus, chaque unité maintient une mémoire interne, appelée "état caché", qui est mise à jour à chaque étape temporelle en fonc-

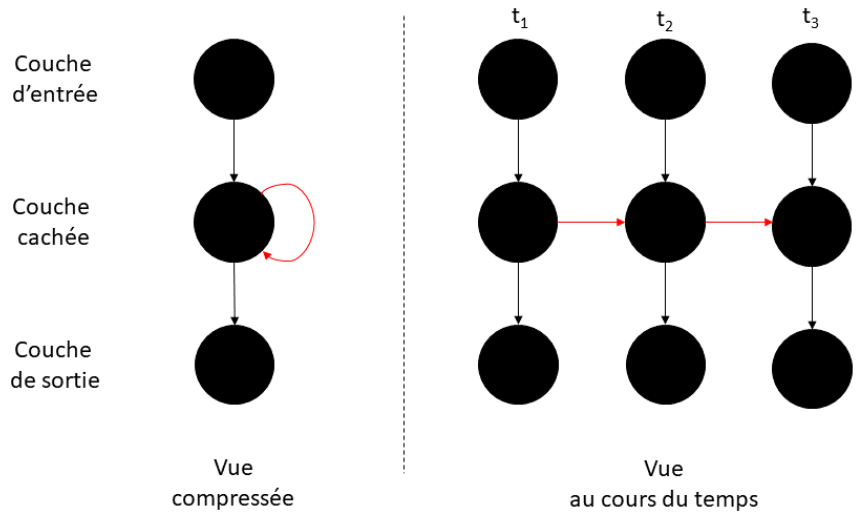


FIGURE 29 – Schéma de deux vues "de coté" d'un réseau de neurones récurrents.

tion de l'entrée actuelle, de l'état caché précédent et des paramètres du modèle comme on peut le voir dans la Figure 29.

Les données séquentielles sont introduites dans le RNN une à une. À chaque étape, l'unité récurrente traite l'entrée en utilisant ses paramètres, et l'état caché précédent pour produire une sortie. Cette sortie est alors également renvoyée comme nouvel état caché pour la prochaine étape. Le RNN est capable de capturer les dépendances temporelles dans les données grâce à la mise à jour récurrente de l'état caché. Cela signifie que le réseau peut prendre en compte les informations passées pour influencer sa sortie actuelle. L'état caché d'une unité récurrente agit comme une forme de mémoire du réseau. Il peut stocker des informations pertinentes pour le contexte actuel de la séquence. Comme pour les modèles précédents, lors de l'apprentissage, une erreur est calculée entre la sortie prédite par le réseau et la sortie attendue. Cette erreur est propagée en sens inverse à travers les étapes temporelles du RNN pour ajuster les paramètres du modèle et améliorer la précision des prédictions. Les RNN sont particulièrement efficaces dans le traitement de séquence de texte.

Un réseau neuronal récurrent ***Long Short-Term Memory*** (LSTM) est une variante des RNN, qui a été développée pour surmonter les limitations des RNN traditionnels, et traiter des séquences de données en conservant des informations à long terme, ce qui les rend aptes à capturer des dépendances temporelles complexes. Contrairement aux RNN traditionnels, qui peuvent souffrir du problème du "gradient qui explose" ou du "gradient qui disparaît" lors de la rétropropagation de l'erreur, les LSTM utilisent des mécanismes spécifiques pour éviter ces problèmes.

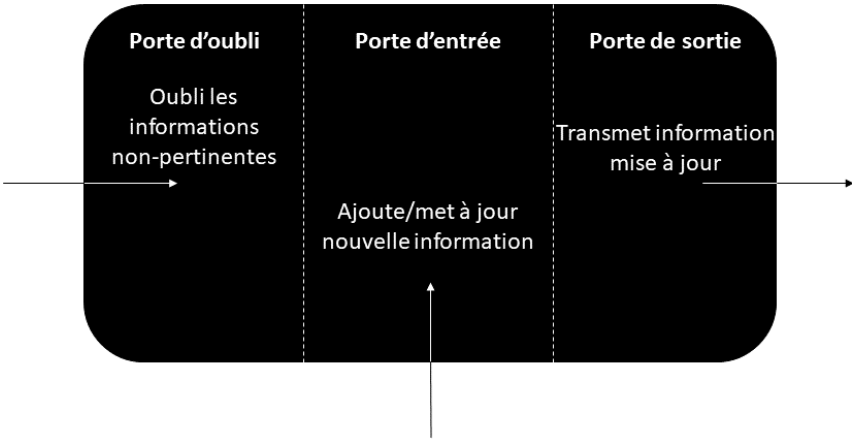


FIGURE 30 – Schéma d'une cellume *long short-term memory*.

Un LSTM est composé de cellules spécifiques comme on peut le voir dans la Figure 30, chaque cellule comportant trois portes principales :

1. **Porte d'oubli** : Cette porte détermine quelles informations précédemment stockées dans la cellule doivent être oubliées ou maintenues. Elle prend en compte l'entrée actuelle et l'état caché précédent pour effectuer cette opération.
2. **Porte d'entrée** : Cette porte permet de décider quelles nouvelles informations de l'entrée actuelle doivent être incluses dans la mémoire de la cellule. Elle prend également en compte l'entrée actuelle et l'état caché précédent.
3. **Porte de sortie** : Cette porte contrôle la sortie de la cellule en se basant sur l'entrée actuelle, l'état caché précédent et les informations mémorisées dans la cellule

Le réseau prend en entrée une séquence de données, que ce soit du texte, de l'audio, des valeurs de capteurs, ou tout autre type de séquence temporelle. Cette entrée de la séquence est transmise à travers les différentes cellules LSTM en respectant l'ordre séquentiel. Chaque cellule effectue des calculs pour gérer les informations à court et à long terme. Après avoir parcouru la séquence, le réseau est entraîné en comparant ses prédictions à la sortie attendue. La rétropropagation de l'erreur permet d'ajuster les poids et les paramètres du modèle pour minimiser l'erreur. Une fois entraîné, le réseau peut être utilisé pour effectuer des prédictions sur de nouvelles séquences, en utilisant la mémoire à long terme qu'il a apprise au cours de l'entraînement.

A.2 DÉFINITION DES TRANSFORMERS

Avant de discuter des transformers, il est important de définir rapidement ce qu'est un plongement lexical ou *embedding* en anglais. Les

plongements lexicaux sont des représentations numériques de mots ou de phrases dans un espace vectoriel continu. Ces représentations sont apprises à partir de données textuelles et captent des relations sémantiques entre les mots. Les plongements lexicaux sont utilisés dans le traitement du langage naturel pour transformer le texte en vecteurs numériques, facilitant ainsi la compréhension et le traitement automatique du langage par les modèles informatiques [179].

Les transformers sont une classe de modèles d'apprentissage automatique introduite en 2017 par Vaswani et al.[234]. Ces modèles ont révolutionné le domaine du traitement du langage naturel et de l'apprentissage profond. Les exemples imagés de cette partie sont inspirés de l'article *The Illustrated Transformer* [5].

Le fonctionnement des transformers repose sur l'idée d'attention, permettant au modèle de se concentrer sur des parties spécifiques de la séquence d'entrée pendant le traitement. Chaque couche d'un transformer a deux principales composantes : le mécanisme d'auto-attention et les réseaux de neurones entièrement connectés. Le mécanisme d'auto-attention permet au modèle de pondérer différemment les parties de la séquence d'entrée, en mettant l'accent sur les éléments les plus pertinents pour la tâche en cours.

En effet, chaque séquence est subdivisée en plusieurs unités de sens, également appelées *tokens*. Pour chaque token dans la séquence, l'auto-attention calcule une série de pondérations d'attention. Ces pondérations déterminent l'importance relative de chaque autre token par rapport au token en question, comme illustré dans la Figure 31. Les pondérations d'attention obtenues sont utilisées pour pondérer les plongements lexicaux de tous les autres tokens de la séquence. En d'autres termes, les tokens qui sont plus pertinents pour le token en cours obtiennent une pondération plus élevée. Ensuite, les réseaux de neurones entièrement connectés, sont utilisés pour sommer ces plongements lexicaux pondérés afin de former un vecteur contextuel qui capture l'information contextuelle des tokens environnants par rapport au token en question.

Contrairement aux approches précédentes, qui traitaient les séquences de manière uniforme, l'attention permet une adaptation dynamique aux contextes changeants, améliorant ainsi la capacité du modèle à comprendre des relations complexes et à effectuer des tâches telles que la traduction automatique, la génération de texte et la compréhension du langage naturel. En intégrant ce phénomène d'attention, les modèles deviennent plus adaptables, robustes et performants dans la manipulation de données séquentielles complexes.

Pour être plus précis, les transformers utilisent de l'auto-attention multi-tête, souvent abrégée en attention multi-têtes. Contrairement à une attention simple qui se concentre sur une combinaison linéaire unique des représentations d'entrée, l'auto-attention multi-tête permet au modèle de prêter attention simultanément à différentes parties

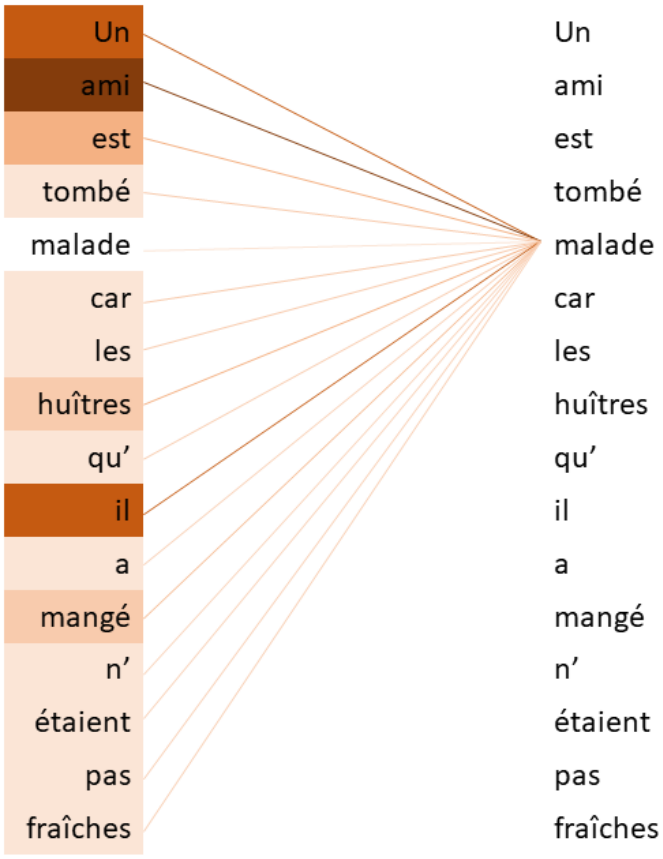


FIGURE 31 – Exemple du procédé d'auto-attention. Le mécanisme d'attention a remarqué que les tokens "Un" et "ami" étaient pertinents pour contextualisé sur le token "malade".

de la séquence à partir de multiples perspectives. Le concept sous-jacent est d'introduire plusieurs mécanismes d'attention, appelés « têtes », qui fonctionnent de manière indépendante mais coopérative. Chaque tête génère une représentation pondérée de l'entrée, et ces différentes représentations sont ensuite concaténées ou combinées de manière linéaire, comme illustré dans la Figure 32. Cela permet au modèle de saisir des motifs complexes et non linéaires dans les données, améliorant ainsi la capacité du transformer à gérer des relations complexes dans des séquences de longueur variable.

Les transformers reposent sur une architecture composée de deux composants, l'encodeur et le décodeur :

— **Encodeur** : Les encodeurs dans les transformers sont responsables de la représentation des données d'entrée. Ils décomposent la séquence d'entrée en une série de vecteurs, capturant ainsi les informations contextuelles et les relations entre les éléments de la séquence. Chaque élément de la séquence est traité indépendamment, puis les résultats sont agrégés en une représentation globale. Ce processus d'encodage permet au modèle de construire des représentations riches des données d'entrée.

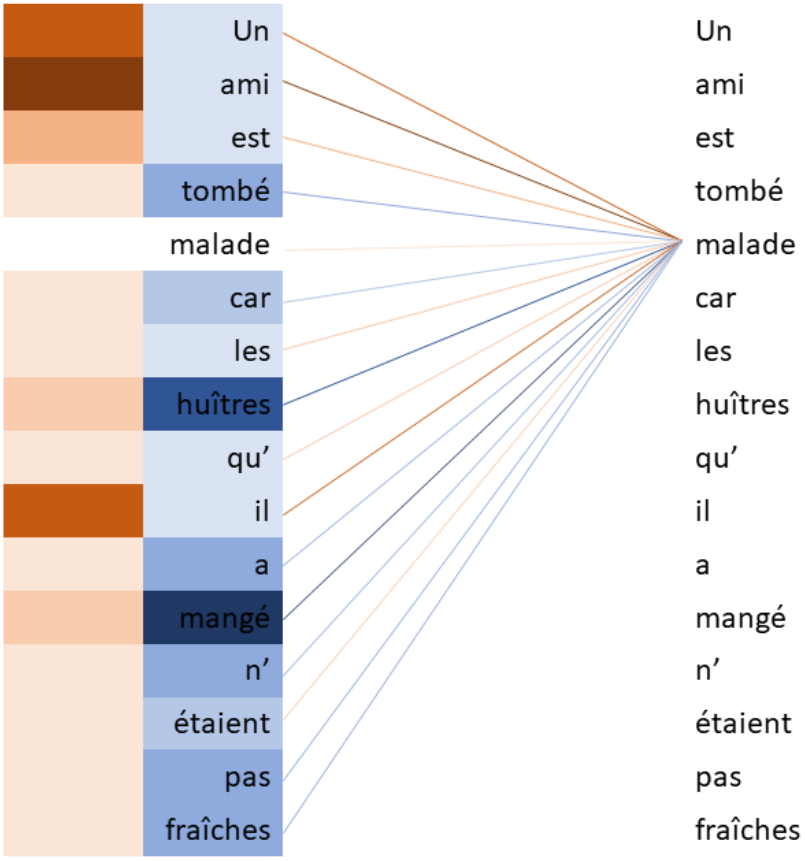


FIGURE 32 – Exemple du procédé d'auto-attention multi-tête. L'une des têtes d'attention se focalise sur "Un" et "ami", tandis que l'autre sur "mangé" et "huîtres", pour contextualiser malade. Cela parait pertinent puisque "malade" réfère à la fois à l'individu et à son état, chaque tête se concentrant sur un sens du token.

— **Décodeur** : Les décodeurs sont chargés de générer la séquence de sortie à partir de la représentation encodée. Ils fonctionnent de manière similaire aux encodeurs, mais sont orientés vers la génération d'une sortie séquentielle plutôt que vers la capture d'information importante d'une entrée. Les décodeurs prennent en compte la représentation encodée ainsi que les parties déjà générées de la séquence de sortie pour produire les éléments suivants de manière séquentielle. Cela permet aux transformers de générer des séquences de manière cohérente et contextuellement informée.

Un exemple du fonctionnement d'un transformers composé d'une partie encodeur et une partie décodeur, effectuant une tâche de traduction automatique, est illustré dans la Figure 33. On peut voir que la partie de l'encodeur permet de transformer une séquence de texte dans une langue source en plongement lexical contextualisé dans un espace vectoriel sémantique commun aux deux langues. La partie décodeur transforme ce plongement lexical en séquence de texte dans la

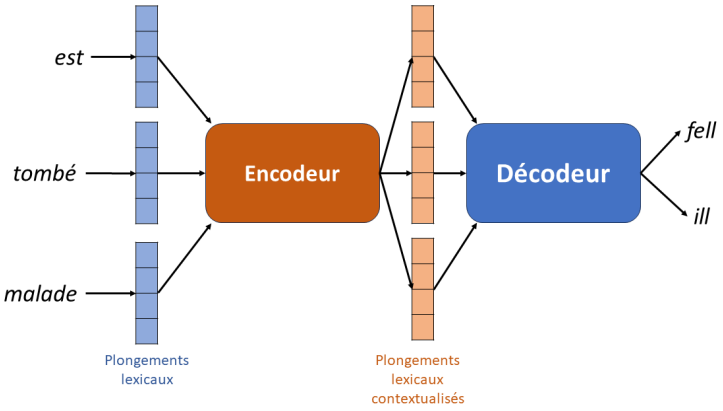


FIGURE 33 – Exemple du fonctionnement d'un transformers sur la tâche de traduction automatique.

langue objectif. De plus, on précise que la principale différence entre le mécanisme d'auto-attention de l'encodeur et du décodeur, est que pour celle du décodeur, seuls les tokens antérieurs dans la séquence sont pris en compte.

Pour des tâches de compréhension de texte, telle que la traduction automatique où la sortie dépend de l'ensemble de l'entrée et des éléments "sorties" déjà générés, on utilise souvent une architecture avec un encodeur suivi d'un décodeur. L'encodeur traite la séquence source, tandis que le décodeur génère la séquence cible étape par étape, en prenant en compte les représentations contextuelles produites par l'encodeur. En revanche, pour des tâches où la sortie dépend uniquement de l'entrée, comme la classification de texte, un modèle composé uniquement d'un encodeur peut être suffisant. L'encodeur capture les caractéristiques importantes de la séquence d'entrée afin de former des représentations, ensuite utilisées pour effectuer la tâche spécifique, telle que la classification. Un exemple concret de l'utilisation exclusive d'un décodeur se trouve dans des tâches génératives, telles que la génération de texte ou d'images. Dans ces cas, le modèle génère une séquence étape par étape en fonction d'un contexte initial, sans recevoir d'entrée préexistante. Cela permet au modèle de créer de nouvelles données basées sur son apprentissage préalable.

BERT *Bidirectional Encoder Representations from Transformers*) et GPT *Generative Pre-trained Transformer* sont deux exemples emblématiques basés sur l'architecture transformers :

— BERT, développé par Google [48], est un modèle d'encodeur qui a révolutionné le traitement du langage naturel. Contrairement aux modèles précédents qui prenaient en compte uniquement le contexte avant ou après un mot, BERT adopte une approche bidirectionnelle, prenant en considération tout le contexte entourant un mot. Cette méthode permet à BERT de capturer des rela-

tions complexes et subtiles entre les mots dans une phrase, améliorant ainsi considérablement la compréhension sémantique.

— GPT, développé par OpenAI [24], se distingue en tant que modèle de décodeur. GPT est pré-entraîné sur un vaste corpus de texte et excelle dans la génération de texte cohérent et contextuellement informé. Le modèle utilise un décodeur transformer qui, à chaque étape, génère le mot suivant dans une séquence en se basant sur les parties précédemment générées et la représentation contextuelle du texte d'entrée.

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Université de Toulouse

Titre : Apprentissage actif multi-labels pour des architectures transformers

Mots clés : Apprentissage actif, Multi-labels, Transformers,

Résumé : L'annotation des données est cruciale pour l'apprentissage automatique, notamment dans les domaines techniques, où la qualité et la quantité des données annotées affectent significativement l'efficacité des modèles entraînés. L'utilisation de personnel humain est coûteuse, surtout lors de l'annotation pour la classification multi-labels, les instances pouvant être associées à plusieurs labels. L'apprentissage actif (AA) vise à réduire les coûts d'annotation en sélectionnant intelligemment des instances pour l'annotation, plutôt que de les annoter de manière aléatoire. L'attention récente portée aux transformers a mis en lumière le potentiel de l'AA dans ce contexte. De plus, le mécanisme de fine-tuning, où seules quelques données annotées sont utilisées pour entraîner le modèle sur une nouvelle tâche, est parfaitement en accord avec l'objectif de l'AA de sélection des meilleures données à annoter. Nous étudions donc l'utilisation de l'AA dans le contexte des transformers pour la tâche de classification multi-labels. Hors, la plupart des stratégies AA, lorsqu'elles sont appliquées à ces modèles, conduisent à des temps de calcul excessifs, ce qui empêche leurs utilisations au cours d'une interaction homme-machine en temps réel. Afin de pallier ce problème, nous utilisons des stratégies d'AA plus rapides, basées sur l'incertitude. D'abord, nous mettons l'accent sur l'application de six stratégies d'AA différentes sur deux modèles transformers. Nos travaux mettent en évidence qu'un certain nombre de stratégies basées sur l'incertitude ne surpassent pas l'échantillonnage aléatoire lorsqu'elles sont appliquées aux modèles transformers. Afin d'évaluer si ces résultats sont dûs à un biais des stratégies basées sur l'incertitude, une approche de pré-clustering est introduite pour ajouter de la diversité dans la sélection des instances. Enfin, nous nous penchons sur les défis pratiques de la mise en œuvre de l'AA dans des contextes industriels. Notamment, l'écart entre les cycles de l'AA laisse du temps inutilisé aux annotateurs. Pour résoudre ce problème, nous étudions des méthodes alternatives de sélection d'instances, visant à maximiser l'efficacité de l'annotation en s'intégrant de manière transparente au processus de l'AA. Nous commençons par adapter deux méthodes existantes aux transformers, en utilisant respectivement un échantillonnage aléatoire et des informations de cycle d'AA périmées. Ensuite, nous proposons notre méthode novatrice basée sur l'annotation des instances pour rééquilibrer la distribution des labels. Notre approche atténue les biais, améliore les performances du modèle (jusqu'à 23% d'amélioration sur le score F1), limite les disparités dépendantes de la stratégie (diminution de près de 50% de l'écart-type) et réduit le déséquilibre des libellés (diminution de 30% du ratio moyen de déséquilibre). Nos travaux ravivent ainsi la promesse de l'AA en montrant que son intégration adaptée dans un projet d'annotation se traduit par une amélioration des performances du modèle final entraîné.

Title: Multi-label active learning applied to transformers architectures

Key words: Active Learning, Multi-labels, Transformers,

Abstract: Data annotation is crucial for machine learning, especially in technical domains, where the quality and quantity of annotated data significantly impact the effectiveness of trained models. Human annotation is costly, particularly for multi-label classification tasks, as instances may be associated with multiple labels. Active Learning (AL) aims to reduce annotation costs by intelligently selecting instances for annotation, rather than annotating randomly. Recent attention on transformers has highlighted the potential of AL in this context. Moreover, the fine-tuning mechanism, where only a few annotated data points are used to train the model for a new task, aligns well with the goal of AL to select the best data for annotation. We investigate the use of AL in the context of transformers for multi-label classification tasks. However, most AL strategies, when applied to these models, lead to excessive computational time, hindering their use in real-time human-machine interaction. To address this issue, we employ faster AL strategies based on uncertainty. First, we focus on applying six different AL strategies to two transformer models. Our work highlights that several uncertainty-based strategies do not outperform random sampling when applied to transformer models. To evaluate if these results stem from a bias in uncertainty-based strategies, we introduce a pre-clustering approach to add diversity to instance selection. Lastly, we tackle the practical challenges of implementing AL in industrial contexts. Particularly, the gap between AL cycles leaves idle time for annotators. To resolve this, we explore alternative instance selection methods aiming to maximize annotation efficiency by seamlessly integrating with the AL process. We start by adapting two existing methods to transformers, using random sampling and outdated AL cycle information, respectively. Then, we propose our innovative method based on instance annotation to rebalance label distribution. Our approach mitigates biases, improves model performance (up to 23% improvement on the F1 score), reduces strategy-dependent disparities (nearly 50% decrease in standard deviation), and decreases label imbalance (30% decrease in the mean imbalance ratio). Our work thus revives the promise of AL by demonstrating that its adapted integration into an annotation project results in improved performance of the final trained model.