Université de Montréal

Sur la génération d'exemples pour réduire le coût d'annotation

par

Frédéric Piedboeuf

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures et postdoctorales

1er décembre 2023

Université de Montréal

Faculté des études supérieures et postdoctorales

Cette thèse intitulée

Sur la génération d'exemples pour réduire le coût d'annotation

présentée par

Frédéric Piedboeuf

a été évaluée par un jury composé des personnes suivantes :

Esma Aïmeur
(président-rapporteur)

Philippe Langlais
(directeur de recherche)

Jian-Yun Nie
(membre du jury)

Éric Gaussier
(examinateur externe)

Alejandro Murua
(représentant du doyen de la FESP)

Thèse acceptée le :
20 mars 2024

Sommaire

L'apprentissage machine moderne s'appuie souvent sur l'utilisation de jeux de données massifs, mais il existe de nombreux contextes où l'acquisition et la manipulation de grandes données n'est pas possible, et le développement de techniques d'apprentissage avec de petites données est donc essentiel. Dans cette thèse, nous étudions comment diminuer le nombre de données nécessaires à travers deux paradigmes d'apprentissage : l'augmentation de données et l'apprentissage par requête synthétisée.

La thèse s'organise en quatre volets, chacun démontrant une nouvelle facette concernant la génération d'exemples pour réduire le coût d'annotation. Le premier volet regarde l'augmentation de données pour des textes en anglais, ce qui nous permet d'établir une comparaison objective des techniques et de développer de nouveaux algorithmes. Le deuxième volet regarde ensuite l'augmentation de données dans les langues autres que l'anglais, et le troisième pour la tâche de génération de mots-clés en français. Finalement, le dernier volet s'intéresse à l'apprentissage par requête synthétisée, où les exemples générés sont annotés, en contraste à l'augmentation de données qui produit des exemples sans coût d'annotation supplémentaire. Nous montrons que cette technique permet de meilleures performances, particulièrement lorsque le jeu de données est large et l'augmentation de données souvent inefficace.

Mots-clés : Intelligence artificielle, traitement des langues, apprentissage supervisé, jeux de données, augmentation de données, modèles génératifs, MQS, petites données.

Summary

Modern machine learning often relies on the use of massive datasets, but there are many contexts where acquiring and handling large data is not feasible, making the development of techniques for learning with small data essential. In this thesis, we investigate how to reduce the amount of data required through two learning paradigms : data augmentation and membership query synthesis.

The thesis is organized into four parts, each demonstrating a new aspect of generating examples to reduce annotation costs. The first part examines data augmentation for English text, allowing us to make an objective comparison of techniques and develop new algorithms. The second one then explores data augmentation in languages other than English, and the third focuses on the task of keyword generation in French. Finally, the last part delves into membership query synthesis, where generated examples are annotated, in contrast to data augmentation, which produces examples without additional annotation costs. We show that this technique leads to better performance, especially when the dataset is large and data augmentation is often ineffective.

Keywords : `Artificial Intelligence, Natural Language Processing, Supervised Learning, Datasets, Data Augmentation, Generative Models, Synthesized Query Learning, Small Data.`

Table des matières

List of tables

List of figures

Remerciements

Réaliser une thèse est un travail de longue haleine, un test d'endurance et de résilience qui serait impossible sans un réseau de support solide, et il me semble important de prendre le temps ici de souligner leurs contributions.

En premier lieu évidemment se trouve mon superviseur Philippe Langlais, qui m'a guidé à travers les hauts et les bas de la recherche, m'a poussé vers une plus grande rigueur scientifique, et m'a permis de voir mes erreurs comme des opportunités d'apprentissages plutôt que des catastrophes.

Je remercie également mes collègues et amis David, Guillaume et Olivier, qui m'ont non seulement accompagné dans le travail de cette thèse, mais rejoint dans ma recherche dans une collaboration qui dure encore après deux articles.

Un merci à mon réseau de support moral, incluant (mais non limité à) ma mère, Djawed, Élodie, Béatrice, Sara, Jade et Marwan, qui m'ont écouté me plaindre pendant quatre ans.

Finalement, je tiens à remercier LexRockAI, qui a financé une grande partie du projet et m'a permis à travers des discussions récurrentes d'approfondir mes idées et surtout de voir la valeur de ma recherche pour un contexte industriel.

Chapitre 1

Introduction

Des voitures qui se conduisent presque seules, des machines qui lisent du texte et qui conversent fluidement avec les humains, ou encore des algorithmes qui peuvent générer d'incroyables morceaux d'arts en ayant uniquement besoin de descriptions. Ces technologies, qu'il y a 50 ans étaient du domaine de la science-fiction, sont maintenant réalité, et le progrès rapide des cinq dernières années laisse percevoir une ère de grands changements dans un futur proche.

Cette révolution a été apportée par plusieurs améliorations successives des réseaux neuronaux, par l'augmentation de la puissance de calcul des GPUs, et finalement par l'augmentation de la quantité de données disponible (ou du moins utilisable) pour entraîner ces réseaux. Il est difficile aujourd'hui de douter que plus de données équivaut à de meilleures performances, ce qui est supporté par la littérature étudiant la question [Feng *et al.*, 2021; Shorten et Khoshgoftaar, 2019; Krizhevsky *et al.*, 2017].

L'obtention de données massives se montre cependant souvent coûteuse, particulièrement lorsque la tâche en elle-même est complexe. Même avec les bonnes performances que plusieurs des algorithmes ont aujourd'hui en *zero-shot learning*, l'utilisation de données manuellement annotées reste souvent plus efficace [Ollion *et al.*, 2023; Kocoń *et al.*, 2023]. De plus, plusieurs recherches récentes montrent

un danger à l'utilisation de données massives de façon non-supervisée, puisqu'il est alors difficile, voire impossible, d'empêcher certains biais de rentrer dans les jeux de données [Bender *et al.*, 2021; Paullada *et al.*, 2021], biais qui peuvent ensuite avoir un impact très réel sur les utilisateurs.

Dû à la difficulté d'obtenir des données annotées de qualité, de nombreuses techniques ont été développées pour pouvoir diminuer la taille des jeux de données nécessaires, tel l'apprentissage semi-supervisé [Su *et al.*, 2021; Yang *et al.*, 2021; Gururangan *et al.*, 2019], l'apprentissage non supervisé [Kingma *et al.*, 2014], ou encore l'apprentissage actif [Bachman *et al.*, 2017; Beygelzimer *et al.*, 2016]. Ces techniques deviennent particulièrement intéressantes lorsque l'on considère les domaines spécialisés où il est difficile d'obtenir des données, tel le domaine de la médecine [Budd *et al.*, 2021] ou le traitement automatique des langues (TAL) pour des langages rares [Feldman et Coto-Solano, 2020].

Cette thèse s'intéresse au problème d'acquisition de données annotées, en mettant l'accent sur l'utilisation des modèles génératifs. Notamment, nous explorons l'augmentation de données pour différentes tâches, ainsi que l'apprentissage par requête synthétisée, un domaine connexe ou des données sont générées puis étiquetées par un humain. Les contributions de cette thèse sont, en ordre présenté :

(1) Étudier l'utilisation des VAEs, ainsi que l'utilisation des modèles pré-entraînés massifs (ChatGPT, Llama2), pour l'augmentation de données [Piedboeuf et Langlais, 2022a, 2023],

(2) Étudier l'utilisation de l'augmentation de données sur les allolangues (langues non anglaises) et montrer que les algorithmes classiques utilisés dans la littérature ne sont pas nécessairement les plus efficaces,

(3) Étudier l'augmentation de données pour la génération de mots-clés en français, incluant la création d'un corpus à cet effet et le développement de nouvelles techniques [Piedboeuf et Langlais, 2022b; Piedboeuf *et al.*, 2023],

(4) Montrer l'efficacité du MQS sur des problèmes réels, ainsi qu'une étude sur la question des *oracles* parfois utilisés en IA pour remplacer les humains étiqueteurs [Piedboeuf et Langlais, 2022c].

La thèse est structurée comme tel. Dans le chapitre 2, les algorithmes essentiels à la compréhension de la thèse ainsi que des notions préalables sont présentées. Puis les trois chapitres suivants explorent l'augmentation de données pour différents domaines. Le chapitre 3 regarde l'aspect de l'augmentation de données sur les textes anglophones, suivi par un regard sur l'augmentation de données sur les langues non-anglaises dans le chapitre 4. Cette exploration se poursuit dans le chapitre 5 en considérant l'augmentation de données pour une tâche différente : la génération de mots-clés, ce qui permet de souligner le besoin de développer des techniques spécifiques à chaque tâche. Le dernier chapitre de cette thèse porte sur l'apprentissage par requête synthétisé, où nous testons plusieurs algorithmes et montrons leur efficacité, notamment en comparaison à l'augmentation de données pour des jeux de données larges.

Il est important de noter que la réalisation de cette thèse a été débutée en 2019, avant l'avènement des modèles tel ChatGPT [OpenAI, 2023] et LLama2 [Touvron *et al.*, 2023b] qui ont des capacités génératives bien supérieures aux auto-encodeurs variationnels que nous utilisons dans cette thèse. Les résultats présentés, bien que novateurs à l'époque, se retrouvent parfois dépassés par ces nouvelles technologies. Lorsque approprié, nous tentons d'intégrer ces technologies à notre thèse, en les contrastant aux méthodes développées.

Chapitre 2

Notions de bases

Ce chapitre présente les notions importantes à la lecture de cette thèse, avec un focus sur les algorithmes qui reviendront régulièrement (section 2.1), ainsi qu'une présentation du principe d'augmentation de données et d'apprentissage actif (section 2.2), qui sont des thèmes sous-jacents à tous les chapitres. Nous tentons ici de donner une vue d'ensemble pour faciliter la lecture de la thèse, et chaque chapitre inclut une revue de littérature qui souligne la littérature nécessaire à la compréhension de la matière présentée.

2.1. Algorithmes

2.1.1. RNNs, LSTMs, et GRUs

Le premier des algorithmes exploré est le réseau neuronal de type récurrent, qui est une famille de réseaux neuronaux conçus spécifiquement pour être entraînés sur des séries temporelles. Trois types principaux existent : Les RNNs (*recurrent neural networks*) [Rumelhart *et al.*, 1985], les LSTMs (*Long Short-Term Memory*) [Hochreiter et Schmidhuber, 1997], et les GRUs (*Gated Recurrent Units)* [Cho *et al.*, 2014]. Bien que ces trois méthodes soient les principales utilisées dans la recherche, bien d'autres types de réseaux récurrents ont été développés, tels

les réseaux récurrents hiérarchiques [Chung *et al.*, 2017] ou les réseaux récurrents pour temps continu [Bown et Lexer, 2006].

Le principe de base de ce type d'algorithme repose sur un réseau neuronal qui est appliqué à chaque étape (dans ce cas-ci chaque mot), prenant comme entrée un vecteur d'information et retournant une sortie (qui peut par exemple être ensuite transformé en mot) ainsi qu'un vecteur d'information supplémentaire qui achemine l'information des étapes précédentes.

Le LSTM est une version plus complexe et plus efficace du RNN, qui rajoute trois moyens de contrôler l'information : un contrôle de l'entrée, un contrôle de la sortie, et un contrôle d'oubli. Ces contrôles permettent d'apprendre des dépendances plus longues entre les séquences et de diminuer le problème du gradient disparu, où le gradient tombe à zéro lorsque la séquence est trop longue [Bengio *et al.*, 1994].

Finalement, le GRU est une simplification du LSTM où il n'y a que deux contrôles (contrôle de mise à jour et d'oubli) et une sortie (alors que le LSTM avait trois contrôles et deux sorties) et qui obtient des performances rivalisant ce dernier [Cho *et al.*, 2014; Khandelwal *et al.*, 2016]. Bien que ces réseaux soient aujourd'hui moins utilisés dans la recherche en TAL (traitement des langues naturelles), remplacés largement par les *transformers* (présentés ci-dessous), ils restent pertinents pour certains algorithmes qui sont utilisés dans cette thèse, notamment les VAEs textuels qui sont à base de GRUs. La figure 2.1 montre les schémas de fonctionnement des trois réseaux.

2.1.2. LLM, encodeur

Les transformers sont des modèles profonds qui intègrent le concept d'attention [Bahdanau *et al.*, 2016] et qui permettent de traiter tous les mots

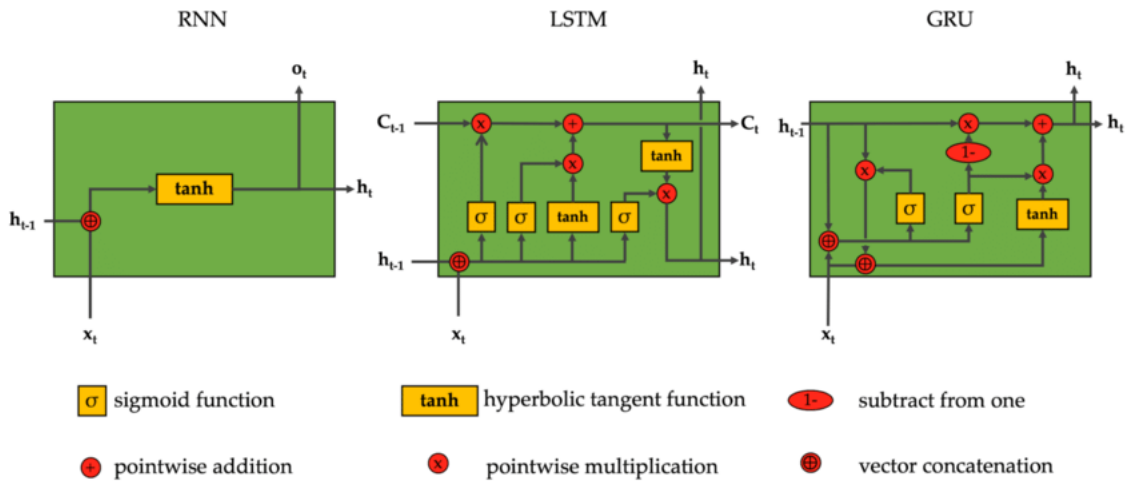

Fig. 2.1. RNN, LSTM, et GRU. Image prise de Sit *et al.* [2020]

d'une séquence en parallèle, augmentant ainsi la rapidité et la puissance du traitement lorsque comparé à des réseaux récurrents [Vaswani *et al.*, 2017].[1]

Cette technologie a également initié une vague de modèles pré-entraînés sur des tâches non supervisées de TAL (Traitement Automatique des Langues), qui ont permis d'atteindre un nouvel état de l'art sur de nombreuses tâches en aval. Ces modèles sont ce que nous appelons ici des LLM, ou *Large Language Models*, et peuvent être séparés en trois catégories [Yang *et al.*, 2023] : les modèles de types encodeurs, les modèles de types décodeurs, et les modèles de types encodeurs-décodeurs. Dans les modèles de types encodeurs, nous considérons ici BERT, qui en est le principal représentant et qui est encore d'usage courant aujourd'hui [Senn *et al.*, 2022]. Pour

[1]Il est à noter que les transformers, bien qu'initialement conçus pour du traitement de texte et utilisés comme tel dans cette thèse, ont également été appliqués à de nombreux autres domaines, tel le traitement d'image [Touvron *et al.*, 2021] ou la météorologie [Zhang *et al.*, 2022a]. De nombreuses variations ont également été proposées, notamment en modifiant la structure pour pouvoir être entraînés sur des documents plus longs [Condevaux et Harispe, 2023].

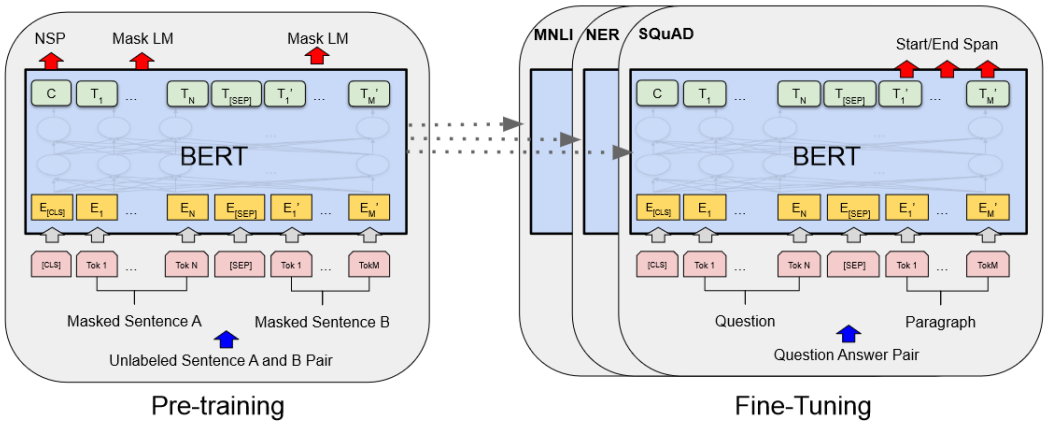

Fig. 2.2. Visualisation de BERT. Image de Devlin *et al.* [2019]

un aperçu global du développement des transformers au cours des dernières années, nous référons à Yang *et al.* [2023].

BERT [Devlin *et al.*, 2019] (*Bidirectional Encoder Representations from Transformers*) est un transformer qui est entraîné pour obtenir des représentations informatives des mots, qui peuvent ensuite être utilisées pour de nombreuses tâches telles de la classification, du question-réponse, ou de la reconnaissance d'entités nommées. À un haut niveau, BERT utilise le principe d'attention pour calculer un score entre les tokens, basé sur trois représentations différentes de ceux-ci (la clé, la requête, et la valeur). Ce processus permet de calculer le score pour chaque token en parallèle, augmentant ainsi de façon importante la rapidité du traitement lorsque comparé à des modèles comme les RNNs qui doivent effectuer les calculs de façon séquentielle. Le même processus est ensuite appliqué aux représentations obtenues, de façon à avoir des niveaux d'abstractions de plus en plus élevés à chaque couche.

Pour tous les transformers modernes, l'architecture reste sensiblement la même (multiples couches d'attention et de couches linéaires empilées menant à une représentation de la phrase), et donc les trois facteurs qui viennent influencer la

performance sont la taille du réseau, la tâche de pré-entraînement, et le nombre de données d'entraînement.[2]

BERT utilise deux tâches de pré-entraînement : la prédiction de mots masqués (remplacer aléatoirement des mots par un masque dans l'entrée et demander à BERT de prédire les mots originaux), et la prédiction de la prochaine phrase (prédire si la phrase qui suit était la bonne phrase dans le texte original ou non). Ces deux tâches ont pour but d'encourager le réseau à apprendre des informations sémantiques au niveau des mots et des phrases. Un token spécifique, "[CLS]" est rajouté en début de phrase, et a pour but d'obtenir une représentation globale de la phrase qui peut ensuite être utilisée dans une tâche de classification en rajoutant une couche de neurones à cet effet par-dessus la représentation obtenue. La version que nous utilisons, BERT-base, est composée de 12 couches avec 12 têtes d'attention, totalisant 110M de paramètres, et est pré-entraînée sur BookCorpus [Zhu *et al.*, 2015] et Wikipédia Anglais. Une représentation en est montrée à la figure 2.2. Nous utilisons également mBERT, un modèle BERT multilingue qui possède la même architecture, mais qui est entraîné sur le wikipédia des 102 langues les plus communes.[3]

2.1.3. LLM, décodeur

Contrairement aux transformers de type encodeurs, les transformers de type décodeurs sont entraînés pour générer du texte de manière auto-régressive. Cela nécessite une adaptation du mécanisme d'attention pour restreindre l'accès uniquement aux jetons précédents. Sans cette restriction, le modèle aurait un aperçu des jetons suivants, ce qui compromettrait sa capacité à générer du texte inédit.

[2]Il s'agit évidemment d'une simplification du processus, et chaque nouveau modèle apporte généralement de petites améliorations supplémentaires au niveau de l'architecture en elle-même.

[3]`https://huggingface.co/bert-base-multilingual-uncased`

GPT-2 (*Generative Pre-trained transformers 2)* fut le premier modèle de ce type à être publié et a montré une capacité impressionnante à générer du texte cohérent [Radford *et al.*, 2019]. GPT-2 est disponible en cinq tailles : petite, moyenne, grande, extra-grande et distillée, et est entraîné sur environ 40G de données. Le modèle que nous utilisons dans cette thèse est le large, qui est composé de 762M de paramètres.

GPT-3, Llama, GPT-Neo, et co.: Le successeur de GPT-2, GPT-3, affiche des performances de génération nettement supérieures. Avec ses 175 milliards de paramètres, ce modèle est cent fois plus grand que la version antérieure et a été entraîné sur le double de données, soit 45G supplémentaires [Brown *et al.*, 2020]. À l'instar de son homologue précédent, il est conçu pour générer du texte de manière auto-régressive.

La publication de GPT-3 a marqué le début de l'ère des réseaux neuronaux pré-entraînés propriétaires, OpenAI (la compagnie qui a entraîné GPT-3) jugeant que la mise à disposition du réseau au grand public était trop risquée en raison des dangers potentiels associés à son utilisation.[4] En réponse à cela, des versions *sources libres* de ces systèmes ont été publiées, notamment par EleutherAI qui ont entraîné GPT-Neo [Black *et al.*, 2021], GPT-J [Wang et Komatsuzaki, 2021], et GPT-NeoX [Black *et al.*, 2022]. D'autres variations, à la fois en sources libres et fermées, ont été proposées par la suite, tel Galactica [Taylor *et al.*, 2022] (un modèle de langue spécialisé pour le domaine scientifique) ou Chinchilla [Hoffmann *et al.*, 2022] (un modèle créé pour être plus petit tout en gardant l'efficacité de GPT-3). Plus récemment, Meta a introduit Llama [Touvron *et al.*, 2023a], un modèle qui rivalise avec GPT-3. Bien que de taille plus réduite, il affiche des performances prétendument

[4] `https://www.independent.co.uk/tech/microsoft-openai-gpt3-exclusive-b550673.html`

supérieures. Un des avantages de ce modèle est qu'il est à source ouverte pour les académiques, alors que GPT-3 est payant.

ChatGPT, BARD, et co: L'évolution naturelle des modèles de types GPT-2/GPT-3 est le développement des LLMs de types "agents conversationnels". Ces algorithmes ont vu leurs débuts avec Instruct-GPT [Ouyang *et al.*, 2022], un affinage de GPT-3 pour le rendre habile à répondre à des instructions (par exemple "Explique moi pourquoi le ciel est bleu").[5] Cela a ensuite mené à des modèles conversationnels plus puissants, notamment ChatGPT (GPT-3.5) et GPT-4 [OpenAI, 2023], qui ont montré d'excellentes capacités de compréhension du langage et de génération de texte [Nori *et al.*, 2023; Katz *et al.*, 2023]. Google a riposté à GPT-3.5 en sortant BARD, un agent conversationnel du même style que ChatGPT. Similairement, Meta a entraîné et publié sa prochaine itération de modèles : Llama2, une série de modèles à la fois de types agents conversationnels et pour de la complétion de texte. Au moment de l'écriture de cette thèse, peu de choses sont connues sur Llama2 à part le fait que la version large (70B de paramètres) utilise GQA (Group-Query Attention), une technique qui permet de grouper les requêtes d'attention pour diminuer le temps d'inférence [Ainslie *et al.*, 2023]. Dans cette thèse, "Llama-2-13b-chat-hf" est utilisé, en inférence uniquement et avec quantization [Dettmers *et al.*, 2023] afin de pouvoir le faire rouler sur un GPU.

2.1.4. LLM, encodeur-décodeur

La troisième et dernière catégorie des transformers sont les modèles qui combinent à la fois un encodeur et un décodeur, permettant ainsi d'affiner les modèles pour des tâches de style *seq-to-seq*, telle la traduction ou la génération de paraphrase. Bien

[5] Il est à noter que les agents conversationnels existaient avant, mais que nous parlons ici des agents basés sur les transformers pré-entraînés, qui sont bien plus performants que leurs prédécesseurs.

que plusieurs modèles existent, nous nous concentrons dans cette thèse sur deux d'entres eux : BART et T5. Ces deux modèles partagent une architecture analogue, consistant en un encodeur de type BERT associé à un décodeur de la famille GPT-2. Ils intègrent à la fois de l'attention auto-dirigée (*self-attention*) et de l'attention dirigée vers les sorties de l'encodeur. Cependant, ils se distinguent considérablement par leurs stratégies respectives d'entraînement.

BART (*Bidirectional Auto-Regressive Transformers*) est entraîné sur cinq tâches : prédiction de mots masqués, prédiction de séquences masquées, prédiction de mots supprimés, réordonnement de phrases, et prédiction du début de texte lorsque la phrase est "enroulée" sur elle-même (par exemple "ceci est une phrase" $\rightarrow$ "une phrase ceci est"), et ce sur 160G de données [Lewis *et al.*, 2020]. Les modèles utilisés au cours de cette thèse sont `BART-large`, un modèle de 336M de paramètres, ainsi que `mBART-large-50`, une extension du modèle multilingue de Lewis *et al.* [2020] qui fait passer le nombre de langages que le modèle peut manipuler de 25 à 50 [Tang *et al.*, 2020].

T5 (*Text-to-Text Transfer Transformer*) est également un modèle de type encodeur-décodeur, mais à la différence de BART il est entraîné à la fois sur des tâches non-supervisées et des tâches supervisées, permettant d'utiliser le modèle non seulement pour des tâches de types *seq-to-seq*, mais également pour des tâches de classification, de QA, de NER, ou autre [Raffel *et al.*, 2020]. Le modèle est entraîné sur 750G de données, donc considérablement plus de données que BART. Pour les tâches supervisées, un préfixe décrivant la tâche est rajouté à l'entrée (*summarize*, *translate*, *cola sentence*, etc), pour indiquer au modèle ce qu'il doit effectuer. Le modèle qui est utilisé à travers cette thèse est `t5-large`, un modèle de 770 millions de paramètres.

2.1.5. Limitation des transformers

Bien qu'étant des outils puissants, les transformers ont été l'objet de plusieurs critiques, notamment par rapport au biais qu'ils peuvent apprendre [Bender *et al.*, 2021; Lalor *et al.*, 2022], au fait qu'ils semblent ignorer certaines informations essentielles lors de l'apprentissage [Pham *et al.*, 2021], ou au coût écologique lié à l'entraînement de modèles de plus en plus gros, sur de plus en plus de données [Schwartz *et al.*, 2019; Strubell *et al.*, 2019], pour un retour seulement logarithmique [Huang *et al.*, 2017]. Ces modèles sont également souvent limités au niveau de plusieurs tâches (notamment montré par McKenzie *et al.* [2023]; Kocoń *et al.* [2023]), et au niveau de la fausse information, que les systèmes ont tendances à donner avec une grande confiance [Tamkin *et al.*, 2021]. Ces considérations sont importantes et justifient plusieurs des choix effectués dans cette thèse. En effet, l'apprentissage sur des petites données permet 1- de réduire considérablement le coût environnemental et 2- d'avoir des données observables et interprétables qui permettent de s'assurer qu'il n'existe pas de biais qui se sont introduits dans le jeu de données.

2.1.6. VAEs

Parmi les modèles génératifs (modèles qui apprennent à modéliser la distribution du jeu de données), les modèles latents ont ces dernières années grandement gagné en popularité, en raison de leur efficacité démontrée dans le domaine des images. Dans cette thèse, nous nous penchons particulièrement sur les VAEs qui sont, des quatre principaux algorithmes modernes latents existants, les plus étudiés, du moins pour les modèles textuels. Ces quatre modèles sont les VAEs, les GANs, les modèles de flots, et les modèles de types diffusion, illustrés à la figure 2.3.

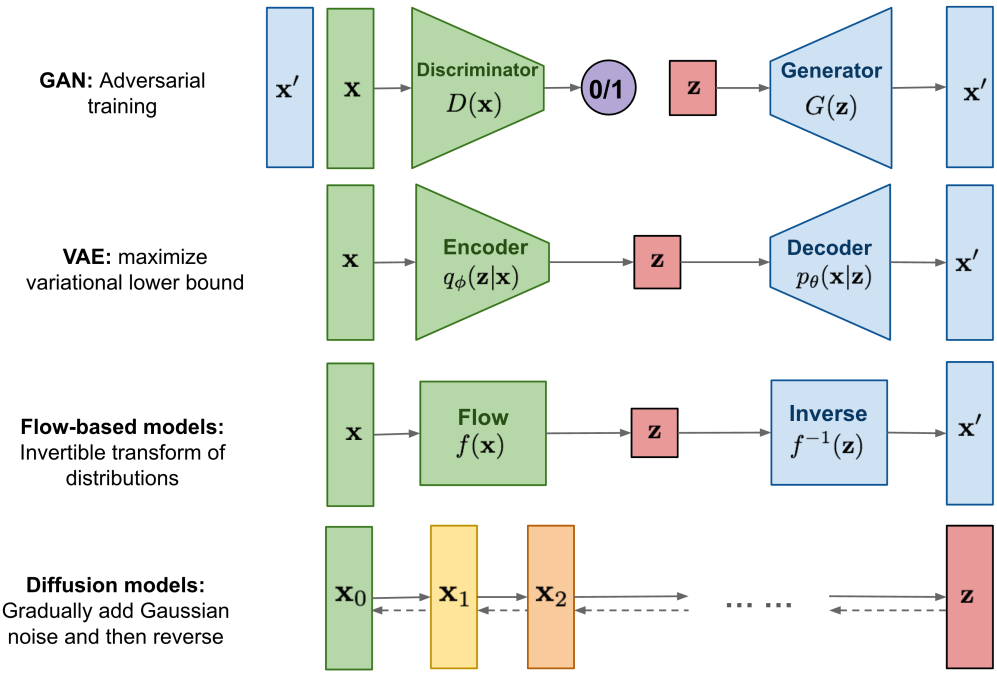

Fig. 2.3. Représentation d'un GAN, VAE, modèle de flot, et d'un modèle de type diffusion. Image de `http://143.89.199.124/paper_reading/The_theory/diffussion_model/`.

Les modèles génératifs sont intéressants puisqu'ils permettent automatiquement d'extraire des caractéristiques latentes des données, qui peuvent ensuite être utilisées pour des applications en aval. Si, par exemple, nous entraînons un VAE sur les données MNIST [Lecun *et al.*, 1998], des données représentants des chiffres écrits à la main, nous obtenons une distribution dans l'espace latent ressemblant à la figure 2.4. Cette distribution, où l'on voit une séparation claire des différentes classes, peut ensuite être utilisée pour des tâches connexes. Le graphique révèle particulièrement la puissance des VAEs pour organiser les données dans un espace latent, puisque les classes sont organisées pour permettre une transition continue d'une classe à l'autre.

Par exemple, la classe des chiffres sept est contiguë à la classe des neuf, qui est elle-même contiguë à la classe des chiffres quatre. Huit, cinq, puis deux peuvent également être produits dans une ligne continue.

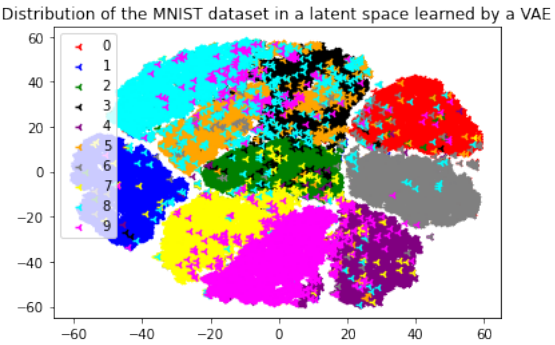

Fig. 2.4. Visualisation de l'espace latent avec T-SNE [Maaten et Hinton, 2008]. Le VAE comprend seul comment séparer les classes et l'on peut observer une logique dans la distribution.

Cette observation souligne les deux caractéristiques intéressantes des espaces latents, du moins pour cette thèse : ils sont *continus* et *denses*. La première des caractéristiques indique qu'un petit changement dans l'espace latent devrait résulter en un petit changement dans l'espace interprétable, et la deuxième, que n'importe quel point de l'espace latent devrait correspondre à un point de l'espace interprétable. Les modèles latents possèdent donc des applications particulièrement intéressantes pour des tâches en aval, tel l'apprentissage actif [Sinha *et al.*, 2019], le regroupement automatique [Graving et Couzin, 2020], ou même la classification [Xu et Tan, 2020].

Tous les modèles latents mentionnés plus haut sont construits pour trouver un espace latent z qui gouverne le processus de génération, comme illustré à la figure 2.5. La différence entre les modèles est la façon dont cet espace latent est trouvé. Les GANs (*Generative auto-encoders)* sont des modèles composés de deux parties : un générateur qui pige au hasard d'une distribution à priori (souvent une gaussienne

isotropique) et transforme le point pigé en donnée interprétable (dans notre cas, en texte), et un discriminateur qui tente de distinguer si l'entrée vient du jeu de données ou du générateur. En entraînant les deux simultanément, le générateur parvient, en théorie, à imiter la distribution du jeu de données [Goodfellow *et al.*, 2014; Yu *et al.*, 2017; Feng *et al.*, 2020].

Les modèles de flots sont des réseaux neuronaux qui créent un *mapping* du jeu de données à l'espace latent, et inversement. Contrairement aux VAEs cependant, l'encodeur et le décodeur ne sont pas deux modèles, mais un seul modèle qui peut faire passer les données des deux côtés. Puisque le gradient doit pouvoir être propagé des deux côtés, les opérations appliquées par chaque couche de réseau sont choisies pour que le déterminant de la matrice Jacobienne soit facilement calculable [Rezende et Mohamed, 2016], ce qui est nécessaire pour calculer le *mapping* inverse. Finalement, les modèles de type diffusion ont récemment gagné beaucoup de popularité grâce à leurs performances en traitement d'images [Sohl-Dickstein *et al.*, 2015; Li *et al.*, 2023; Saharia *et al.*, 2021], et consistent à l'application d'un processus brownian qui va venir graduellement bruiter l'entrée, pour ensuite être renversé pour retrouver la représentation originale.

Les VAEs, ou *Variational auto-encoders* fonctionnent en combinant les modèles auto-encodeurs avec l'inférence variationnelle [Kingma et Welling, 2014]. Concrètement, le VAE est composé d'un encodeur $q_\phi(z|x)$ et d'un décodeur $p_\theta(x|z)$, regroupés en une architecture inspirée des auto-encodeurs. Statistiquement, l'encodeur représente la distribution variationnelle qui vient approximer la vraie distribution postérieure $p(z|x_i)$, celle-ci étant *intractable*. En effet, l'évaluation de $p(z_i|x_i) = \frac{p(x_i|z_i)p(z_i)}{\int_z p(x_i|z_i)p(z_i)dz}$ est trop complexe, en raison de l'évaluation de l'intégrale du dénominateur. Les méthodes variationnelles permettent d'approximer ce calcul, en utilisant un modèle $q_\phi(z|x_i)$ approximant le vrai postérieur.

Comme dans toute génération de texte, le but de l'entraînement du modèle est de trouver les paramètres θ et ϕ qui maximisent $\log_\theta p(x)$. La dérivation classique pour retrouver la perte (le ELBO, ou *Evidence Lower Bound*) est

$$\log_\theta p(x) = \log \int_z p_\theta(x,z)dz \tag{2.1.1}$$

$$= \log \int_z p_\theta(x,z)\frac{q_\phi(z|x)}{q_\phi(z|x)}dz \tag{2.1.2}$$

$$= \log \mathbb{E}_{z\sim q_\theta(z|x)}\frac{p_\theta(x,z)}{q_\phi(z|x)} \tag{2.1.3}$$

$$\geq \mathbb{E}_{z\sim q_\theta(z|x)} \log \frac{p_\theta(x,z)}{q_\phi(z|x)} \tag{2.1.4}$$

$$= \mathbb{E}_{z\sim q_\theta(z|x)} \log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} \tag{2.1.5}$$

$$= \mathbb{E}_{z\sim q_\theta(z|x)} \log p_\theta(x|z) - KL(q_\phi(z|x)||p(z)) \tag{2.1.6}$$

La dérivation utilise l'inégalité de Jensen qui affirme, entre autres, que le log d'une espérance est plus grand ou égal à l'espérance du log. On se retrouve donc avec deux termes que l'on peut calculer : $\mathbb{E}_{z\sim q_\theta(z|x)} \log p_\theta(x|z)$, qui peut être approximé avec une pige de Monte Carlo, et la divergence KL entre $q_\theta(z|x)$ et $p(z)$ qui, avec un choix judicieux du priori $p(z)$ et du modèle $q_\phi(z|x)$, peut être résolu analytiquement. Dans la plupart des VAEs, l'apriori $p(z)$ est une gaussienne isotropique, et le modèle $q_\phi(z|x)$ produit les variances σ et les moyennes μ d'une gaussienne diagonale, ce qui permet de calculer analytiquement la divergence[6] avec

$$KL(q_\phi(z|x)||p(z)) = \frac{1}{2}\sum_{j=1}^{J}(1 + log(\sigma_j^2) - \mu_j^2 - \sigma_j^2) \tag{2.1.7}$$

[6]voir Kingma et Welling [2013] pour la dérivation complète

À ce stade du VAE, l'entraînement se produirait de la façon suivante : pour chaque point d'une itération, l'encodeur trouverait les paramètres associés à la gaussienne. Une pige au hasard serait faite pour retrouver la valeur latente, qui serait ensuite passée au décodeur qui tenterait de reconstruire l'entrée. Le problème de cette technique repose dans la pige au hasard, qui mène à une grande variance lorsque l'on essaye de faire passer le gradient du décodeur à l'encodeur. Pour régler ce problème, le truc de la *reparamétrisation* est utilisé, où $z = \mu + \epsilon\sigma$, et ϵ est pigé d'une gaussienne isotropique. Cela assure la même moyenne et déviation standard que $\mathcal{N}(\mu, \sigma^2)$, en gardant une dépendance directe par laquelle le gradient peut être calculé facilement.

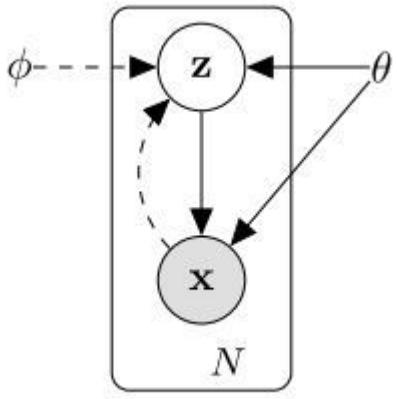

Fig. 2.5. Représentation en plaque d'un VAE. Image de Wang *et al.* [2019].

Finalement, il faut noter que l'adaptation du VAE pour le texte cause généralement le problème dénommé le *KL collapse*, où le terme KL tombe à 0 lors de l'entraînement. Comme établi précédemment, la perte est le ELBO, écrite mathématiquement comme

$$ELBO = \mathbb{E}_{z\sim q_\theta(z|x)} \log p_\theta(x|z) - KL(q_\phi(z|x)||p(z)) \tag{2.1.8}$$

Il est facilement notable que les deux termes se battent entre eux : d'un côté le terme de reconstruction tente de maximiser l'information passée par l'espace latent, menant à un grand terme KL. De l'autre, le terme de la divergence tente de faire correspondre l'espace latent à la distribution à priori, ce qui voudrait dire que tous les points sont générés à partir d'une gaussienne isotropique. Dans le cas des images, le processus d'entraînement fonctionne bien. Cependant, le décodeur classique d'un VAE textuel est de type récurrent, ce qui signifie qu'il peut délaisser les données de l'espace latent pour se concentrer uniquement sur l'optimisation à partir des plongements de mots fournis à chaque étape du décodage, à la manière d'un modèle de langue traditionnel. Pratiquement, c'est ce qui arrive le plus souvent, le modèle se coinçant dans un minimum local où la divergence est à zéro et la reconstruction est basée sur le décodeur uniquement.

Deux solutions sont généralement utilisées pour diminuer l'ampleur de ce problème : le *dropout* du décodeur, où un pourcentage des mots sont remplacés au hasard par un jeton de mot inconnu, et le recuit de la divergence (*KL Annealing*), où la perte est réécrite comme $\mathbb{E}_{z\sim q_\theta(z|x)} \log p_\theta(x|z) - \alpha KL(q_\phi(z|x)||p(z))$, et le terme α est lentement amené de 0 à 1 pendant l'entraînement [Bowman *et al.*, 2016].

Une note finale importante nous amène à souligner les nombreuses améliorations qui ont été apportés aux VAEs, mais sont largement ignorées dans le cadre de cette thèse. Les améliorations proposées (et potentiellement pertinentes) reposent soit sur essayer de diminuer le problème de la chute du terme KL [Lucas *et al.*; Zhu *et al.*, 2020; Dai *et al.*, 2020], sur améliorer globalement le VAE [Burda *et al.*, 2016; Dai et Wipf, 2019; Larsen *et al.*, 2016], ou essayer (pour les VAEs textuels) d'utiliser de

meilleurs réseaux que les LSTMs ou GRUs, comme le transformer [Zhao *et al.*, 2021] ou les modèles larges pré-entraînés [Park et Lee, 2021; Li *et al.*, 2020].

Dans cette thèse, nous entraînons les VAEs avec les deux stratégies de base mentionnées plus haut : le *dropout* de mots du décodeur et le recuit de la divergence, et en utilisant des GRUs comme encodeurs et décodeurs. Nombre de modèles alternatifs ont été essayés, mais nous avons trouvé que trois problèmes majeurs nous empêchait d'utiliser ces modèles. Tout d'abord, beaucoup des modèles sont difficiles à réimplémenter, car les articles ne fournissent pas tous les détails nécessaires pour cela et le code n'est pas utilisable (lorsque présent), ce qui fait de chaque implémentation un projet entier. Deuxièmement, la plupart de ces modèles ne sont pas adaptés aux petites données, domaine dans lequel nous travaillons. Finalement, une bonne partie des articles se fient sur des métriques que nous avons trouvé erronées ou non significatives, et n'observent pas de façon consistante les sorties de ces modèles. Ces facteurs font que pour tous les modèles que nous avons réimplémentés, nous avons observé soit des performances sub-optimales, ou un gain non significatif comparé aux modèles de base utilisés.[7]

2.2. Concepts importants

Nous finissons ce chapitre en décrivant deux concepts qui sont importants pour la compréhension de cette thèse : l'augmentation de données (AD) et l'apprentissage actif (AL, ou *Active Learning*). Bien que ces concepts soient revisités dans les chapitres concernés, une brève introduction est présentée ici pour permettre une lecture plus aisée de la thèse.

[7]Il faut aussi signaler que les modèles VAEs textuels sont très délicats à entraîner, et donc qu'implémenter et apprendre à utiliser un nouveau type est un investissement significatif de temps.

2.2.1. Augmentation de données

Le but de l'augmentation de données (AD) est de générer de nouvelles paires de données (x', y) à partir de données existantes, sans coût d'annotation supplémentaire. Il n'existe pas de taxonomie définitive, mais il est généralement possible de la séparer en deux grandes familles : l'AD interprétable et l'AD non interprétable. L'augmentation interprétable regroupe les techniques qui créent de nouvelles données qui peuvent être observées par un être humain et analysées. Dans le domaine des images, la rotation ou la translation sont des exemples typiques de techniques interprétables, alors que pour le texte des exemples seraient le remplacement par synonyme ou la création de paraphrases.

L'AD non interprétable comporte des techniques comme l'apprentissage adversatif, où de nouvelles représentations vectorielles sont créées pour renforcer le modèle. Bien que l'AD non interprétable puisse être très efficace pour augmenter la performance, il a été noté que l'AD interprétable est important pour réduire les biais qui peuvent s'introduire dans les modèles [Minot *et al.*, 2021; Kamalloo *et al.*, 2021]. Comme noté précédemment, cette thèse se concentre sur cette famille due aux nombreux avantages qu'elle possède, telle la réduction de biais ou le coût d'entraînement généralement plus bas.

L'augmentation de données se fait dans le cadre d'une tâche spécifique. Dans cette thèse, deux types de tâches pour l'augmentation de données sont considérées : la classification de phrases, et la génération de mots-clés, les deux étant des tâches supervisées. Dans ces tâches, un jeu de données X composé de données $x_0, x_1, ..., x_n$ avec les étiquettes associées $y_0, y_1, ..., y_n$ est fourni, et les modèles d'AD tentent de générer de nouveaux exemples qui viennent renforcer le réseau. L'étude de ces deux tâches est intéressante car elle permet de souligner la variabilité existante dans les techniques d'AD utilisées. Notamment, pour la génération de mots-clés il est possible

de générer de nouveaux exemples (x', y') qui viennent apporter de la variabilité au jeu de données, alors que pour la classification, l'algorithme ne peut que générer des exemples (x', y), l'étiquette étant une classe spécifique associée à la donnée.

2.2.2. Apprentissage actif

Nous finissons ce chapitre en décrivant brièvement le processus d'apprentissage actif (AL, ou *Active Learning).* Tout d'abord, les tâches supervisées sont encore considérées ici, impliquant qu'il faut initialement un jeu de données étiqueté. Alors que l'augmentation de données s'occupe de ce problème après la collecte de données, l'apprentissage actif intervient lors de la création du jeu, pour diminuer la taille de celui-ci tout en maximisant la performance.

Dans l'apprentissage actif, l'utilisateur commence avec un petit ensemble de données étiquetées $\mathcal{L}$, un ensemble de données non étiquetées $\mathcal{U}$, et un algorithme à entraîner $\mathcal{M}$. Le but est de maximiser la performance de $\mathcal{M}$ avec le nombre minimal d'annotations $\mathcal{U} \rightarrow \mathcal{L}$.

L'AL est séparé en trois catégories : l'apprentissage actif en continu (*stream-based*), l'apprentissage actif basé sur un ensemble (*pool-based*), et l'apprentissage par requête synthétisée (*Membership Query Synthesis*) [Schröder et Niekler, 2020].

Dans le premier cas, les exemples sont reçus en temps réel et, à chaque instant t, il faut décider si l'on conserve ou non l'exemple pour l'étiquetage. Dans le deuxième cas, nous possédons un ensemble de données non annotées, et l'on va itérativement chercher des données dans cet ensemble, les annoter, et les transférer à l'ensemble de données $\mathcal{L}$. La sélection de quel exemple choisir est au cœur de l'AL, et les stratégies considérées se basent souvent soit sur la confiance du modèle [Ravanbakhsh *et al.*, 2019], soit sur la distribution de $\mathcal{L}$ [Dasgupta et Hsu, 2008].

Finalement, le cas qui nous intéresse dans cette thèse est l'apprentissage par requête synthétisée (MQS), où l'on génère des nouveaux exemples à annoter. Cette

catégorie a une efficacité potentiellement beaucoup plus grande que l'AL, car elle permet de générer des points à partir de n'importe où dans la distribution du jeu de données. Bien que l'efficacité théorique du MQS ait été démontrée [Chen *et al.*, 2017; Hopkins *et al.*, 2020], son applicabilité reste en doute. D'un côté, les modèles de type latent qui permettent de transformer n'importe quel point dans l'espace de données en données interprétables ne fonctionnent que médiocrement sur le texte, apprenant souvent un espace latent ambigu. De l'autre, les modèles de générations de type GPT et transformers permettent la génération de textes très fluides, mais contrôler la génération selon la distribution du jeu de données se retrouve à être très difficile.

Le cas du MQS est intéressant pour cette thèse dû à son affinité naturelle avec l'augmentation de données. En effet, les deux paradigmes diffèrent principalement au niveau d'un seul élément : l'annotation des exemples générés. Alors que dans l'augmentation de données, les nouvelles données sont directement rajoutées à $\mathcal{L}$, dans le MQS ces données sont d'abord étiquetées par un humain ou un oracle. Comme démontrée au chapitre 6, cette distinction apporte plusieurs avantages et une augmentation significative de la performance, puisque le plus gros facteur déterminant l'efficacité d'un algorithme d'augmentation de données textuelles est la cohérence de classe, c'est-à-dire, la génération d'exemples (x', y) qui ne changent pas le sens de la phrase en exemple (x', y').

2.3. Conclusion

Dans ce chapitre, nous présentons plusieurs concepts clés. Nous abordons les transformers, ainsi que deux méthodes visant à réduire le coût d'annotation : l'augmentation de données et l'apprentissage actif. Dans le prochain chapitre, nous explorons l'augmentation de données, montrant notamment comment utiliser les VAEs à cette fin et en les comparant à d'autres algorithmes.

Chapitre 3

Augmentation de données textuelles pour la classification de textes courts

La première et plus simple des applications qui est explorée dans cette thèse est l'augmentation de données pour la classification de textes courts, en anglais.

Dans le domaine des images, les techniques d'AD bénéficient d'opérations simples, comme la rotation ou le changement de taille, pour créer des images qui gardent la même classe tout en apportant de la diversité à l'ensemble de données. Cependant, le TAL est plus sensible aux petits changements, et une modification négligente d'un mot peut complètement changer la classe de la phrase. Par exemple, une mauvaise modification de la phrase "This movie is good" pourrait la transformer en "This movie is bad", ce qui change complètement sa classe, si la tâche est de déterminer la polarité de la phrase. Les techniques d'augmentation de données textuelles doivent donc porter une attention particulière à maintenir la *cohérence de classe*.

Ce chapitre se penche sur le phénomène d'augmentation de données textuelles sous plusieurs angles. Dans un premier temps, divers algorithmes d'augmentation de données sont implémentés et comparés, en portant une attention particulière à l'utilisation des modèles génératifs. La littérature sur le sujet manque de comparaisons objectives, et la recherche de ce chapitre permet d'observer la

performance des algorithmes sur des jeux de données communs et avec un protocole commun, et de montrer qu'au final, il existe peu de différences entre les performances des algorithmes testés. De plus, plusieurs facteurs qui ont été largement ignorés jusqu'à maintenant sont explorés. Notamment, nous regardons la performance de l'AD en fonction du ratio de phrases originales vs phrases augmentées, la performance en fonction de la taille du noyau (nombre de phrases dans le jeu de données), et les erreurs produites par les algorithmes d'AD, ce qui permet d'isoler certaines caractéristiques définissant un bon algorithme. Nous intégrons aussi les résultats obtenus à l'aide de ChatGPT et Llama2, selon différentes stratégies de génération de données, et montrons que générer des données en fournissant simplement une description de la tâche est la méthode la plus efficace, si celle-ci est bonne. Les contributions de ce chapitre sont :

(1) Produire l'étude la plus exhaustive comparant des méthodes d'AD sur divers jeux de données,

(2) Proposer plusieurs méthodes originales (VAE-Linked, T5, ChatGPT), qui apportent un nouvel état de l'art à l'AD,

(3) Produire une première étude qui montre l'efficacité des VAEs pour l'augmentation de données,

(4) Montrer que pour les modèles génératifs, avoir un modèle par classe est plus efficace que les autres stratégies possibles (paraphrase, modèle conditionnel),

(5) Analyser les méthodes et faire ressortir plusieurs aspects de l'AD qui avaient été ignorés jusqu'à maintenant dans la littérature,

(6) Tester l'utilisation de ChatGPT et LLama2 et montrer que générer des données est plus efficace que créer des variations de données existantes, mais moins efficace que la collection de données externes,

(7) Montrer plusieurs biais dans les jeux de données communément utilisés en augmentation de données textuelles qui rendent la sollicitation des VLLMs (*Very Large Language Models*) difficile.

Le chapitre est composé comme suit. Les algorithmes qui sont utilisés ainsi que les jeux de données étudiés sont d'abord présentés dans les sections 3.2 et 3.3. Puis, dans les sections 3.4 et 3.5, les résultats et l'analyse de ces résultats sont explicités. Un article a été écrit et publié à COLING 2022 concernant notamment l'utilisation des VAEs pour l'augmentation de données [Piedboeuf et Langlais, 2022a], et un autre article a été publié à EMNLP 2023 concernant l'utilisation de ChatGPT pour l'augmentation de données [Piedboeuf et Langlais, 2023].

3.1. Revue de littérature

Cette section présente la littérature sur l'augmentation de données, avec un focus sur la classification de phrases et l'AD interprétable[1], en essayant de souligner les travaux séminaux. Pour une revue plus complète de la littérature sur l'AD pour le TAL, nous référons à Feng *et al.* [2021]; Bayer *et al.* [2023]; Pluščec et Šnajder [2023].

La taxonomie que nous adoptons est une séparation des techniques en trois catégories : les opérations sur les mots (par exemple remplacer un mot par un synonyme), les opérations sur les phrases (paraphraser ou interpoler entre deux phrases), et les modèles génératifs (générer directement des phrases d'une classe spécifique).

Toutes ces techniques partent d'un jeu de données $\mathcal{L}$ (appelé le *noyau*) composé de données (x, y), et tentent de générer des exemples (x', y) qui vont venir apporter de la diversité au jeu de données. La différence entre les différentes techniques repose dans la manière de générer ces exemples, qui apporteront différentes caractéristiques

[1]Voir Shorten et Khoshgoftaar [2019]; Yang *et al.* [2022] pour une revue de littérature sur l'augmentation de données sur les images.

au jeu final. Par exemple, les opérations sur les mots produisent souvent des exemples moins divers que les deux autres catégories, et souvent moins fiables, mais sont plus simples à implémenter et utiliser.

À travers la revue de littérature, la taille du noyau utilisée est également soulignée. Cette taille est importante pour analyser les résultats correctement, puisque les gains obtenus vont grandement en dépendre. Basé sur notre expérience, les tailles de noyaux sont séparées en trois catégories : petites, ou *few-shot learning* (jusqu'à 100 données), moyennes (entre 100 et 2000 données) et grandes (en haut de 2000 données).

3.1.1. Opérations sur les mots

La première famille d'AD considérée est celle basée sur les opérations au niveau des mots, qui représente sans doute la famille la plus simple et directe d'AD textuelle. Le représentant classique de cette famille est EDA (*Easy Data Augmentation*) [Wei et Zou, 2019; Liesting *et al.*, 2021]. Dans la méthode EDA, quatre opérations peuvent être utilisées : remplacer des mots par leurs synonymes venant d'un dictionnaire, supprimer des mots, insérer des mots reliés sémantiquement aux mots précédents, ou changer deux mots de place. Une opération et un texte d'entrée sont choisis au hasard, et le texte modifié par cette opération est généré. Cette méthode a montré des bons résultats sur des RNNs et des CNNs, avec un gain d'approximativement 1% sur les moyennes et grandes données, et est utilisée de façon importante dans la littérature sur l'augmentation de données, étant souvent définie comme un des résultats de référence à battre.

L'opération de remplacement de mots a été explorée en détail par Marivate et Sefara [2020], qui regardent notamment le remplacement par des synonymes

venant de WordNet[2], ou encore par des voisins dans un plongement de mots pré-entraînés, ce qui est particulièrement utile lorsqu'un dictionnaire de synonymes n'est pas disponible. Sur des jeux de données larges (>10K exemples), les auteurs montrent une augmentation de la performance d'environ 1%.

Une autre technique est de remplacer les mots selon le contexte et la classe de la phrase, en utilisant un RNN [Kobayashi, 2018] (gains d'un peu moins de 1% sur grandes données) ou encore un BERT conditionnel, dénommé CBERT [Wu *et al.*, 2019] (gains d'environ 2% sur des grandes données). L'algorithme CBERT est défini comme une modification de BERT auquel on fournit la classe de l'exemple avec le texte, ce qui lui permet d'apprendre à prédire les mots masqués conditionnellement à la classe.

Une méthode qui repose uniquement sur l'insertion de signes de ponctuation est l'AEDA, où les signes sont insérés dans la phrase au hasard [Karimi *et al.*, 2021] (gains entre 1 et 3% sur moyennes et grandes données). Cette méthode est aussi dans les premières qui soulignent que les phrases générées n'ont pas besoin d'être grammaticalement cohérentes et fluides pour être informatives.

Finalement, Kumar *et al.* [2021] proposent le BART Conditionnel (CBART), où BART est affiné sur une tâche de mots masqués, avec la classe de la phrase fournie en début de texte. La différence principale avec CBERT est que CBART peut prédire des séquences de mots au lieu de mots individuels, lui apportant une plus grande flexibilité. Kumar *et al.* [2021] montrent entre 5 et 20% de gains sur des petites données en l'utilisant, selon la tâche.

[2] `https://WordNet.princeton.edu/`

3.1.2. Opérations sur les phrases

La famille d'opérations sur les phrases est une famille de techniques où la phrase entière est considérée pour créer de nouvelles données, souvent en la paraphrasant. Une technique séminale ayant montré son efficacité est l'augmentation par retour de traduction (Back-translation, ou BT), où une phrase est traduite dans une langue seconde, puis retraduite dans la première langue, créant ainsi une paraphrase [Hayashi *et al.*, 2018; Yu *et al.*, 2018; Corbeil et Ghadivel, 2020; Edunov *et al.*, 2018]. Cette technique est une des plus populaires dans la littérature, dû sans doute à sa simplicité ainsi qu'à son efficacité, et se retrouve souvent à être difficile à battre par les nouvelles techniques proposées, comme observé dans ce chapitre.

Une autre technique utilisée est la création de nouvelles données grâce à des VAEs, en encodant et décodant une donnée [Mesbah *et al.*, 2019; Yerukola *et al.*, 2021; Nishizaki, 2017]. Cette technique a cependant plus été utilisée dans le monde des images que dans le monde du texte, dû principalement à la complexité d'obtenir un espace latent désambiguïsé pour le texte. La création de paraphrases par modification de l'arbre syntaxique a été explorée par Coulombe [2018], qui montre que cette technique fonctionne mieux que le BT, malgré sa complexité additionnelle (gains entre 4 et 20% sur des grandes données).

Finalement, une autre technique qui a été proposée est l'utilisation d'un modèle de type SeqToSeq pour directement générer des paraphrases. Kumar *et al.* [2019] testent plusieurs techniques de décodage sur les RNNs pour créer des paraphrases, et arrivent à la conclusion que leur objectif submodulaire pour la génération arrive à créer de meilleures paraphrases et est plus efficace pour l'augmentation de données (2% sur données larges) que les autres techniques qu'ils testent (décodage par faisceaux, décodage par renforcement, etc). Plus récemment, Okur *et al.* [2022] utilisent un modèle BART affiné sur un corpus de paraphrases, lui-même créée avec du BT

d'un corpus externe, afin de générer de nouvelles phrases pour la compréhension du langage naturel (*Natural language understanding* ou NLU en anglais).

3.1.3. Modèles génératifs

Les articles sur l'augmentation de données utilisant les modèles génératifs sont nombreux et la plupart tournent autour de l'utilisation de modèles *conditionnels*, à la fois pour les images [Zhuang *et al.*, 2019; Wang *et al.*, 2020] et pour le texte [Malandrakis *et al.*, 2019]. Lorsque des modèles non-conditionnels sont utilisés, la méthode la plus courante pour générer de nouvelles données est par encodage et décodage de données existantes (voir section précédente), mais d'autres façons de le faire ont également été explorées. Par exemple, Islam *et al.* [2021] déterminent la frontière de décision dans l'espace latent, pour pouvoir générer des données de la classe minoritaire, et montrent que cette technique est plus efficace que d'autres techniques plus classiques comme SMOTE [Chawla *et al.*, 2002] ou ADASYN [He *et al.*, 2008] des techniques d'AD spécialisées pour des données non balancées. Les VAEs ont également été utilisés pour certaines tâches spécifiques, comme pour la détection d'erreurs grammaticales [Wan *et al.*, 2020], en perturbant les représentations latentes pour générer de nouvelles phrases avec des erreurs.

Les VAEs n'ont été que peu explorés pour l'augmentation de données et la classification de phrases. Le seul article que nous avons trouvé explorant l'utilisation des VAEs à cet effet est celui de Qiu *et al.* [2020], qui regarde l'utilisation des VAEs pour balancer des jeux de données, et qui conclut que l'utilisation des modèles génératifs peine à mieux performer que la pige au hasard pour balancer les classes. Dans ce chapitre, l'utilisation des VAEs dans un contexte pur d'augmentation de données est exploré. Nous démontrons que cet algorithme peut être efficace, tout en ayant l'avantage de ne pas utiliser de données externes, une caractéristique qui sera importante pour le prochain chapitre sur les données non-anglaises.

De façon similaire, les GANs ont aussi été utilisés pour l'augmentation de données, mais majoritairement dans le domaine des images, avec des modèles conditionnels [Antoniou *et al.*, 2018; Tanaka et Aranha, 2019; Bozorgtabar *et al.*, 2019], ou par transfert de style pour générer des nouvelles données à partir d'exemples d'une autre classe [Hong *et al.*, 2021; Jin *et al.*, 2021; Katsuma *et al.*, 2022; Zhang *et al.*, 2022b]. Certains auteurs se sont également intéressés à l'entraînement d'un GAN de façon jointe au classificateur pour générer des données qui en augmentent la performance [Tran *et al.*], ou pour maximiser l'information encodée dans l'espace latent [Tronchin *et al.*, 2023]. Certains articles considèrent l'utilisation des GANs pour l'AD textuelle [Shang *et al.*, 2021; Gupta, 2019], mais dû à la difficulté d'affiner les GANs pour le texte, ces recherches se produisent dans des domaines dans lesquels beaucoup de données sont accessibles, ou alors elles pré-entraînent le modèle sur des milliers de données connexes. Dans les domaines où ces méthodes sont appliquées, elles amènent tout de même une augmentation de la performance par rapport au résultat de référence.

Une autre solution est l'utilisation de larges modèles de langues pré-entraînés, tel le CGPT (GPT conditionnel) [Kumar *et al.*, 2020]. Dans CGPT, l'étiquette est rajoutée avant les phrases préalablement à l'entraînement[3], GPT-2 est affiné, et de nouvelles phrases sont générées en fournissant seulement l'étiquette comme entrée, avec possibilité de rajouter des pertes d'apprentissage par renforcement [Liu *et al.*, 2020]. Une utilisation intéressante de GPT-2 est proposée par Yang *et al.* [2020], qui explorent le filtrage de données générées par GPT-2 à l'aide de fonctions d'influences. Travaillant sur des données de questions-réponses, ils montrent que filtrer les données avec différentes stratégies (diversité, fonctions d'influences, etc.) améliore légèrement la performance, gagnant un peu moins de 1% sur des grandes données.

[3]Par exemple pour la phrase positive "Le film est bon" l'exemple d'entraînement deviendrait "pos: Le film est bon"

Outre le travail de Qiu *et al.* [2020] sur les VAEs, les travaux s'appariant le plus à ce chapitre sont ceux qui tentent également de réaliser des comparaisons objectives d'algorithmes d'augmentation de données. Dai et Adel [2020] comparent plusieurs méthodes d'augmentation des données simples pour le NER, et les auteurs montrent que combiner plusieurs méthodes est souvent plus efficace, que l'augmentation fonctionne mieux lorsque le noyau initial est petit, et qu'il est important d'évaluer sur des modèles pré-entraînés. Plus récemment, Chen *et al.* [2021] comparent plusieurs variations de l'augmentation par mots ainsi que quelques techniques plus récentes, sur des tâches supervisées et semi-supervisées, et concluent qu'il n'existe pas de techniques universellement meilleures que d'autres. Leur étude se penche cependant uniquement sur les très petites données (10 données de départ) et regardent surtout des techniques simples par mots, et donc il n'existe pas de garantie que les résultats soient applicables aux autres domaines. Okimura *et al.* [2022] regardent plusieurs algorithmes, dont BT, W2V (remplacement des mots par des voisins dans un plongement de mots pré-entraîné), et remplacement par BERT, et concluent que pour les jeux de données larges, il est difficile d'avoir une augmentation, une observation récurrente dans la littérature et dans cette thèse.

La plupart des expériences de cette thèse ont été réalisées avant l'apparition des "*very large language models*" (VLLMs), tels ChatGPT (GPT-3.5), GPT-4 [OpenAI, 2023], Llama [Guo *et al.*, 2023], Llama2 [Touvron *et al.*, 2023b], et autres. Bien que nous intégrions ces algorithmes dans le chapitre, nombre des contributions incrémentales réalisées pendant la thèse sont maintenant obsolètes due à la présence de ces modèles, notamment les expériences réalisées sur les VAEs et publiés à COLING 2022 [Piedboeuf et Langlais, 2022a]. Nous séparons donc dans cette section les articles utilisant ces technologies des autres, pour bien comprendre le sillon que l'apparition de ces technologies ont présenté.

Yoo *et al.* [2021] testent l'augmentation de données avec GPT-3, en lui passant une liste d'exemples avec la classe et le laissant compléter la liste, montrant ainsi que sa performance dépasse le BT et l'EDA de jusqu'à 20% sur les petites données. Une contribution importante de leur article est de montrer que les résultats ne sont pas dus seulement à de la mémorisation du jeu de test[4], en créant un nouveau jeu de données à partir de critiques de cinéma écrites *après* l'entraînement de GPT-3. Sahu *et al.* [2022] s'intéressent également à l'utilisation de GPT-3 pour l'augmentation de données, cette fois pour une tâche de classification d'intention. Similairement à Yoo *et al.* [2021], ils utilisent GPT-3 pour compléter une liste d'exemples, mais la différence principale est qu'ils séparent la génération par classe, alors que Yoo *et al.* [2021] demandent à GPT-3 de fournir l'étiquette avec le nouvel exemple. ChatGPT a aussi été étudié pour générer des nouvelles données, en demandant de paraphraser des données existantes [Dai *et al.*, 2023], mais les auteurs affinent au préalable le classificateur avec un large ensemble de données de la même distribution, ce qui rend difficile de déterminer l'efficacité réelle de la technique. Finalement, Møller *et al.* [2023] comparent la performance de GPT-3.5 et GPT-4 pour l'augmentation de données comparée à de l'annotation humaine, et concluent que pour des jeux de données simples (comme de l'analyse de critiques de produits), ChatGPT est meilleur, mais que dans les autres cas l'annotation humaine bat l'AD.

Depuis nos expériences réalisées pour EMNLP, d'autres recherches se sont intéressé à l'utilisation de ChatGPT pour l'augmentation de données. Fang *et al.* [2023] paraphrasent des données avec ChatGPT pour une tâche de détection d'intention et rapportent une petite augmentation de la performance. Frick [2023] génère des données pour une tâche de détection de subjectivité en demandant à ChatGPT de faire du transfert de style entre les classes, mais ultimement rapporte

[4]Les VLLMs sont entraînés sur des données massives, et c'est donc une supposition raisonnable qu'il pourrait y avoir du *peaking*.

que les phrases générées se retrouvent à être trop éloignées de la distribution du jeu de données pour être pertinentes, et que la technique utilisée n'augmente pas la performance. Ubani *et al.* [2023] montrent sur de petits jeux de données que la génération *zero-shot* performe mieux que d'autres modèles comme CBERT, CGPT, ou CBART. Shushkevich et Cardiff [2023] créent des données additionnelles pour une tâche de détection de subjectivité en anglais et en italien, montrant une légère amélioration de la performance sur le jeu de développement. Sharma et Feldman [2023] génèrent des nouveaux dialogues entre patients et docteurs, pour une tâche de classification de sujet des conversations, et montrent une petite amélioration sur un jeu de 1201 exemples, mais ne comparent pas à d'autres techniques. Ces expériences confirment les résultats de ce chapitre, montrant que la génération de nouvelles données est très efficace mais que l'utilisation de ChatGPT pour la paraphrase ne semble pas fonctionner globalement mieux que les autres techniques plus simples.

3.2. Algorithmes

La recherche menant à ce chapitre a d'abord pris naissance dans un désir d'évaluer les capacités des VAEs pour la génération de données, mais a ensuite grandi dans une tentative d'évaluer objectivement les algorithmes existants pour essayer de comprendre lesquels fonctionnaient le mieux et pourquoi. Nous présentons donc ici les algorithmes utilisés pour cette comparaison, en portant une attention particulière à l'utilisation des VAEs pour la génération de données (les résultats de ces expériences ayant été publiés à COLING 2022 [Piedboeuf et Langlais, 2022a]). Le paramétrage des algorithmes d'AD est également un facteur important pour leur efficacité, et les hyper-paramètres considérés pour chaque algorithme sont décrits. Pour paramétrer correctement les algorithmes, un mélange d'observations manuelles des exemples générés et de la performance sur l'ensemble de développement est utilisé. Dû aux différentes tailles de noyaux utilisées, il est difficile d'avoir un paramétrage parfait

pour toutes les expériences, et virtuellement impossible de paramétrer séparément pour chacun des jeux de données et des tailles utilisées.

Les algorithmes évalués sont sélectionnés basés sur leurs popularités et leurs efficacités rapportées, et également en visant l'obtention d'un ensemble diversifié de types de techniques. Les algorithmes testés sont les suivants: AEDA, EDA, CBERT, CBART, CGPT, VAE, CVAE, VAE-Par, BT, T5, ainsi que quelques variations décrites ci-dessous.[5] De ces algorithmes, certains sont des modifications que nous proposons (AEDA-R, T5, VAE-Linked), alors que d'autres sont bien établis dans la littérature. Les méthodes sont illustrées à la figure 3.1.

Tout d'abord, AEDA est un algorithme simple basé sur l'insertion de ponctuations au hasard dans le texte. Les signes de ponctuation à insérer sont "?", ".", ";", ":", "!", et ",", et le nombre d'insertions est calculé selon la formule `RANDINT(1, len(sentence)/3)`. Une variation dénommée AEDA-R (pour AEDA-*Regularization*) est également testée, et consiste en une modification simple de l'AEDA où AEDA est appliqué à la volée sur chaque itération, transformant 50% des exemples. Cela permet d'évaluer rapidement l'impact de la diversité sur l'entraînement, puisque appliquer de nouveau la méthode va produire des exemples plus variés, comparativement à créer des exemples transformés une fois au début. Comme les résultats le montrent cependant, si le ratio d'exemples générés vs exemples originaux et que l'affinage des hyper-paramètres est bien effectué, appliquer l'AD à chaque itération n'est pas plus efficace.

EDA est basé sur quatre opérations : insertion de synonymes, échanger la position des mots, suppression de mots, et remplacement par des synonymes. Pour

[5]Certains algorithmes ont également été testés brièvement, mais abandonnés, dû notamment au temps qu'ils prenaient ainsi qu'à leurs mauvaises performances. Parmi ces algorithmes, nous comptons l'utilisation des GANs (conditionnels ou non) ainsi que W2V, une méthode qui remplace les mots par des voisins dans un plongement de mots pré-entraîné.

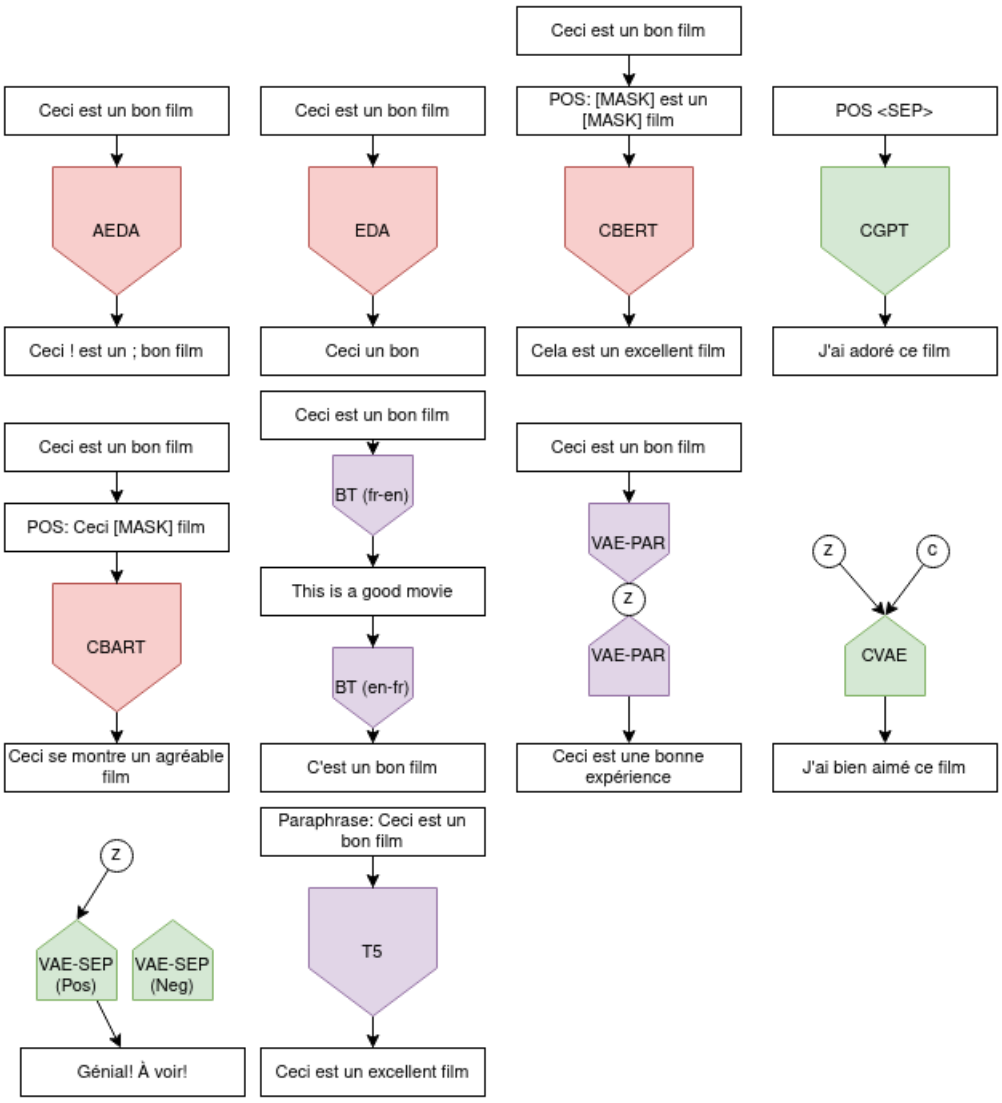

Fig. 3.1. Illustration des différentes méthodes décrites dans cette section. VAE-Linked génère de la même façon que VAE-Sep. T5-Quora, T5-Tapaco, et T5BT génèrent selon l'illustration pour T5. Les méthodes au niveau des mots sont en rouge, celle au niveau des phrases, en mauve, et les méthodes génératives sont en vert. z représente l'espace latent, et c la classe.

chaque phrase, une opération est sélectionnée au hasard et appliquée. Chacune des opérations est contrôlée de façon indépendante par un ratio qui indique quel pourcentage des mots de la phrase originale sera touché. L'échange et la suppression se font de façon directe, mais l'insertion et le remplacement utilisent WordNet (de NLTK[6]) pour trouver les synonymes. De plus, les mots d'arrêt sont exclus des opérations, utilisant la liste de mots d'arrêt de spacy[7]. Les paramètres d'EDA consistent en quatre ratios, un pour chaque opération, qui contrôlent le nombre de mots affectés par cette opération lorsqu'elle est appliquée. Ces quatre ratios sont fixés à 0.1, suivant une recherche d'hyperparamètres ou chacun des ratios est modifié en isolation pour en voir l'impact.

Wei et Zou [2019] montrent que les opérations d'insertions et de remplacement apportent une moins bonne augmentation de la performance que les deux autres, et donc une version utilisant uniquement les opérations de changement de position et de suppression de mots est testée, dénommée dans cette thèse EDA-SD. Cette technique a également l'avantage de ne pas nécessiter de ressources externes, ce qui se montrera pratique lors de l'étude de l'AD sur les langues non-anglaises. Les ratios de ces deux paramètres restent fixés à 0.1, suivant les résultats trouvés pour EDA.

CBERT [Wu *et al.*, 2019] est un algorithme populaire d'augmentation de données où des mots sont masqués et prédits par BERT, conditionnellement à la classe de l'exemple. Lors de l'entraînement et de la génération, la classe est rajoutée au début de la phrase (voir figure 3.1), et BERT est affiné sur la tâche de prédiction des mots masqués. Deux hyper-paramètres importants sont le nombre d'époques d'affinage de CBERT, ainsi que le pourcentage des mots masqués. Contrairement au CBERT original décrit dans [Wu *et al.*, 2019], une condition est également rajoutée pour être

[6] `https://www.nltk.org/`

[7] `https://spacy.io/`

sûr que les mots prédits ne sont pas les mêmes que les mots masqués, en choisissant le deuxième mot le plus probable dans ce cas. À travers des expériences, cela semble augmenter légèrement la performance de CBERT.

CBART est également considéré pour cette thèse. Comme mentionné, CBART est une technique très similaire à CBERT, avec la différence principale que BART permet le un-masquage d'une *suite de mots* au lieu de mots individuels, apportant ainsi une plus grande diversité aux phrases générées. Tout comme CBERT, le paramétrage se fait au niveau du nombre d'époques pour l'affinage, du *learning rate*, et du nombre d'exemples par itération. Les paramètres de génération sont une recherche en faisceaux de 10 avec une séquence maximale de 50.

L'entraînement de CGPT [Kumar *et al.*, 2020] consiste à rajouter la classe au début de la phrase, comme dans CBERT et CBART, et à affiner GPT-2 pour qu'il génère des exemples conditionnellement à celle-ci. Pour générer de nouveaux exemples, il est alors possible de simplement fournir en entrée la classe et le jeton de séparation, et laisser GPT-2 compléter la phrase. Dans notre cas, nous utilisons la méthode du *nucleus sampling*, avec un paramètre de *no-repeat-n-grams* de trois, ce qui force une diversité lors de la génération et empêche le système de tomber dans une boucle où il génère répétitivement la même chose. Le paramétrage se fait également au niveau du nombre d'époques et du *learning rate*.

Back-translation est une technique populaire où une phrase est traduite dans une deuxième langue, puis retraduite dans la langue originelle, créant ainsi une paraphrase. Nous utilisons l'allemand comme seconde langue, suivant [Edunov *et al.*, 2018], et FSMT comme modèle de traduction [Ng *et al.*, 2019], avec une recherche en faisceaux d'une largeur de 10. Puisque les modèles sont pré-entraînés, il n'y a pas de paramétrage à faire pour cette technique.

Comme décrit au chapitre 2, les VAEs sont des algorithmes génératifs de type latents qui permettent la génération à partir de la distribution préalable (dans ce cas

une gaussienne isotropique). Un facteur à considérer cependant est l'adaptation de ces modèles pour pouvoir générer des exemples conditionnellement à la classe.

Similairement à Qiu *et al.* [2020], trois stratégies majeures sont considérées. Dans la stratégie VAE-Par, la phrase est encodée dans l'espace latent, puis passée à travers le décodeur. Si le VAE a appris correctement à organiser son espace latent, la phrase décodée devrait être proche sémantiquement de la phrase encodée, tout en apportant de la variation. Dans la stratégie CVAE, un VAE *Conditionnel* est entraîné, modification du VAE qui consiste simplement à fournir la classe de l'exemple au décodeur et le laisser apprendre par lui-même à utiliser l'information de classe pour générer de meilleures phrases.[8]

Les deux derniers modèles de types VAEs sont VAE-Sep et VAE-Linked. Dans VAE-Sep, un VAE séparé est entraîné pour chaque classe, permettant une génération directement de l'espace latent avec peu de risque de générer de la mauvaise classe. VAE-Linked est une modification de cette idée, ou l'encodeur est partagé à travers les modèles, permettant l'apprentissage d'un meilleur espace latent, mais avec des décodeurs uniques à la classe (donc un encodeur et plusieurs décodeurs). Pour tous ces modèles, nous utilisons des GRUs comme encodeurs et décodeurs, un espace latent de 15, et nous faisons une hausse graduelle du terme KL de 0 à 1, en fonction du nombre d'exemples, ce qui donne comme résultat que les jeux de données plus

[8]Les CVAEs dans la littérature ont souvent la classe fournie à la fois à l'encodeur et au décodeur. Dans cette thèse, la classe est fournie uniquement au décodeur, laissant l'encodeur libre d'organiser l'espace latent comme il le désire. En fournissant la classe à l'encodeur, il est possible qu'il utilise l'information pour organiser l'espace latent de façon désambiguïsée, avec les classes bien séparées, et que le décodeur ignore l'information de classe. Puisque le but ici est d'avoir des exemples similaires en style, mais opposés en sémantique, se cotoyant dans l'espace latent, il est logique de fournir uniquement la classe au décodeur. Nos expériences nous amènent à voir une légère augmentation de la performance en faisant ainsi.

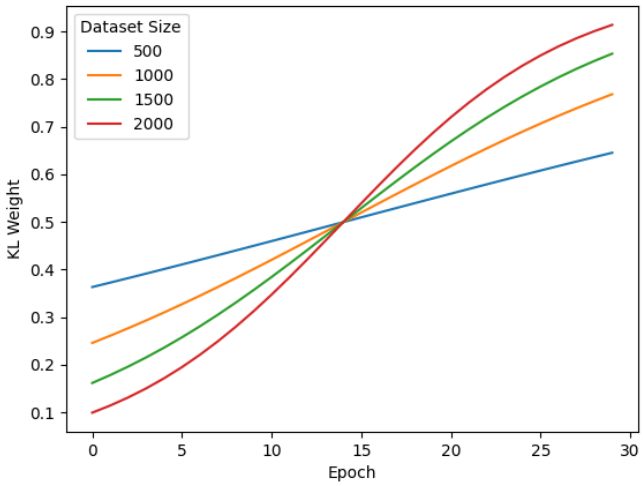

Fig. 3.2. Poids du terme KL à travers les époques pour différentes tailles de départ de jeux de données, pour les méthodes d'augmentation basée sur les VAEs.

petits finiront avec un terme KL plus petit. Cette hausse du KL est montrée dans la figure 3.2.

Les trois prochaines techniques considérées sont inspirées de la littérature sur l'augmentation de données dans le domaine des agents conversationnels. Beaucoup des techniques d'AD de ce domaine sont basées sur la création de paraphrases, puisque les agents classiques (c'est-à-dire pré ChatGPT) reposent sur l'idée de pouvoir mapper différentes façons de formuler une idée à une même réponse. Les deux techniques utilisées ici, qui sont des techniques de notre création, utilisent des réseaux neuronaux pré-entraînés sur des tâches de paraphrases pour directement générer des paraphrases de nos entrées. Nous utilisons deux modèles disponibles sur *huggingface*, T5-Tapaco et T5-Quora, qui sont respectivement entraînés sur les corpus de paraphrases Tapaco [Scherrer, 2020] et Quora [Chandra et Stefanus], et peuvent donc être directement utilisés pour générer de nouvelles phrases sans affinage

additionnel. Pour la génération, une pige au hasard de type *nucleus* (*top-p sampling*) est utilisée avec un facteur de 0.7.

La dernière technique utilisée est une combinaison de T5-Tapaco et BT, que nous dénommons T5BT. T5 (à la fois la version Tapaco et Quora) peut parfois créer des phrases qui sont plus éloignées de la distribution du jeu de données. Par exemple, une des paraphrases de "so mind-numbingly awful that you hope britney won't do it one more time, as far as movies are concerned." par T5-Tapaco est "You're so crazy about you if you don't do it again.", ce qui n'est pas une phrase valide dans le contexte de critiques de cinéma. Nous nous inspirons donc de Okur *et al.* [2022], qui créent d'abord un corpus de paraphrases avec BT pour ensuite affiner BART sur celui-ci, afin qu'il puisse directement générer des paraphrases. Cependant, à la différence de Okur *et al.* [2022] nous affinons T5-Tapaco sur ce corpus de paraphrases nouvellement créées, dans l'idée de forcer T5 à générer des phrases plus proches du jeu de données. Le paramétrage pour cette technique se fait au niveau du nombre d'époques d'affinage et du nombre d'exemples par itération.

3.3. Jeux de données et protocole

Nous choisissons cinq jeux de données pour tester objectivement les algorithmes : SST-2, FakeNews, Irony, IronyB et TREC6. SST-2 [Socher *et al.*, 2013] est un jeu de données de classification de critiques de cinéma, avec les classes de critiques négatives ou positives. FakeNews[9] est un jeu de données où l'on essait de détecter si la nouvelle est vraie ou fausse (désinformation) à partir du titre. Irony et IronyB [Van Hee *et al.*, 2018] sont des tâches de classification de gazouillis. Irony consiste à classifier les gazouillis selon s'ils sont ironiques ou non, et IronyB est la version multiclasse de la tâche, où les gazouillis doivent aussi être classifiés selon le type d'ironie

[9] `https://www.kaggle.com/c/fake-news/overview`

qu'ils présentent (contraste de polarité, ironie situationnelle, autre). Finalement, TREC6 [Li et Roth, 2002] est un jeu de données où le but est de classifier des questions en six catégories différentes (abréviation, description, entités, humains, lieux et valeurs numériques). Certaines caractéristiques des jeux de données sont présentées à la table 3.1.

Nom	SST2	Irony	FakeNews	IronyB	TREC6
nb. classes	2	2	2	4	6
|entraînement|	6920	2683	12799	2681	5452
longueur. sents.	19.3	13.7	12.5	13.7	10.2
|test|	1821	3834	3293	3832	492

Tab. 3.1. Quelques caractéristiques des jeux de données. Pour la longueur des phrases, l'espace blanc est utilisé comme délimiteur de mots.

Ces jeux de données ont été choisis pour avoir une variété de tâches, et parce qu'ils sont communément utilisés dans la littérature sur l'augmentation de données. SST-2 et TREC6 sont utilisées régulièrement, par exemple par Kumar *et al.* [2020]; Quteineh *et al.* [2020]; Regina *et al.* [2021]; Kobayashi [2018]. Similairement, Irony et IronyB ont reçu beaucoup d'attention pour l'AD, par exemple par Liu *et al.* [2020]; Turban et Kruschwitz [2022]; Yao et Yu [2021]. Finalement, FakeNews n'a jamais, à notre connaissance, été testé avec de l'augmentation de données, mais il reste un jeu de données communément utilisé pour la détection de fausses nouvelles [Verma *et al.*, 2023; Chakraborty *et al.*, 2023; Iceland, 2023].

Le premier problème étudié dans ce chapitre est tout simplement l'efficacité des algorithmes décrits à la section précédente. Deux facteurs importants vont cependant affecter l'efficacité des algorithmes, c'est-à-dire, la taille du noyau (nombre de données

étiquetées au départ) et le ratio de phrases générées vs phrases originales. Les six expériences suivantes sont définies :

(1) Petites données (10, 20) avec un ratio de 10,

(2) Moyennes données (100, 500) avec un ratio de (5, 1),

(3) Grandes données (1000, toutes) avec un ratio de 1.

Ces domaines couvrent globalement ce qui se fait dans la littérature, et nous permet de voir l'efficacité sur différentes tailles de jeux de données. Bien qu'il soit parfois standard dans la littérature sur l'AD d'utiliser la justesse à la fois pour les tâches binaires et multiclasses [Marivate et Sefara, 2020; Kumar *et al.*, 2021], nous utilisons ici le macro-f1 pour les tâches multiclasses, ce qui nous permet de mieux évaluer l'impact sur la performance. Les choix de ratio sont justifiés à la section 3.5.2.

Finalement, nous utilisons BERT [Devlin *et al.*, 2019] comme classificateur, et l'ensemble de développement avec arrêt précoce pour affiner le modèle. Nous rapportons la moyenne ainsi que la déviation standard sur 15 expériences.

3.4. Résultats

Nous présentons dans cette section les résultats sur les différents jeux de données et en utilisant les ratios précisés à la section précédente. Dans la prochaine section, nous analysons l'impact du ratio et de la taille du noyau. Les tables regroupent les résultats par taille de noyau (petites données, moyennes données, grandes données).

La table 3.2 montre les résultats pour les petites données, c'est-à-dire, des tailles de noyau de 10 et 20, avec un ratio de 10. La majorité des algorithmes fonctionnent bien, malgré le fait que certains (CVAE, VAE-PAR, CGPT, les algorithmes génératifs en général) semblent avoir plus de difficulté à obtenir des bonnes performances. Les algorithmes qui semblent le mieux performer globalement sont AEDA-R et T5. Les autres algorithmes fonctionnent parfois bien, mais ne se démarquent pas de façon spectaculaire. Une chose à noter est que la stratégie T5BT, qui avait pour but

	SST2	FakeNews	Irony	IronyB	Trec6	Moyenne
Baseline	54.9/61.5	50.7/51.8	54.1/56.2	25.8/31.1	32.1/38.6	43.5/47.8
EDA	59.2/64.4	54.0/56.0	55.2/57.0	30.0/35.2	38.3/50.6	47.3/52.6
EDASD	60.1/64.7	54.0/**56.4**	54.9/57.8	30.3/36.1	37.3/49.9	47.3/53.0
AEDA	59.6/65.3	53.3/55.0	55.6/57.5	32.4/35.2	36.7/50.6	47.5/52.7
AEDAR	**60.9**/66.7	53.6/54.9	**55.7**/57.7	33.7/34.9	38.5/52.8	**48.5**/53.4
VAESep	57.7/64.7	**54.3**/56.5	**55.7**/57.9	31.7/33.8	32.8/44.8	46.4/51.5
VAELink	57.9/64.5	53.1/55.6	55.4/57.7	32.0/35.6	33.3/43.0	46.3/51.3
CVAE	57.3/61.0	53.0/54.2	55.9/56.0	29.1/34.6	34.6/38.7	46.0/48.9
VAEPar	53.5/59.5	50.6/54.3	53.4/55.4	28.8/31.9	29.3/39.6	43.1/48.1
BT	58.0/63.4	53.2/55.0	56.1/57.2	31.8/33.0	**40.4**/53.8	47.9/52.5
CBERT	58.7/66.5	52.9/53.1	55.8/57.5	32.8/31.9	35.4/48.9	47.1/51.6
CGPT	52.7/58.0	51.3/54.3	52.7/56.1	28.7/33.0	37.4/50.8	44.6/50.4
CBART	58.5/65.2	51.2/53.0	54.4/56.8	28.5/31.5	37.0/49.0	45.9/51.1
T5-Tapaco	59.9/66.1	54.2/55.8	53.8/56.1	**33.9**/36.4	40.3/**54.9**	48.4/**53.9**
T5Quora	57.7/64.5	52.9/54.4	54.0/**58.1**	33.6/**37.5**	38.0/50.9	47.2/53.1
T5BT	59.1/**67.5**	52.8/54.4	53.7/55.8	32.9/36.3	36.8/49.7	47.1/52.7

Tab. 3.2. Résultats sur les petites données (10, 20) avec un ratio de 10. Les déviations standards sont entre 2.8 et 9.0.

d'apporter une augmentation de la performance par rapport à T5, semble au contraire la dégrader, bien que de peu. Comme nous le montrons à la section 3.5, il semble que le facteur le plus important pour l'efficacité des algorithmes soit que les phrases générées ne donnent pas de l'information erronée au modèle, et donc il semblerait que T5 soit assez bon pour faire cela sans avoir besoin d'être affiné en plus. Il se peut

également que T5BT, bien qu'il diminue les erreurs, n'apporte pas assez de diversité pour être efficace.

	SST2	FakeNews	Irony	IronyB	Trec6	Moyenne
Baseline	71.9/87.1	61.0/72.4	60.9/69.0	38.2/55.7	72.4/90.1	60.9/74.9
EDA	81.1/87.7	63.4/73.7	60.9/69.2	44.0/55.8	76.7/90.3	65.2/75.3
EDASD	81.0/**87.8**	63.3/**73.8**	60.6/69.5	43.4/56.5	75.0/89.2	64.7/**75.4**
AEDA	**82.3**/87.1	63.1/73.2	60.9/69.2	43.6/55.3	77.1/90.1	**65.4**/75.0
AEDAR	80.1/87.0	62.5/72.6	61.1/69.2	43.4/55.7	**78.0**/**90.4**	65.0/75.0
VAESep	78.1/87.3	63.6/73.0	60.8/69.1	42.0/56.3	69.4/88.6	62.8/74.9
VAELink	80.0/86.4	62.1/73.7	61.1/69.2	42.0/53.8	70.2/88.9	63.1/74.4
CVAE	74.7/86.0	62.1/72.2	60.5/69.2	40.8/53.6	73.0/89.2	62.2/74.0
VAEPar	74.7/85.2	61.5/72.2	60.1/68.4	37.4/50.8	74.4/89.5	61.6/73.2
BT	81.2/87.7	63.0/73.1	**61.5**/**69.4**	41.6/55.4	77.5/90.3	65.0/75.2
CBERT	80.5/86.5	62.3/72.5	60.1/68.8	39.5/55.7	72.8/88.2	63.0/74.3
CGPT	74.1/86.7	59.4/72.7	57.8/68.5	40.0/53.2	75.3/89.4	61.3/74.1
CBART	81.7/86.9	60.8/72.2	58.4/68.5	39.6/53.4	75.0/89.7	63.1/74.1
T5-Tapaco	81.6/86.6	**63.8**/73.4	60.2/68.7	**44.3**/56.4	75.3/88.6	65.0/74.7
T5Quora	79.6/87.0	62.8/73.3	60.6/68.1	43.5/**56.6**	73.6/87.5	64.0/74.5
T5BT	80.5/**87.8**	62.7/73.1	60.5/69.1	43.9/55.3	73.6/89.3	64.2/74.9

Tab. 3.3. Résultats sur les moyennes données (100, 500) avec un ratio de 5 et de 1. Les déviations standards sont entre 0.4 et 3.1.

Similairement, la table 3.3 montre les résultats pour les données moyennes (100 et 500). Cette fois, et bien que T5-Tapaco reste un compétiteur très performant, les méthodes par mots semblent globalement plus efficaces. Cependant, il est à noter que l'amélioration sur la taille de noyau de 500 est petite et que la différence pour

la taille de 100 n'est pas très grande.[10] Dans la section 3.5, nous effectuons des tests statistiques pour évaluer si cette différence est significative.

	SST2	FakeNews	Irony	IronyB	Trec6	Moyenne
Baseline	88.6/91.2	76.2/88.0	75.2/90.9	66.2/86.9	92.5/**95.9**	79.7/90.6
EDA	88.5/91.3	76.3/88.6	75.5/91.0	65.8/**87.2**	92.4/95.4	79.7/90.7
EDASD	88.3/91.1	**77.4**/**88.6**	75.9/91.0	65.3/**87.2**	92.1/95.6	79.8/90.7
AEDA	88.3/91.3	76.9/88.3	75.2/91.1	**66.9**/87.0	92.5/95.6	**80.0**/90.7
AEDAR	88.4/91.3	77.0/**88.6**	75.6/**91.2**	65.9/**87.2**	**93.0**/95.6	**80.0**/**90.8**
VAESep	88.3/90.3	77.3/88.2	75.0/90.9	64.0/86.7	92.3/95.8	79.4/90.4
VAELink	88.3/90.7	76.5/88.0	**75.7**/91.0	65.4/86.5	91.5/95.6	79.5/90.4
CVAE	88.1/91.0	76.7/88.4	75.2/90.9	63.2/86.9	92.7/95.7	79.2/90.6
VAEPar	86.1/90.4	75.2/87.7	74.2/90.1	63.8/85.5	92.6/92.3	78.4/89.2
BT	88.4/**91.8**	76.7/87.7	75.5/90.9	66.0/**87.2**	**93.0**/95.6	79.9/90.6
CBERT	87.6/90.4	76.5/88.2	75.3/90.8	64.2/86.7	92.2/95.5	79.2/90.3
CGPT	**89.2**/90.8	76.8/87.9	74.6/90.3	63.6/87.0	92.1/95.7	79.3/90.3
CBART	87.8/91.0	71.5/87.9	75.3/90.8	63.8/86.7	92.1/95.6	78.1/90.4
T5-Tapaco	87.8/91.0	72.7/87.6	75.0/90.6	65.7/86.5	91.1/95.2	78.5/90.2
T5Quora	88.2/91.0	77.2/87.9	75.0/90.4	64.4/86.8	90.7/94.3	79.1/90.1
T5BT	88.4/91.6	77.0/87.8	75.1/91.1	66.4/86.9	92.5/95.5	79.9/90.6

Tab. 3.4. Résultats sur les grandes données (1000, All) avec un ratio de 1. Les déviations standard sont entre 0.3 et 2.9.

Finalement, la table 3.4 montre les résultats pour les grands jeux de données. Les tendances observées aux tables précédentes semblent se poursuivre. L'augmentation

[10]Nous montrons à la section 3.5.2 qu'il ne s'agit pas d'une question de ratio, mais plutôt de taille de noyau.

est globalement moins efficace, et les méthodes par mots semblent gagner encore une fois sur les méthodes par phrases, qui ne semblent pas être capables d'apporter une amélioration.

3.5. Analyse des résultats

3.5.1. Exemples de phrases générées

Malgré le fait qu'il ne semble pas y avoir d'algorithmes qui surpassent les autres de façon définitive, certains des algorithmes peuvent être plus intéressants que d'autres, notamment au niveau de l'interprétabilité des données, de la fluidité des phrases, ou de la diversité. La table 3.5 montre un exemple de phrases générées pour SST-2 avec la phrase d'entrée "makes a joke out of car chases for an hour and then gives us half an hour of car chases.", lorsque l'algorithme nécessite une phrase d'entrée.

Dans la plupart des cas, les phrases sont proches de la phrase d'origine, ce qui n'est pas surprenant compte tenu de la nature des algorithmes d'augmentation. Nous profitons également de cette analyse pour étiqueter manuellement certains des exemples générés (50 par classes, par algorithme, pour un total de 1400 phrases), afin de voir si des erreurs sont produites, et si oui, à quelle fréquence et par quels algorithmes. La table 3.6 présente le décompte des erreurs par classe et par algorithme. Une erreur est définie ici comme une phrase de la mauvaise classe ou dont la classe est ambiguë.

La conclusion la plus évidente est qu'il ne semble pas y avoir de lien entre le nombre d'erreurs générées par un algorithme d'AD et son efficacité, à condition évidemment que cela reste en deça d'un certain seuil. Notamment, EDA et EDA-SD, qui performent de façon similaire, se retrouvent avec une performance bien différente si l'on regarde les erreurs générées. T5 génère presque autant d'erreurs que CGPT,

Input	makes a joke out of car chases for an hour and then gives us half an hour of car chases.
EDA	makes a joke out dirigeant of car chases for an hour and then gives us half an hour of car chases.
EDA-SD	makes a joke out of car chases for an hour then gives us half an hour chases.
AEDA	makes , a joke out of ? car chases for an ! hour and , then gives us ! half an hour of : car , chases .
VAE-Sep	it's just merely very bad.
VAE-Linked	it's not just a bad premise, just a bad movie.
CVAE	a sour attempt to please with a farrelly of a movie that wasn't much to begin with a crucial third act miscalculation.
VAE-Par	an ill-conceived jumble that ' s not scary and not engaging .
BT	turns car chases into a joke for an hour and then gives us half an hour of car chases.
CBERT	is a whole series of car chases for an hour and it gives us all an hour on car chases.
CGPT	it's the kind of movie you'll never forget.
CBART	makes a joke out of car chases for an hour and then gives us a car chase.
T5-Tapaco	The car chases for half hour is a joke.
T5-Quora	Is it possible to make a joke out of car chases for an hour?
T5BT	He has a joke out of car chases for a hour and then gives us half an hour of car chases.
ChatGPT-P	It turns car chases into a comedic spectacle for an entire hour, followed by another 30 minutes of non-stop car action.
ChatGPT-D	The film was a major disappointment, lacking any coherent plot or engaging characters.
Llama2-P	The movie takes a lighthearted approach to car chases, with comedic moments and thrilling action sequences that last for an hour.
Llama2-D	Underwhelming experience with too much repetition.

Tab. 3.5. Exemples de phrases générées par les différents algorithmes considérés. Quand l'algorithme a besoin d'être entraîné, nous utilisons 1000 phrases du jeu de données SST-2, et lorsque l'algorithme ne prend pas de phrase en entrée (VAE-Sep, VAE-Link, CVAE, GPT), nous générons de la classe négative. Les stratégies de ChatGPT/Llama2 sont décrites à la section 3.5.5.

mais performe bien mieux. CVAE ne génère aucune erreur, tout comme l'AEDA, mais sa performance est moindre.

	Neg	Pos		Neg	Pos
EDA	22	8	BT	8	2
EDA-SD	8	4	CBERT	34	20
AEDA	0	0	CGPT	10	24
VAE-Sep	2	2	CBART	10	2
VAE-Link	4	8	T5-Tapaco	28	2
CVAE	0	0	T5-Quora	34	20
VAE-Par	40	56	T5BT	2	2

Tab. 3.6. Pourcentage d'erreurs dans les exemples générées, où une erreur est une phrase de la mauvaise classe ou de la classe ambiguë. 50 exemples par classe sont étiquetés et le jeu de données SST-2 avec une taille de noyau de 1000 est utilisé pour générer les exemples.

La table 3.7 montre des exemples d'erreurs générées par les algorithmes. En regardant les phrases, nous pouvons identifier certaines tendances. Le CVAE, qui ne fait pas d'erreurs, génère beaucoup de variations de la même phrase (dans ce cas-ci des variations de "the movie is undone by a filmmaking methodology that's not experimental to alienate the mainstream audience ringing ringing cliched to hardened indie-heads."). Cela explique pourquoi son nombre d'erreurs est bas, mais aussi pourquoi sa performance l'est : il apporte au final peu de variation au jeu de données. En général, les modèles génératifs de type VAEs vont créer des variations de phrases existantes, joignant des phrases différentes ensemble pour en faire une sorte d'interpolation.

CBERT, qui génère beaucoup d'erreurs, semble avoir de la difficulté avec les phrases courtes. Cela est logique puisque CBERT se base sur le contexte pour générer de nouveaux mots, contexte qui apporte peu d'information lorsque la phrase fait

EDA	not bond film goes off the beaten path , this necessarily for the better .
	... in no way original , or even all that memorable destination , but as downtown saturday matinee brain candy , it does n' t disappoint destination .
EDA-SD	wo if you ' an re elvis person , you even n' t find anything to get excited about on this dvd .
	this is the that disney movies are made of .
VAE-SEP	it ' s a mindless action flick with a twist - - far better than the multiplex .
	a special kind of movie , this melancholic noir me me me , is a thriller , and the performances are odd and pixilated .
VAE-Link	mr . wollter ms ms seldhal give strong and convincing performances , but the deepest recesses of the character to unearth the quaking essence of passion , grief and fear .
	a searing epic treatment treatment blight seems nationwide blight that seems to be on the rise .
VAE-PAR	... pays tribute to heroes the way julia roberts hands out awards - - but it ' s so successful , and you ' ll wonder a good job of the first .
	it ' s just merely bad bad .
BT	You can see almost immediately that Collinwood won't moan.
	This Disney cartoon has a subversive element that causes unexpected confusion.
CBERT	- garb. a - pb - b.
	a film of the friendship that men are, and women will talk to for them,
CBART	this bond film is on the right path, not necessarily for you.
	it's not worth catching solely on its own.
CGPT	if nothing else, " rollerball " 2002 may go down in cinema history as the
	-lrb- clooney's -rrb- debut can be accused of
T5	I pay tribute to her heroes like julia roberts gives out awards.
	The original wasn't a lot of any other saturday night party.
T5-Quora	Is it hard to connect with a puzzle?
	What is your favorite wedding and why?
T5BT	A chiller with chills.
	I'm introspective and panoramic.

Tab. 3.7. Exemples de phrases erronées (de la mauvaise classe ou de classe ambiguë), pour SST-2 et une taille de noyau de 1000. La première ligne pour chaque algorithme est un exemple "négatif" généré par l'algorithme et la deuxième, un exemple "positif".

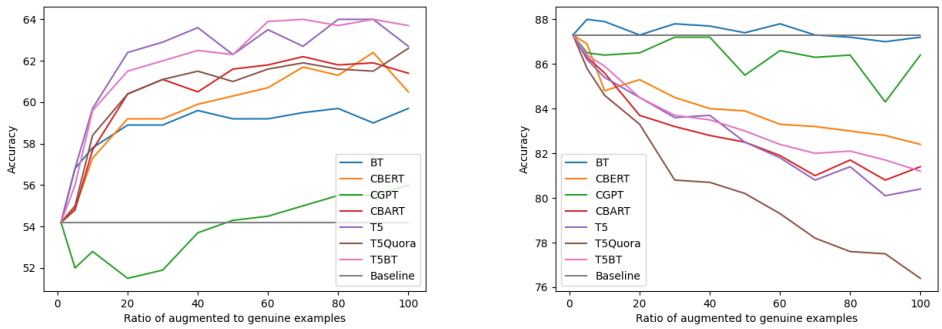

Fig. 3.3. Influence du paramètre de ratio pour les tailles de noyau de 10 (gauche) et 500 (droite), pour SST-2.

seulement 2 ou 3 mots. Finalement, CGPT génère des phrases qui sont plutôt bonnes, ce qui est surprenant vu la performance médiocre de l'algorithme. Si l'on observe les performances en détail, on peut voir que cela est dû surtout à sa performance sur les petites données, où les algorithmes génératifs en général ont de la difficulté à bien performer.

3.5.2. Impact du ratio sur la performance de l'AD

Dans cette section, nous justifions certains de nos choix concernant le paramètre du ratio. Tout d'abord, nous montrons qu'une taille de départ inférieure bénéficie de ratios plus élevés puisque, dans la majorité des cas, la performance relative des algorithmes reste la même, peu importe le ratio utilisé. Ce point est important, car il permet de justifier l'utilisation d'un ratio non idéal lors de nos expériences (par exemple, nous utilisons un ratio de 10 pour la taille de départ de 10, mais lors de nos expériences, nous voyons qu'un ratio de 60 ou 70 fonctionne mieux). Puisque nous nous intéressons ici à l'impact relatif des algorithmes, utiliser un plus petit ratio ne modifie pas les résultats.

La figure 3.3 montre les résultats de différents ratios (de 1 à 100), pour les tailles de départ de 10 et 500, et pour une sous-sélection des algorithmes. Comme mentionné, nous voyons que pour la taille de noyau de 10, la performance continue à augmenter avec le ratio jusqu'à un ratio d'environ 60. À l'opposé, pour la taille de départ de 500 la performance diminue au fur et à mesure que le ratio augmente, de façon plutôt constante, pour tous les algorithmes.

3.5.3. Impact de la taille du noyau

Nos expériences montrent que l'efficacité de l'AD dépend grandement de la taille du noyau, notamment au niveau de deux phénomènes. Premièrement, plus le noyau est gros, moins la performance globale semble efficace. Puis, certains types d'algorithmes fonctionnent mieux sur des grosses tailles de noyau, notamment les algorithmes génératifs et pré-entraînés (CBERT, CGPT, etc), et d'autres au niveau des petites tailles de noyau (EDA, CBART, etc). Pour obtenir une image plus claire du phénomène, nous utilisons des tailles de noyau de 10, 20, 50, 75, 100, 200, 300, 400 et 500, en ajustant le ratio à chacune pour obtenir des résultats optimaux.

Les résultats sont présentés à la figure 3.4, et un rendement rapidement décroissant pour tous les algorithmes est facilement observable. Bien qu'il soit facile d'obtenir une augmentation significative à des tailles inférieures à 100, à partir de 300, nous pouvons voir que le résultat de référence devient aussi efficace, voir plus par moment, que les résultats avec augmentation.

3.5.4. Piste d'analyse quantitative

Comprendre pourquoi l'augmentation de donnée fonctionne (ou pas) est un problème complexe qui reste encore sans réponse claire. Il est évident qu'il s'agit d'une forme de régularisation, qui vient renforcer le réseau neuronal, mais comment

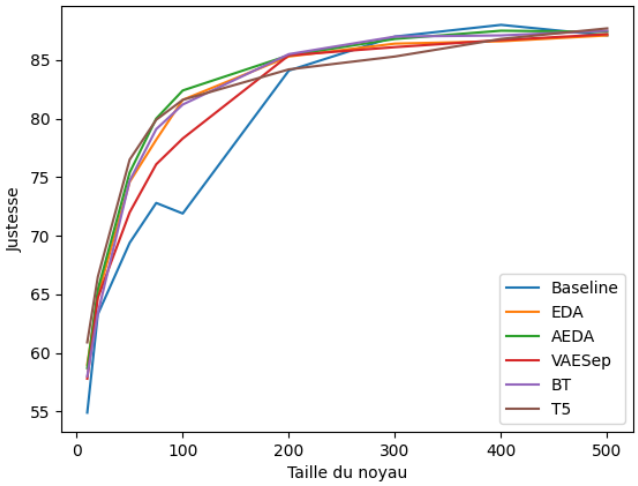

Fig. 3.4. Impact de la taille du noyau sur la performance pour SST-2.

savoir quels exemples seront les plus efficaces à cet effet ? Les résultats obtenus dans cette thèse offrent déja quelques pistes de réflexion, notamment :

(1) La performance exceptionnelle d'AEDA nous indique que la fluidité des phrases n'est pas importante, et le fait que T5-Tapaco performe mieux que ChatGPT malgré des phrases moins fluides vient renforcer cette idée.

(2) Le nombre d'exemples erronés ne semble pas corrélé à la performance, au vu des résultats de la table 3.6.

(3) La performance moyenne des algorithmes génératifs versus les manipulations par mots ou par phrases indique que soit 1- des exemples trop éloignés du jeu de données ne sont pas informatifs, ou 2- les indices de classes disparaissent si la fluidité est trop mauvaise.

Approfondir la compréhension de l'AD ouvrirait la voie à la génération d'exemples plus pertinents et à une réduction significative des coûts d'annotation. Cette section

détaille diverses tentatives d'analyse quantitative qui ont été mises en œuvre et évaluées.

Tout d'abord, l'utilisation de techniques de filtrage a été tentée, afin d'essayer de déterminer quelles données étaient utiles ou pas. Parmi les techniques que nous avons essayées se trouvent le filtrage des données doubles, l'utilisation de la distribution du jeu de données (garder les données générées qui apportent, ou pas, de la diversité), utiliser la valeur du gradient d'un exemple généré pour déterminer si on le garde, ou utiliser des techniques de similarité comme le TF-IDF, pour garder les exemples générés proche des exemples du noyau (haute fidélité) ou éloignés (haute diversité). Malheureusement, aucune de ces techniques n'a apporté un changement significatif aux résultats de l'augmentation.

Les fonctions d'influences ont également été considérées [Koh et Liang, 2020], fonctions qui permettent de déterminer l'impact d'une donnée d'entraînement sur la décision du réseau neuronal. Cela aurait permis de voir sur quel genre d'exemple le réseau se fie lorsqu'il prend une décision. Nos recherches nous laissent cependant penser que bien que ces fonctions soient utiles pour les petits réseaux linéaires, les résultats semblent moins cohérents pour les gros réseaux pré-entraînés que nous utilisons dans ce chapitre.

Finalement, nous avons également étudié les techniques de distillation, où le filtrage des données est parfois appliqué [Kamalloo *et al.*, 2021; Wang *et al.*, 2022]. Malheureusement, les techniques utilisées reposent généralement sur l'utilisation de données massives, et nos tentatives de transférer ces techniques aux petites données n'ont amené que des résultats négatifs.

Au vu des échecs de ces techniques, l'analyse qualitative semble rester le meilleur moyen d'analyser les résultats de l'AD. L'augmentation de données reste, même aujourd'hui, un phénomène peu compris au niveau des réseaux neuronaux. Il est

probable qu'il faudra cependant attendre que les techniques d'explicabilité soient plus avancées avant de mieux comprendre l'AD.

3.5.5. ChatGPT et Llama2

Le développement récent des modèles de langues a apporté un grand changement dans l'état de l'art du TAL. Notamment, ChatGPT et Llama2 ont bouleversé ce qui était connu par rapport à la construction de jeux de données, et certains chercheurs se sont même demandés si leur existence annonçaient la fin de l'annotation humaine [Kuzman *et al.*, 2023; Gilardi *et al.*, 2023]. Cette section décrit les expériences qui ont été faites avec ChatGPT et publiées à EMNLP 2023 [Piedboeuf et Langlais, 2023], et réplique ces expériences avec Llama2.

Nous reprenons une partie des algorithmes testés à la section précédente (EDA, AEDA, BT, CBERT, CGPT, CBART, et T5), et testons deux stratégies avec les VLLMs : la création de paraphrases et la génération de nouvelles phrases à partir des descriptions du jeu de données et de la classe voulue (les messages de sollicitations exacts sont donnés en annexe). Nous roulons les expériences pour les petites données (10 et 20 avec un ratio de 10) et moyennes données (500 et 1000 avec un ratio de 1). Nous notons aussi que nous avons raffiné le processus de nettoyage de FakeNews pour en faire une tâche plus difficile et plus proche de la réalité, ce qui diminue globalement les résultats. La tâche originale laissait souvent les noms des journaux intégrés dans les titres "ex : California Today: $8 Million in Tech Money for the Bay Area Arts - The New York Times", ce qui en fait une tâche mixte entre prédire si l'article est fiable et mémoriser quels journaux sont bons ou non. Dans cette section, les titres des journaux sont enlevés. Nous notons également que le processus d'entraînement a été optimisé, en affinant les hyper-paramètres de façon plus précise, ce qui mène à une hausse globale des résultats.

	SST2	FakeNews	Irony	IronyB	Trec6	Average
Baseline	56.4/60.6	52.2/53.3	54.4/57.8	24.4/24.6	27.9/34.7	43.0/46.2
Perfect	87.4/85.2	63.4/68.7	62.0/64.3	38.0/43.2	54.8/77.9	61.1/67.9
EDA	59.4/63.2	55.0/56.6	56.2/57.4	27.8/29.6	30.8/43.7	45.8/50.1
AEDA	59.3/64.6	53.4/55.3	**56.4**/58.1	28.3/28.8	28.9/44.2	45.3/50.2
BT	59.0/64.6	55.1/56.2	55.8/58.1	**28.5**/28.5	32.9/**46.3**	46.3/50.7
CBERT	57.6/63.1	54.5/55.5	**56.4**/**58.9**	**28.5**/29.5	29.0/40.6	45.2/49.5
CGPT	55.6/61.2	52.4/54.7	54.2/54.8	25.0/26.8	23.3/38.1	42.1/47.1
CBART	60.5/64.9	**55.8**/**57.2**	55.8/57.9	28.4/29.3	**34.1**/46.1	46.9/51.1
T5	60.4/64.6	54.5/56.6	54.4/57.2	27.2/29.5	34.0/**46.3**	46.1/50.8
GPT3.5-P	62.5/69.0	53.8/54.9	55.2/57.8	28.3/29.8	31.3/44.8	46.2/51.2
GPT3.5-D	**78.6**/**82.6**	51.5/52.8	53.6/55.9	27.6/28.0	31.4/42.9	**48.5**/**52.4**
Llama2-P	64.9/52.9	52.2/54.7	55.0/56.4	27.3/**29.9**	30.1/40.1	45.9/46.8
Llama2-D	74.5/73.9	53.9/56.7	53.9/55.7	22.9/24.1	32.0/44.8	47.4/51.1

Tab. 3.8. Résultats sur les petites données (10,20) pour les expériences EMNLP avec un ratio de 10.

Finalement, nous rajoutons un résultat de référence, que nous dénommons “Perfect”, qui consiste à simuler l’ajout de données externes étiquetées. Pour ce faire, nous allons simplement chercher dans le jeu de données des données non utilisées et nous les rajoutons à l’entraînement. Cette stratégie nous permet d’établir une borne supérieure sur la performance des stratégies de description.

Les résultats sont montrés aux tables 3.8 et 3.9. On peut voir que les techniques de paraphrases performent au même niveau que les meilleurs algorithmes, malgré le fait que les phrases générées sont généralement plus fluides.

	SST2	FakeNews	Irony	IronyB	Trec6	Average
Baseline	87.7/88.8	73.3/77.0	69.6/74.9	52.1/61.5	81.0/85.4	72.7/77.5
Perfect	88.7/89.8	77.1/80.4	75.0/84.5	60.9/66.9	86.7/91.1	77.7/82.5
EDA	87.9/88.9	73.7/77.6	69.5/75.6	52.5/63.3	81.3/86.1	73.0/78.3
AEDA	88.0/89.0	73.5/77.6	69.4/75.5	52.1/63.0	**82.7**/86.4	73.1/78.3
BT	**88.2**/**89.1**	73.6/77.4	69.7/**76.2**	52.6/**63.6**	81.7/86.1	**73.2**/78.5
CBERT	87.5/88.3	73.6/77.5	68.4/76.0	49.8/61.5	80.9/85.4	72.1/77.7
CGPT	87.8/88.7	73.2/77.6	68.0/75.1	51.2/61.1	82.1/**87.2**	72.4/77.9
CBART	87.7/88.6	**73.9**/**77.9**	69.1/75.9	51.9/61.0	78.6/83.4	72.2/77.4
T5	87.9/88.7	73.8/73.3	69.0/74.8	**52.9**/60.7	79.9/85.0	72.7/76.5
GPT3.5-P	**88.2**/**89.1**	73.8/77.7	69.2/76.0	52.4/63.1	82.4/87.1	**73.2**/**78.6**
GPT3.5-D	87.4/88.9	71.9/75.9	68.7/75.0	50.5/62.5	79.8/84.0	71.7/77.3
Llama2-P	87.8/88.9	73.4/77.4	68.4/74.9	51.3/62.2	81.8/86.3	72.5/77.9
Llama2-D	88.1/88.8	73.5/77.1	**68.8**/74.9	51.6/61.7	79.4/85.7	72.3/77.6

Tab. 3.9. Résultats sur les moyennes données (500, 1000) pour les expériences EMNLP avec un ratio de 1.

En ce qui concerne la stratégie de description, Llama2 et ChatGPT affichent tous deux des performances exceptionnelles sur SST-2, bien que Llama2 soit en général moins efficace. Cependant, pour les autres jeux de données, ces deux algorithmes sont surpassés par les performances des autres méthodes. Comme nous le montrons à la section 3.5.6, en observant les jeux de données, nous avons découvert des biais importants qui empêchent de solliciter de façon efficace les VLLMs. Cependant, le résultat de référence “Perfect” et les résultats positifs sur SST-2 montrent que si l’on

est capable de générer correctement des données variées de la même distribution, cela reste la stratégie la plus efficace.

3.5.6. Biais des jeux de données

La première hypothèse pour justifier la basse performance des stratégies de description est que certains jeux de données sont compliqués à modéliser. Notamment, FakeNews et Irony se retrouvent à être des jeux difficiles, FakeNews parce que la classification se base sur des indices fallacieux (de par la définition de tâche) et Irony parce que la tâche en elle-même est complexe. Cependant, cela ne constitue pas une explication complète, car TREC6 est un jeu de données qui devrait être simple à modéliser.

En observant plus précisement les jeux de données et les phrases générées, nous notons qu'il existe un biais qui, pour tous les jeux de données sauf SST-2, fait que la tâche *voulue* est différente de la tâche *réelle*. Ces biais viennent principalement de défaults/problèmes dans la construction des jeux de données.

Pour Irony et IronyB, deux jeux de données venant de la compétition SemEVAL 2018, les données ont été moisonnées de Twitter en cherchant pour une série de hashtags (#Irony, #not, #sarcasm), hashtags qui ont ensuite été enlevés pour former la classe Irony[11]. Pour la classe non-ironique, d'autres gazouillis ont été moissonnés[12]. Cette technique cause cependant un biais majeur qui change la tâche considérablement. En effet, sans les hashtags qui donnent un contexte, beaucoup des textes deviennent neutres. La table 3.10 montre des exemples qui pourraient

[11]Bien que les exemples aient été manuellement ré-étiquettés par des étudiants, le biais vient du processus de la collection de données. Nos expériences du chapitre 6 nous confirment que, sachant le processus de collection, l'annotation est facile (en se demandant si un hashtag #Irony pourrait aller avec le gazouillis).

[12]La totalité des données ont été moissonnées entre 2014 et 2015.

Ironic	Shoutout to my mom for being hella supportive of me
Ironic	Luv this
Non-Ironic	@alyssaanicoleL this Friday lit
Non-Ironic	they don't sing live, but they sure are hella good looking #smh

Tab. 3.10. Exemples de phrases du jeu de données Irony qui sont ambiguës, selon notre évaluation manuelle.

facilement être ironiques ou ne pas l'être. Comme la tâche IronyB est une déclinaison de Irony, elle garde le même biais.

TREC6 est un jeu de données de la conférence Text REtrieval Conference et consiste en une classification des questions en six catégories différentes. Bien que toutes les données de la tâche aient été annotées manuellement, de nombreuses incohérences ont été observées dans les annotations. Par exemple, la question "What is narcolepsy?" est étiquetée comme Description, mais "What is a fear of motion?" comme Entité. D'autres ambiguïtés sont par exemple "What is the oldest profession?" et "What team did baseball's St. Louis Browns become?" étiquetées comme Humain versus "What do you call a professional map drawer?" comme Entité, ou encore "Where did Indian Pudding come from?" étiqueté comme Description mais "Where does chocolate come from" comme un Lieu. Comme les mêmes ambiguïtés existent dans le jeu de test (ex "Where does dew come from?" étiquetée comme un lieu), les VLLMs générant des phrases de la classe décrite ne vont pas aider à résoudre les ambiguïtés du jeu de données. Il est à noter que ces problèmes ont déjà été soulevés par Li et Roth [2002] qui recommandent d'utiliser une classification multilabel pour réduire l'impact de l'ambiguïté sur le classificateur. Cependant, dans toutes les recherches sur l'AD où nous avons recensé l'utilisation de TREC6, la tâche utilisée était de la classification directe.

Finalement, FakeNews est une tâche Kaggle qui a été reprise dans plusieurs recherches sur la détection de fausses nouvelles. Nous avons décidé d'utiliser ce jeu de données, car la tâche est intéressante et difficile, mais nous avons trouvé plusieurs biais qui changent la définition de tâche, à l'instar d'Irony. De ce que nous avons pu comprendre de la création du jeu de données, les nouvelles ont été collectées de plusieurs sources et ensuite divisées en nouvelles réelles ou fausses selon la source. Cela cause un biais, car bien que certains journaux ont une tendance à publier de la désinformation, cela n'indique pas que toutes les nouvelles de ce journal sont fausses. De plus, nous avons trouvé certains choix étranges dans l'étiquetage. Par exemple, tous les articles de Breitbart sont étiquetés comme de vraies nouvelles, même si le score de "factual reporting" est mixte[13] , alors que Consortium, qui reçoit le même score global, est étiqueté comme faux[14].

Finalement, en améliorant les messages de sollicitation (pour essayer de mieux les faire correspondre à la tâche), nous pouvons augmenter le score sur TREC6 à 68.6%, ce qui sous-performe tout de même comparé à BERT. Dans nos expériences, nous avons observé que ChatGPT avait de la difficulté à comprendre les concepts de questions sur les entités et les humains, les étiquetant souvent comme description à la place.

3.5.7. Est-ce que les résultats sont statistiquement significatifs ?

Nous avons jusqu'ici suivi les protocoles utilisés de façon classique dans la littérature, en utilisant un arrêt précoce et en rapportant les moyennes ainsi que les déviations standards. Cependant, au vu de la variance élevée ainsi que les gains assez modestes observés, nous nous interrogeons sur la significativité des résultats, et sur si un des algorithmes performe statistiquement mieux que les autres.

[13] `https://mediabiasfactcheck.com/breitbart/`

[14] `https://mediabiasfactcheck.com/consortium-news/`

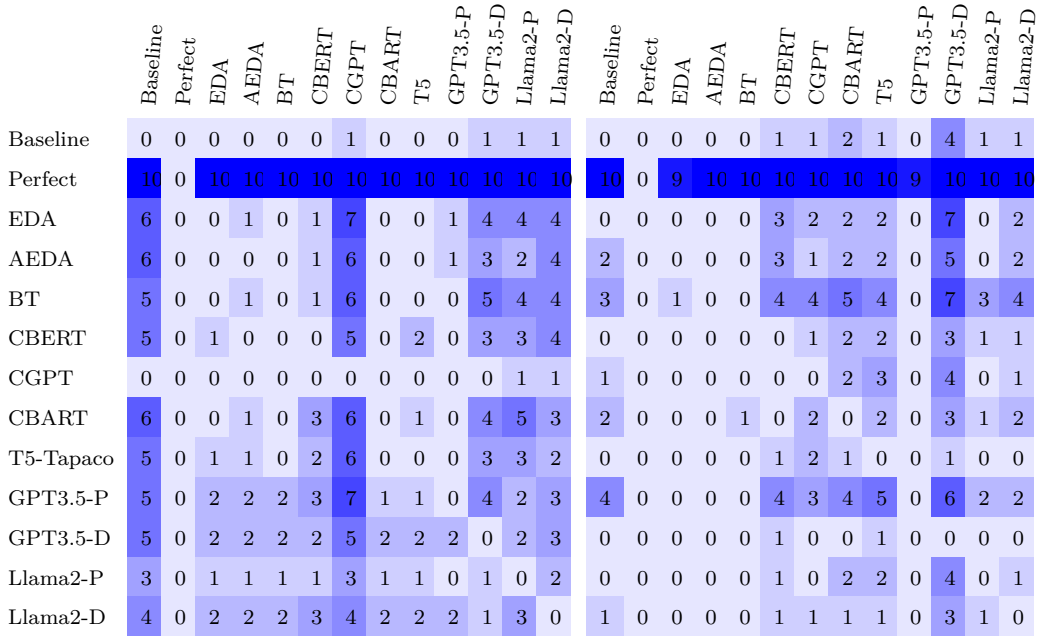

Fig. 3.5. Nombre de fois que l'algorithme de la ligne réussit mieux que l'algorithme en colonne de façon statistiquement significative, avec un seuil de p-value de 0.05.

Pour déterminer cela, nous calculons des "two-tailed paired t-test" entre chaque paire de résultats et entre chaque algorithme (une paire de résultats étant par exemple les 15 résultats AEDA sur SST-2 et les 15 résultats pour EDA sur SST-2).

Nous reprenons les résultats de la section précédente, qui sont les résultats les plus à jour, et nous regroupons les résultats par domaine (10 et 20 données initiales pour le *few-shot learning* et 500 et 1000 pour classification avec moyennes données. Les résultats sont présentés à la figure 3.5, et chaque entrée représente le nombre de fois où l'algorithme de la ligne bat l'algorithme de la colonne **et** que le test statistique indique que la différence est significative.

À notre surprise, nous constatons que, statistiquement, il n'y a que deux classes d'algorithmes : ceux qui surpassent le résultat de référence de manière assez constante et ceux qui sous-performent (CGPT). En comparant les algorithmes entre eux, nous

pouvons également remarquer qu'aucun algorithme ne semble surpasser constamment les autres. Les meilleurs résultats sont obtenus par ChatGPT-D et LLama2-D, mais uniquement grâce à leurs hautes performances sur SST-2.

Par ailleurs, les tests avec des tailles initiales de 500 et 1000 révèlent que, généralement, les gains de performance ne sont pas statistiquement significatifs. Cela suggère que, dans la majorité des cas, les améliorations pourraient résulter davantage du hasard que d'une réelle optimisation du processus d'apprentissage.

À ce stade, nous tenons à souligner que cela ne signifie pas que ces algorithmes sont de *mauvaises* méthodes d'augmentation de données, mais plutôt qu'aucune méthode d'augmentation de données ne fonctionne de manière constante sur tous les jeux de données, une distinction importante. Nous tenons également à souligner que ces résultats sont calculés en utilisant 15 répétitions, ce qui est la limite supérieure de ce que nous avons trouvé dans la littérature [Kumar *et al.*, 2021]. Il est très possible qu'en augmentant le nombre de répétitions, une image plus claire pourrait émerger et certains algorithmes se montreraient alors plus efficaces que d'autres.

3.6. Conclusion

L'augmentation de données textuelles est un domaine de recherche très actif, dû notamment aux avantages qui peuvent être amenés pour les industries. Les contributions qui sont présentées au cours de ce chapitre sont les suivantes : tout d'abord, nous testons plusieurs types de modèles génératifs de types VAE pour l'augmentation de données et concluons qu'avoir un modèle par classe est plus efficace que d'avoir un modèle unique de type conditionnel ou des techniques de types encodage-décodage. Nous testons également de nouveaux algorithmes de type transformer, comme T5, et montrons que leurs performances égalent celle de BT, l'algorithme le plus populaire pour l'AD, tout en étant plus facile à utiliser. Nous produisons l'étude la plus complète sur les méthodes d'AD, comparant les algorithmes

les plus efficaces entre eux sur plusieurs jeux de données. Finalement, nous analysons certains aspects critiques de l'AD qui ont été laissés de côté dans la littérature, notamment au niveau du nombre d'erreurs générées et du ratio utilisé.

Dans le prochain chapitre, nous utilisons la connaissance acquise pour tester l'augmentation de données sur des jeux non anglais, ce qui nécessite une adaptation des algorithmes puisque beaucoup d'entre eux utilisent des données externes qui ne sont disponibles que dans certaines langues.

Chapitre 4

Augmentation de données pour les allolangues

4.1. Introduction

Comme le démontre le chapitre précédent, ces dernières années en apprentissage automatique ont été marquées par un grand intérêt pour l'augmentation de données, une technique qui permet la génération de nouvelles données synthétiques sans étiquetage. Cependant, la plupart des techniques testées et recensées reposent sur des ressources externes, telles que des dictionnaires ou des plongements de mots pré-entraînés, ce qui pourrait limiter leur utilisation sur des langues autres que l'anglais (appelées dans ce chapitre *allolangages* ou *allolangues*), et particulièrement lorsque l'on regarde au niveau des langues rares comme l'inuktitut ou le swahili. Même sachant cela, l'adaptation de l'AD aux allolangues est sous-étudiée et notre revue de littérature n'a révélé aucune étude se penchant sur la question de quelle méthode fonctionne le mieux, similairement à la constatation faite pour l'anglais dans le chapitre précédent. Compte tenu de l'intérêt actuel de la recherche sur les systèmes multilingues et de la difficulté de collecter des données étiquetées dans plusieurs allolangages rares, il s'agit d'un domaine de recherche important qui mérite plus d'attention.

Ce chapitre est conçu comme une première étude de l'AD pour les allolangues, avec un accent sur les tâches de classification. Nous sélectionnons et adaptons cinq méthodes présentées dans le chapitre précédent, à savoir Back-translation (BT) [Hayashi *et al.*, 2018; Yu *et al.*, 2018], AEDA [Karimi *et al.*, 2021], VAE-Sep [Piedboeuf et Langlais, 2022a; Qiu *et al.*, 2020], CBERT [Wu *et al.*, 2019], et EDA [Wei et Zou, 2019; Liesting *et al.*, 2021], et les testons sur quatre jeux de données dans différentes langues (français, allemand, coréen, swahili) et avec différentes tailles de noyau, dans le but de voir lesquels de ces algorithmes fonctionnent mieux sur les allolangues, et d'analyser les résultats selon les langues.

Ce chapitre apporte deux contributions par rapport à la compréhension de l'AD sur les allolangues. Les expériences démontrent que, premièrement, les méthodes d'AEDA et d'AEDA-R surpassent les autres techniques, et deuxièmement que les méthodes qui n'utilisent pas de données externes (VAE-Sep, EDA-SD, et AEDA), fonctionnent mieux que celles qui le font (CBERT, EDA, et BT). Bien que le deuxième point semble être une conclusion naturelle, il n'est pas représenté dans la littérature sur l'AD dans les allolangues, où BT est l'une des techniques les plus couramment employées. Cette utilisation du BT est similaire aux résultats des expériences pour l'anglais, avec la différence importante que pour l'anglais, les expériences montrent que BT reste une des méthodes les plus efficaces.

Le chapitre est organisé comme tel. La littérature pertinente sur l'augmentation de données pour les allolangages est présentée à la section 4.2. Puis, dans les sections 4.3 et 4.4, les méthodes utilisées et leurs adaptations aux allolangages sont décrites, et nous présentons également les jeux de données utilisés. Les résultats et l'analyse sont présentés dans les sections 4.5 et 4.6, avant de conclure à la section 4.7.

4.2. Revue de littérature

Bien que les études sur le phénomène de l'AD pour les allolangues soient minimes, la technique en elle-même reste couramment utilisée sur les jeux de données non anglophones pour augmenter les performances. Dans le chapitre précédent, la revue de littérature présentait les différentes techniques existantes pour l'augmentation de données pour la classification de texte. Dans cette section, nous présentons les recherches qui ont essayé d'appliquer ces mêmes méthodes aux allolangues.

En raison de la diversité des langues, des tâches, des modèles et de la taille des ensembles de données, comme montrée dans le tableau 4.1, il est difficile de tirer une conclusion claire sur ce qui fonctionne, les études ayant souvent des résultats contradictoires. Cependant, et comme les résultats de ce chapitre le démontrent, montrer qu'un algorithme est efficace en anglais ne veut pas dire qu'il est pour autant efficace sur des langues qui ont moins de ressources.

La taxonomie utilisée dans le chapitre 4 est conservée dans ce chapitre, c'est-à-dire une catégorisation en trois familles principales : la création de paraphrases, les opérations au niveau des mots et les modèles génératifs. Tout comme pour l'AD anglophone, le *Back-translation* (BT) est le représentant principal de la famille des paraphrases. Ma et Li [2020] l'utilisent sur plusieurs tâches de classification en chinois et montrent que BT surpasse systématiquement l'EDA. Pour l'ensemble de données de taille moyenne qu'ils étudient (574 exemples), ils montrent une amélioration de 1,2% sur un CNN et de 1,4% sur un RNN. En revanche, l'étude de Vu *et al.* [2022] compare le BT et l'EDA pour de la similarité sémantique en coréen et pour de l'inférence du langage naturel (NLI), et les auteurs concluent que dans la plupart des cas, l'EDA fonctionne mieux. L'utilisation du BT pour la détection d'émotions (multiclasse) en espagnol a été étudiée sur des ensembles d'entraînement assez larges (5723 phrases) [Luo, 2021]. Les auteurs rapportent une amélioration d'environ 3%

sur l'ensemble de développement, mais ne rapportent pas les résultats de référence sur l'ensemble de test, ce qui rend difficile d'évaluer l'impact de l'AD. Son utilisation pour d'autres tâches a également été explorée, comme pour le NER arabe [Sabty *et al.*, 2021] où les auteurs combinent BT avec des substitutions de mots, ou encore pour la similarité sémantique en arabe [AlAwawdeh et Abandah, 2021], montrant une augmentation de la performance.

Comme mentionné précédemment, les opérations sur les mots ont aussi été étudiées pour l'augmentation sur les allolangues, telles que l'adaptation de l'EDA pour la détection d'agressivité en espagnol [Guzman-Silverio *et al.*], qui démontre un gain de 1,6% en utilisant à la fois un modèle d'ensemble et l'EDA. La méthode W2V, qui implique le remplacement de mots par des voisins issus d'un plongement de mots pré-entraîné, a aussi été mise en œuvre, notamment dans les travaux de Marivate *et al.* [2020]. Les auteurs expliquent comment ils ont constitué un corpus de nouvelles en Setswana et Sepedi, ont entraîné des plongements de mots dans ces langues, puis ont utilisé des représentations similaires pour substituer des mots dans les phrases. Les performances sont seulement rapportées graphiquement, et il est donc difficile de déterminer exactement l'impact de l'AD bien que les résultats semblent fluctuer entre -5% et 10% d'augmentation de la performance. Un désavantage de cette méthode est cependant le fait qu'il faut recueillir un corpus externe suffisamment gros pour entraîner un plongement de mot W2V.

L'utilisation des modèles génératifs est intéressante car ces modèles ne demandent pas de ressources externes, et peuvent donc fonctionner, à priori, sur n'importe quelle langue. Chang *et al.* [2019] génèrent de nouvelles phrases pour la modélisation du langage avec de l'alternance codique (*code-switching*)[1] (entre le mandarin et l'anglais). Pour ce faire, un GAN est utilisé pour générer des phrases multilingues

[1]L'alternance codique fait référence à un phénomène où le langage utilisé alterne rapidement entre deux ou plusieurs langages.

à partir de phrases monolingues, et les auteurs montrent une amélioration des performances en utilisant cette méthode. Les GAN ont également été utilisés pour générer de nouvelles phrases pour l'identification des dialectes arabes [Carrasco *et al.*, 2021], en entraînant un GAN pour chaque dialecte afin de pouvoir générer de nouvelles données, similairement à la stratégie VAE-SEP du chapitre précédent. Cependant, Carrasco *et al.* [2021] utilisent de grands ensembles de données pour entraîner les GANs (environ 61 000 données partagées entre les classes), ce qui est beaucoup plus volumineux que ce qui est disponible pour la plupart des applications d'AD. L'utilisation des VAEs pour l'augmentation pour allolangages a de même été explorée, par exemple dans la classification des intentions et l'étiquetage des créneaux en paraphrasant avec un VAE conditionnel [Panda *et al.*, 2021], obtenant également des améliorations d'environ 2% sur la classification en aval. La table 4.1 présente un résumé des méthodes recensées, résumé qui aide à montrer l'incohérence entre les différentes conclusions que les recherches atteignent, selon le jeu de données utilisé, de la taille des données de départ, et du classificateur.

4.3. Adaptation aux allolangues

Nous décrivons dans cette section l'adaptation aux allolangues de chacune des méthodes choisies (AEDA, EDA, VAE-Sep, CBERT et BT). Nous sélectionnons EDA et BT puisqu'ils sont souvent utilisés pour l'AD sur les allolangages. En tant que membre des méthodes génératives, nous choisissons d'utiliser VAE-Sep, car les approches avec des GANs qui ont montré de bonnes performances sont entraînées sur plusieurs milliers d'exemples, tandis que nous étudions l'AD pour des ensembles de données limités. Finalement, nous testons également AEDA et CBERT, qui sont des algorithmes populaires pour l'augmentation de données en anglais et qui ont montré de bons résultats.

Approche	Tâche	Langue	Taille	Gain
BT [Ma et Li, 2020]	Classification non-balancée	Chinois	[574, 4000]	2.3%
BT [Luo, 2021]	Détection d'émotion	Espagnol	5723	1.1%
BT+Ensemble [AlAwawdeh et Abandah, 2021]	Similitude sémantique	Arabe	[11K, 45K]	[3.1%, 4.1%]
BT et EDA [Sabty *et al.*, 2021]	NER	Arabic	5306	1.51%
BT et EDA [Vu *et al.*, 2022]	Multiple	Coréen	[1725, 3927]	[-1.1%, 2.2%]
EDA+Ensemble [Guzman-Silverio *et al.*]	Détection d'agressivité	Espagnol	5865	0.8%
W2V [Marivate *et al.*, 2020]	Classification de nouvelles	Setswana, Sepidi	[219, 491]	[-5, 10]
AEDA [Karimi *et al.*, 2021]	Classification	Anglais	[500,5000]	[0.4%, 3.2%]
CBERT [Wu *et al.*, 2019]	Classification de texte	Anglais	[5000, 1000]	2.0%
CBERT [Kumar *et al.*, 2021]	Classification de texte	Anglais	10	[3.4%, 16.1%]
GAN [Chang *et al.*, 2019]	Génération de *code-switching*	Mandarin, Anglais	[12K, 94K]	-
GAN [Carrasco *et al.*, 2021]	Identification de dialecte	Arabe	61K	[0.5%, 1.1%]
CVAE [Panda *et al.*, 2021]	Systèmes de dialogue	Multiple	5871	-

Tab. 4.1. Résumé de certaines des approches d'AD qui ont été utilisées dans le passé. La comparaison est souvent difficile en raison de nombreux facteurs, notamment les différentes tâches, métriques ou classificateurs utilisés, ce qui met en évidence la nécessité d'études centralisées concernant l'AD pour les allolangues.

Comme montré au chapitre précédent, AEDA fonctionne en insérant des signes de ponctuations aléatoires (parmi ”?”, ”.”, ”;”, ”:”, ”!” et ”,”) dans le texte, et donc aucune modification n'est nécessaire pour l'augmentation de données multilingue. La tokénisation se fait en utilisant les espaces, mais une adaptation supplémentaire devrait être faite pour des langues telles que le chinois qui n'utilisent pas les espaces comme délimiteur de mot, par exemple en utilisant un tokenizer spécialisé [Rust *et al.*, 2021]. Nous testons également AEDA-R, la version d'AEDA où AEDA est appliqué lors de l'affinage du modèle, en transformant au hasard la moitié des phrases de chaque itération.

Le chapitre précédent testait deux versions d'EDA : EDA standard (utilisant les quatre opérations) et EDA-SD (seulement les opérations de suppression et de changement de place). EDA-SD peut être utilisé sans modification supplémentaire, à condition que la langue se tokenise aux espaces blancs. EDA, de son côté, est difficile à adapter aux allolangues en raison des ressources nécessaires pour le faire fonctionner (WordNet et une liste de mots d'arrêts). Pour adapter EDA, la liste de mots d'arrêts de Spacy[2] est utilisée lorsque disponible (non disponible seulement pour le swahili) et pour wordNet nous utilisons la version multilingue de WordNet de NLTK [3]. Malheureusement, sur les quatre langues considérées, seul le français est disponible pour WordNet, et l'anglais est utilisé lorsque la langue n'est pas disponible. La version multilingue de WordNet est disponible pour 32 langues et pourrait être une bonne option dépendamment du langage du jeu de données. Cependant les résultats, à la fois sur les allolangues et l'anglais (section 4.6) mènent à la conclusion que l'utilisation de la substitution et de l'insertion n'entraîne pas d'augmentation des performances significative par rapport à EDA-SD.

VAE-Sep fonctionne en entraînant un modèle génératif (à savoir, un VAE) sur chaque classe, ce qui permet la génération directe de nouveaux exemples. De par sa nature, il ne nécessite aucune modification pour fonctionner avec les allolangages. Nous utilisons VAE-Sep au lieu de VAE-Linked (version de VAE-Sep ou l'encodeur est commun à toutes les classes), car il n'existait pas de différences significatives

[2] `https://spacy.io/`

[3] `https://www.nltk.org/`

entre les deux et donc nous suivons le principe du rasoir d'Occam en choisissant le modèle le plus simple à implémenter et entraîner.

Finalement, BT et CBERT ont simplement besoin d'un système multilingue adapté aux allolangues. Nous utilisons `mbart-large-50` [Tang *et al.*, 2020] pour BT (qui est entraîné sur 50 langues) et l'anglais comme langue pivot, en supposant que les performances seront meilleures en raison d'une plus grande quantité de données disponibles pour l'anglais. La nature de `mbart-large-50` en fait un modèle avec lequel il est possible de traduire dans plusieurs langues sans affinage supplémentaire. Pour CBERT multilingue (dénommé mCBERT), le modèle mBERT [Devlin *et al.*, 2019] est utilisé au lieu de BERT, masquant 40% des mots de la phrase. mBERT est entraîné sur 102 langues et donc permet l'utilisation de CBERT sans difficulté supplémentaire. Le paramétrage pour tous les hyper-paramètres se fait de façon analogue au chapitre précédent, et les hyper-paramètres pour les expériences sont disponibles dans le github lié à ce chapitre[4].

4.4. Ensemble de données et protocole

Pour avoir une idée globale de la performance des algorithmes sur les différents jeux de données, quatre jeux dans quatre langues différentes sont utilisés, c'est-à-dire SB10k, un ensemble de données de classification des sentiments de Twitter en allemand [Cieliebak *et al.*, 2017] (avec les classes de tweets négatifs, neutres ou positifs), SwaNews, un ensemble de données de classification des actualités en swahili[5] (local, international, finance, santé, sports et divertissement), koHateSpeech, un ensemble de données sur la détection des discours haineux en coréen [Moon *et al.*, 2020] (discours haineu, neutre ou offensant), et CLS, un jeu de données de produits en

[4] `https://github.com/smolPixel/DAAllolangue-ICANN2023`

[5] `https://zenodo.org/record/5514203#.Y20HvblyZhE`

	CLS	SB10K	koHateSpeech	SwaNews
langues	Français	Allemand	Coréen	Swahili
nb. classes	2	3	3	6
nb. ex. d'entraînement	5400	5233	7106	21042
nb. ex. développement	612	748	790	1105
longueur phrase	23.6	13.5	8.4	31.9

Tab. 4.2. Tâches utilisées dans ce chapitre. La longueur des phrases est définie en fonction du nombre d'espaces.

français d'Amazon [Prettenhofer et Stein, 2010] (négatif ou positif). Pour ce dernier ensemble, comme les critiques de produits sont beaucoup plus longues que les textes des autres tâches (longueur moyenne de 103.8 mots), les données sont artificiellement raccourcies en ajoutant une phrase à la fois jusqu'à ce que la longueur soit supérieure à 20 mots (tokenisés aux espaces). Les caractéristiques des jeux de données sont présentés à la table 4.2.

Différentes tailles de noyau sont testées pour pouvoir voir l'impact global de l'AD selon le nombre de données d'entraînement disponibles. Nous testons sur des tailles de 100, 500, 1000 et 1500, ainsi qu'avec l'ensemble d'entraînement complet, et avec mBERT comme classificateur. Pour réduire les variations dues à la sélection de l'ensemble d'entraînement initial, chaque expérience est répétée 15 fois et nous rapportons la métrique moyenne ainsi que la plage des écarts-types. Pour CLS, une tâche binaire, la justesse est utilisée, alors que le macro-f1 est utilisé pour les tâches multiclasses. Finalement, les jeux de développements sont utilisées pour affiner mBERT, avec de l'arrêt précoce.

L'AD ne conduit généralement qu'à une augmentation mineure des performances lorsque la taille du noyau est au-delà d'un certain seuil [Dai et Adel, 2020], et donc

les résultats sur les jeux complets sont inclus par soucis de complétude plutôt qu'avec attente que l'augmentation de donnée donne des résultats positifs. Pour toutes les expériences, un ratio de un est utilisé, menant à un doublage de la taille des jeux de données. Dans la section 4.6, l'impact d'un ratio plus grand est examiné.

4.5. Résultats

La table 4.3 montre les résultats pour toutes les méthodes et toutes les tailles de départ et la table 4.4, la métrique moyenne par jeu de données. Globalement, nous pouvons observer que AEDA-R est de loin le meilleur algorithme.

Pour les autres algorithmes, il est observable que CBERT et BT sont les deux techniques les moins performantes, dégradant en moyenne les performances. C'est plutôt surprenant puisque, comme mentionné, BT est souvent la technique de pointe pour les allolangues et qu'elle donne des bons résultats en anglais. Si nous examinons les performances par langue, BT fonctionne bien sur CLS et KoHateSpeech, mais mal sur SB10K et SwaNews. Cela suggère que BT pourrait rester une bonne méthode d'augmentation selon la langue et, possiblement, de l'algorithme utilisé pour traduire. Cependant, son utilisation pour les allolangages demanderait de tester plusieurs systèmes et langages intermédiaires pour en trouver un qui fonctionne bien avec le langage de l'ensemble de données, nécessitant beaucoup plus de travail que certains des autres algorithmes.

EDA fonctionne bien pour koHateSpeech et SwaNews, et légèrement mieux en utilisant uniquement la méthode d'échange et de suppression, comme nous l'avions supposé plus tôt dans ce chapitre, mais mal sur CLS et SB10K. L'utilisation des opérations de substitutions et d'insertions, qui utilisent des ressources externes, semblent dégrader légèrement les résultats lorsque comparé avec l'utilisation uniquement des opérations d'échange et de suppression.

	100	500	1000	1500	Moyenne	Jeu complet
Référence	53.7	61.2	64.4	66.3	61.4	**72.9**
AEDA	54.8	63.0	**65.3**	66.5	62.4	72.8
AEDA-R	**55.6**	**63.2**	**65.3**	**66.9**	**62.8**	72.8
EDA	54.6	62.0	64.7	66.0	61.8	63.6
EDA-SD	54.6	62.0	65.2	66.4	62.1	63.5
VAE-Sep	54.5	61.7	65.0	66.6	62.0	72.6
mCBERT	53.8	61.4	64.2	65.5	61.2	71.6
BT	54.1	61.4	63.7	65.4	61.2	72.8

Tab. 4.3. Résultats moyens sur les quatre ensembles de données en fonction de la taille de l'ensemble d'entraînement. La dernière colonne (Jeu complet) n'est pas inclut dans la moyenne puisqu'elle présente un intérêt moins grand pour l'AD. Les écarts-types par langue sont de l'ordre de [0,5, 3,7] (fr), [0,6, 5,0] (de), [1,3, 3,2] (ko), [0,6, 16,8] (sw), avec les valeurs plus élevées de stds associées aux plus petites tailles de noyau. Les valeurs de 16,8 et 13,2 ont été obtenues par EDA et EDA-SD sur SwaNews. Les résultats **inférieurs** aux résultats de référence (pas d'augmentation) sont soulignés.

Finalement, même si VAE-Sep ne surpasse pas l'AEDA, il reste tout de même l'un des meilleurs algorithmes parmi toutes les options testées. En regardant les performances sur les ensembles de données individuels, nous pouvons voir que c'est en coréen que cela fonctionne le moins bien, n'apportant que 0,3% d'augmentation par rapport au résultat de référence. Cela pourrait être dû à la complexité de la langue coréenne, qui est peut-être plus difficile à modéliser pour VAE-Sep. Comme mentionné, un inconvénient de l'AEDA est que les données générées ne ressemblent

	CLS (fr)	SB10K (de)	KoHateSpeech (ko)	SwaNews (sw)
Référence	65.6	62.6	48.0	69.5
AEDA	66.3	62.9	48.9	71.4
AEDA-R	**66.6**	**63.3**	49.1	**72.0**
EDA	65.6	62.5	48.5	70.6
EDA-SD	65.4	62.5	**49.2**	71.2
VAE-Sep	66.0	63.1	48.3	70.5
mCBERT	65.4	62.6	47.8	69.0
BT	66.0	62.4	48.7	67.5

Tab. 4.4. Résultats moyens pour les quatre tailles de départ (100, 500, 1000, 1500). Les déviations standards sont les mêmes que dans le tableau 4.3. Les résultats soulignés sont les résultats sous le résultat de référence, les résultats en gras représentent les meilleurs résultats.

pas à des données humaines, donc VAE-Sep pourrait être une bonne alternative si l'interprétabilité des données est quelque chose d'important pour le jeu de données.

4.6. Discussion

4.6.1. Exemples de phrases générées et facilité d'utilisation

Nous montrons dans le tableau 4.5 des exemples de phrases générées par tous les algorithmes, pour le jeu de données CLS et une taille de départ de 1000. Il n'y a pas beaucoup de surprise dans les phrases générées, et AEDA, EDA et mCBERT fonctionnent exactement comme prévu. VAE-Sep construit des phrases qui ne sont pas toujours grammaticalement correctes, mais qui apportent de la variation par rapport aux phrases de l'ensemble d'apprentissage. Les résultats

AEDA	kiss . est une institution, c'est pas un scoop ! alors je ? me . suis , laissé tenter (avoir).
EDA	kiss est une fondation , c'est pas un scoop! alors je me suis laissé rendre (avoir).
EDA-SD	kiss c'une institution, est est pas un scoop! me je alors suis laissé tenter (avoir).
VAE-Sep	je suis très déçu par ce livre qui survole très très haut de l'angleterre.
mCBERT	kiss, une institution. c'etait pas le scoop, mais que je suis a tenter (avoir peur.
BT	Je me suis laissé tenter (avoir).

Tab. 4.5. Exemple de phrases générées pour le jeu de données CLS, avec une taille de départ de 1000 et pour la phrase d'entrée négative "kiss est une institution, c'est pas un scoop ! alors je me suis laissé tenter (avoir).", lorsque nécessaire (tous excepté VAE-Sep).

laissent penser qu'avoir des phrases grammaticalement incorrectes peut être une force pour l'augmentation de données. Enfin, BT dans ce cas génère une phrase beaucoup plus courte que celle d'entrée, mais semble préserver dans une certaine mesure la sémantique de la phrase.

4.6.2. Comparaison avec l'AD pour l'anglais

Il est pertinent de se demander comment les performances se comportent lorsque les algorithmes sont appliqués sur des jeux en anglais versus en allolangages. La table 4.6 rappelle les résultats du chapitre précédent pour faciliter la comparaison.

Comparé à l'AD pour l'anglais, une image différente émerge quant à l'efficacité des algorithmes. Tout d'abord, AEDA-R n'a pas l'air de performer de façon significativement différente de AEDA. Puis, à l'exception de VAE-Sep et CBERT, tous les algorithmes ont l'air globalement de performer de façon assez proche l'un de l'autre. Cela, selon nous, souligne le besoin d'études spécialisées sur les allolangues.

	100	500	1000	1500	Average	All data
Référence	60.9	74.9	79.7	83.0	74.6	90.6
AEDA	**65.4**	75.0	**80.0**	**83.2**	**75.9**	90.7
AEDAR	65.0	75.0	**80.0**	83.1	75.8	**90.8**
EDA	65.2	75.3	79.7	82.8	75.8	90.7
EDA-SD	64.7	**75.4**	79.8	82.8	75.7	90.7
VAE-Sep	62.8	74.9	79.4	82.3	74.8	90.4
CBERT	63.0	74.3	79.2	82.0	74.6	90.3
BT	65.0	75.2	79.9	83.3	75.8	90.6

Tab. 4.6. Moyenne des résultats sur les jeux anglais utilisés dans le chapitre précédent (SST-2, FakeNews, Irony, IronyB, et TREC6). Les résultats sous le résultat de référence sont soulignés et les meilleurs résultats sont en gras. Les valeurs des stds varient entre 0,4 et 3,2, les stds plus grandes étant associées à des tailles d'entraînement plus petites.

4.6.3. Nombre de données générées

Jusqu'à présent, les expériences de ce chapitre utilisaient un ratio de un, doublant la taille du jeu de donnée. Dans le chapitre précédent, nous montrons que le ratio idéal est généralement un paramètre de la taille du noyau plutôt que de l'algorithme ou du jeu de données. Afin de vérifier cela pour les allolangues, nous réexécutons tous les algorithmes avec une taille de départ de 1000 sur l'ensemble de données CLS, en utilisant un rapport entre exemples générés et exemples authentiques de 1, 2, 3, 4 et 5.

La figure 4.1 montre qu'il semble y avoir peu d'avantages à ajouter plus de données, à l'exception de EDA, EDA-SD, et BT pour qui cela apporte des gains

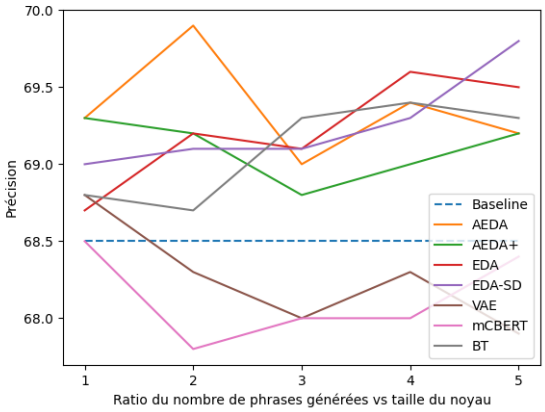

Fig. 4.1. Justesse vs le ratio d'exemples de départ vs générés sur CLS, avec une taille de départ de 1000, et pour tous les algorithmes utilisés dans ce chapitre.

faibles mais constants. Il est également à noter que dans certains cas, comme le VAE-Sep, ajouter plus de données semble nuire aux performances. Notre hypothèse pour justifier cela est que si l'algorithme se retrouve à générer des phrases trop loin de la distribution, le modèle se mélange quant à quelle distribution prioriser lorsque le nombre d'exemples artificiels devient trop grand par rapport aux exemples originaux.

4.7. Conclusion

Bien qu'elle soit parfois utilisée dans la recherche, l'augmentation des données pour les allolangues manque d'études où les différentes méthodes sont comparées sur des jeux de données communs. Dans ce chapitre, cinq techniques populaires d'augmentation de données sont adaptées aux allolangues et testées sur plusieurs jeux de données, afin d'estimer lequel des algorithmes pourrait être le plus efficace.

Selon les résultats obtenus, AEDA pourrait être la meilleure option pour effectuer de l'augmentation des données, mais il est possible que d'autres techniques soient plus efficaces ou même que certaines de nos hypothèses soient fausses, comme l'utilisation

de l'anglais comme langue pivot pour BT. Nous espérons que la recherche effectuée suscitera plus d'intérêt pour l'AD multilingue, un domaine de recherche essentiel en raison de la difficulté d'obtenir des données annotées dans les allolangues.

Chapitre 5

Augmentation de données pour la génération de mots-clés

Alors que les chapitres précédents exploraient l'augmentation de données pour des tâches de classification, ce chapitre étudie l'AD pour une autre application : la génération de mots-clés, et plus spécifiquement la génération de mots-clés académiques francophones. Ce domaine est un cas d'étude intéressant pour deux raisons : il représente un problème réel pour lequel les données existantes sont minimes, et la spécificité des données demandent l'utilisation d'algorithmes d'AD différents que ceux utilisés dans le chapitre 4, ce qui permet d'appuyer la difficulté de développer des techniques universelles pour l'AD.

Les mots-clés sont définis comme des unités lexicales d'un ou de plusieurs mots qui permettent de résumer le contenu d'un document [Gallina *et al.*, 2019]. Ces mots-clés sont utiles pour une pléthore d'applications, tels l'indexation de documents [Medelyan et Witten, 2006], le regroupement de documents (*document clustering)* [Hammouda *et al.*, 2005], la création automatique de résumés [Qazvinian *et al.*, 2010] ou encore l'extraction d'opinion [Berend, 2011]. Pour en donner un exemple, les mots-clés représentant cette thèse sont `Augmentation de données, modèles génératifs, MQS, petites données,`

`jeux de données, apprentissage supervisé, traitement des langues, intelligence artificielle`.

Malgré l'utilité indéniable des mots-clés, la recherche sur leur génération automatique tourne principalement autour de la langue anglaise, un constat récurrent dans cette thèse. Les jeux de données non anglophones sont rares et contiennent peu de données, ce qui empêche l'entraînement d'algorithmes efficaces pour le KPG (*keyphrase generation*). Le but de ce chapitre est donc double. Dans un premier temps, nous créons un nouveau jeu de données pour la génération de mots-clés francophones, et dans un second temps, nous comparons différentes méthodes d'augmentation de données, démontrant que l'augmentation de performance des capacités *génératives* mènent souvent à une détérioration des capacités *extractives* des méthodes, et que les méthodes existantes apportent souvent des gains minimes.

Le chapitre est organisé comme suit. La section 5.1 définit le problème et présente la littérature pertinente, puis la section 5.2 décrit le processus de collecte de données. Les sections 5.3 et 5.4 présentent le protocole, les résultats de références, et les résultats de nos expériences sur l'AD, suivies par une discussion à la section 5.5.

5.1. Définition de la tâche et revue de littérature

Les travaux pertinents à notre recherche peuvent être séparés en deux catégories : les travaux sur le KPG en anglais, qui a reçu beaucoup d'attention, et les travaux sur le KPG et l'extraction de mots-clés (KPE) pour les allolangues, cette dernière branche de recherche n'ayant commencé à être sérieusement explorée que récemment. La différence entre ces deux techniques est que le KPE tente uniquement d'extraire les mots-clés *présents*, c'est-à-dire, les mots-clés qui se trouvent dans le texte en entrée. Des exemples de techniques pour la tâche du KPE sont l'utilisation du TF-IDF pour extraire les termes importants [Li *et al.*, 2007] ou encore l'entraînement d'un BERT pour délimiter les mots-clés [Priyanshu et Vijay, 2022], similairement

à ce qui est fait en question-réponse. La tâche de KPG, quant à elle, considère également les mots-clés *absents*, c'est-à-dire les mots-clés qui ne sont pas présents dans le texte. Ces mots-clés représentent souvent une proportion importante des mots-clés écrits par les auteurs des thèses et articles [Meng *et al.*, 2017], et sont une source d'information pertinente pour les tâches en aval, telles la classification ou la recherche de documents.

Comparer les différents systèmes entre eux est complexe, car l'évaluation du KPG est elle-même limitée. En effet, il n'existe pas de critère universel qui indique si un mot-clé est bon ou non, et même en comparant aux références, plusieurs critères viennent affecter la qualité de l'évaluation. À titre informatif, voici quelques questions qui pourraient se poser pour une évaluation :

(1) Est-ce qu'un modèle qui génère peu de mots-clés, mais de haute qualité, est meilleur qu'un modèle qui en génère beaucoup de qualité moyenne?

(2) Est-ce que les mots-clés "génération de texte" et "génération de textes" sont équivalents ?

(3) Est-ce que le mot-clé "génération de textes francophone" est suffisamment similaire aux mots-clés précédents pour recevoir un bon score ?

(4) Comment évaluer un mot-clé qui est bon pour le texte, mais qui n'est pas dans la référence ?

Ces questions sont, pour la plupart, sans réponse, mais il est important de les garder à l'esprit lors de l'application du KPG. Au fil du temps, les techniques d'évaluation ont tout de même progressé afin de tenter d'observer objectivement la performance des modèles.

Les métriques utilisées aujourd'hui sont la précision, le rappel et le F1@K, où K dénote le nombre de mots-clés considérés. Soit un ensemble de mots-clés références $\mathcal{Y}$ et un ensemble de mots-clés $\hat{\mathcal{Y}}$ générés, ces métriques sont calculées de la façon suivante :

$$P@K = \frac{|\hat{\mathcal{Y}} \cap \mathcal{Y}|}{|\hat{\mathcal{Y}}|}, \quad R@K = \frac{|\hat{\mathcal{Y}} \cap \mathcal{Y}|}{|\mathcal{Y}|}, \quad F_1@K = \frac{2 * P@K * R@K}{P@K + R@K} \tag{5.1.1}$$

Autrement dit, la précision est la fraction du nombre de mots-clés générés qui sont dans la référence, le rappel est la fraction du nombre de mots-clés de la référence qui sont générées et le F1 est la moyenne harmonique des deux.

Au départ, l'évaluation se faisait avec des K différents d'une étude à l'autre, en séparant les mots-clés présents et absents. À titre d'exemple, Meng *et al.* [2017] utilisent le F1@5 et @10 pour les mots-clés présents, et le R@10 et @50 pour les mots-clés absents. Chowdhury *et al.* [2022] utilisent le $F_1@5$ et @10 pour les mots-clés présents, et le R@10 pour les mots-clés absents.

Cette multiplication des métriques rend cependant la comparaison difficile, et donc récemment la communauté a adopté un standard selon lequel les métriques sont rapportées selon le $F_1@5$ et le $F_1@M$, où M correspond au nombre de mots-clés générés, et toujours en séparant les métriques pour les mots-clés présents et les mots-clés absents. Nous rapportons dans cette section le $F_1@M$, qui permet d'avoir une meilleure idée des performances que le $F1@5$. Si le $F1@M$ n'est pas disponible, nous précisons alors quelle métrique est rapportée, et nous notons les résultats sur Kp20K [Meng *et al.*, 2017], un jeu de données de 500K articles du domaine de l'informatique, qui est le seul corpus utilisé de façon consistante à travers la littérature. Il est à souligner que ces résultats peuvent ne pas être représentatifs de la performance globale, qui est souvent évaluée sur une myriade de corpora en aval de l'apprentissage, tel Inspec [Hulth, 2003], Krapivin [Krapivin *et al.*, 2009], NUS [Nguyen et Kan, 2007] ou encore SemEval-2010 [Kim *et al.*, 2010].

5.1.1. KPG pour l'anglais

L'histoire du KPG anglophone démontre de manière frappante l'importance d'avoir un corpus large et facilement accessible, la recherche commençant uniquement après la création et la publication de Kp20K [Meng *et al.*, 2017], un corpus pour le KPG contenant environ 500K exemples.[1] Les premières recherches sur le KPG utilisaient principalement des modèles Seq-to-Seq de type récurrent, chaque article apportant une petite amélioration, comme CatSeqD [Yuan *et al.*, 2020], qui manipule les états cachés du décodeur pour améliorer la diversité, atteignant 36.0 F_1@5 présents et 11.7 R@10 absents, ou Meng *et al.* [2019, 2021] qui montrent que l'ordre des mots-clés dans les réseaux Seq-to-Seq est important pour l'entraînement, et que le réseau apprend mieux si les mots-clés sont ordonnés de façon à ce que les mots-clés présents soient générés en premier, suivis des mots-clés absents (29.0 F_1@10 présents et 14.0 R@50 absents). Chan *et al.* [2019] étudient l'utilisation de l'apprentissage par renforcement (38.6 présents, 5.0 absents[2]), Ahmad *et al.* [2021]; Mahata *et al.* [2022] le KPG pour des documents plus longs (37.9 présents, 3.6 absents), Çano et Bojar [2019] l'utilisation de techniques de résumé automatique pour mieux générer les mots-clés, et Swaminathan *et al.* [2020a,b] l'utilisation de GANs pour encourager la génération de mots-clés absents (37.8 présents, 4.5 absents).

L'utilisation de transformers a suivi, par exemple avec Ye *et al.* [2021] qui proposent SetTrans, un modèle transformer entraîné à générer un ensemble non ordonné de mots-clés plutôt qu'une séquence, éliminant ainsi l'importance de l'ordre des mots-clés dans l'équation (39.2 présents, 5.8 absents). Récemment, l'attention s'est tournée vers l'utilisation de modèles de langue pré-entraînés. Chowdhury *et al.*

[1] Nous référons à Xie *et al.* [2023] pour une revue plus complète de la littérature.

[2] Comme mentionné, lorsque la métrique n'est pas précisée, nous utilisons le F_1@M rapporté par les auteurs.

[2022] montrent que l'utilisation de BART peut donner des résultats équivalents pour le KPG tout en étant beaucoup plus facile à utiliser (31.1 F1@10 présent, 6.1 R@10 absent), ce qui est confirmé par Wu *et al.* [2022a] (38.8 présents, 4.7 absents), qui ajoutent qu'avoir un transformer spécialisé pour le domaine, tel que SciBERT [Beltagy *et al.*, 2019], ou le SciBART qu'ils entraînent, est plus efficace qu'utiliser un modèle générique (39.6 présents, 5.2 absents). SciBART a été entraîné sur OAGKX [Çano et Bojar, 2020], un jeu de données de KPG de 22 millions d'entrées en anglais, avant d'être affiné sur Kp20K pour évaluation. Kulkarni *et al.* [2022] proposent un autre système d'affinage qu'ils appellent KeyBART (39.8 présents, 4.3 absents) et utilisant également l'ensemble de données OAGKX. Ce modèle performe cependant moins bien que SciBART, bien que la différence soit minime. Dans un autre travail récent, Wu *et al.* [2022b] examinent l'utilisation de BART pour le KPG dans de petits domaines de données, en modifiant l'objectif d'entraînement classique pour que le décodeur se concentre sur les phrases saillantes dans l'encodeur. Avec 20K exemples d'entraînement, ils obtiennent 35.4 présents et 3.8 absents.

Song *et al.* [2023] testent l'utilisation de ChatGPT pour le KPG et démontrent son potentiel pour la tâche (16.5 présents, 2.5 absents), malgré le fait qu'il ne surpasse pas les modèles spécialisés pour le KPG. Cette conclusion est partagée par Wu *et al.* [2023], où les auteurs proposent et examinent différentes métriques pour le KPG et concluent que pour le domaine des actualités et selon certaines métriques non-supervisées, telles que la naturalité ou la diversité, ChatGPT surpasse les modèles spécialisés. Cependant, il est difficile de comparer cet article aux autres en raison des métriques spécialisées qu'ils utilisent. D'autres chercheurs cependant, par exemple Martínez-Cruz *et al.* [2023], rapportent une conclusion contraire où ChatGPT obtient de meilleurs résultats que le SOTA, et particulièrement pour les longs documents. Cependant, contrairement aux autres articles de KPG, les auteurs ne séparent pas les métriques entre les mots-clés présents et les mots-clés absents,

mais séparent plutôt les résultats entre documents longs et documents courts, ce qui rend difficile une comparaison directe.

Il est à noter que plusieurs autres systèmes ont été proposés pour le KPG, notamment les GANs pour les domaines à ressources faibles [Lancioni *et al.*, 2020], l'étude de KPG pour les documents plus longs [Ahmad *et al.*, 2021] ou l'utilisation de techniques de génération de résumés automatiques [Çano et Bojar, 2019]. Le KPG a également été exploré pour d'autres domaines, notamment l'actualité [Gallina *et al.*, 2019] et le commerce [Gao *et al.*, 2022]. Ces travaux ont tendance à utiliser les mêmes algorithmes que dans le domaine du KPG académique, les entraînant simplement sur des jeux de données du domaine spécifique.

5.1.2. KPG et KPE pour les allolangues

L'aspect des mots-clés pour les allolangues a majoritairement été exploré pour la tâche d'extraction des mots-clés (KPE, ou *Keyphrase Extraction*), la principale différence étant que le KPE ne s'intéresse qu'à la récupération de mots-clés *présents* dans l'entrée. Les recherches en KPE se basent toutefois sur une gamme diversifiée de corpus, rendant ainsi les comparaisons directes entre leurs résultats problématiques.

De nombreuses techniques pour le KPE utilisent des fonctionnalités indépendantes de la langue, par exemple YAKE [Campos *et al.*, 2020], une méthode statistique utilisant des attributs comme la fréquence ou la position normalisée des termes, ou singleRank [Wan et Xiao, 2008b], qui modifie TextRank [Mihalcea et Tarau, 2004] pour l'extraction de mots-clés. D'autres travaux utilisent également des systèmes multilingues, tels que KeyBERT [Grootendorst, 2020], une technique utilisant BERT pour extraire des n-grammes importants. Ce n'est que récemment que des recherches ont commencé à examiner KPG dans d'autres langues, avec Gao *et al.* [2022] qui présentent des corpus pour la génération de mots-clés multilingues,

pour le KPG de commerce (en français, espagnol, allemand et italien) et pour le KPG académique (en coréen et chinois).

Dans ce chapitre, nous collectons d'abord un corpus pour le KPG académique francophone, moissonnant les données de *Papyrus*, le dépôt de thèse de l'Université de Montréal. Puis, nous nous intéressons à la performance de l'augmentation de données sur ce corpus. Nous notons que bien que ce chapitre se concentre sur les données francophones, des expériences ont également été réalisées sur les données anglophones et multilingues, comme nous le détaillons dans [Piedboeuf et Langlais, 2022b].

5.1.3. Augmentation de données

Peu d'articles se sont intéressés à l'augmentation de données pour la tâche de KPG ou de KPE, et la plupart des études le faisant reposent sur l'utilisation de corpora massifs externes pour venir augmenter la taille du jeu de données [Liu *et al.*, 2018; Gero et Ho, 2021; Shvets et Wanner, 2020; Veyseh *et al.*, 2022], ce qui est ultimement peu utile pour les allolangues [Feldman et Coto-Solano, 2020].

Pour le KPE, Mahfuzh *et al.* [2019] montrent que remplacer les mots par des synonymes permet une grande augmentation de la performance par rapport au résultat de référence. Pour le KPG, deux techniques à notre connaissance ont été développées. KPDrop [Ray Chowdhury *et al.*, 2022] masque certains des mots-clés présents, les transformant ainsi en mots-clés absents et forçant ainsi le modèle à se concentrer sur ce type de mots-clés. Les auteurs montrent sur plusieurs algorithmes de génération qu'appliquer KPDrop donne une augmentation d'environ 1% présent et absent. KPSR [Garg *et al.*, 2023] remplace les mots-clés présents par des synonymes, encourageant ainsi la génération de mots-clés absents tout en gardant la phrase naturelle. Les auteurs ne regardent pas la performance sur Kp20K, mais étudient

plutôt les performances sur des petites données, montrant une légère augmentation de la performance (moins de 1% présent, jusqu'à 1.5% absent).

Ces articles représentent, à notre connaissance, l'intégralité des méthodes existantes pour combiner l'AD et le KPG. Dans ce chapitre, nous tentons d'utiliser ces méthodes pour notre corpus ainsi que certaines méthodes novatrices, afin de voir ce qui pourrait impacter positivement la performance.

5.2. Collecte du corpus

La première étape est d'assembler un corpus de mots-clés francophones, la tâche qui nous intéresse ici. Pour ce faire, nous tournons notre regard vers Papyrus[3], un dépôt de documents institutionnel, comportant majoritairement des thèses mais également d'autres documents mis en ligne par des membres du corps professoral, étudiant ou administratif. La plupart de ces documents sont accompagnés de méta données qui permettent l'identification facile des mots-clés et des résumés, les deux données qui nous intéressent. Les mots-clés, notamment, sont écrits par les auteurs des thèses, assurant ainsi une haute qualité et une grande diversité [Zhao *et al.*, 2022].

Bien que les lignes directrices de l'Université de Montréal indiquent que la rédaction des thèses devrait être faite en français[4], ceci n'est pas appliqué de façon consistante, et nombres d'auteurs choisissent également d'avoir plusieurs résumés et mots-clés dans diverses langues. Pour compliquer la collecte, le fait que les mots-clés et résumés sont écrits par les auteurs mène parfois à un non-alignement des langues des mots-clés et résumés (par exemple, des mots-clés peuvent être présents en français alors que le résumé l'est seulement en anglais). Un exemple d'une thèse de maîtrise est montré à la figure 5.1.

[3] `https://papyrus.bib.umontreal.ca/xmlui/`

[4] Information prise du Guide de présentation des mémoires et thèses

Personality extraction through LinkedIn

dc.contributor.advisor	Langlais, Philippe	
dc.contributor.advisor	Lapalme, Guy	
dc.contributor.author	Piedboeuf, Frédéric	
dc.date.accessioned	2019-11-19T19:23:16Z	
dc.date.available	NO_RESTRICTION	fr
dc.date.available	2019-11-19T19:23:16Z	
dc.date.issued	2019-10-30	
dc.date.submitted	2019-05	
dc.identifier.uri	http://hdl.handle.net/1866/22536	
dc.subject	Extraction de personalité	fr
dc.subject	MBTI	fr
dc.subject	DiSC	fr
dc.subject	LinkedIn	fr
dc.subject	Réseau sociaux	fr
dc.subject	Profilage d'auteur	fr
dc.subject	Personality Extraction	fr
dc.subject	Social Network	fr
dc.subject	Author Profiling	fr
dc.subject.other	Applied Sciences - Computer Science / Sciences appliqués et technologie - Informatique (UMI : 0984)	fr
dc.title	Personality extraction through LinkedIn	fr
dc.type	Thèse ou mémoire / Thesis or Dissertation	
etd.degree.discipline	Informatique	fr
etd.degree.grantor	Université de Montréal	fr
etd.degree.level	Maîtrise / Master's	fr
etd.degree.name	M. Sc.	fr
dcterms.abstract	L'extraction de personnalité sur les réseaux sociaux est un domaine qui n'a que récemment commencé à capturer l'attention des chercheurs. La tâche consiste à, en partant d'un corpus de profils d'utilisateurs de réseaux sociaux, être capable de classifier leur personnalité correctement, selon un modèle de personnalité tel que défini en psychologie. Ce mémoire apporte trois innovations au domaine. Premièrement, la collecte d'un corpus d'utilisateurs LinkedIn. Deuxièmement, l'extraction sur deux modèles de personnalités, MBTI et DiSC, l'extraction sur DiSC n'ayant pas encore été faite dans le domaine, et finalement, la possibilité de passer d'un modèle de personnalité à l'autre est explorée, dans l'idée qu'il serait ainsi possible d'obtenir les résultats de multiples modèles de personnalités en partant d'un seul test.	fr
dcterms.abstract	Personality extraction through social networks is a field that only recently started to capture the attention of researchers. The task consists in, starting with a corpus of user profiles on a particular social network, classifying their personalities correctly, according to a specific personality model as described in psychology. In this master thesis, three innovations to the domain are presented. Firstly, the collection of a corpus of LinkedIn users. Secondly, the extraction of the personality according to two personality models, DiSC and MBTI, the extraction with DiSC having never been done before. Lastly, the idea of going from one personality model to the other is explored, thus creating the possibility of having the results on two personality models with only one personality test.	fr
dcterms.language	eng	fr

Fig. 5.1. Exemple des métadonnées disponibles dans Papyrus.

Nous moissonnons les 26508 pages web qui étaient en ligne au moment de la collecte, le 7 avril 2022. De ces 26508 pages, 657 étaient des pages d'erreurs et 9602 des documents sans résumé ou sans mot-clés, laissant un total de 16249 documents

utilisables. Ces documents sont composés d'un ou plusieurs résumés et de plusieurs mots-clés. Cependant, bien que Papyrus identifie les langues de ces mots-clés et des résumés, cette identification donne généralement l'étiquette de *Français* à toutes les entrées, peu importe la véritable langue. Ce phénomène est reflété à la figure 5.1 où à la fois les mots-clés anglais (Author Profiling, Personality Extraction, etc) et le résumé anglophone sont identifiés comme francophone. Afin d'utiliser ces données, il nous faut donc identifier les langues des résumés et des mots-clés.

Le protocole que nous adoptons est le fruit de nombreuses expériences antérieures, qui nous ont permis de déterminer la méthode la plus efficace pour l'identification des langues. Pour identifier la langue des résumés, nous utilisons `langdetect`[5], et nous développons une heuristique simple pour les mots-clés. Soit un mot-clé m et les résumés $r_{1,2,..,n}$ d'un document écrit dans n langues différentes, pour chaque r_i dans lequel m est présent, m est assigné à r_i. Plusieurs mots-clés, comme des noms propres ou des noms de molécules, peuvent être attribués à plus d'une langue et donc cette attribution permet de ne pas se limiter. Pour les mots-clés absents cependant, nous utilisons `fasttext`[6] pour identifier la langue et prenons la plus probable qui correspond également à une langue d'un résumé. Pour 119 des mots-clés, le top-15 des langues identifiés ne correspond à aucun des résumés (par exemple avec "間 ma", "1000", or "Leptoquark"). Dans ce cas, nous considérons que les langues associées à ces mots-clés sont l'ensemble des langues des résumés du document.

Il est à noter également que nous enlevons automatiquement deux types de mots-clés qui viennent des standards UMI et JEL de classification (par exemple le mot-clé "Applied Sciences - Computer Science / Sciences appliqués et technologie - Informatique (UMI : 0984)" ou "Philosophy / Philosophie (UMI : 0422)"). Nous

[5] `https://pypi.org/project/langdetect/`

[6] `https://fasttext.cc/docs/en/language-identification.html`

décidons de les retirer puisque 1) ces mots-clés correspondent davantage à la tâche d'indexation[7] que de KPG, et 2) dans un contexte de génération de mots-clés, ces standards ne représentent pas la sortie généralement désirée.

Étant donné que cette approche admet la possibilité d'erreur, nous regardons 100 exemples au hasard et vérifions manuellement si les langues des résumés et mots-clés ont été correctement identifiés. Nous trouvons une justesse de 100% pour les résumés et de 98.9% pour les mots-clés. Sur l'ensemble des documents, il y a 1761 documents monolingues, 9391 bilingues, 134 trilingues, 3 quadrilingues et un qui a six langues. Pour ce qui est des langages utilisés, il y a 15289 résumés anglophones, 14826 francophones, 172 espagnol, 29 en allemand, 20 en italien, 17 en portugais, 7 en arabe, 5 en tagalog, 3 en catalan et greque, 2 en turque et russe, et 1 en polonais, farsi, indonésien, lingala, suédois, finlandais, roumain et coréen. Deux résumés étaient dans un langage que `langdetect` n'a pas réussi à identifier. Au meilleur de notre jugement, l'un de ces résumés est en Inuktitut, et l'autre est une erreur où le texte "2002-10" a été marqué comme un résumé, en plus du résumé anglais et français. Pour vérifier qu'aucun autre artefact de la sorte n'a été introduit, nous vérifions manuellement que les autres entrées courtes sont des résumés corrects de la thèse.

Finalement, une fois cette séparation en langues faites, nous pouvons définir les deux tâches qui nous intéressent. Nous montrons un exemple fictif des tâches à la table 5.1.

- **Papyrus-f** : Génération automatique de mots-clés francophones à partir des résumés francophones.
- **Papyrus-e**: Génération automatique de mots-clés anglophones à partir des résumés anglophones.

[7]La tâche d'indexation consiste à assigner des mots-clés à un texte, parmi un ensemble prédéterminé de mots-clés.

Tâche	Résumé	Mots-clés
Papyrus-f	Ce document parle de l'extraction et de la génération de mots-clés francophone et multilingue.	Extraction de mots-clés, génération de mots-clés, tâche francophone, tâche multilingue
Papyrus-e	This document is about keyphrase generation and extraction, for French and multilingual corpora.	keyphrase generation, keyphrases extraction, Multilingual keyphrases, document indexing.

Tab. 5.1. Exemple fictif d'un document bilingue et de la séparation des résumés et mots-clés pour les deux tâches. Les mots-clés *présents* sont soulignés.

Nous séparons les documents en ensemble d'entraînement, de développement et de test, avec un ratio 70/10/20. Le nombre de données pour chaque corpus et chaque ensemble est montré à la table 5.2, et différentes statistiques sont présentées à la table 5.3. Pour valider notre corpus et les approches utilisées, nous nous comparons également à Kp20K, qui est le corpus d'usage pour le KPG anglophone.

Jeu de données	**#Train**	**#Dev**	**#Test**
Papyrus-f	10299	1488	2981
Papyrus-e	10508	1539	3046

Tab. 5.2. Nombre d'exemples dans chaque ensemble du jeu de données (entraînement, développement, test) et pour chaque jeu de données.

Papyrus présente plusieurs caractéristiques intéressantes par rapport aux corpus existants. Tout d'abord, et le plus important, est le fait qu'il vient en version francophone et multilingue (voir Piedboeuf et Langlais [2022b] pour une description plus complète de la tâche multilingue), ce qui n'existe pas dans la littérature. Ensuite,

Jeu de donnée	long. résumés	#kp	long.kp	% présent kp	% Pr. Kp. brisé
Kp20K	148	5.3	2.1	50.9	60.8
Papyrus-f	323	7.0	1.9	65.0	72.2
Papyrus-e	290	7.4	1.7	60.8	67.6

Tab. 5.3. Statistiques pour les différentes tâches. Les valeurs représentent la moyenne sur tous les exemples du jeu d'entraînement. *Pr. KP. brisé* représente les mots-clés *brisés*, c'est-à-dire, les mots-clés où tous les mots sont individuellement présents dans le résumé, mais pas nécessairement de façon contigus.

par rapport à Kp20K, le corpus qui est le plus utilisé dans la littérature de KPG, Papyrus est multidomaine. Kp20K se concentre majoritairement sur le domaine de l'informatique, ce qui en fait un corpus mal adapté au KPG général.

Il est également pertinent de s'attarder aux mots-clés présents dans le corpus, au delà des statistiques présentées à la table 5.3. En comptant les fréquences des mots-clés, on peut voir que relativement à sa taille, Papyrus est beaucoup plus divers. Par exemples, Papyrus-f présente 3.7 mots-clés différents par entrée (total 38392 mots-clés différents), Papyrus-e, 3.9 (total 41368) et Kp20K, 1.4 (total 720348). Le nombre de mots-clés qui apparaissent juste une fois est de 29208 pour Papyrus-f, 31161 pour Papyrus-e et 510044 pour Kp20K.

5.3. Modèles et résultats de références

Nous nous concentrons ici sur le corpus francophone, qui est plus intéressant au niveau de l'AD et de la nouveauté présentée.

Avant de nous lancer dans l'AD, nous vérifions dans un premier temps la pertinence de Papyrus-f, puis validons nos algorithmes en comparant les résultats avec ceux obtenus sur le corpus Kp20K.

Comme mentionné, les études sur le KPG pour les allolangues reposent surtout sur les méthodes extractives, et nous désirons savoir si avoir un corpus large abstractif permet une augmentation des performances. Nous reprenons donc les meilleurs systèmes extractifs de Giarelis *et al.* [2021], qui comparent différents systèmes de KPE sur des corpus multilingues, et les comparons à deux systèmes : BARTHez et mBARTHez, entraînés sur Papyrus-f. Les trois systèmes extractifs utilisés pour notre comparaison sont SingleRank, YAKE, et KeyBERT. SingleRank (SR) [Wan et Xiao, 2008a] utilise des algorithmes de graphes pour extraire des mots-clés ayant un grand poids dans les documents. YAKE [Campos *et al.*, 2020] utilise plusieurs traits statistiques (position du terme, contexte, etc) pour déterminer la valeur d'un mot-clé et retourne les mots-clés les mieux évalués. Finalement, KeyBERT[8] utilise la similarité cosinus entre les candidats et le document (avec BERT) pour trouver les candidats qui ont la plus grande similarité.

Pour vérifier l'efficacité de notre modèle entraîné sur Papyrus-f, trois corpus sont utilisés en aval : le jeu de test de Papyrus-f, WikiNews [Bougouin *et al.*, 2013], un jeu de données francophone du domaine des nouvelles et ecommerce [Gao *et al.*, 2022], qui est une partie d'un corpus multilingue où les entrées sont des descriptions de produits et les mots-clés sont des termes de recherches pour ces produits. Des exemples des trois corpus de tests sont montrés à la table 5.4.

Les résultats pour les mots-clés présents et absents sont montrés aux tables 5.5 et 5.6. Nous roulons tous les modèles 3 fois afin d'obtenir une moyenne.

Comme nous pouvons l'observer, mBARThez obtient les meilleurs résultats à la fois au niveau extractif et au niveau abstractif, ce qui démontre l'utilité des méthodes génératives. Pour la suite de ce chapitre, nous dénotons mBARThez entraîné sur

[8]https://maartengr.github.io/KeyBERT/

Tâche	Texte d'entrée	Mots-clés
Papyrus-f	Représentation et fétichisation des femmes trans racisées dans les médias audiovisuels Ce mémoire porte sur la représentation [...]	télévision ; Cinéma ; Trans ; Queer ; Séries télévisées ; Intersectionnalité ; Performativité ; Performance
WikiNews	29 mai 2012. - Un postier travaillant sur le site de Noyal-sur-Vilaine (Ille-et-Vilaine) a été retrouvé mort [...]	la poste ; postier ; suicide ; demande de changement de service ; refus de congés ; syndicat ; vie au travail ; difficultés personnelles
Ecommerce	Le transmetteur Bluetooth HomeSpot pour Nintendo Switch permet la diffusion audio sans fil [...]	adaptateur bluetooth pour ; adaptateur bluetooth switch ; switch lite ; nintendo switch ; casque ; usb ; audio

Tab. 5.4. Exemple de document pour les trois corpus de test utilisés.

Papyrus-f par Bart-f. De façon similaire, nous dénotons BART entraîné sur Papyrus-e par Bart-e.

Nous rapportons également les résultats des différents systèmes sur leur propre corpus de test dans la table 5.7 (c'est-à-dire BART entraîné sur Kp20K et testé sur Kp20K, Bart-e testé sur Papyrus-e test, et Bart-f testé sur Papyrus-f test). Étant donné que les ensembles de test sont différents, cette comparaison sert davantage à faire une vérification globale de la qualité et de notre système que pour en faire une analyse fine. Nous pouvons en retirer que la difficulté de notre corpus semble être comparable à celle de Kp20K, et que la performance de notre modèle est globalement comparable à l'état de l'art pour Kp20K.

	Papyrus-f		WikiNews		ecommerce	
	F@5	**F@M**	**F@5**	**F@M**	**F@5**	**F@M**
SR	5.4	7.5	10.3	14.6	3.0	3.9
YAKE	10.3	11.6	18.6	26.5	7.4	8.9
KeyBERT	2.3	3.2	4.5	6.1	1.7	2.0
BARThez	22.1	26.9	17.4	20.3	4.2	6.1
mBARThez	**29.5**	**33.8**	**22.5**	**25.6**	**7.2**	**9.6**

Tab. 5.5. F1@5/F1@M pour les mots-clés présents et différents systèmes extractifs/génératifs.

	Papyrus-f		WikiNews		ecommerce	
	F@5	**F@M**	**F@5**	**F@M**	**F@5**	**F@M**
SR	0	0	0	0	0	0
YAKE	0	0	0	0	0	0
KeyBERT	0	0	0	0	0	0
BARThez	2.7	4.9	1.0	**2.4**	0.3	0.6
mBARTHez	**3.7**	**6.5**	**1.1**	**2.4**	**0.5**	**0.8**

Tab. 5.6. F1@5/F1@M pour les mots-clés absents et différents systèmes extractifs/génératifs.

5.4. Augmentation de données

Les résultats laissent supposer une grande place à l'amélioration, surtout au niveau des performances sur les mots-clés absents, c'est-à-dire, au niveau des

	Present		Absent	
	F@5	**F@M**	**F@5**	**F@M**
Kp20K	23.3	31.5	4.5	8.4
Papyrus-f	29.5	33.8	3.7	6.5
Papyrus-e	32.8	35.8	4.1	7.1

Tab. 5.7. Évaluation des modèles sur leurs propres ensemble de test. Les déviations standards sont entre 0.2 et 0.5.

capacités abstractives des modèles. Dans cette section, nous testons différentes techniques d'AD pour tenter d'améliorer la performance des modèles.

Nous séparons nos techniques d'AD en deux catégories. Partant d'un exemple (x, y) où x est le résumé et y les mots-clés, la première catégorie de techniques crée des nouveaux résumés sans toucher aux mots-clés, créant des exemples (x', y), et la deuxième catégorie modifie les mots-clés sans toucher aux résumés, créant des exemples (x, y'). Dans les deux cas, l'intérêt est surtout mis sur les capacités *génératives*, puisque c'est à ce niveau que les systèmes peinent généralement. Les différentes techniques sont illustrées à la table 5.8, utilisant un exemple fictif.

Dans les techniques de la première catégorie, nous utilisons AEDA, EDA et VAE-Par, techniques décrites aux chapitres précédents. Nous utilisons également KPDrop avec différents ratios, qui remplace une partie des mots-clés présents dans le résumé par un masque, ainsi que KPSR, qui remplace tous les mots-clés présents dans le résumé par des synonymes de WordNet.

Dans les techniques de la deuxième catégorie, nous utilisons cinq stratégies. Tout d'abord, nous testons l'utilisation du VAE pour paraphraser les mots-clés (VAE-Labels), rajoutant ceux qui n'étaient pas déjà présents. L'utilisation de techniques de recherche d'information est également testée, techniques illustrées à la figure 5.2.

Résumé	Cette thèse parle d'augmentation de données et de modèles génératifs.
Mots-clés	augmentation de données, modèles génératifs, petites données
AEDA	Cette thèse ; parle d'augmentation de données ! et de modèles génératifs ..
EDA	Cette maîtrise parle d'augmentation de données et de systèmes génératifs.
VAE-Par	Ce document discute d'augmentation de données et de systèmes génératifs.
KPDrop	Cette thèse parle d'[MASK] et de [MASK]
KPSR	Cette thèse parle de majoration de points et de systèmes de génération
(x, y')	apprentissage semi-supervisé, apprentissage

Tab. 5.8. Exemples fictifs de l'effet des différentes méthodes d'augmentation de données. L'entrée (x,y') regroupe toutes les méthodes de la deuxième catégorie, c'est-à dire les deux méthodes utilisant le PRF, les deux méthodes utilisant la distance de Jaccard, VAE-Labels et Bootstrap. Nous regroupons ces entrées car la sortie n'est pas prévisible, dépendant entièrement de ce que l'algorithme décide de générer.

L'idée derrière cette famille de techniques est d'utiliser l'information présente dans le corpus pour aller chercher des mots-clés supplémentaires. L'algorithme de base se réalise en deux étapes. Tout d'abord, une fonction donne un score aux documents du corpus selon un critère prédeterminé. Deux critères sont testés pour donner un score aux documents : BM25 [Robertson et Zaragoza, 2009] (dénommé PRF pour le reste du chapitre, pour *pseudo-relevance feedback*) et la distance de Jaccard entre le document source et le document du corpus.

Dans la seconde étape, l'algorithme extrait les nouveaux mots-clés des documents les plus importants. Ici encore, deux variations sont testées: la méthode TF-IDF extrait les termes ayant le plus de poids dans les documents récupérés, alors que la méthode "All" récupère l'entièreté des mots-clés présents dans ces documents. Au

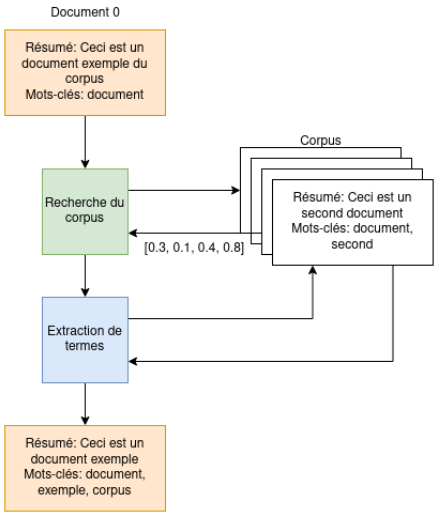

Fig. 5.2. Illustration des méthodes inspirées de la recherche d'information (RI).

total, cela résulte en quatre méthodes: PRF-TFIDF, PRF-ALL, Jaccard-TFIDF, et Jaccard-All.

La dernière technique testée est la technique *Bootstrap*, où Bart-f est entraîné une première fois sur le corpus, avant d'être utilisé pour générer des mots-clés additionnels pour chaque document, puis entraîné de nouveau sur les documents augmentés.

Nous regardons maintenant la performance de nos systèmes avec les différentes méthodes d'augmentation de données. Les résultats pour les mots-clés présents sont montrés à la table 5.9 et pour les mots-clés absents, à la table 5.10.

Il existe un compromis entre la performance des mots-clés absents et présents. Plus concrètement, à l'exception d'EDA, aucune des méthodes ne semble apporter d'augmentation significative de la performance sur les mots-clés présents, ni @5, ni @M. Comme mentionné, les méthodes proposées ne sont toutefois pas adaptées à l'augmentation de performance sur les mots-clés présents, puisqu'elles modifient soit le texte en entrée sans toucher aux mots-clés ou les mots-clés sans toucher au texte,

	Papyrus-f		WikiNews		ecommerce	
	F@5	**F@M**	**F@5**	**F@M**	**F@5**	**F@M**
Bart-f	29.5	33.8	22.5	25.6	7.2	9.6
AEDA	30.1	34.0	25.1	27.6	8.3	10.7
EDA	29.6	33.8	**25.9**	**28.4**	8.1	10.6
VAE-Par	27.9	32.8	19.8	22.9	7.8	10.7
VAE Labels	28.8	32.7	19.9	22.3	7.2	9.4
KPDDrop 0.5	27.1	32.0	20.5	23.5	6.8	9.4
KPDDrop 0.7	26.8	32.0	21.2	24.3	6.9	9.6
KPSR	28.1	32.9	23.0	26.1	7.5	10.3
PRF-TFIDF	28.7	30.7	23.9	25.6	8.5	10.2
PRF-ALL	27.0	31.3	21.1	23.9	7.4	9.5
Jaccard-TFIDF	28.9	28.2	24.7	25.9	7.4	8.2
Jaccard-All	26.1	29.4	18.5	20.6	6.0	7.1
Bootstrap	**30.7**	**34.1**	25.4	27.9	**8.8**	**11.0**

Tab. 5.9. F1@5/F1@M pour les mots-clés présents pour les différentes méthodes d'augmentation de données.

ce qui dans les deux cas encourage les fonctions *abstractives* du système au détriment des fonctions *extractives*.

Pour les mots-clés absents, nous pouvons noter que les méthodes qui affectent les résumés sont plus efficaces que les méthodes qui affectent les mots-clés. Une explication possible est que les mots-clés générés par les méthodes de la deuxième catégorie sont plus loin du résumé, alors que les méthodes telles KPDRop encouragent simplement la capacité de synthèse du modèle. Ceci se reflète également par le fait

	Papyrus-f		WikiNews		ecommerce	
	F@5	**F@M**	**F@5**	**F@M**	**F@5**	**F@M**
Bart-f	3.7	6.5	1.1	2.4	0.5	0.8
AEDA	4.7	7.5	1.8	**4.3**	0.7	1.0
EDA	4.7	7.5	**2.1**	**4.3**	0.7	1.0
VAE-Par	4.0	6.7	1.7	3.5	0.7	1.0
VAE Labels	4.0	6.7	1.0	2.2	0.5	0.7
KPDDrop 0.5	**5.2**	**8.0**	1.8	3.9	0.7	1.0
KPDDrop 0.7	**5.2**	7.9	1.6	3.0	0.7	1.1
KPSR	4.7	7.6	1.7	3.5	0.7	1.1
PRF-TFIDF	4.0	2.6	1.1	0.8	0.6	0.5
PRF-ALL	3.3	1.3	1.3	0.5	0.5	0.2
Jaccard-TFIDF	4.9	3.1	1.2	1.1	**1.2**	**1.2**
Jaccard-All	4.8	3.2	1.7	1.1	0.6	0.6
Bootstrap	4.3	7.0	1.1	2.4	0.6	0.9

Tab. 5.10. F1@5/F1@M pour les mots-clés absents pour les différentes méthodes d'augmentation de données.

que KPDrop fonctionne mieux que KPSR, bien que les deux techniques soient très similaires l'une de l'autre.

5.5. Discussion

Dans cette section nous tentons d'analyser les sorties de nos algorithmes, afin d'essayer d'extraire des tendances et de mieux comprendre l'effet de l'AD. Nous

choisissons au hasard un document du jeu de test, qui a le résumé et les mots-clés suivants :

"***Résumé :*** *Ce mémoire porte sur la représentation et la fétichisation des femmes trans dans les médias audiovisuels. Cette étude décrit tout d'abord l'évolution de la représentation trans-féminine des années 1950 aux années récentes. Les femmes trans racisées sont pour ainsi dire invisibles lors des premières décennies. Le mémoire distingue la représentation anti-assimilationniste de la représentation intégrationniste. Le film Paris Is Burning (1991, Livingston) apporte un changement à cet effet et présente une nouvelle réalité dans les médias. L'influence du documentaire du Nouveau Cinéma Queer est encore majeure. Près de trente ans plus tard, la série télévisée Pose (Ryan Murphy, Brad Falchuk, Steven Canals, FX, 2018-2021) rend hommage au film de la réalisatrice Jennie Livingston. Le mémoire se penche sur la théorie féministe, intersectionnelle et queer des années 1990 afin de comprendre l'évolution de ces études. Cette analyse interroge la remise en question des rôles de genre et l'impact de la performance et de la performativité dans les images médiatiques trans.*

Mots-clés : *télévision ; cinéma ; trans ; queer ; séries télévisées ; intersectionnalité ; performativité ; performance* "

La table 5.12 montre des exemples de mots-clés générés par les différents systèmes, et la table 5.13, le nombre moyen de mots-clés générés, ainsi que le nombre moyen de mots-clés générés présents et absents. Nous notons que toutes les techniques semblent augmenter légèrement le nombre de mots-clés générés, et que les techniques qui modifient les mots-clés pour en rajouter permettent une augmentation significative du nombre moyen de mots-clés, mais que la performance augmente peu due à une trop grosse distance entre les nouveaux mots-clés et les mots-clés originaux. Les exemples illustrent aussi bien la difficulté de l'évaluation en KPG. Beaucoup des mots-clés générés (medias, femmes trans, fétichisation, etc) sont des mots-clés valides, mais

Référence	télévision ; cinéma ; trans ; queer ; séries télévisées ; intersectionnalité ; performativité ; performance
Bart-f	femmes trans ; médias audiovisuels ; fétichisation ; intersectionnalité ; performance
AEDA	femmes trans ; médias ; documentaire ; intersectionnalité ; fétichisation ; performance ; performativité ; pose
EDA	femmes trans ; médias ; documentaire ; intersectionnalité ; fétichisation ; performance ; performativité ; pose
VAE-Par	femmes trans ; médias audiovisuels ; intersectionnalité ; fétichisation
VAE Labels	cinéma ; trans ; intersectionnalité ; fétichisation ; performance ; performativité ; nouveau cinéma queer ; pose
KPDrop 0.5	cinéma ; documentaire ; femme trans ; intersectionnalité ; fétichisation ; performance ; performativité ; pose
KPDrop 0.7	cinéma ; télévision ; femmes trans ; intersectionnalité ; fétichisation ; performance ; performativité ; pose
KPSR	femmes trans ; médias audiovisuels ; performance ; performativité ; féminisme ; intersectionnalité
PR-TFIDF	femmes trans ; médias audiovisuels ; fétichisation ; intersectionnalité ; performance ; performativité ; théorie féministe ; nouveau cinéma queer ; jennie Livingston ; pose ; rhua ; île ; trnasec ; îles ; arnt
PRF All	cinéma québécois ; documentaire à la première personne ; auto biographie ; autoethnographie ; problème des n corps ; configuration centrale ; configuration en toile d'araignée ; preuve assistée par ordinateur ; théorie
Jaccard TFIDF	femmes trans ; médias audiovisuels ; fétichisation ; intersectionnalité ; performance ; performativité ; nouveau cinéma queer ; vloggers ; vlog ; gais ; bisexuel ; lesbiennes ; sein couple ; sein ; sein équipe recherche ; couple homosexuel
Jaccard All	cinéma ; identité ; transgenre ; non-binaire ; intersectionnalité ; études de genre ; études féministes ; études postcoloniales ; montréal ; stéréotypes ; clichés
Bootstrap	femmes trans ; médias audiovisuels ; fétichisation ; intersectionnalité ; performance ; performativité

Tab. 5.11. Exemples de mots-clés générés par Bart-f entraînés avec les différentes techniques d'augmentation de données. Certains mots-clés récurrents sont en couleur pour faciliter la lecture

Référence	télévision ; cinéma ; trans ; queer ; séries télévisées ; intersectionnalité ; performativité ; performance
mBARThez	femmes trans ; médias audiovisuels ; fétichisation ; intersectionnalité ; performance
AEDA	femmes trans ; médias ; documentaire ; intersectionnalité ; fétichisation ; performance ; performativité ; pose
EDA	femmes trans ; médias ; documentaire ; intersectionnalité ; fétichisation ; performance ; performativité ; pose
VAE-Par	femmes trans ; médias audiovisuels ; intersectionnalité ; fétichisation
VAE Labels	cinéma ; trans ; intersectionnalité ; fétichisation ; performance ; performativité ; nouveau cinéma queer ; pose
KPDrop	cinéma ; documentaire ; femme trans ; intersectionnalité ; fétichisation ; performance ; performativité ; pose
KPSR	femmes trans ; médias audiovisuels ; performance ; performativité ; féminisme ; intersectionnalité
PRF	cinéma ; identité ; transgenre ; non-binaire ; intersectionnalité ; études de genre ; études féministes ; études postcoloniales ; montréal ; stéréotypes ; clichés
Bootstrap	femmes trans ; médias audiovisuels ; fétichisation ; intersectionnalité ; performance ; performativité

Tab. 5.12. Exemples de mots-clés générés par Bart-f entraînés avec les différentes techniques d'augmentation de données. Certains mots-clés récurrents sont en couleur pour faciliter la lecture

vont recevoir un score de 0. Malheureusement, et comme mentionné précédemment, il s'agit ici de la limite des protocoles d'évaluation utilisés.

5.6. Conclusion

Dans ce chapitre, nous regardons l'augmentation de données pour la tâche de génération de mots-clés, ce qui nous permet de mieux souligner le fait que différentes méthodes doivent être développées pour différents domaines. Pour ce faire, nous collectons d'abord un corpus de l'Université de Montréal pour créer un premier corpus

	KP gen	Présent	Absent
Bart-f	4.7	3.9	0.9
AEDA	5.7	4.3	1.3
EDA	5.7	4.3	1.4
VAE-Par	4.8	3.6	1.2
VAE Labels	5.4	4.3	1.1
KPDrops 0.5	5.3	3.6	1.6
KPDrop 0.7	5.4	3.5	1.9
KPSR	5.1	3.7	1.4
PRF TFIDF	13.3	5.2	8.1
PRF All	7.7	3.9	3.8
Jaccard TFIDF	16.9	7.2	9.7
Jaccard All	7.1	3.6	3.4
Bootstrap	5.6	4.4	1.1

Tab. 5.13. Nombre moyen de mots-clés générés par les différents systèmes, ainsi que le nombre moyen de mots-clés présents et absents générés.

francophone de KPG académique. Nous établissons ensuite plusieurs résultats de références, concluant que mBARThez fonctionne le mieux.

Nous testons finalement plusieurs algorithmes d'augmentation de données, certains existant dans la littérature et certains non. Nous concluons que, premièrement, il semble être difficile d'avoir une augmentation significative à la fois sur les mots-clés présents et les mots-clés absents. En effet, la plupart des méthodes explorées encouragent la génération de mots-clés absents, mais souvent au détriment des mots-clés présents. Nous observons ensuite qu'il est difficile de

guider la génération pour que les mots-clés rajoutés soient pertinents. Les méthodes qui ajoutent des mots-clés supplémentaires encouragent souvent la génération de mots-clés trop peu reliés au sujet du résumé pour qu'ils permettent d'augmenter les métriques.

Nous concluons en soulignant que l'AD pour le KPG est encore à ses débuts, et qu'il reste beaucoup de recherche à faire dans ce domaine. Notamment, il est possible qu'un meilleur affinage de nos algorithmes d'augmentation mène à des performances plus élevées. Il y aurait également la possibilité de séparer les modèles extractifs et abstractifs pour obtenir de meilleures performances sur les deux, ou qu'intégrer de l'information supplémentaire (titre, domaine, etc) aide la génération de mots-clés de façon significative.

Chapitre 6

Apprentissage par requête synthétisée

Le dernier chapitre de cette thèse s'intéresse à un paradigme d'apprentissage qui est à mi-chemin entre l'augmentation de données et l'apprentissage actif (*Active Learning* ou AL) : l'apprentissage par requête synthétisée (*Membership Query Synthesis* ou MQS). Ce type d'apprentissage est une sous-branche de l'apprentissage actif où, au lieu de sélectionner les exemples les plus informatifs à étiqueter (comme dans l'AL) on génère des nouveaux exemples que l'on espère hautement informatifs. Ce processus est illustré à la figure 6.1, contrasté au processus d'augmentation de données et de l'apprentissage actif. Nous utilisons également le terme *apprentissage passif* pour désigner la méthode classique de création de jeux de données, c'est-à-dire, l'étiquetage au hasard parmi un ensemble d'exemples non étiquetés. Lorsque le terme *apprentissage actif* est utilisé sans précision, il fait référence à l'apprentissage actif par ensemble, même si l'apprentissage actif est techniquement un regroupement de trois méthodes : l'apprentissage actif par ensemble (le plus courant), le MQS et l'apprentissage actif en temps réel.

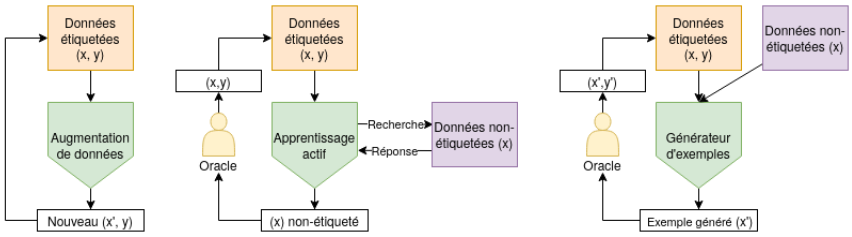

Fig. 6.1. Illustration générale de l'augmentation des données (à gauche), de l'apprentissage actif (au milieu) et du MQS. Ces figures représentent les algorithmes généraux, mais sont modifiables. Par exemple, il serait facile de concevoir un algorithme de DA qui utilise des données non étiquetées, ou un algorithme MQS qui n'a accès qu'aux données étiquetées.

Alors que le MQS peut, en théorie, générer des points très informatifs [Chen *et al.*, 2017], il demeure en pratique un sujet peu étudié ou pratiqué, dû principalement à deux problèmes. Tout d'abord, l'expérimentation sur le MQS exige l'étiquetage des exemples générés. La solution idéale serait de mettre un humain dans la boucle, mais cela demanderait un nombre massif de ressources (en temps d'étiquetage), et donc les études récentes s'appuient sur un algorithme - dénommé *oracle* - pour simuler l'étiquetage humain. En contraste, dans l'apprentissage actif, les exemples à étiqueter sont sélectionnés de l'ensemble de données non étiquetées, et donc des simulations peuvent être menées sur de vraies données étiquetées où les étiquettes sont initialement masquées.

Le deuxième problème est l'inefficacité générale du MQS par rapport à l'apprentissage actif, car ni générer de nouveaux exemples informatifs, ni les étiqueter, n'est trivial. En fait, seulement deux articles montrent des résultats positifs sur le MQS avec des jeux de données non filtrés[1]. L'article de Quteineh *et al.*

[1]Nous utilisons le terme données filtrées dans ce chapitre pour faire référence à la construction d'un *toy dataset* par Schumann et Rehbein [2019]

[2020], qui montre que générer des exemples avec GPT-2 sur TREC6 et SST-2 et les annoter permet effectivement d'augmenter la performance, et notre article publié à Canadian AI 2022 [Piedboeuf et Langlais, 2022c], qui regarde la performance sur SST-2 et les VAEs (voir section 6.1).

Nous nous attaquons donc à ces problèmes dans ce chapitre en trois étapes. Dans un premier temps, nous montrons que le MQS peut fonctionner sur un jeu de données couramment utilisé dans la littérature et non filtré : SST-2. Ces résultats sont significatifs, car ils montrent que des données générées peuvent être plus informatives que des données sélectionnées au hasard lors du processus d'étiquetage, ou encore que des données sélectionnées avec un algorithme de base d'AL (confiance).

Puis, armé de la connaissance de l'efficacité du MQS, nous nous intéressons à savoir si nous pouvons l'utiliser dans un paradigme d'extension de jeux de données déjà complètement étiqueté. Concrètement, au lieu d'avoir au départ un ensemble $\mathcal{L}$ étiqueté et un ensemble $\mathcal{U}$ non étiqueté (voir section 6.1), le paradigme est modifié pour ne donner accès qu'à un ensemble $\mathcal{L}$ composé de toutes les données disponibles pour ce jeu de données. Ce changement est important, car le coût d'annotation supplémentaire demandé pour l'étiquetage devient plus facilement justifiable, puisqu'il n'existe pas de données alternatives à étiqueter. Les cinq jeux de données utilisés au long de cette thèse sont encore une fois étudiés, pour conclure qu'utiliser MQS de cette façon permet d'augmenter légèrement la performance, et surpasse l'augmentation de données qui, comme montré aux chapitres précédents, à tendance à fonctionner de façon médiocre sur les jeux de données larges.

Un problème demeure avec cette approche : elle demande un coût d'annotation large pour pouvoir voir une différence significative dans la performance (dans ce chapitre, les algorithmes génèrent 15K et 50K données). Il y a plusieurs domaines où les coûts supplémentaires sont justifiés par des améliorations de performance significatives, comme dans l'analyse médicale ou l'exploration spatiale où chaque

avancée est cruciale. Cependant, pour les secteurs industriels plus traditionnels, une telle justification peut s'avérer plus difficile. Pour tenter de diminuer le coût, le dernier volet de ce chapitre s'intéresse à la jointure de l'apprentissage semi-supervisé et du MQS. Malheureusement, ces expériences donnent peu de résultats positifs, mais nous prenons le temps d'analyser les résultats et dressons une liste de pistes à explorer pour un futur travail.

Le chapitre est écrit comme suit. Tout d'abord, la revue de la littérature est présentée à la section 6.1. Puis, la section 6.2 présente le protocole expérimental, suivi des résultats expérimentaux dans les sections 6.3 et 6.4. Finalement, la section 6.5 présente une analyse et la section 6.6 présente les expériences avec l'apprentissage semi-supervisé.

6.1. Revue de littérature et notions préliminaires

Dans l'apprentissage actif basé sur un ensemble, nous considérons un ensemble de données non étiqueté $\mathcal{U}$ et un ensemble étiqueté $\mathcal{L}$, dans un espace $\mathcal{X}$, ainsi qu'un classificateur $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$, où $\mathcal{Y}$ désigne l'ensemble des étiquettes possibles. $\mathcal{L}$ et $\mathcal{F}$ sont formés de manière itérative où, à chaque itération, les instances les plus informatives dans $\mathcal{U}$ sont sélectionnées et étiquetées, les transférant dans $\mathcal{L}$.[2] Le défi de l'apprentissage actif est de définir correctement ce que signifie "le plus informatif". Cette notion d'information est souvent basée sur la distribution des données dans $\mathcal{X}$ [Dasgupta et Hsu, 2008] ou sur un manque de confiance du classificateur $\mathcal{F}$ [Ravanbakhsh *et al.*, 2019]. En termes plus simples, le but est de sélectionner parmi un ensemble de données non étiquetées le plus petit nombre d'exemples à annoter afin de maximiser les performances du classificateur. Dans

[2]Nous nous concentrons ici sur la classification, mais des systèmes d'apprentissage actif ont également été développés pour d'autres types de tâches, comme le NER [Shen *et al.*, 2017] ou la traduction automatique [Haffari *et al.*, 2009].

une application réelle, un étiqueteur humain étiquette les exemples, mais pour les besoins de la recherche, cela est souvent évité en utilisant un ensemble de données déjà étiquetées et en "masquant" les étiquettes jusqu'à ce que l'exemple soit ajouté à $\mathcal{L}$.

MQS est une variante de cette idée où, au lieu de sélectionner les exemples les plus informatifs de $\mathcal{U}$, les points les plus informatifs de $\mathcal{X}$ sont sélectionnés. Cela implique un générateur $\mathcal{G}$ qui transforme les points $\mathcal{X}$ en données interprétables (dans ce cas-ci en phrases), générateur qui est souvent entraîné sur l'ensemble des données $\mathcal{U}U\mathcal{L}$. Cela implique également que, même dans un contexte de recherche, nous avons besoin d'un étiqueteur ou d'un oracle O, puisque les points nouvellement générés ne sont pas accompagnés d'une étiquette. Pour la réalisation de la recherche, cependant, l'oracle peut être remplacé par un classificateur qui simule l'humain [Zarecki et Markovitch, 2020; Sahu *et al.*, 2022].

Deux autres choses importantes dont il faut tenir compte sont 1- le moment où s'arrêter et 2- le nombre de données étiquetées initiales à utiliser, ou la taille du noyau. Le noyau est utilisé pour aider l'algorithme de sélection à choisir les prochains points à étiqueter au début du processus lorsque aucune donnée générée n'est dans $\mathcal{L}$. Le point d'arrêt, quant à lui, est utilisé pour déterminer le budget d'annotation, c'est-à-dire, combien d'exemples sommes-nous prêts à annoter avant d'arrêter. Dans un contexte réel d'utilisation du MQS, la meilleure stratégie serait un budget flexible, en fonction de la performance que le classificateur obtient à ce moment et du budget monétaire restant, mais pour pouvoir comparer plusieurs expériences entre elles, un budget fixe est utilisé dans ce chapitre. Le processus classique du MQS est montré à la figure 6.2.

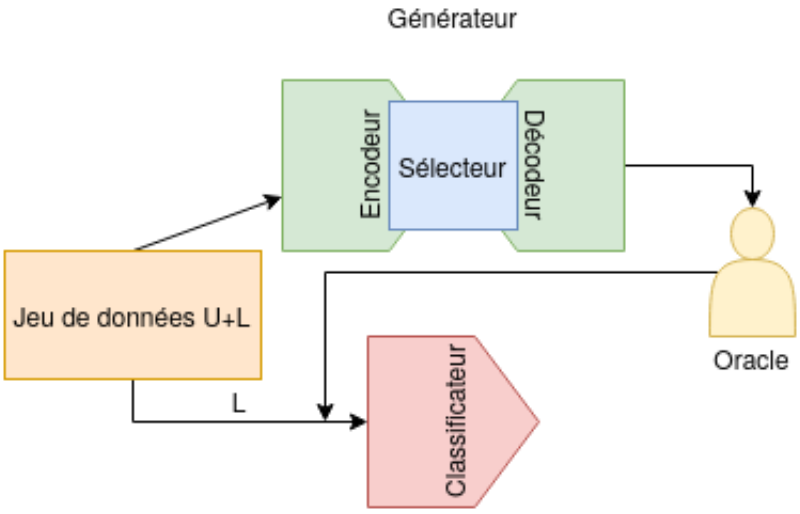

Fig. 6.2. Représentation du processus de base du MQS avec un modèle latent. Le sélecteur choisi le prochain point à étiqueter selon la distribution du jeu de données dans l'espace latent $\mathbf{z}$. Le décodeur transforme le point en phrase, et l'oracle étiquette cette phrase avant de la rajouter dans le jeu de données $\mathcal{L}$.

Le problème de la dimensionnalité du texte fait en sorte que la plupart des données se retrouvent sur un petit *manifold*, dont la caractérisation précise est ardue, rendant ainsi l'échantillonnage difficile. Pour contourner ce problème, la plupart des systèmes MQS introduisent un espace latent continu $\mathbf{z}$ à partir duquel les points sont transformés en données lisibles. Dans l'ensemble, un système MQS totalise cinq composants, à savoir le générateur (composé de l'encodeur et du décodeur), le sélecteur, le classificateur (le modèle à entraîner) et l'oracle. Certains algorithmes modifient légèrement ce processus. Par exemple, Zhu et Bento [2017] utilisent un GAN et Quteineh *et al.* [2020] utilisent GPT-2 comme générateur, ne s'appuyant donc pas sur un encodeur (voir chapitre 2).

Pour trouver les points à étiqueter, Zhu et Bento [2017] formulent un problème d'optimisation qui projette la frontière de décision dans l'espace $\mathcal{X}$ à l'espace latent $\mathbf{z}$, et étiquètent ensuite des points proches de cette frontière de décision. Quteineh *et al.* [2020], de leur côté, testent deux stratégies de décodage et filtrent les exemples par la suite pour maximiser l'entropie.

Comme mentionné, le MQS pour les données textuelles a été moins étudié, en partie à cause des limites des modèles génératifs pour le texte. Au total, trois articles s'intéressent à l'utilisation du MQS textuel : sur des données textuelles extrêmement limitées (moins de 50 exemples au total) avec de l'édition de phrases [Zarecki et Markovitch, 2020] ou de l'utilisation de modèles pré-entraînés [Quteineh *et al.*, 2020], et sur des données textuelles filtrées [Schumann et Rehbein, 2019].

L'algorithme principal étudié dans [Schumann et Rehbein, 2019] se base sur l'algorithme de Wang *et al.* [2015], où les auteurs présentent un algorithme d'apprentissage actif basé sur un ensemble. Cet algorithme recherche dans un espace latent des points proches de la frontière de décision et étiquette les données de $\mathcal{U}$ qui s'y retrouvent. La modification pour l'appliquer au MQS consiste simplement à transformer les points de z en données et à rajouter celle-ci dans $\mathcal{L}$. Nous notons cet algorithme DB, (pour *decision boundary*). Il est à noter que cet algorithme fonctionne uniquement sur les données de classification binaire.

L'algorithme DB commence par trouver une paire de points (un de chaque classe) qui est proche de la frontière de décision. Pour ce faire, les points à mi-distance des centroïdes des deux classes du noyau sont interrogés (c'est-à-dire, transformés en phrases et étiquetés) pour un nombre fixe d'itérations, en ajoutant les nouveaux points à $\mathcal{L}$ et en rapprochant les centroïdes de la frontière à chaque fois. Puis, partant des deux derniers points de classes opposées trouvés (x_+, x_-), de façon itérative l'algorithme : 1- génère un vecteur de grandeur λ perpendiculaire à la médiane des deux points, 2- interroge ce point pour obtenir sa classe y_i,

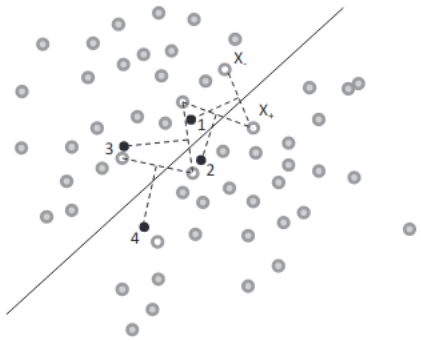

Fig. 6.3. Sélection du point suivant dans l'espace latent à transformer en phrase. Dans [Schumann et Rehbein, 2019], le point est directement interrogé, alors que dans [Wang *et al.*, 2015], le voisin le plus proche dans $\mathcal{U}$ est interrogé. Image de Wang *et al.* [2015].

et 3- remplace le point de la classe y_i dans le couple (x_+, x_-) par ce nouveau point. Cela a pour effet d'échantillonner des points le long de la frontière, des deux côtés de celle-ci. L'algorithme, illustré à la figure 6.3, est arrêté lorsque le budget d'étiquetage est atteint. Dans l'article de Wang *et al.* [2015], la version d'apprentissage actif de l'algorithme (étiqueter les points de $\mathcal{U}$ proche des points retrouvés) est testé sur plusieurs ensembles de données médicales, ainsi que des versions binaires de MNIST, et les auteurs montrent une amélioration par rapport à d'autres algorithmes d'apprentissage actif, notamment une stratégie basée sur la confiance ou la distribution du jeu de données.

Schumann et Rehbein [2019] utilisent cet algorithme dans l'espace latent d'un VAE, pour générer et étiqueter directement le point sélectionné, sur un ensemble de données filtrées de classification des sentiments composé de phrases courtes (moins de 15 mots) de SAR14 et SST-2. L'algorithme DB est comparé à deux techniques d'apprentissage actif (échantillonnage aléatoire et moindre confiance), et les auteurs

rapportent que les deux techniques MQS surpassent l'AL lors de la comparaison du coût d'annotation.

Le filtrage du jeu de données crée cependant un jeu de données assez irréaliste, qui rend difficile de savoir si l'algorithme fonctionnerait bien en pratique, surtout considérant la difficulté des VAEs à modéliser le texte (voir chapitre 3). Par exemple, si nous ne regardons que l'ensemble de données SST-2, nous constatons que la longueur moyenne des phrases (définie par le nombre d'espaces) est de 19.3, versus 9.4 en gardant uniquement les phrases de moins de 15 mots, et il reste donc à vérifier si cette inégalité des longueurs n'invaliderait pas l'approche proposée.

Un autre travail sur le MQS textuel est celui de Zarecki et Markovitch [2020], où les phrases sont éditées pour perturber les exemples, similairement à la méthode EDA présentée au chapitre 4. Pour ce faire, les mots sont remplacés par des substituts sémantiquement proches (dans un embedding Word2Vec), et ces phrases sont ensuite étiquetées pour voir la nouvelle classe de l'exemple modifié. Bien qu'il s'agisse techniquement de MQS, l'accent n'est pas mis sur la meilleure façon de sélectionner et de créer un jeu de données à partir d'un espace continu, mais plutôt sur la génération de nouveaux exemples lorsque l'ensemble de données non étiquetées est très petit.

La dernière recherche sur le MQS textuel que nous avons trouvé est celle de Quteineh *et al.* [2020], qui utilise deux stratégies de décodage avec des GANS (normal et avec une recherche d'arbre de type Monte-Carlo) sur SST-2 et TREC6, montrant avec un très petit ensemble de départ (20 et 30 exemples) une augmentation significative de la performance (respectivement 5% et 25%). Cependant, les auteurs ne semblent faire qu'une seule répétition et ne comparent pas d'autres stratégies, ce qui rend difficile de savoir si la stratégie qu'ils proposent est la meilleure ou non.

Finalement, l'étude de Zhu et Bento [2017] s'intéresse au MQS pour les images. La stratégie utilisée dans l'étude est d'entraîner un GAN sur le jeu de données $\mathcal{U} + \mathcal{L}$, et ensuite de projeter la barrière de décision dans l'espace latent pour sélectionner

des points qui y sont proches. Cependant, comme noté dans le chapitre 2, les GANs sont difficiles à entraîner sur le texte, menant souvent à des espaces latents peu représentatifs. Pour cette raison, les GANs ne sont pas considérés dans ce chapitre.

6.2. Méthodologie

Nous présentons dans cette section la méthodologie utilisée pour tester le MQS, dont le processus est illustré à l'algorithme 2. Dans un premier temps, nous testons différents paramètres, afin de voir ce qui performe bien. Puis, dans un second temps, l'utilisation du MQS pour des jeux de données complets et une absence de données non étiquetées est testé.

6.2.1. Jeux de données

Les cinq jeux de données utilisés au long de cette thèse (SST-2, FakeNews, Irony, IronyB et TREC6) sont à nouveau repris dans ce chapitre. Garder les mêmes jeux nous permet de nous comparer aux techniques d'augmentation de données développées au cours des chapitres précédents, et également de voir ce qui performe bien sur plusieurs tâches différentes.

6.2.2. Générateurs

Le générateur est la composante importante qui permet de générer de nouvelles données. Tout d'abord, deux algorithmes sont repris du chapitre 3 sur l'augmentation de données: le VAE et CVAE. Comme cette fois les exemples se font annoter par un oracle, le VAE est entraîné sur la totalité des données, sans séparation par classe. Deux autres algorithmes sont testés : l'utilisation de GPT-2 suivant [Quteineh *et al.*, 2020] et le SSVAE, le *Semi-supervised VAE*. Ce dernier est entraîné pour générer des exemples conditionnellement à la certitude du classificateur, une modification directe du CVAE. Il est à noter que le CVAE ne peut être entraîné que sur les

données annotées, alors que le VAE, le SSVAE, et GPT-2 peuvent être entraînés sur toutes les données.

Pour ce qui est de GPT-2, Quteineh *et al.* [2020] testent deux stratégies de décodage : décodage normal et décodage avec une recherche de type Monte-Carlo, mais rapportent une différence non significative entre les deux stratégies (entre 1 et 2% de différence sur une seule expérience). Selon ces résultats, la stratégie de décodage normale (c'est-à-dire, avec un top-p) est suivie, stratégie simple à utiliser.

6.2.3. Sélecteurs

Le sélecteur est la composante du système en charge de choisir quel point transformer en phrase. Le contrôle de la génération est un problème complexe et il n'existe pas de méthode universelle, surtout dans un cas comme le MQS où l'important est d'avoir des phrases informatives, ce qui est difficile à mesurer. Pour les modèles de types VAEs, le contrôle de la génération se fait à partir de l'espace latent, alors que pour GPT-2 le contrôle se fait à partir de techniques de décodage.

La première méthode de sélection testée est une pige au hasard (R). Pour les VAEs, cela implique simplement de piger à partir du prior et de passer à travers le décodeur, similairement à la technique d'augmentation de données VAE-SEP du chapitre 4. Pour GPT-2, cela implique de faire une pige au hasard à travers le top_p (nucleus sampling), fixé à 0.95.

Ensuite, nous tentons une pige au hasard pour le VAE, mais à partir du postérieur plutôt que de la distribution préalable (A). Le postérieur est défini comme une mixture de gaussiennes composées des points du jeu de données, et donne théoriquement des meilleurs résultats que la distribution préalable car il représente mieux la distribution du jeu de données dans l'espace latent [Xu *et al.*, 2020].

Pour le VAE, nous testons également la méthode DB, où l'hyperplan de décision dans l'espace latent est déterminé et des points sont sélectionnés de chaque côté

de cet hyper-plan (DB). Le CVAE permet également une technique de génération d'exemples balancés (B), où à chaque itération l'on essait de balancer le jeu de données $\mathcal{L}$. Finalement, pour le SSVAE, nous essayons de générer des points qui sont directement évalués comme ayant une confiance faible selon le classificateur (C). Dans les sections qui suivent, nous utilisons la dénomination Générateur-Sélection pour parler de la combinaison des deux. Par exemple, la sélection au hasard avec un VAE est dénommée VAE-R.

6.2.4. Oracle

Ce chapitre s'inspire de [Zarecki et Markovitch, 2020; Sahu *et al.*, 2022] en utilisant un oracle dans nos expériences, algorithme qui simule l'humain étiquetant des exemples. Étant donné que nous répétons plusieurs fois les expériences pour diminuer les variations dues au hasard, manuellement étiqueter toutes les phrases générées serait impossible. En effet, le total de phrases à annoter pour les expériences se retrouveraient bien au-delà de 10M.

L'oracle utilisé est BERT, entraîné sur la totalité du jeu de donnée (entraînement, développement et test) pour simuler l'humain. Deux exceptions sont rajoutées au processus d'étiquetage : tout d'abord, si l'exemple généré est déjà présent dans le jeu de données $\mathcal{L}$, il n'est pas passé à l'oracle et il n'est pas compté dans le budget d'annotation. Puis, si l'oracle est incertain (moins de 0.7 de confiance), l'exemple est rejeté, mais compté dans le budget d'annotation. Ce seuil sert à simuler le rejet d'un humain en présence de phrases mal formées ou ambiguës (par exemple, la phrase "J'ai été voir ce film" n'est ni positive, ni négative, et serait rejetée lors de l'annotation).

Nous analysons plus en détail l'impact de l'oracle dans le MQS à la section 6.5, notamment au niveau de la puissance de l'oracle, du seuil de filtrage et de l'étiquetage humain.

Algorithm 1 Algorithme MQS pour le VAE. L'algorithme est modifié pour les autres générateurs tel que décrit à la section 6.2.2.

procedure MEMBERSHIP QUERY SYNTHESIS($\mathcal{L}$, $\mathcal{U}$, $\mathcal{B}$):
    Budget $\leftarrow \mathcal{B} - |\mathcal{L}|$
    Classifier $\leftarrow$ BERT
    Oracle $\leftarrow$ BERT-FULL ▷ L'oracle est entraîné sur train+dev+test
    Generator $\leftarrow$ VAE
    Generator.train($\mathcal{U} + \mathcal{L}$)
    while *Budget*>0 **do**
        exemple $\leftarrow$ *Generator.generate()*
        label, *confidence* $\leftarrow$ *Oracle.label(example)*
        if *exemple* in $\mathcal{L}$ **then**
            Continue
        else if *confidence*>0.7 **then**
            $\mathcal{L}$.append(*exemple, label*)
        Budget $\leftarrow$ *Budget* $- 1$
    Classifier.finetune($\mathcal{L}$)

6.3. Étude du MQS pour créer des jeux de données

Dans cette section, nous regardons l'impact des différents algorithmes pour la création d'un jeu de données à partir de données artificielles, dans le but de déterminer si le MQS est une méthode efficace avec les algorithmes génératifs disponibles aujourd'hui. Nous utilisons un petit noyau (10, 50 ou 100 exemples) et un budget de 500 exemples total (incluant le noyau qui aurait été étiqueté au préalable), avec le jeu de données SST-2.

Deux résultats de référence sont établis pour évaluer la performance, et deux autres paradigmes d'apprentissage roulés avec les mêmes données. Dans le résultat de référence *noyau*, le modèle est entraîné avec uniquement le noyau, c'est-à-dire, la petite proportion d'exemples qui sont déjà étiquetés au départ. Le résultat de référence AE est le processus de MQS en utilisant un encodeur simple au lieu d'un VAE, et en générant également en pigeant au hasard de l'espace caché. Pour ce qui est des autres paradigmes de création de jeux de données, nous nous comparons à l'apprentissage passif, où l'on sélectionne des exemples au hasard pour les étiqueter, et une stratégie simple d'apprentissage actif par confiance. Il est à noter cependant que l'apprentissage actif par confiance avec les réseaux neuronaux profonds ont montré des résultats mitigés, ce qui peut expliquer les résultats décevants que nous obtenons pour cette stratégie [Ren *et al.*, 2021].

Les résultats montrent assez clairement que le MQS performe mieux que les deux résultats de références. De plus, il y a une différence claire entre les algorithmes qui performent mal (VAE-DB, CVAE-R, CVAE-B), et les algorithmes qui performent bien. Par contre, parmi les algorithmes de MQS qui obtiennent des bons résultats, il est difficile de savoir lequel est le meilleur dû à l'écart-type élevé. VAE-A et GPT-R semblent tout de même être les meilleurs algorithmes globalement.

Nous montrons à la table 6.2 des exemples de phrases générées par chaque algorithme pour une taille de départ de 100, et étiquetées comme positive ou négative par l'oracle. À la vue de ces exemples, une hypothèse qui pourrait expliquer pourquoi certaines techniques de piges au hasard dans l'espace $\mathcal{X}$ donnent des meilleurs résultats que l'apprentissage passif est que ces nouveaux exemples sont plus informatifs, car moins bien formés, et donc demandent au classificateur un effort d'apprentissage plus grand lors de l'entraînement. Ceci est similaire à la conclusion du chapitre 4 où l'on montre que des exemples générés par VAEs peuvent être plus informatifs que des exemples bien formés par paraphrase pour l'entraînement. Il est

	10	50	100
Noyau	52.1 (2.1)	53.2 (3.0)	61.6 (2.6)
AE-R	70.1 (7.4)	75.6 (4.4)	73.5 (6.7)
Passive Learning	82.5 (1.6)	82.6 (2.3)	82.5 (2.6)
Confidence AL	83.8 (2.1)	80.9 (2.2)	84.7 (1.4)
VAE-R	82.2 (4.7)	82.5 (4.3)	84.7 (4.1)
VAE-A	83.9 (6.3)	**86.2** (1.7)	**85.3** (2.8)
VAE- DB	50.8 (3.6)	58.5 (4.1)	65.1 (5.4)
CVAE-R	51.2 (0.9)	54.5 (4.5)	60.3 (2.6)
CVAE-B	57.4 (3.6)	62.7 (3.6)	77.0 (3.5)
SSVAE-R	83.4 (2.3)	81.9 (2.8)	82.1 (1.0)
SSVAE-C	81.3 (2.8)	82.0 (1.2)	82.1 (3.5)
GPT-R	**85.1** (1.4)	84.9 (1.0)	84.1 (1.6)

Tab. 6.1. Résultats pour les trois tailles de noyau et un budget d'annotation total de 500. Les écart-types sont indiqués entre parenthèses, calculés à travers 6 expériences. Les meilleurs résultats pour le MQS sont indiqués en gras, et les meilleurs résultats indépendamment de l'algorithme sont soulignés.

également à noter que le CVAE fonctionne très mal, ce qui n'est pas étonnant étant donné la quantité de données très limitée à laquelle il a accès. Ceci est en contraste avec les exemples de la table 6.4, ou le CVAE a accès à toutes les données et génère alors des données cohérentes.

6.4. Étude du MQS pour étendre les jeux de données

Dans cette section, nous étudions maintenant un paradigme du MQS un peu différent, l'utilisation du MQS lorsque l'ensemble $\mathcal{U}$ est inexistant et l'ensemble $\mathcal{L}$

Algorithme	Phrase générée	Polarité
VAE	an authentic and deeply felt work of the worst kind of an artist.	Positive
	the movie is a desperate miscalculation.	Negative
CVAE	it ' s , and it ' s , and it ' s , and it ' s , and it ' s , and it ' s , and it ' s , and it ' s , and it ' s , and it ' s , and it ' s , and it ' s , and it ' s , and it ' s , and	Positive
	a waste.	Negative
SSVAE	as the director ' s most refreshing and most likeable , the film is a smart , unforced intimacy.	Positive
	in all the way , it ' s just tediously bad.	Negative
GPT	the film's plot is a little off-putting, but it works.	Positive
	iced tea is a good movie, but it's not a great one.	Negative

Tab. 6.2. Exemples de phrases générées pour une taille de *noyau* de 100 et la polarité telle que déterminée par l'oracle.

est de taille plus significative. Concrètement, nous étudions le cas où l'on voudrait appliquer le MQS sur un jeu de données existant pour augmenter les performances du classificateur. Dans la section précédente, la supposition était qu'une partie significative du jeu était non étiquetée, mais en pratique, il existe peu de raisons de ne pas d'abord étiqueter l'ensemble des données existantes, avant d'en générer des nouvelles, surtout quand ces données sont plutôt limitées.

Étant donné les résultats de la section précédente, nous choisissons de ne pas utiliser le SSVAE ou VAE-DB, qui se sont avérés inefficaces et longs à exécuter. Comme résultats de références, le résultat sans MQS ainsi que l'augmentation de données avec l'AEDA et l'EDA sont utilisés. Pour avoir une idée globale de la performance, des expériences en ajoutant 15K et 50K données sont roulées, et nous

	SST-2	Irony	FakeNews	IronyB	TREC6	Moyenne
Référence	91.0 (0.6)	90.7 (0.5)	88.6 (0.4)	46.9 (1.6)	95.3 (1.7)	82.5
AEDA	91.0 (0.8)	91.4 (0.5)	87.8 (0.5)	44.4 (3.1)	95.3 (0.7)	82.0
	90.8 (0.7)	91.5 (0.8)	87.8 (0.6)	43.4 (2.3)	93.7 (2.1)	81.4
EDA	91.2 (0.5)	91.5 (0.5)	86.5 (1.1)	47.1 (1.4)	94.7 (1.3)	82.2
	89.5 (1.1)	90.6 (0.4)	86.9 (0.3)	43.2 (2.9)	93.3 (1.0)	80.7
VAE-R	**92.0 (0.8)**	91.5 (0.7)	**89.0 (0.3)**	49.5 (2.3)	95.2 (0.7)	**83.4**
	92.5 (0.4)	91.9 (0.5)	**89.4 (0.6)**	51.2 (1.6)	**95.9 (0.7)**	**84.2**
VAE-A	91.3 (0.5)	92.0 (0.3)	88.2 (0.4)	49.0 (1.5)	95.2 (2.0)	83.1
	92.1 (4.6)	91.3 (0.4)	88.4 (0.5)	**51.5 (2.0)**	95.6 (1.9)	83.8
CVAE-R	91.3 (0.6)	91.1 (1.0)	88.5 (0.3)	47.8 (3.8)	96.1 (0.6)	83.0
	91.8 (0.5)	**92.0 (0.5)**	89.7 (0.3)	51.2 (1.0)	94.2 (4.7)	83.8
CVAE-B	91.1 (0.9)	**91.8 (0.5)**	88.9 (0.6)	**50.3 (2.8)**	**96.2 (0.6)**	83.7
	92.2 (0.8)	**92.0 (0.5)**	89.0 (0.9)	51.3 (1.3)	95.3 (1.1)	84.0
GPT-R	90.6 (0.5)	91.2 (1.0)	87.7 (0.2)	48.1 (1.2)	94.8 (1.6)	82.5
	90.8 (0.6)	90.8 (1.0)	87.6 (0.4)	44.0 (2.4)	94.2 (3.3)	81.5

Tab. 6.3. Résultats pour 15K (première ligne) et 50K (deuxième ligne) données générées. Les valeurs en gras représentent la meilleure performance pour le jeu de données et nombre de données générées, et celles soulignées représentent les valeurs en dessous du résultat de référence. La référence est le résultat sans MQS ou DA. Les valeurs sont la moyenne sur 6 expériences, et l'écart-type est indiqué entre parenthèses.

analysons aussi en détail comment le processus évolue au fur et à mesure que nous ajoutons des données. Les résultats sont présentés à la table 6.3.

La première chose à noter est que les résultats, à la fois pour 15K et 50K données rajoutées, surpassent le résultat de référence ainsi que l'augmentation de données. Nous voyons même que dans ce cas, l'augmentation en général nuit à la performance, la dégradant en moyenne de 0.3%, alors que le MQS augmente la performance de 0.5% à 1.7%.

L'exception notable est GPT-R, qui est maintenant l'algorithme le moins efficace de tous. Nous supposons que cela est dû à la difficulté de générer un grand nombre de phrases diverses en utilisant le top-p et en partant du même token (<bos>). Cela expliquerait aussi pourquoi, pour GPT-R, la performance est meilleure généralement pour 15K phrases générées que pour 50K phrases. Une solution serait de commencer la génération par un token aléatoire sélectionné du corpus d'entraînement, ce qui encouragerait GPT-2 à générer des phrases plus diverses. La table 6.4 montre des exemples de phrases générées pour tous les jeux de données.

Finalement, nous nous intéressons également à la façon dont le classificateur réagit lorsque davantage de données sont ajoutées. Alors que la table 6.3 montre les résultats lors de l'ajout de 15K et 50K exemples, ces nombres ont été choisis uniquement pour donner une idée générale de la performance. Pour avoir une meilleure idée, nous traçons à la figure 6.4 la justesse par rapport au nombre de phrases pour l'ensemble de données SST-2, en générant 50 000 phrases et en les ajoutant par lot de 5 000 phrases (les deux premiers points de données sont cependant 6690, le nombre de points dans l'ensemble de données SST-2, et 10 000). Cela nous donne non seulement une idée de la stabilité du processus MQS, mais également du taux d'augmentation de la précision.

Contrairement à ce qui pourrait être déduit de la table 6.3, ajouter plus de phrases ne conduit pas à une meilleure précision passé un certain seuil (après environ 30 000 données la performance ne semble plus augmenter). Cela suggère que pour une utilisation dans un environnement réel, une inspection régulière des performances du

VAE	SST-2 (negative)	The pianist's script is a rambling incoherence and his innocence.
	Irony (non-ironic)	#phillhughes #63 #out we miss u always respect for u [LINK]
	FakeNews (real news)	FBI Investigates Saudi Wife-Abusing to Shine Light on Safety Record
	IronyB (clashing irony)	The answers often lay the word " you just got a bit of the game : flushed_face :
	TREC6 (abbreviation)	What bay sparkles next to Miami, Florida?
CVAE	SST-2 (negative)	A bore and derivative variation of the animal planet.
	Irony (non-ironic)	Going to the Rec tonight and ya of the year
	FakeNews (real news)	Feeling Cornered, Coal Industry Borrows From Tobacco Playbook, Activists Say
	IronyB (clashing irony)	@user no one. #HowTheGrinchStoleChristmas
	TREC6 (abbreviation)	What does the name of the ' ' blue ribbon ' ' stand for?
GPT	SST-2 (negative)	It's not a bad movie, but i'm not sure i 'd like it as much as i would like it to be.
	Irony (non-ironic)	Is a great way to start off a day :smiling_face_with_open_mouth_and_cold_sweat:
	FakeNews (real news)	Professional Activists Are Behind 'Chaos' in US
	IronyB (clashing irony)	A bad game last night. Way to go Packers! #Gophers
	TREC6 (abbreviation)	What is the abbreviation of the " New York Yankees

Tab. 6.4. Exemples de phrases générées pour les différents jeux de données, selon les étiquettes de l'oracle.

classificateur est importante. Ce n'est pas un problème car, dans la construction d'un ensemble de données, l'annotation est le goulot d'étranglement du processus, tandis que dans le cadre expérimental, c'est l'entraînement d'algorithme et la génération qui ralentissent le processus d'apprentissage. Nous constatons également que, comme montré à la table 6.3, VAE-R, CVAE-R et CVAE-B fonctionnent bien, tandis que GPT-R n'apporte pas de nouvelle information.

6.5. Analyses

6.5.1. Performances et limites des générateurs

Cette section s'intéresse aux performances des générateurs, soulignant les forces et les faiblesses de chacun d'entre eux.

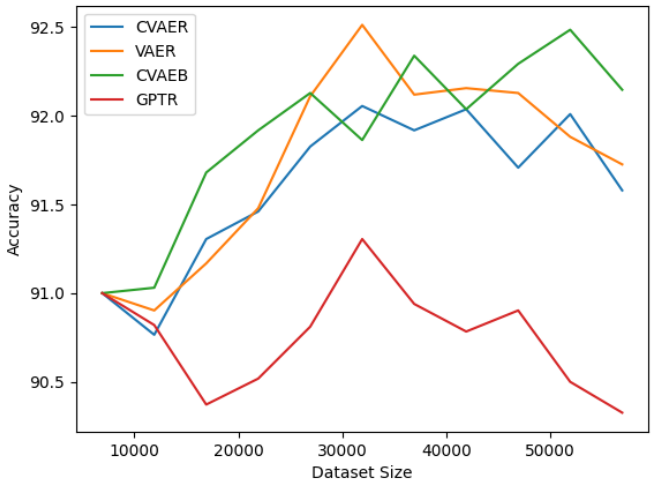

Fig. 6.4. Justesse vs le nombre de points dans le jeu de données utilisé pour entraîner le classificateur, pour SST-2.

VAE : L'auto-encodeur variationnel se retrouve à être un excellent algorithme pour générer de nouvelles phrases pour le MQS. Cependant, son efficacité est grandement dépendante de la stratégie utilisée, et seul l'échantillonnage au hasard et l'échantillonnage du postérieur s'avèrent efficaces. Comme montré au chapitre 3, les VAEs se retrouvent à être peu efficaces dans l'organisation de leur espace latent lorsque utilisé avec du texte, ce qui explique pourquoi VAE-BD se retrouve à être plutôt inefficace.

CVAE : Le VAE conditionnel, pour sa part, se retrouve de façon non surprenante extrêmement inefficace lorsque le nombre de données étiquetées est petit. Lorsqu'il a accès à toutes les données, le CVAE performe bien, mais n'arrive cependant pas à surpasser les stratégies avec les VAEs, et ce, malgré la stratégie d'échantillonnage balancé qui devrait l'avantager. En observant les données générées, il est notable que ceci est dû au fait que les algorithmes non-conditionnels apprennent à générer une

bonne balance de données naturellement, et donc il n'y a aucun avantage à utiliser cette stratégie.

SSVAE : Bien que SSVAE-C est une stratégie proposée pour surpasser le VAE-R en travaillant directement avec la confiance du classificateur, il s'avère en fait que les résultats ne sont pas meilleurs, voir un peu moins bons que ceux du VAE-R. Nous remarquons que ceci est dû à la confiance extrême que BERT a pour ses prédictions. Cette différence entre la confiance de prédiction et la justesse est souvent dénommée un problème de *calibrage* [Guo *et al.*, 2017], et dans ce cas précis a comme conséquence que le SSVAE ne voit que trop rarement des exemples de confiance faible. L'impact de cela est que le SSVAE n'arrive ainsi pas à générer ce type d'exemples, et donc ne peut performer mieux que le VAE-R.

GPT : L'utilisation de GPT-2 pour la génération de données semble naturelle puisque, contraîrement au VAE cet algorithme est entraîné sur une quantité massive de données. Lorsque nous l'utilisons dans le cas du MQS classique (peu de données étiquetées), GPT-R émerge comme le meilleur algorithme, mais lorsque nous l'utilisons pour étendre un jeu de données, GPT-R ne performe pas bien. Comme mentionné, nous supposons que ceci est dû à la difficulté de générer un grand nombre de phrase diverses en partant du même token de départ ('<bos>').

6.5.2. Utilisation de BERT comme oracle

À travers ce chapitre, BERT a été utilisé comme oracle. Cette substitution de l'annotateur humain était évidemment nécessaire pour réaliser les expériences montrées précédemment. Sans cela, le coût associé serait beaucoup trop grand pour être réalisable dans le cadre d'une thèse. Cependant, l'utilisation d'un oracle, et surtout d'un oracle entraîné sur (entre autres) le jeu de test, amène beaucoup de questions, notamment :

(1) Comment le processus se compare-t-il à l'étiquetage humain ?

(2) Est-ce que le fait d'utiliser BERT à la fois comme oracle et comme classificateur réduit le MQS à un processus de *bootstrapping*?

(3) Quel est l'influence du seuil de filtrage de l'oracle, qui filtre les exemples mal formés ?

Pour répondre à la première question, nous produisons une expérience avec un budget total de 500, une taille de départ de 100, et VAE-R comme stratégie de MQS, tout cela sur SST-2. Bien que cela ne soit pas une réponse définitive, puisqu'il ne s'agit que d'une expérience, nous obtenons 86.5% quand l'oracle est humain, et 82.5% pour BERT-FULL avec les mêmes données. Cela laisse fortement supposer que les performances obtenues plus haut sous-estiment la vraie performance du MQS.

Pendant l'annotation, nous remarquons que l'oracle est efficace globalement, mais fait des erreurs sur certaines entrées un peu plus ambiguës. Par exemple, l'oracle a annoté "What was the player's largest department store in the Virgin of the road" comme "Human Being", alors qu'un humain l'annoterait probablement comme "Entities", pour le jeu TREC6. Un autre exemple est la phrase "The movie is long, and the tabloid setpieces." pour le jeu SST-2, que l'oracle a étiqueté comme positif, alors que nous l'étiquetons comme négatif, ou du moins comme ambigüe.

Le seuil de confiance (qui détermine à partir de quel seuil les exemples sont considérés comme "incertains" et sont rejetés) cause également des problèmes dans l'annotation : les phrases qui devraient être étiquetées, mais ne le sont pas, et vice-versa. Dans la première catégorie, on retrouve par exemple "It's a lot of a good time to be a cheap, biting, and the lush of a sick and twisted.", que nous étiquetons comme négatif, et dans la deuxième catégorie, nous trouvons par exemple "You're not the truth of the plot's best films – and I'm not sure any of the year", que l'oracle étiquette comme négatif. Évidemment, l'entièreté de ces exemples représente des cas ambigus et sont rapportés ici pour montrer la difficulté de la tâche. Il est à noter

Oracle	SST-2	Irony	FakeNews	IronyB	TREC6
BERT-Full	98.2	97.3	99.8	97.8	98.6
BERT-TrainDev	91.8	91.5	88.1	43.7	97.0
BERT- Train	90.9	90.3	87.0	45.0	95.9

Tab. 6.5. Justesse des différents oracles sur l'ensemble de test.

que cette annotation n'a été réalisée que par un annotateur (l'auteur de cette thèse) et donc il est possible que des erreurs s'y soient glissées.

Pour répondre à la deuxième question, qui concerne la puissance de l'oracle vs celle du classificateur, nous reroulons les expériences avec un budget de 15K et VAE-R pour les cinq jeux de données, ce qui nous permet de mieux voir les variations de performances. Nous prenons ici trois oracles: BERT entraîné sur seulement le jeu d'entraînement, BERT entraîné sur le jeu de développement et d'entraînement, et BERT entraîné sur la totalité des données. La table 6.5 montre la justesse des différents systèmes sur le jeu de test, et la figure 6.5, selon l'oracle utilisé, avec 15K données générées.

Nous observons qu'il faut intégrer le test pour obtenir des gains lors du MQS expérimental. Il est également possible d'observer qu'utiliser train+dev apporte des légers gains par rapport à utiliser seulement train. Il est assez naturel que l'oracle doive être considérablement plus performant que le classificateur pour pouvoir apporter des gains. Cela semble indiquer que ce que nous faisons n'est pas seulement du bootstrapping, ce qui est également supporté par les études de MQS utilisant un annotateur humain qui obtiennent des résultats positifs [Schumann et Rehbein, 2019; Quteineh *et al.*, 2020].

Finalement nous testons différentes valeurs pour le filtrage de l'oracle, c'est-à-dire, la valeur en dessous de laquelle nous rejetons l'exemple car il est mal formé

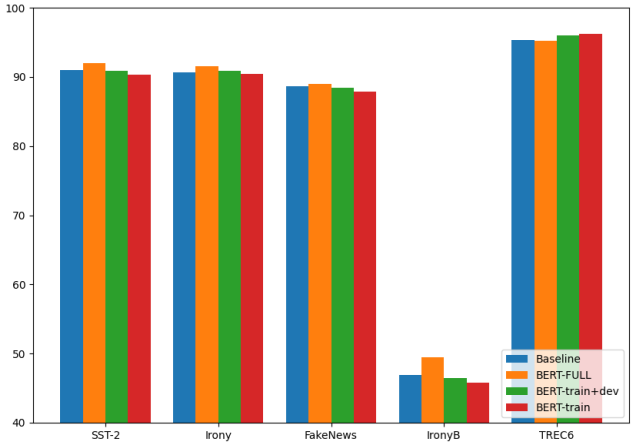

Fig. 6.5. Justesse pour les cinq jeux de données, VAE-R, et 15K données générées, et différents oracles.

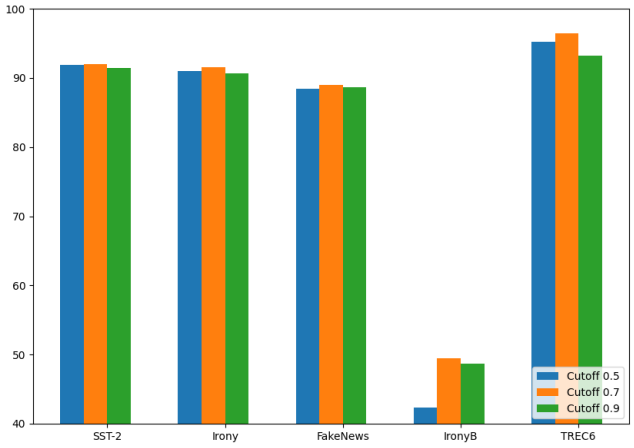

Fig. 6.6. Justesse pour les cinq jeux de données, VAE-R, et 15K données générées, avec différents seuils de filtrage pour l'oracle.

Algorithm 2 Combinaison du MQS et de l'apprentissage semi-supervisé. Si la confiance du classificateur est au-dessus du seuil t, alors on ajoute l'exemple à $\mathcal{L}$ avec la classe prédite par le classificateur. Sinon, on passe l'exemple à l'oracle et on suit le protocole normal du MQS.

procedure MQS+SEMI-SUPERVISÉ($\mathcal{L}$, $\mathcal{B}$, t):
    Budget $\leftarrow \mathcal{B}$
    Classificateur $\leftarrow$ BERT
    Threshold $\leftarrow t$
    Oracle $\leftarrow$ BERT-FULL ▷ L'oracle est entraîné sur train+dev+test
    Generator $\leftarrow$ VAE
    Generator.train($\mathcal{U}$)
    while *Budget*>0 **do**
        exemple $\leftarrow$ *Generator.generate()*
        label, *confidence* $\leftarrow$ *Classificateur.label(exemple)*
        if *exemple* in $\mathcal{L}$ **then**
            Continue
        else if *confidence*>Threshold **then**
            $\mathcal{L}$.append(*exemple, label*)
            Budget $\leftarrow$ *Budget* $-$ 1
            Continue
        label, *confidence* $\leftarrow$ *Oracle.label(exemple)*
        if *exemple* in $\mathcal{L}$ **then**
            Continue
        else if *confidence*>0.7 **then**
            $\mathcal{L}$.append(*exemple, label*)
        Budget $\leftarrow$ *Budget* $-$ 1
    Classifier.finetune($\mathcal{L}$)

ou neutre au niveau de la classe. Les résultats sont montrés à la figure 6.6, et nous pouvons confirmer que la valeur de 0.7 qui avait été choisie est en effet la valeur qui semble donner les meilleurs résultats. Il faut noter cependant que ceci n'indique rien au niveau de la correspondance entre le seuil et un comportement "humain", mais ces expérimentations sont laissées pour de la recherche future.

6.6. MQS et apprentissage semi-supervisé

Nous concluons ce chapitre en testant un dernier paradigme du MQS, à savoir la combinaison avec l'apprentissage semi-supervisé pour réduire les coûts d'annotation. Cette technique pourrait être particulièrement utile au niveau de l'extension de jeux de données, où il serait alors possible d'augmenter la performance sur des jeux existants avec un coût d'annotation réduit.

Nous reprenons un algorithme classique d'apprentissage semi-supervisé, "l'auto-étiquetage", dans lequel un classificateur annote lui-même les exemples non étiquetés, les rejetant s'ils sont en dessous d'un certain seuil de confiance. Cependant, dans le cadre de la jointure du MQS et de l'apprentissage semi-supervisé, nous apportons deux modifications.

Premièrement, nous générons les données à étiqueter avec un VAE et deuxièmement, au lieu de rejeter les exemples qui ont une confiance trop basse, nous les passons à l'oracle pour être annotés. Ceci apporte donc théoriquement une baisse du coût d'annotation puisque seul les exemples incertains auront à être annotés. Le classificateur reste le même que dans les sections précédentes, c'est-à-dire, un BERT entraîné sur le noyau $\mathcal{L}$ et ensuite itérativement sur le jeu de données produit avec MQS.

La table 6.6 montre les résultats de nos expériences. Malheureusement, le constat est que la mauvaise calibration de BERT fait en sorte qu'il est très difficile de trouver un seuil de filtrage approprié. Même à un seuil de confiance de 0.99 (c'est-à-dire que

		SST-2	Irony	FakeNews	IronyB	TREC6
	Baseline	91.0 (0.6)	90.7 (0.5)	88.6 (0.4)	63.3 (2.4)	96.4 (0.7)
	AEDA	91.0 (0.8)	91.4 (0.5)	87.8 (0.5)	64.4 (1.4)	96.3 (0.4)
	MQS+SemiSup	90.3 (0.9)	83.5 (14.3)	87.8 (0.5)	63.8 (1.2)	96.4 (0.4)
Seuil de confiance pour le classificateur	0.5	90.3 (0.9)	83.5 (14.3)	87.8 (0.5)	61.2 (2.4)	96.3 (0.5)
		0	0	0	128	68
	0.9	90.7	75.4 (18.0)	87.7 (0.4)	62.0 (2.3)	96.5 (0.7)
		2249	4842	2262	2574	1038
	0.95	90.3 (0.5)	69.2 (19.8)	88.0 (0.6)	63.0 (2.4)	96.1 (0.6)
		3109	6557	3054	3440	1569
	0.98	90.2 (0.5)	61.7 (18.7)	87.8 (0.3)	61.8 (2.4)	96.3 (0.3)
		4289	9145	4168	4662	2827
	0.99	90.1 (0.5)	48.5 (0.0)	87.5 (0.5)	61.5 (1.2)	95.4 (1.0)
		5329	11517	5091	5726	6235
	1.0	91.8 (0.4)	**91.6** (0.7)	**89.0** (0.2)	65.3 (1.7)	**97.0** (0.3)
		15000	15000	15000	15000	14695
	MQS	**92.0 (0.8)**	91.5 (0.7)	**89.0** (0.3)	**65.7** (1.5)	96.2 (0.3)

Tab. 6.6. Combinaison du MQS et de l'apprentissage semi-supervisé, avec VAE-R et 15000 exemples générés. La première ligne représente la justesse finale, et la deuxième, le nombre d'exemples étiquetés par l'oracle, sur 15000. Le reste est étiqueté par le classificateur si sa confiance est au-dessus du seuil. Comme vérification de notre processus, nous lançons aussi l'expérience avec un seuil de 1 (ce qui revient à du MQS sauf pour TREC6 pour lequel certains exemples obtiennent la confiance nécessaire pour être étiquetés par le classificateur).

tous les exemples qui ont une confiance plus petite que 0.99 par le classificateur sont étiquetés manuellement), nous nous retrouvons à devoir étiqueter autour de 5000 exemples sur 15000 générés.

Bien que nous laissions l'idée de la calibration et la preuve de concept pour un travail futur, nous espérons tout de même que l'idée persiste, et éventuellement soit

reprise pour son potentiel, notamment à la lumière de la poussée vers des algorithmes génératifs puissants cette dernière année.

6.7. Conclusion

Ce chapitre présente l'évolution naturelle de l'augmentation de données avec des modèles génératifs, c'est-à-dire, l'apprentissage par requête synthétisée. Cette technique consiste à générer des nouveaux exemples qui sont ensuite annotés et rajoutés au jeu de données étiqueté $\mathcal{L}$.

Pendant l'écriture de cette thèse, la technologie de génération de texte à fait des bonds importants, avec l'apparition notamment de ChatGPT et GPT-4 [OpenAI, 2023]. Il est probable que le processus d'annotation se concentre exponentiellement sur l'apprentissage *zero* ou *few-shot* [Reynolds et McDonell, 2021], qui permettent d'avoir de bonnes performances avec peu ou pas de données.

Nous espérons donc que le matériel présenté dans ce chapitre vienne appuyer le développement d'outils d'annotation efficaces combinant ces nouvelles technologies et techniques classiques d'apprentissage actif. Notamment, nous proposons des paradigmes pour étendre des jeux de données qui pourraient être utilisés dans un contexte industriel avec un avantage réel. Il serait particulièrement intéressant d'observer dans ce cadre la performance de ChatGPT ou GPT-4 pour générer des nouvelles données que l'on pourrait ensuite venir annoter.

Le domaine du MQS reste jeune. Comme nous l'avons mentionné, il n'y a que très peu d'études qui s'intéressent à cette branche, en raison de sa difficulté. Cependant, elle présente un grand avantage au niveau de l'annotation, et il y a peu de doutes que dans le futur, les compagnies qui utilisent des processus d'annotation plus efficaces auront la main haute sur leurs compétiteurs.

Chapitre 7

Conclusion

Les données sont au cœur même de l'intelligence artificielle, qu'elles soient annotées ou non. Bien que les avancées récentes aient introduit des méthodes telles que l'apprentissage avec peu de données (*few-shot learning*) ou même sans donnée (*zero-shot learning*), la réalité est que les données étiquetées demeurent cruciales pour de nombreuses applications.

Dans cette thèse, nous nous penchons sur la question de comment réduire le nombre d'annotations tout en gardant une performance élevée. La thèse tourne donc autour de l'augmentation de données, faisant un tour de diverses applications, notamment : l'augmentation de données pour l'anglais, pour les allolangues, et pour la génération de mots-clés. À travers ces trois chapitres, plusieurs contributions sont présentées.

Tout d'abord, l'utilisation des VAEs pour l'augmentation de mots-clés est étudiée en détail, proposant de nouvelles façons d'utiliser ceux-ci, et montrant que la cohérence de classe est un des critères les plus importants pour obtenir une augmentation de performance lors de l'AD. Ces expériences ont été publiées à COLING 2022 [Piedboeuf et Langlais, 2022a]. Puis, nous nous intéressons à l'augmentation de données utilisant ChatGPT et LLama2, les récents modèles

de langues de types conversationnels, et montrons que la performance globale ne dépasse pas celle de modèles autres, comme T5-Tapaco, un nouveau modèle que nous proposons pour l'augmentation de données ou un modèle T5 pré-entraîné sur une tâche de paraphrase. Ces derniers résultats font l'objet d'un court article à EMNLP Findings 2023 [Piedboeuf et Langlais, 2023], et un article de suivi a été mis en source libre récemment et est en court de soumission [Piedboeuf et Langlais, 2024].

L'augmentation de données pour les allolangues (les langues non anglaises) est ensuite étudiée. Nous regardons quatre jeux de données différents et montrons que deux des stratégies normalement utilisées pour ce domaine, Back-translation et EDA, ne sont en fait pas les plus efficaces globalement et que des méthodes complètement indépendantes de la langue, comme l'AEDA ou les VAEs, fonctionnent de façon plus consistante.

Le dernier volet de l'augmentation de données étudié concerne la génération de mots-clés, et plus particulièrement, la génération de mots-clés académiques francophones. L'étude de ce domaine est intéressante car elle demande le développement de techniques complètement différentes que dans les deux chapitres précédents, permettant ainsi de souligner la nécessité de développer des techniques par tâche. De plus, l'intégration de méthodes d'AD nous permet de publier un premier modèle pré-entraîné pour la génération de mots-clés francophones, et donc l'impact des techniques d'AD utilisées est importante pour les futurs utilisateurs de ce modèle. Nous commençons le chapitre en décrivant le processus de collection de données pour la création d'un corpus, puis l'entraînement de modèles ainsi que les différentes techniques d'augmentation de données utilisées. Plusieurs techniques sont explorées, certaines novatrices et d'autres tirées de la littérature. Deux articles ont été publiés sur ce chapitre, à NEURIPS 2022 [Piedboeuf et Langlais, 2022b] et

à DCMI 2023 [Piedboeuf *et al.*, 2023]. Un article en collaboration sur le KPG pour le domaine légal multilingue a également été écrit récemment [Salaün *et al.*, 2024].

Le dernier chapitre de cette thèse se penche sur un volet très proche, mais légèrement différent de l'AD: l'apprentissage par requêtes d'exemples synthétisés, ou le MQS. Dans ce domaine, des exemples sont générés, similairement à l'AD, mais sont ensuite étiquetés par un oracle. Ce domaine est encore peu étudié, dû à la difficulté d'étiqueter le nombre d'exemples nécessaire pour réaliser une étude objective. Nos contributions à ce domaine sont donc une première étude objective qui montre que le MQS peut fonctionner sur des jeux de données textuelles, ainsi que le développement de plusieurs applications novatrices du MQS (absence de données non étiquetées, jointure du MQS à l'apprentissage non supervisé, etc). Nous nous penchons également en détail sur le rôle de l'oracle, qui dans notre chapitre est un classificateur entraîné sur tout le jeu de données disponible. Un article sur ce chapitre a été publié à Canadian AI 2022 [Piedboeuf et Langlais, 2022c].

Le domaine de l'apprentissage machine évolue extrêmement vite, et ces dernières années ont vu des bouleversements majeurs à cet égard. Notamment, l'apparition de ChatGPT a rendu l'utilisation des VAEs plus ou moins obsolètes. Bien que les VAEs possèdent des avantages au niveau de ce qu'ils peuvent faire, grâce à l'espace latent, les phrases qu'ils génèrent ne sont pas aussi fluides et naturelles que les modèles larges pré-entraînés. Il s'agit bien entendu d'une limitation que tous les chercheurs en IA doivent confronter, et nous avons essayé de mettre à jour cette thèse en suivant les développements technologiques. Malgré tout, nos contributions restent importantes au niveau de l'analyse apportée, qui permet notamment de comprendre l'utilité d'un exemple rajouté au jeu de données.

Le travail établi dans cette thèse permet d'approfondir la connaissance sur de nombreux aspects, mais la compréhension de l'AD et du MQS est loin d'être complète. Au niveau de l'AD, nous ouvrons la piste que son efficacité serait liée

davantage aux propriétés du classificateur qu'à la forme des exemples générés, mais cette compréhension reste superficielle, et va aussi à l'encontre de plusieurs articles qui démontrent une augmentation de la performance en filtrant les données. Au niveau de l'AD pour les allolangues, nous avons établi quelques algorithmes de comparaison, mais l'intégration des VLLMs serait sans doute plus efficace. Le KPG francophone et multilingue bénéficierait de corpora plus larges, et de techniques d'AD plus poussées que le simple masquage des mots-clés, par exemple en dupliquant les réseaux et en entraînant un réseau pour l'extraction, et un pour la génération, avec chacun leur algorithme d'AD. Finalement, nous avons établi que le MQS avec les VAEs fonctionnait, mais il faudrait le mettre à jour avec les VLLMs et trouver de meilleurs algorithmes de génération. Il serait particulièrement intéressant de regarder leur efficacité avec les biais des jeux que nous avons trouvés, et de voir comment on peut générer des exemples qui vont dans le sens de la tâche réelle plutôt que la description fautive de la tâche.

Cette thèse s'intéresse à la diminution globale du coût d'annotation, et apporte plusieurs contributions significatives, mais la recherche sur le sujet est loin d'être complète. Nous dressons dans ce qui suit une liste des pistes de recherche à court et plus long terme.

À court terme, certaines parties du protocole que nous avons suivi pourraient être améliorées. Dans la plupart de nos expériences, nous avons mesuré la moyenne et la déviation standard, mais comme le chapitre 3 le montre, ceci n'est pas toujours révélateur. Il serait pertinent de réaliser des tests statistiques sur les résultats des autres chapitres afin d'en extraire des tendances plus solides. L'affinage des modèles d'AD pour le KPG pourrait également être amélioré. En effet, la création même du corpus s'est révélée plutôt chronophage et une attention plus grande aurait pu être portée à la sélection d'hyper-paramètres. La calibration de BERT pour le MQS

pourrait également être explorée, ce qui permettrait de générer des exemples qui sont difficiles pour le classificateur.

Parlant de génération d'exemples, les méthodes basées sur les VAEs sont aujourd'hui plutôt dépassées par les VLLMs. L'avantage principal des méthodes latentes textuelles était la génération d'exemples novateurs, grâce à la pige dans l'espace latent. Cependant, la taille de contexte élevé des VLLMs leur permet une "mémorisation" des exemples déjà générés et donc une prédiction d'exemples plus diversifiés. Il serait donc pertinent de tester l'application des VLLMs à l'AD pour les allolangues et le KPG, ainsi que pour le MQS.

Finalement, à plus long terme, il faudra trouver les meilleures stratégies de construction de jeux de données, particulièrement dans un contexte où les VLLMs peuvent en générer une partie. Réduire drastiquement le coût d'annotation permettra non seulement une démocratisation de l'apprentissage machine, mais également une meilleure inspection des jeux de données et ainsi une réduction des biais qu'ils contiennent et que nous avons souligné. Nous restons convaincu à l'issue de nos travaux que l'humain devra rester impliqué dans le processus d'augmentation de données; aussi tirer le meilleur profit de l'humain et de la machine demeure un sujet de recherche plus que pertinent.

Bibliography

Wasi Ahmad, Xiao Bai, Soomin Lee et Kai-Wei Chang : Select, Extract and Generate: Neural Keyphrase Generation with Layer-wise Coverage Attention. *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)*, pages 1389–1404, Online, août 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.acl-long.111.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón et Sumit Sanghai : GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints, mai 2023. URL http://arxiv.org/abs/2305.13245. arXiv:2305.13245 [cs].

Shorouq M. AlAwawdeh et Gheith A. Abandah : Improving the Accuracy of Semantic Similarity Prediction of Arabic Questions Using Data Augmentation and Ensemble. *In 2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 272–277, novembre 2021.

Antreas Antoniou, Amos Storkey et Harrison Edwards : Data Augmentation Generative Adversarial Networks. *arXiv:1711.04340 [cs, stat]*, mars 2018. URL http://arxiv.org/abs/1711.04340. arXiv: 1711.04340.

Philip Bachman, Alessandro Sordoni et Adam Trischler : Learning Algorithms for Active Learning. *In Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 301–310. JMLR.org, 2017. URL http://dl.acm.org/citation.cfm?id=3305381.3305413. event-place: Sydney, NSW, Australia.

Dzmitry Bahdanau, Kyunghyun Cho et Yoshua Bengio : Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, mai 2016. URL http://arxiv.org/abs/1409.0473. arXiv: 1409.0473.

Markus Bayer, Marc-André Kaufhold et Christian Reuter : A Survey on Data Augmentation for Text Classification. *ACM Computing Surveys*, 55(7):1–39, juillet 2023. ISSN 0360-0300, 1557-7341. URL https://dl.acm.org/doi/10.1145/3544558.

Iz Beltagy, Kyle Lo et Arman Cohan : SciBERT: A Pretrained Language Model for Scientific Text, septembre 2019. URL http://arxiv.org/abs/1903.10676. arXiv:1903.10676 [cs].

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major et Shmargaret Shmitchell : On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? . *In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623,

Virtual Event Canada, mars 2021. ACM. ISBN 978-1-4503-8309-7. URL https://dl.acm.org/doi/10.1145/3442188.3445922.

Y. Bengio, P. Simard et P. Frasconi : Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, mars 1994. ISSN 1941-0093. Conference Name: IEEE Transactions on Neural Networks.

Gábor Berend : Opinion Expression Mining by Exploiting Keyphrase Extraction. *In Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1162–1170, Chiang Mai, Thailand, novembre 2011. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I11-1130.

Alina Beygelzimer, Daniel J Hsu, John Langford et Chicheng Zhang : Search Improves Label for Active Learning. *In* D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon et R. Garnett, éditeurs : *Advances in Neural Information Processing Systems 29*, pages 3342–3350. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6183-search-improves-label-for-active-learning.pdf.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang et Samuel Weinbach : GPT-NeoX-20B: An Open-Source Autoregressive Language Model, avril 2022. URL http://arxiv.org/abs/2204.06745. arXiv:2204.06745 [cs].

Sid Black, Leo Gao, Phil Wang, Connor Leahy et Stella Biderman : GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, mars 2021. URL https://doi.org/10.5281/zenodo.5297715.

Adrien Bougouin, Florian Boudin et Béatrice Daille : TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. *In Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, Nagoya, Japan, octobre 2013. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I13-1062.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz et Samy Bengio : Generating Sentences from a Continuous Space. *arXiv:1511.06349 [cs]*, mai 2016. URL http://arxiv.org/abs/1511.06349. arXiv: 1511.06349.

Oliver Bown et Sebastian Lexer : Continuous-Time Recurrent Neural Networks for Generative and Interactive Musical Performance. *In* Franz Rothlauf, Jürgen Branke, Stefano Cagnoni, Ernesto Costa, Carlos Cotta, Rolf Drechsler, Evelyne Lutton, Penousal Machado, Jason H. Moore, Juan Romero, George D. Smith, Giovanni Squillero et Hideyuki Takagi, éditeurs : *Applications of Evolutionary Computing*, Lecture Notes in Computer Science, pages 652–663, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-33238-1.

Behzad Bozorgtabar, Dwarikanath Mahapatra, Hendrik von Teng, Alexander Pollinger, Lukas Ebner, Jean-Phillipe Thiran et Mauricio Reyes : Informative sample generation using class aware generative adversarial networks for classification of chest Xrays. *arXiv:1904.10781 [cs]*, avril 2019. URL http://arxiv.org/abs/1904.10781. arXiv: 1904.10781.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini

Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever et Dario Amodei : Language Models are Few-Shot Learners, juillet 2020. URL http://arxiv.org/abs/2005.14165. arXiv:2005.14165 [cs].

Samuel Budd, Emma C. Robinson et Bernhard Kainz : A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71:102062, juillet 2021. ISSN 1361-8415. URL https://www.sciencedirect.com/science/article/pii/S1361841521001080.

Yuri Burda, Roger Grosse et Ruslan Salakhutdinov : Importance Weighted Autoencoders, novembre 2016. URL http://arxiv.org/abs/1509.00519. arXiv:1509.00519 [cs, stat].

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes et Adam Jatowt : YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, janvier 2020. ISSN 0020-0255. URL https://www.sciencedirect.com/science/article/pii/S0020025519308588.

Xavier A. Carrasco, Ashraf Elnagar et Mohammed Lataifeh : A Generative Adversarial Network for Data Augmentation: The Case of Arabic Regional Dialects. *Procedia Computer Science*, 189:92–99, janvier 2021. ISSN 1877-0509. URL https://www.sciencedirect.com/science/article/pii/S1877050921011674.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha et Furong Huang : On the Possibilities of AI-Generated Text Detection, juin 2023. URL http://arxiv.org/abs/2304.04736. arXiv:2304.04736 [cs].

Hou Pong Chan, Wang Chen, Lu Wang et Irwin King : Neural Keyphrase Generation via Reinforcement Learning with Adaptive Rewards. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2163–2174, Florence, Italy, juillet 2019. Association for Computational Linguistics. URL https://aclanthology.org/P19-1208.

Andreas Chandra et Ruben Stefanus : Experiments on Paraphrase Identification Using Quora Question Pairs Dataset.

Ching-Ting Chang, Shun-Po Chuang et Hung-Yi Lee : Code-switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation, juin 2019. URL http://arxiv.org/abs/1811.02356. arXiv:1811.02356 [cs].

N. V. Chawla, K. W. Bowyer, L. O. Hall et W. P. Kegelmeyer : SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, juin 2002. ISSN 1076-9757. URL https://www.jair.org/index.php/jair/article/view/10302.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal et Diyi Yang : An Empirical Survey of Data Augmentation for Limited Data Learning in NLP. *arXiv:2106.07499 [cs]*, juin 2021. URL http://arxiv.org/abs/2106.07499. arXiv: 2106.07499.

Lin Chen, Hamed Hassani et Amin Karbasi : Near-Optimal Active Learning of Halfspaces via Query Synthesis in the Noisy Setting. *In Thirty-First AAAI Conference on Artificial Intelligence*, février 2017. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14230.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau et Yoshua Bengio : On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, octobre 2014. URL http://arxiv.org/abs/1409.1259. arXiv:1409.1259 [cs, stat].

Md Faisal Mahbub Chowdhury, Gaetano Rossiello, Michael Glass, Nandana Mihindukulasooriya et Alfio Gliozzo : Applying a Generic Sequence-to-Sequence Model for Simple and Effective Keyphrase Generation. *arXiv:2201.05302 [cs]*, janvier 2022. URL http://arxiv.org/abs/2201.05302. arXiv: 2201.05302.

Junyoung Chung, Sungjin Ahn et Yoshua Bengio : Hierarchical Multiscale Recurrent Neural Networks, mars 2017. URL http://arxiv.org/abs/1609.01704. arXiv:1609.01704 [cs].

Mark Cieliebak, Jan Milan Deriu, Dominic Egger et Fatih Uzdilli : A Twitter Corpus and Benchmark Resources for German Sentiment Analysis. *In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 45–51, Valencia, Spain, avril 2017. Association for Computational Linguistics. URL https://aclanthology.org/W17-1106.

Charles Condevaux et Sébastien Harispe : LSG Attention: Extrapolation of Pretrained Transformers to Long Sequences. *In* Hisashi Kashima, Tsuyoshi Ide et Wen-Chih Peng, éditeurs : *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pages 443–454, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-33374-3.

Jean-Philippe Corbeil et Hadi Abdi Ghadivel : BET: A Backtranslation Approach for Easy Data Augmentation in Transformer-based Paraphrase Identification Context, septembre 2020. URL http://arxiv.org/abs/2009.12452. arXiv:2009.12452 [cs].

Claude Coulombe : Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs, décembre 2018. URL http://arxiv.org/abs/1812.04718. arXiv:1812.04718 [cs].

Bin Dai, Ziyu Wang et David Wipf : The Usual Suspects? Reassessing Blame for VAE Posterior Collapse. *In Proceedings of the 37th International Conference on Machine Learning*, pages 2313–2322. PMLR, novembre 2020. URL https://proceedings.mlr.press/v119/dai20c.html. ISSN: 2640-3498.

Bin Dai et David Wipf : DIAGNOSING AND ENHANCING VAE MODELS. page 12, 2019.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Quanzheng Li, Dinggang Shen, Tianming Liu et Xiang Li : ChatAug: Leveraging ChatGPT for Text Data Augmentation, février 2023. URL http://arxiv.org/abs/2302.13007. arXiv:2302.13007 [cs].

Xiang Dai et Heike Adel : An Analysis of Simple Data Augmentation for Named Entity Recognition. *In Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867, Barcelona, Spain (Online), décembre 2020. International Committee on Computational Linguistics. URL https://aclanthology.org/2020.coling-main.343.

Sanjoy Dasgupta et Daniel Hsu : Hierarchical sampling for active learning. *In Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 208–215, Helsinki, Finland, 2008. ACM Press. ISBN 978-1-60558-205-4. URL http://portal.acm.org/citation.cfm?doid=1390156.1390183.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman et Luke Zettlemoyer : QLoRA: Efficient Finetuning of Quantized LLMs, mai 2023. URL http://arxiv.org/abs/2305.14314. arXiv:2305.14314 [cs].

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)*, pages 4171–4186, Minneapolis, Minnesota, juin 2019. Association for Computational Linguistics. URL https://aclanthology.org/N19-1423.

Sergey Edunov, Myle Ott, Michael Auli et David Grangier : Understanding Back-Translation at Scale, octobre 2018. URL http://arxiv.org/abs/1808.09381. arXiv:1808.09381 [cs].

Yihao Fang, Xianzhi Li, Stephen W. Thomas et Xiaodan Zhu : ChatGPT as Data Augmentation for Compositional Generalization: A Case Study in Open Intent Detection, août 2023. URL http://arxiv.org/abs/2308.13517. arXiv:2308.13517 [cs].

Isaac Feldman et Rolando Coto-Solano : Neural Machine Translation Models with Back-Translation for the Extremely Low-Resource Indigenous Language Bribri. *In Proceedings of the 28th International Conference on Computational Linguistics*, pages 3965–3976, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.coling-main.351.

Ruili Feng, Deli Zhao et Zhengjun Zha : On Noise Injection in Generative Adversarial Networks. *arXiv:2006.05891 [cs, stat]*, juin 2020. URL http://arxiv.org/abs/2006.05891. arXiv: 2006.05891.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura et Eduard Hovy : A Survey of Data Augmentation Approaches for NLP. *In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online, août 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.findings-acl.84.

Raphael Antonius Frick : Fraunhofer sit at CheckThat! 2023: can LLMs be used for data augmentation & few-shot classification? detecting subjectivity in text using chatGPT. *Working Notes of CLEF*, 2023.

Ygor Gallina, Florian Boudin et Béatrice Daille : KPTimes: A Large-Scale Dataset for Keyphrase Generation on News Documents. *arXiv:1911.12559 [cs]*, novembre 2019. URL http://arxiv.org/abs/1911.12559. arXiv: 1911.12559.

Yifan Gao, Qingyu Yin, Zheng Li, Rui Meng, Tong Zhao, Bing Yin, Irwin King et Michael R. Lyu : Retrieval-Augmented Multilingual Keyphrase Generation with Retriever-Generator Iterative Training, juin 2022. URL http://arxiv.org/abs/2205.10471. arXiv:2205.10471 [cs].

Krishna Garg, Jishnu Ray Chowdhury et Cornelia Caragea : Data Augmentation for Low-Resource Keyphrase Generation, mai 2023. URL http://arxiv.org/abs/2305.17968. arXiv:2305.17968 [cs].

Zelalem Gero et Joyce C. Ho : Uncertainty-based Self-training for Biomedical Keyphrase Extraction. *In 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–4, juillet 2021. ISSN: 2641-3604.

Nikolaos Giarelis, Nikos Kanakaris et Nikos Karacapilidis : A Comparative Assessment of State-Of-The-Art Methods for Multilingual Unsupervised Keyphrase Extraction. *In Artificial Intelligence Applications and Innovations*, pages 635–645. Springer, Cham, 2021. URL https://link.springer.com/chapter/10.1007/978-3-030-79150-6_50.

Fabrizio Gilardi, Meysam Alizadeh et Maël Kubli : ChatGPT Outperforms Crowd-Workers for Text-Annotation Tasks, mars 2023. URL http://arxiv.org/abs/2303.15056. arXiv:2303.15056 [cs].

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville et Yoshua Bengio : Generative Adversarial Nets. page 9, 2014.

Jacob M. Graving et Iain D. Couzin : VAE-SNE: a deep generative model for simultaneous dimensionality reduction and clustering. preprint, Animal Behavior and Cognition, juillet 2020. URL http://biorxiv.org/lookup/doi/10.1101/2020.07.17.207993.

Maarten Grootendorst : KeyBERT: Minimal keyword extraction with BERT., 2020. URL https://doi.org/10.5281/zenodo.4461265. Version Number: v0.3.0.

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*, pages 1321–1330. PMLR, juillet 2017. URL https://proceedings.mlr.press/v70/guo17a.html. ISSN: 2640-3498.

Zhen Guo, Peiqi Wang, Yanwei Wang et Shangdi Yu : Dr. LLaMA: Improving Small Language Models in Domain-Specific QA via Generative Data Augmentation, mai 2023. URL http://arxiv.org/abs/2305.07804. arXiv:2305.07804 [cs].

Rahul Gupta : Data augmentation for low resource sentiment analysis using generative adversarial networks, février 2019. URL http://arxiv.org/abs/1902.06818. arXiv:1902.06818 [cs, stat].

Suchin Gururangan, Tam Dang, Dallas Card et Noah A. Smith : Variational Pretraining for Semi-supervised Text Classification. *arXiv:1906.02242 [cs]*, juin 2019. URL http://arxiv.org/abs/1906.02242. arXiv: 1906.02242.

Mario Guzman-Silverio, Ángel Balderas-Paredes et Adrián Pastor López-Monroy : Transformers and Data Augmentation for Aggressiveness Detection in Mexican Spanish.

Gholamreza Haffari, Maxim Roy et Anoop Sarkar : Active learning for statistical phrase-based machine translation. *In Proceedings of Human Language Technologies: The 2009 Annual*

Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '09, page 415, Boulder, Colorado, 2009. Association for Computational Linguistics. ISBN 978-1-932432-41-1. URL http://portal.acm.org/citation.cfm?doid=1620754.1620815.

Khaled M. Hammouda, Diego N. Matute et Mohamed S. Kamel : CorePhrase: Keyphrase Extraction for Document Clustering. *In* Petra Perner et Atsushi Imiya, éditeurs : *Machine Learning and Data Mining in Pattern Recognition*, Lecture Notes in Computer Science, pages 265–274, Berlin, Heidelberg, 2005. Springer. ISBN 978-3-540-31891-0.

T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo et K. Takeda : Back-Translation-Style Data Augmentation for end-to-end ASR. *In 2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 426–433, décembre 2018.

Haibo He, Yang Bai, Edwardo A. Garcia et Shutao Li : ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, juin 2008. ISSN: 2161-4407.

Sepp Hochreiter et Jürgen Schmidhuber : Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997. Publisher: MIT Press.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals et Laurent Sifre : Training Compute-Optimal Large Language Models, mars 2022. URL http://arxiv.org/abs/2203.15556. arXiv:2203.15556 [cs].

Minui Hong, Jinwoo Choi et Gunhee Kim : StyleMix: Separating Content and Style for Enhanced Data Augmentation. page 9, 2021.

Max Hopkins, Daniel Kane, Shachar Lovett et Gaurav Mahajan : Point Location and Active Learning: Learning Halfspaces Almost Optimally. *In 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1034–1044, novembre 2020. ISSN: 2575-8454.

Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama et Kevin Murphy : Speed/accuracy trade-offs for modern convolutional object detectors, avril 2017. URL http://arxiv.org/abs/1611.10012. arXiv:1611.10012 [cs].

Anette Hulth : Improved Automatic Keyword Extraction Given More Linguistic Knowledge. *In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223, 2003. URL https://aclanthology.org/W03-1028.

Matthew Iceland : How Good Are SOTA Fake News Detectors, août 2023. URL http://arxiv.org/abs/2308.02727. arXiv:2308.02727 [cs].

Zubayer Islam, Mohamed Abdel-Aty, Qing Cai et Jinghui Yuan : Crash data augmentation using variational autoencoder. *Accident Analysis & Prevention*, 151:105950, mars 2021. ISSN 0001-4575. URL https://www.sciencedirect.com/science/article/pii/

S000145752031770X.

Zengrui Jin, Mengzhe Geng, Xurong Xie, Jianwei Yu, Shansong Liu, Xunying Liu et Helen Meng : Adversarial Data Augmentation for Disordered Speech Recognition. *arXiv:2108.00899 [cs, eess]*, août 2021. URL http://arxiv.org/abs/2108.00899. arXiv: 2108.00899.

Ehsan Kamalloo, Mehdi Rezagholizadeh, Peyman Passban et Ali Ghodsi : Not Far Away, Not So Close: Sample Efficient Nearest Neighbour Data Augmentation via MiniMax. *In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3522–3533, Online, août 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.findings-acl.309.

Akbar Karimi, Leonardo Rossi et Andrea Prati : AEDA: An Easier Data Augmentation Technique for Text Classification. *In Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic, novembre 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.findings-emnlp.234.

Daiki Katsuma, Hiroharu Kawanaka, V B Surya Prasath et Bruce J Aronow : Data Augmentation Using Generative Adversarial Networks for Multi-Class Segmentation of Lung Confocal IF Images. page 9, 2022.

Daniel Martin Katz, Michael James Bommarito, Shang Gao et Pablo Arredondo : GPT-4 Passes the Bar Exam, mars 2023. URL https://papers.ssrn.com/abstract=4389233.

Shubham Khandelwal, Benjamin Lecouteux et Laurent Besacier : COMPARING GRU AND LSTM FOR AUTOMATIC SPEECH RECOGNITION. Research Report, LIG, janvier 2016. URL https://hal.science/hal-01633254.

Su Nam Kim, Olena Medelyan, Min-Yen Kan et Timothy Baldwin : SemEval-2010 Task 5 : Automatic Keyphrase Extraction from Scientific Articles. *In Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Uppsala, Sweden, juillet 2010. Association for Computational Linguistics. URL https://aclanthology.org/S10-1004.

Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed et Max Welling : Semi-supervised learning with deep generative models. *In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3581–3589, Cambridge, MA, USA, décembre 2014. MIT Press.

Diederik P. Kingma et Max Welling : Auto-Encoding Variational Bayes. décembre 2013. URL https://openreview.net/forum?id=33X9fd2-9FyZd&source=post_page----------.

Diederik P. Kingma et Max Welling : Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, mai 2014. URL http://arxiv.org/abs/1312.6114. arXiv: 1312.6114.

Sosuke Kobayashi : Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana, juin 2018. Association for Computational Linguistics. URL https://aclanthology.org/N18-2072.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szyd\lo, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz et others : ChatGPT: Jack of all trades, master of none. *Information Fusion*, page 101861, 2023. Publisher: Elsevier.

Pang Wei Koh et Percy Liang : Understanding Black-box Predictions via Influence Functions, décembre 2020. URL http://arxiv.org/abs/1703.04730. arXiv:1703.04730 [cs, stat].

Mikalai Krapivin, Aliaksandr Autaeu et Maurizio Marchese : Large Dataset for Keyphrases Extraction. Departmental Technical Report, University of Trento, 2009. URL http://eprints.biblio.unitn.it/1671/.

Alex Krizhevsky, Ilya Sutskever et Geoffrey E. Hinton : ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, mai 2017. ISSN 0001-0782. URL https://doi.org/10.1145/3065386.

Mayank Kulkarni, Debanjan Mahata, Ravneet Arora et Rajarshi Bhowmik : Learning Rich Representation of Keyphrases from Text. *In Findings of the Association for Computational Linguistics: NAACL 2022*, pages 891–906, Seattle, United States, juillet 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-naacl.67.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari et Partha Talukdar : Submodular Optimization-based Diverse Paraphrasing and its Effectiveness in Data Augmentation. *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)*, pages 3609–3619, Minneapolis, Minnesota, juin 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N19-1363.

Varun Kumar, Ashutosh Choudhary et Eunah Cho : Data Augmentation using Pre-trained Transformer Models. *In Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China, décembre 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.lifelongnlp-1.3.

Varun Kumar, Ashutosh Choudhary et Eunah Cho : Data Augmentation using Pre-trained Transformer Models, janvier 2021. URL http://arxiv.org/abs/2003.02245. arXiv:2003.02245 [cs].

Taja Kuzman, Igor Mozetič et Nikola Ljubešić : ChatGPT: Beginning of an End of Manual Linguistic Data Annotation? Use Case of Automatic Genre Identification, mars 2023. URL http://arxiv.org/abs/2303.03953. arXiv:2303.03953 [cs].

John P Lalor, Yi Yang, Kendall Smith, Nicole Forsgren et Ahmed Abbasi : Benchmarking intersectional biases in NLP. *In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3598–3609, 2022.

Giuseppe Lancioni, Saida S.Mohamed, Beatrice Portelli, Giuseppe Serra et Carlo Tasso : Keyphrase Generation with GANs in Low-Resources Scenarios. *In Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 89–96, Online, novembre 2020. Association for Computational Linguistics. URL

https://aclanthology.org/2020.sustainlp-1.12.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle et Ole Winther : Autoencoding beyond pixels using a learned similarity metric. *arXiv:1512.09300 [cs, stat]*, février 2016. URL http://arxiv.org/abs/1512.09300. arXiv: 1512.09300.

Y. Lecun, L. Bottou, Y. Bengio et P. Haffner : Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, novembre 1998. ISSN 1558-2256. Conference Name: Proceedings of the IEEE.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov et Luke Zettlemoyer : BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, juillet 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.acl-main.703.

Bo Li, Kaitao Xue, Bin Liu et Yu-Kun Lai : BBDM: Image-to-image Translation with Brownian Bridge Diffusion Models, mars 2023. URL http://arxiv.org/abs/2205.07680. arXiv:2205.07680 [cs, eess].

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang et Jianfeng Gao : Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699, Online, novembre 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.emnlp-main.378.

Juanzi Li, Qi'na Fan et Kuo Zhang : Keyword extraction based on tf/idf for Chinese news document. *Wuhan University Journal of Natural Sciences*, 12(5):917–921, septembre 2007. ISSN 1993-4998. URL https://doi.org/10.1007/s11859-007-0038-4.

Xin Li et Dan Roth : Learning question classifiers. *In Proceedings of the 19th international conference on Computational linguistics* -, volume 1, pages 1–7, Taipei, Taiwan, 2002. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?doid=1072228.1072378.

Tomas Liesting, Flavius Frasincar et Maria Mihaela Truşcă : Data augmentation in a hybrid approach for aspect-based sentiment analysis. *In Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, pages 828–835, New York, NY, USA, mars 2021. Association for Computing Machinery. ISBN 978-1-4503-8104-8. URL https://doi.org/10.1145/3412841.3441958.

Qianying Liu, Daisuke Kawahara et Sujian Li : Scientific Keyphrase Extraction: Extracting Candidates with Semi-supervised Data Augmentation. *In* Maosong Sun, Ting Liu, Xiaojie Wang, Zhiyuan Liu et Yang Liu, éditeurs : *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, Lecture Notes in Computer Science, pages 183–194, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01716-3.

Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang et Soroush Vosoughi : Data Boost: Text Data Augmentation Through Reinforcement Learning Guided Conditional

Generation. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9031–9041, Online, novembre 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.emnlp-main.726.

James Lucas, George Tucker, Roger B Grosse et Mohammad Norouzi : Don't Blame the ELBO! A Linear VAE Perspective on Posterior Collapse.

Hongxin Luo : Emotion Detection for Spanish with Data Augmentation and Transformer-Based Models. *In IberLEF@ SEPLN*, pages 35–42, 2021.

Jun Ma et Langlang Li : Data Augmentation For Chinese Text Classification Using Back-Translation. *Journal of Physics: Conference Series*, 1651(1):012039, novembre 2020. ISSN 1742-6596. URL https://dx.doi.org/10.1088/1742-6596/1651/1/012039. Publisher: IOP Publishing.

Laurens van der Maaten et Geoffrey Hinton : Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. ISSN 1533-7928. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Debanjan Mahata, Navneet Agarwal, Dibya Gautam, Amardeep Kumar, Swapnil Parekh, Yaman Kumar Singla, Anish Acharya et Rajiv Ratn Shah : LDKP: A Dataset for Identifying Keyphrases from Long Scientific Documents. *arXiv:2203.15349 [cs]*, avril 2022. URL http://arxiv.org/abs/2203.15349. arXiv: 2203.15349.

Miftahul Mahfuzh, Sidik Soleman et Ayu Purwarianti : Improving Joint Layer RNN based Keyphrase Extraction by Using Syntactical Features. *In 2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, pages 1–6, septembre 2019.

Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi et Angeliki Metallinou : Controlled Text Generation for Data Augmentation in Intelligent Artificial Agents. *In Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 90–98, Hong Kong, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D19-5609.

Vukosi Marivate et Tshephisho Sefara : Improving Short Text Classification Through Global Augmentation Methods. *In* Andreas Holzinger, Peter Kieseberg, A Min Tjoa et Edgar Weippl, éditeurs : *Machine Learning and Knowledge Extraction*, Lecture Notes in Computer Science, pages 385–399, Cham, 2020. Springer International Publishing. ISBN 978-3-030-57321-8.

Vukosi Marivate, Tshephisho Sefara, Vongani Chabalala, Keamogetswe Makhaya, Tumisho Mokgonyane, Rethabile Mokoena et Abiodun Modupe : Investigating an approach for low resource language dataset creation, curation and classification: Setswana and Sepedi, février 2020. URL http://arxiv.org/abs/2003.04986. arXiv:2003.04986 [cs, stat].

Roberto Martínez-Cruz, Alvaro J. López-López et José Portela : ChatGPT vs State-of-the-Art Models: A Benchmarking Study in Keyphrase Generation Task, juin 2023. URL http://arxiv.org/abs/2304.14177. arXiv:2304.14177 [cs].

Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, Tom Tseng, Tomasz Korbak, Xudong Shen, Yuhui Zhang, Zhengping Zhou, Najoung Kim, Samuel R. Bowman et Ethan Perez : Inverse Scaling: When Bigger Isn't Better, juin 2023. URL http://arxiv.org/abs/2306.09479. arXiv:2306.09479 [cs].

Olena Medelyan et Ian H. Witten : Thesaurus based automatic keyphrase indexing. *In Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '06, pages 296–297, New York, NY, USA, juin 2006. Association for Computing Machinery. ISBN 978-1-59593-354-6. URL https://doi.org/10.1145/1141753.1141819.

Rui Meng, Xingdi Yuan, Tong Wang, Peter Brusilovsky, Adam Trischler et Daqing He : *Does Order Matter? An Empirical Study on Generating Multiple Keyphrases as a Sequence*. septembre 2019.

Rui Meng, Xingdi Yuan, Tong Wang, Sanqiang Zhao, Adam Trischler et Daqing He : An Empirical Study on Neural Keyphrase Generation. *arXiv:2009.10229 [cs]*, avril 2021. URL http://arxiv.org/abs/2009.10229. arXiv: 2009.10229.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky et Yu Chi : Deep Keyphrase Generation. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Vancouver, Canada, 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-1054.

Sepideh Mesbah, Jie Yang, Robert-Jan Sips, Manuel Valle Torre, Christoph Lofi, Alessandro Bozzon et Geert-Jan Houben : Training Data Augmentation for Detecting Adverse Drug Reactions in User-Generated Content. *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)*, pages 2349–2359, Hong Kong, China, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D19-1239.

Rada Mihalcea et Paul Tarau : TextRank: Bringing Order into Text. *In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, juillet 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-3252.

Joshua R. Minot, Nicholas Cheney, Marc Maier, Danne C. Elbers, Christopher M. Danforth et Peter Sheridan Dodds : Interpretable bias mitigation for textual data: Reducing gender bias in patient notes while maintaining classification performance, mars 2021. URL http://arxiv.org/abs/2103.05841. arXiv:2103.05841 [cs, stat].

Jihyung Moon, Won Ik Cho et Junbum Lee : BEEP! Korean Corpus of Online News Comments for Toxic Speech Detection, mai 2020. URL http://arxiv.org/abs/2005.12503. arXiv:2005.12503 [cs].

Anders Giovanni Møller, Jacob Aarup Dalsgaard, Arianna Pera et Luca Maria Aiello : Is a prompt and a few samples all you need? Using GPT-4 for data augmentation in low-resource classification tasks, avril 2023. URL http://arxiv.org/abs/2304.13861. arXiv:2304.13861

[physics].

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli et Sergey Edunov : Facebook FAIR's WMT19 News Translation Task Submission, juillet 2019. URL http://arxiv.org/abs/1907.06616. arXiv:1907.06616 [cs].

Thuy Dung Nguyen et Min-Yen Kan : Keyphrase Extraction in Scientific Publications. *In* Dion Hoe-Lian Goh, Tru Hoang Cao, Ingeborg Torvik Sølvberg et Edie Rasmussen, éditeurs : *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, Lecture Notes in Computer Science, pages 317–326, Berlin, Heidelberg, 2007. Springer. ISBN 978-3-540-77094-7.

H. Nishizaki : Data augmentation and feature extraction using variational autoencoder for acoustic modeling. *In 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1222–1227, décembre 2017.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan et Eric Horvitz : Capabilities of GPT-4 on Medical Challenge Problems, avril 2023. URL http://arxiv.org/abs/2303.13375. arXiv:2303.13375 [cs].

Itsuki Okimura, Machel Reid, Makoto Kawano et Yutaka Matsuo : On the Impact of Data Augmentation on Downstream Performance in Natural Language Processing. *In Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 88–93, Dublin, Ireland, mai 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.insights-1.12.

Eda Okur, Saurav Sahay et Lama Nachman : Data Augmentation with Paraphrase Generation and Entity Extraction for Multimodal Dialogue System, mai 2022. URL http://arxiv.org/abs/2205.04006. arXiv:2205.04006 [cs].

Etienne Ollion, Rubing Shen, Ana Macanovic et Arnault Chatelain : Chatgpt for Text Annotation? Mind the Hype! 2023. Publisher: SocArXiv.

OpenAI : GPT-4 Technical Report, mars 2023. URL http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs].

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike et Ryan Lowe : Training language models to follow instructions with human feedback, mars 2022. URL http://arxiv.org/abs/2203.02155. arXiv:2203.02155 [cs].

Subhadarshi Panda, Caglar Tirkaz, Tobias Falke et Patrick Lehnen : Multilingual Paraphrase Generation For Bootstrapping New Features in Task-Oriented Dialog Systems. *In Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 30–39, Online, novembre 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.nlp4convai-1.4.

Seongmin Park et Jihwa Lee : Finetuning Pretrained Transformers into Variational Autoencoders. *In Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 29–35,

Online and Punta Cana, Dominican Republic, novembre 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.insights-1.5.

Amandalynne Paullada, Inioluwa Deborah Raji, Emily M. Bender, Emily Denton et Alex Hanna : Data and its (dis)contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11):100336, novembre 2021. ISSN 2666-3899. URL https://www.sciencedirect.com/science/article/pii/S2666389921001847.

Thang Pham, Trung Bui, Long Mai et Anh Nguyen : Out of Order: How important is the sequential order of words in a sentence in Natural Language Understanding tasks? *In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1145–1160, Online, août 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.findings-acl.98.

Frédéric Piedboeuf et Philippe Langlais : Effective Data Augmentation for Sentence Classification Using One VAE per Class. *In Proceedings of the 29th International Conference on Computational Linguistics*, pages 3454–3464, Gyeongju, Republic of Korea, octobre 2022a. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.305.

Frédéric Piedboeuf et Philippe Langlais : A new dataset for multilingual keyphrase generation. octobre 2022b. URL https://openreview.net/forum?id=47qVX2pa-2&referrer=%5Bthe%20profile%20of%20Fr%C3%A9d%C3%A9ric%20Piedboeuf%5D(%2Fprofile%3Fid%3D~Fr%C3%A9d%C3%A9ric_Piedboeuf1).

Frédéric Piedboeuf et Philippe Langlais : A working model for textual Membership Query Synthesis. *Proceedings of the Canadian Conference on Artificial Intelligence*, mai 2022c. URL https://caiac.pubpub.org/pub/f6b0scvi/release/1. Publisher: Canadian Artificial Intelligence Association (CAIAC).

Frédéric Piedboeuf et Philippe Langlais : Is ChatGPT the ultimate Data Augmentation Algorithm. *In Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Frédéric Piedboeuf et Philippe Langlais : Data Augmentation is Dead, Long Live Data Augmentation, février 2024. URL http://arxiv.org/abs/2402.14895. arXiv:2402.14895 [cs].

Frédéric Piedboeuf, Guillaume Le Berre, David Alfonso-Hermelo, Olivier Charbonneau et Philippe Langlais : The state of OAI-PMH repositories in Canadian Universities. *In International conference on dublin core and metadata applications*, 2023.

Domagoj Pluščec et Jan Šnajder : Data Augmentation for Neural NLP, février 2023. URL http://arxiv.org/abs/2302.11412. arXiv:2302.11412 [cs].

Peter Prettenhofer et Benno Stein : Cross-Language Text Classification Using Structural Correspondence Learning. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1127, Uppsala, Sweden, juillet 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-1114.

Aman Priyanshu et Supriti Vijay : AdaptKeyBERT: An Attention-Based approach towards Few-Shot & Zero-Shot Domain Adaptation of KeyBERT, novembre 2022. URL http://arxiv.org/abs/2211.07499. arXiv:2211.07499 [cs].

Vahed Qazvinian, Dragomir R. Radev et Arzucan Özgür : Citation Summarization Through Keyphrase Extraction. *In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 895–903, Beijing, China, août 2010. Coling 2010 Organizing Committee. URL https://aclanthology.org/C10-1101.

Siyuan Qiu, Binxia Xu, Jie Zhang, Yafang Wang, Xiaoyu Shen, Gerard de Melo, Chong Long et Xiaolong Li : EasyAug: An Automatic Textual Data Augmentation Platform for Classification Tasks. *In Companion Proceedings of the Web Conference 2020*, pages 249–252. Association for Computing Machinery, New York, NY, USA, avril 2020. ISBN 978-1-4503-7024-0. URL https://doi.org/10.1145/3366424.3383552.

Husam Quteineh, Spyridon Samothrakis et Richard Sutcliffe : Textual Data Augmentation for Efficient Active Learning on Tiny Datasets. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7400–7410, Online, 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-main.600.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei et Ilya Sutskever : Language Models are Unsupervised Multitask Learners. page 24, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li et Peter J. Liu : Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. ISSN 1533-7928. URL http://jmlr.org/papers/v21/20-074.html.

Mahdyar Ravanbakhsh, Tassilo Klein, Kayhan Batmanghelich et Moin Nabi : Uncertainty-Driven Semantic Segmentation through Human-Machine Collaborative Learning. avril 2019. URL https://openreview.net/forum?id=rkgnwY04cV.

Jishnu Ray Chowdhury, Seo Yeon Park, Tuhin Kundu et Cornelia Caragea : KPDROP: Improving Absent Keyphrase Generation. *In Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4853–4870, Abu Dhabi, United Arab Emirates, décembre 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.357.

Mehdi Regina, Maxime Meyer et Sébastien Goutal : Text Data Augmentation: Towards better detection of spear-phishing emails, mars 2021. URL http://arxiv.org/abs/2007.02033. arXiv:2007.02033 [cs].

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen et Xin Wang : A Survey of Deep Active Learning. *ACM Computing Surveys*, 54(9): 180:1–180:40, octobre 2021. ISSN 0360-0300. URL https://doi.org/10.1145/3472291.

Laria Reynolds et Kyle McDonell : Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm, février 2021. URL http://arxiv.org/abs/2102.07350. arXiv:2102.07350 [cs].

Danilo Jimenez Rezende et Shakir Mohamed : Variational Inference with Normalizing Flows. *arXiv:1505.05770 [cs, stat]*, juin 2016. URL http://arxiv.org/abs/1505.05770. arXiv: 1505.05770.

Stephen Robertson et Hugo Zaragoza : The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. ISSN 1554-0669, 1554-0677. URL http://www.nowpublishers.com/article/Details/INR-019.

David E Rumelhart, Geoffrey E Hinton et Ronald J Williams : Learning internal representations by error propagation. Rapport technique, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder et Iryna Gurevych : How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models, juin 2021. URL http://arxiv.org/abs/2012.15613. arXiv:2012.15613 [cs].

Caroline Sabty, Islam Omar, Fady Wasfalla, Mohamed Islam et Slim Abdennadher : Data Augmentation Techniques on Arabic Data for Named Entity Recognition. *Procedia Computer Science*, 189:292–299, janvier 2021. ISSN 1877-0509. URL https://www.sciencedirect.com/science/article/pii/S1877050921012126.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet et Mohammad Norouzi : Image Super-Resolution via Iterative Refinement, juin 2021. URL http://arxiv.org/abs/2104.07636. arXiv:2104.07636 [cs, eess].

Gaurav Sahu, Pau Rodriguez, Issam H. Laradji, Parmida Atighehchian, David Vazquez et Dzmitry Bahdanau : Data Augmentation for Intent Classification with Off-the-shelf Large Language Models, avril 2022. URL http://arxiv.org/abs/2204.01959. arXiv:2204.01959 [cs].

Olivier Salaün, Frédéric Piedboeuf, Guillaume Le Berre, David Alfonso Hermelo et Philippe Langlais : EUROPA: A Legal Multilingual Keyphrase Generation Dataset, février 2024. URL http://arxiv.org/abs/2403.00252. arXiv:2403.00252 [cs].

Yves Scherrer : TaPaCo: A Corpus of Sentential Paraphrases for 73 Languages, mars 2020. URL https://zenodo.org/record/3707949.

Christopher Schröder et Andreas Niekler : A Survey of Active Learning for Text Classification using Deep Neural Networks. *arXiv:2008.07267 [cs]*, août 2020. URL http://arxiv.org/abs/2008.07267. arXiv: 2008.07267.

Raphael Schumann et Ines Rehbein : Active Learning via Membership Query Synthesis for Semi-Supervised Sentence Classification. *In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 472–481, Hong Kong, China, novembre 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/K19-1044.

Roy Schwartz, Jesse Dodge, Noah A. Smith et Oren Etzioni : Green AI, août 2019. URL http://arxiv.org/abs/1907.10597. arXiv:1907.10597 [cs, stat].

Saskia Senn, ML Tlachac, Ricardo Flores et Elke Rundensteiner : Ensembles of BERT for Depression Classification. *In 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 4691–4694, juillet 2022. ISSN: 2694-0604.

Yu Shang, Xiaohui Su, Zhifeng Xiao et Zidong Chen : Campus Sentiment Analysis with GAN-based Data Augmentation. *In 2021 13th International Conference on Advanced Infocomm Technology (ICAIT)*, pages 209–214, octobre 2021. ISSN: 2770-1603.

Ashwyn Sharma et David I Feldman : Team Cadence at MEDIQA-Sum 2023: Using ChatGPT as a Data Augmentation Tool for Classifying Clinical Dialogue. CLEF, 2023.

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*, pages 252–256, Vancouver, Canada, août 2017. Association for Computational Linguistics. URL https://aclanthology.org/W17-2630.

Connor Shorten et Taghi M. Khoshgoftaar : A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60, juillet 2019. ISSN 2196-1115. URL https://doi.org/10.1186/s40537-019-0197-0.

Elena Shushkevich et John Cardiff : Tudublin at CheckThat! 2023: Chatgpt for data augmentation. *Working Notes of CLEF*, 2023.

Alexander Shvets et Leo Wanner : Concept Extraction Using Pointer–Generator Networks and Distant Supervision for Data Augmentation. *In* C. Maria Keet et Michel Dumontier, éditeurs : *Knowledge Engineering and Knowledge Management*, volume 12387, pages 120–135. Springer International Publishing, Cham, 2020. ISBN 978-3-030-61243-6 978-3-030-61244-3. URL http://link.springer.com/10.1007/978-3-030-61244-3_8. Series Title: Lecture Notes in Computer Science.

Samrath Sinha, Sayna Ebrahimi et Trevor Darrell : Variational Adversarial Active Learning. *In 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5971–5980, Seoul, Korea (South), octobre 2019. IEEE. ISBN 978-1-72814-803-8. URL https://ieeexplore.ieee.org/document/9009538/.

Muhammed Sit, Bekir Demiray, Zhongrun Xiang, Gregory Ewing, Yusuf Sermet et Ibrahim Demir : A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology*, 82, août 2020.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng et Christopher Potts : Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, octobre 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan et Surya Ganguli : Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *In Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265. PMLR, juin 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html. ISSN: 1938-7228.

Mingyang Song, Haiyun Jiang, Shuming Shi, Songfang Yao, Shilong Lu, Yi Feng, Huafeng Liu et Liping Jing : Is ChatGPT A Good Keyphrase Generator? A Preliminary Study, mars 2023. URL http://arxiv.org/abs/2303.13001. arXiv:2303.13001 [cs].

Emma Strubell, Ananya Ganesh et Andrew McCallum : Energy and Policy Considerations for Deep Learning in NLP. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1355.

Jong-Chyi Su, Zezhou Cheng et Subhransu Maji : A Realistic Evaluation of Semi-Supervised Learning for Fine-Grained Classification. *In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12961–12970, juin 2021. ISSN: 2575-7075.

Avinash Swaminathan, Raj Kuwar Gupta, Haimin Zhang, Debanjan Mahata, Rakesh Gosangi et Rajiv Ratn Shah : Keyphrase Generation for Scientific Articles Using GANs (Student Abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(10):13931–13932, avril 2020a. ISSN 2374-3468. URL https://ojs.aaai.org/index.php/AAAI/article/view/7238. Number: 10.

Avinash Swaminathan, Haimin Zhang, Debanjan Mahata, Rakesh Gosangi, Rajiv Ratn Shah et Amanda Stent : A Preliminary Exploration of GANs for Keyphrase Generation. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8021–8030, Online, 2020b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-main.645.

Alex Tamkin, Miles Brundage, Jack Clark et Deep Ganguli : Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models, février 2021. URL http://arxiv.org/abs/2102.02503. arXiv:2102.02503 [cs].

Fabio Henrique Kiyoiti dos Santos Tanaka et Claus Aranha : Data Augmentation Using GANs. *arXiv:1904.09135 [cs, stat]*, avril 2019. URL http://arxiv.org/abs/1904.09135. arXiv:1904.09135.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu et Angela Fan : Multilingual Translation with Extensible Multilingual Pretraining and Finetuning, août 2020. URL http://arxiv.org/abs/2008.00401. arXiv:2008.00401 [cs].

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez et Robert Stojnic : Galactica: A Large Language Model for Science, novembre 2022. URL http://arxiv.org/abs/2211.09085. arXiv:2211.09085 [cs, stat].

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve et Hervé Jégou : Going Deeper With Image Transformers. pages 32–42, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Touvron_Going_Deeper_With_Image_Transformers_ICCV_2

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave et Guillaume Lample : LLaMA: Open and Efficient Foundation Language Models, février 2023a. URL http://arxiv.org/abs/2302.13971.

arXiv:2302.13971 [cs].

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov et Thomas Scialom : Llama 2: Open Foundation and Fine-Tuned Chat Models, juillet 2023b. URL http://arxiv.org/abs/2307.09288. arXiv:2307.09288 [cs].

Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer et Ian Reid : A Bayesian Data Augmentation Approach for Learning Deep Models. page 10.

Lorenzo Tronchin, Minh H. Vu, Paolo Soda et Tommy Löfstedt : LatentAugment: Data Augmentation via Guided Manipulation of GAN's Latent Space, juillet 2023. URL http://arxiv.org/abs/2307.11375. arXiv:2307.11375 [cs, eess].

Christoph Turban et Udo Kruschwitz : Tackling irony detection using ensemble classifiers. *In Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6976–6984, 2022.

Solomon Ubani, Suleyman Olcay Polat et Rodney Nielsen : ZeroShotDataAug: Generating and Augmenting Training Data with ChatGPT, avril 2023. URL http://arxiv.org/abs/2304.14334. arXiv:2304.14334 [cs].

Cynthia Van Hee, Els Lefever et Veronique Hoste : SemEval-2018 Task 3: Irony Detection in English Tweets. *In Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana, 2018. Association for Computational Linguistics. URL http://aclweb.org/anthology/S18-1005.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser et Illia Polosukhin : Attention Is All You Need. *arXiv:1706.03762 [cs]*, décembre 2017. URL http://arxiv.org/abs/1706.03762. arXiv: 1706.03762.

Pawan Kumar Verma, Prateek Agrawal, Vishu Madaan et Radu Prodan : MCred: multi-modal message credibility for fake news detection using BERT and CNN. *Journal of Ambient Intelligence and Humanized Computing*, 14(8):10617–10629, août 2023. ISSN 1868-5145. URL https://doi.org/10.1007/s12652-022-04338-2.

Amir Pouran Ben Veyseh, Nicole Meister, Franck Dernoncourt et Thien Huu Nguyen : Improving Keyphrase Extraction with Data Augmentation and Information Filtering, septembre 2022. URL http://arxiv.org/abs/2209.04951. arXiv:2209.04951 [cs].

Dang Thanh Vu, Gwanghyun Yu, Chilwoo Lee et Jinyoung Kim : Text Data Augmentation for the Korean Language. *Applied Sciences*, 12(7):3425, mars 2022. ISSN 2076-3417. URL https://www.mdpi.com/2076-3417/12/7/3425.

Xiaojun Wan et Jianguo Xiao : CollabRank: towards a collaborative approach to single-document keyphrase extraction. *In Proceedings of the 22nd International Conference on Computational Linguistics - COLING '08*, volume 1, pages 969–976, Manchester, United Kingdom, 2008a. Association for Computational Linguistics. ISBN 978-1-905593-44-6. URL http://portal.acm.org/citation.cfm?doid=1599081.1599203.

Xiaojun Wan et Jianguo Xiao : Single document keyphrase extraction using neighborhood knowledge. *In Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, AAAI'08, pages 855–860, Chicago, Illinois, juillet 2008b. AAAI Press. ISBN 978-1-57735-368-3.

Zhaohong Wan, Xiaojun Wan et Wenguang Wang : Improving Grammatical Error Correction with Data Augmentation by Editing Latent Representation. *In Proceedings of the 28th International Conference on Computational Linguistics*, pages 2202–2212, Barcelona, Spain (Online), décembre 2020. International Committee on Computational Linguistics. URL https://aclanthology.org/2020.coling-main.200.

Ben Wang et Aran Komatsuzaki : GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model, mai 2021. URL https://github.com/kingoflolz/mesh-transformer-jax.

Crystal Wang, Yaateh Richardson et Ryan Sander : Unsupervised Image Clustering and Topic Modeling for Accelerated Annotation. 2019. URL http://rgdoi.net/10.13140/RG.2.2.27176.52484. Publisher: Unpublished.

Huan Wang, Suhas Lohit, Michael N Jones et Yun Fu : What makes a" good" data augmentation in knowledge distillation-a statistical perspective. *Advances in Neural Information Processing Systems*, 35:13456–13469, 2022.

Liantao Wang, Xuelei Hu, Bo Yuan et Jianfeng Lu : Active learning via query synthesis and nearest neighbour search. *Neurocomputing*, 147:426–434, janvier 2015. ISSN 0925-2312. URL http://www.sciencedirect.com/science/article/pii/S0925231214008145.

Qian Wang, Fanlin Meng et T. Breckon : Data Augmentation with norm-VAE for Unsupervised Domain Adaptation. *ArXiv*, 2020.

Jason Wei et Kai Zou : EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China, novembre 2019. Association for Computational Linguistics. URL https://aclanthology.org/D19-1670.

Di Wu, Wasi Uddin Ahmad et Kai-Wei Chang : Pre-trained Language Models for Keyphrase Generation: A Thorough Empirical Study, décembre 2022a. URL http://arxiv.org/abs/2212.10233. arXiv:2212.10233 [cs].

Di Wu, Wasi Uddin Ahmad, Sunipa Dev et Kai-Wei Chang : Representation Learning for Resource-Constrained Keyphrase Generation, octobre 2022b. URL http://arxiv.org/abs/

2203.08118. arXiv:2203.08118 [cs].

Di Wu, Da Yin et Kai-Wei Chang : KPEval: Towards Fine-grained Semantic-based Evaluation of Keyphrase Extraction and Generation Systems, mars 2023. URL http://arxiv.org/abs/2303.15422. arXiv:2303.15422 [cs].

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han et Songlin Hu : Conditional BERT Contextual Augmentation. *In* João M. F. Rodrigues, Pedro J. S. Cardoso, Jânio Monteiro, Roberto Lam, Valeria V. Krzhizhanovskaya, Michael H. Lees, Jack J. Dongarra et Peter M.A. Sloot, éditeurs : *Computational Science – ICCS 2019*, Lecture Notes in Computer Science, pages 84–95, Cham, 2019. Springer International Publishing. ISBN 978-3-030-22747-0.

Binbin Xie, Jia Song, Liangying Shao, Suhang Wu, Xiangpeng Wei, Baosong Yang, Huan Lin, Jun Xie et Jinsong Su : From statistical methods to deep learning, automatic keyphrase prediction: A survey. *Information Processing & Management*, 60(4):103382, 2023. Publisher: Elsevier.

Peng Xu, Jackie Chi Kit Cheung et Yanshuai Cao : On Variational Learning of Controllable Representations for Text without Supervision. *arXiv:1905.11975 [cs]*, février 2020. URL http://arxiv.org/abs/1905.11975. arXiv: 1905.11975.

Weidi Xu et Ying Tan : Semisupervised Text Classification by Variational Autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, 31(1):295–308, janvier 2020. ISSN 2162-2388. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin et Xia Hu : Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond, avril 2023. URL http://arxiv.org/abs/2304.13712. arXiv:2304.13712 [cs].

Suorong Yang, Weikang Xiao, Mengcheng Zhang, Suhan Guo, Jian Zhao et Furao Shen : Image Data Augmentation for Deep Learning: A Survey, avril 2022. URL http://arxiv.org/abs/2204.08610. arXiv:2204.08610 [cs].

Xiangli Yang, Zixing Song, Irwin King et Zenglin Xu : A Survey on Deep Semi-supervised Learning. *arXiv:2103.00550 [cs]*, février 2021. URL http://arxiv.org/abs/2103.00550. arXiv: 2103.00550.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi et Doug Downey : Generative Data Augmentation for Commonsense Reasoning. *arXiv:2004.11546 [cs]*, novembre 2020. URL http://arxiv.org/abs/2004.11546. arXiv: 2004.11546.

Zonghai Yao et Hong Yu : Improving Formality Style Transfer with Context-Aware Rule Injection, mai 2021. URL http://arxiv.org/abs/2106.00210. arXiv:2106.00210 [cs].

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu et Qi Zhang : One2Set: Generating Diverse Keyphrases as a Set. *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)*, pages 4598–4608, Online, août 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.acl-long.354.

Akhila Yerukola, Mason Bretan et Hongxia Jin : Data Augmentation for Voice-Assistant NLU using BERT-based Interchangeable Rephrase. *arXiv:2104.08268 [cs]*, avril 2021. URL http://arxiv.org/abs/2104.08268. arXiv: 2104.08268.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee et Woomyoung Park : GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation. *In Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239, Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.findings-emnlp.192.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi et Quoc V. Le : QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *arXiv:1804.09541 [cs]*, avril 2018. URL http://arxiv.org/abs/1804.09541. arXiv: 1804.09541.

Lantao Yu, Weinan Zhang, Jun Wang et Yong Yu : SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *arXiv:1609.05473 [cs]*, août 2017. URL http://arxiv.org/abs/1609.05473. arXiv: 1609.05473.

Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He et Adam Trischler : One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7961–7975, Online, 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.710.

Jonathan Zarecki et Shaul Markovitch : Textual Membership Queries. volume 3, pages 2662–2668, juillet 2020. URL https://www.ijcai.org/proceedings/2020/369. ISSN: 1045-0823.

Huan Zhang, Yibin Yao, Chaoqian Xu, Wei Xu et Junbo Shi : Transformer-Based Global Zenith Tropospheric Delay Forecasting Model. *Remote Sensing*, 14(14):3335, janvier 2022a. ISSN 2072-4292. URL https://www.mdpi.com/2072-4292/14/14/3335. Number: 14 Publisher: Multidisciplinary Digital Publishing Institute.

Yipeng Zhang, Quan Wang et Bingliang Hu : MinimalGAN: diverse medical image synthesis for data augmentation using minimal training data. *Applied Intelligence*, juin 2022b. ISSN 1573-7497. URL https://doi.org/10.1007/s10489-022-03609-x.

Kun Zhao, Hongwei Ding, Kai Ye et Xiaohui Cui : A Transformer-Based Hierarchical Variational AutoEncoder Combined Hidden Markov Model for Long Text Generation. *Entropy*, 23(10):1277, septembre 2021. ISSN 1099-4300. URL https://www.mdpi.com/1099-4300/23/10/1277.

Minyi Zhao, Lu Zhang, Yi Xu, Jiandong Ding, Jihong Guan et Shuigeng Zhou : EPiDA: An Easy Plug-in Data Augmentation Framework for High Performance Text Classification. *In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4742–4752, Seattle, United States, juillet 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.naacl-main.349.

Jia-Jie Zhu et José Bento : Generative Adversarial Active Learning. *ArXiv*, 2017.

Qile Zhu, Jianlin Su, Wei Bi, Xiaojiang Liu, Xiyao Ma, Xiaolin Li et Dapeng Wu : A Batch Normalized Inference Network Keeps the KL Vanishing Away, mai 2020. URL http://arxiv.org/abs/2004.12585. arXiv:2004.12585 [cs].

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba et Sanja Fidler : Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. *In 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, Santiago, Chile, décembre 2015. IEEE. ISBN 978-1-4673-8391-2. URL http://ieeexplore.ieee.org/document/7410368/.

Peiye Zhuang, Alexander G. Schwing et Oluwasanmi Koyejo : FMRI Data Augmentation Via Synthesis. *In 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1783–1787, avril 2019. ISSN: 1945-8452.

Erion Çano et Ondřej Bojar : Keyphrase Generation: A Text Summarization Struggle. *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)*, pages 666–672, Minneapolis, Minnesota, juin 2019. Association for Computational Linguistics. URL https://aclanthology.org/N19-1070.

Erion Çano et Ondřej Bojar : Two Huge Title and Keyword Generation Corpora of Research Articles, février 2020. URL http://arxiv.org/abs/2002.04689. arXiv:2002.04689 [cs].

Messages de sollicitations pour les expériences du chapitre 3

Nous utilisons trois types de messages de sollicitations pour interroger ChatGPT, deux pour la technique de paraphrase et une pour la technique de description, afin d'obtenir les meilleurs résultats possibles tout en minimisant le nombre de requêtes. Pour la paraphrase, si le ratio est supérieur à un, alors la requête est du type :"Create X paraphrases of the following sentence : ".

Si le ratio est de un, alors nous traitons les exemples par lots, avec la requête : " Create a paraphrase for each of the following sentences: 1. [...], 2. [...]".

Enfin, pour la stratégie de description, le modèle que nous utilisons est : "Generate 10 new sentences that you haven't generated before for a dataset of DATASET_DESCRIPTION which would be CLASS_DESCRIPTION". Nous avons constaté que préciser "new sentences that you haven't generated before" aide ChatGPT à créer des phrases plus variées. Les descriptions de jeux de données données sont"movie review", "headline Fake/Real news classification", "Ironic tweet detection", et "Question Classification".

Les valeurs de classe sont "negative or somewhat negative" ou "positive or somewhat positive" pour SST-2, "Real" et "Fake" pour FakeNews, "Non Ironic Tweets" et "Ironic Tweets" pour Irony, "Tweets ironic by polarity contrast, where the polarity is inverted between the literal and intended evaluation", "Tweets ironic by Situational Irony, where a situation fails to meet some expectation", "Tweets

ironic by Other type of Irony, where the Irony is neither by Polarity Contrast or by Situational Irony", et "Tweets that are not ironic" pour IronyB, et enfin pour TREC6 nous utilisons "Question about an abbreviation", "Question about an entity (event, animal, language, etc)", "Question concerning a description (of something, a definition, a reason, etc)", "Question about a human (description of someone, an individual, etc)", "Question about a location", et "Question about something numerical (weight, price, any other number)".

Nous nous sommes référés aux descriptions fournies dans les articles originaux de chaque jeu de données pour élaborer des invitations informatives.