STATAPP 49 : PETITS MODÈLES DE LANGUE

M. AMAZO, R. DELHOMMAIS, V. GIMENES, G. LÉVY

Résumé. Ce rapport présente une étude sur l'évaluation des performances des modèles de langue dans des environnements de ressources limitées, en se concentrant sur l'utilisation du modèle GPT-2 sur des données d'entraînement provenant de pages Wikipédia en anglais. Notre étude explore l'impact de divers hyperparamètres des modèles ainsi que de la diversité du corpus sur les performances des modèles de langue.

Table des matières

1. Introduction

1.1. **Contexte et motivation.**

Dans le domaine en constante évolution du traitement automatique du langage naturel (NLP), l'efficacité des modèles de langue repose souvent sur deux facteurs clés : la taille des données d'entraînement et le nombre de paramètres du modèle. Alors que les avancées récentes ont permis de développer des modèles de langue de plus en plus puissants, ceux-ci nécessitent souvent des ensembles de données massifs et des architectures complexes, ce qui peut poser des défis en termes de stockage, de traitement et de calcul.

1.2. **Objectifs du projet.**

Notre projet de recherche se concentre sur l'évaluation des performances des modèles de langue avec des contraintes strictes en termes de taille des données d'entraînement et de nombre de paramètres. Plus précisément, nous nous intéressons à la façon dont ces modèles se comportent lorsqu'ils sont entraînés sur des sous-ensembles de pages Wikipédia en anglais, représentant des volumes de données allant de 1 à 15 Go. Notre objectif est de déterminer dans quelle mesure ces modèles parviennent à généraliser et à produire des résultats de qualité malgré des ressources limitées.

1.3. **Plan du document.**

Ce compte-rendu technique vise à présenter en détail notre méthodologie expérimentale, les résultats obtenus et les analyses qui en découlent. Nous commencerons par établir les bases théoriques du NLP et des modèles de langue, en mettant en lumière l'importance de la taille des données d'entraînement dans le processus d'apprentissage. Ensuite, nous décrirons en détail notre approche méthodologique, y compris la collecte et le prétraitement des données, ainsi que le choix des modèles de langue à évaluer. Les sections suivantes seront consacrées à la présentation de nos expérimentations, où nous détaillerons les différentes configurations testées, les métriques d'évaluation utilisées et les résultats obtenus. Nous analyserons ensuite ces résultats, en mettant en évidence les tendances observées, les performances relatives des différents modèles et les implications de nos découvertes.
En conclusion, nous synthétiserons les principaux enseignements tirés de cette étude, en discutant des limitations rencontrées et des pistes de recherche futures. Notre travail vise à apporter des éclaircissements sur l'efficacité des modèles de langue dans des contextes de ressources limitées, ouvrant ainsi la voie à de nouvelles approches pour le développement de systèmes de NLP plus efficaces et plus accessibles.

2. Fondements théoriques

Dans cette section, nous posons les bases théoriques nécessaires à la compréhension de notre étude sur l'évaluation des modèles de langue dans des conditions de ressources limitées. Nous abordons les principaux concepts du traitement automatique du langage naturel (NLP) ainsi que les aspects fondamentaux des modèles de langue.

2.1. **Bases du traitement du langage naturel (NLP).**

Le traitement automatique du langage naturel (en anglais, le NLP pour *Natural language processing*) est une branche de l'intelligence artificielle qui vise à permettre aux ordinateurs de comprendre, interpréter et générer un langage humain de manière naturelle. Les tâches courantes du NLP incluent la classification de texte, l'extraction d'informations, la traduction automatique et la génération de texte.

2.2. **Littérature antérieure.**

La recherche sur les modèles de langue de petite échelle a connu un essor significatif ces dernières années, en grande partie grâce aux avancées dans le domaine de l'apprentissage automatique et du traitement du langage naturel. Cependant, malgré ces progrès, il existe encore des lacunes importantes dans notre compréhension et notre capacité à développer efficacement de tels modèles. Cette revue de littérature vise à explorer les travaux existants dans ce domaine et à identifier les tendances, les défis et les opportunités de recherche futures.

Dans un premier temps, il est important de noter que la plupart des travaux de recherche sur les modèles de langue se concentrent sur des architectures de grande échelle, telles que BERT, GPT et Transformer XL, qui sont conçues pour traiter de vastes quantités de données textuelles et nécessitent des ressources informatiques considérables pour l'entraînement et le déploiement. Cependant, il existe un intérêt croissant pour le développement de modèles de langue plus petits, adaptés à des environnements avec des contraintes de mémoire et de calcul, tels que les appareils mobiles ou les systèmes embarqués.

Dans cette optique, plusieurs approches ont été explorées. Certains chercheurs se sont penchés sur des techniques de compression de modèles, telles que la quantification des poids, la pruning [1] et la distillation de connaissances [2], dans le but de réduire la taille des modèles tout en préservant leurs performances. D'autres ont exploré des architectures spécifiquement conçues pour les appareils à faible puissance, telles que les réseaux de neurones convolutifs légers (LCNN) qui peuvent être plus efficaces en termes de ressources. [3]

Malgré ces avancées, il reste encore des défis importants à relever. Par exemple, la réduction de la taille des modèles peut entraîner une perte de performances, en particulier pour les tâches complexes nécessitant une compréhension profonde du langage. De plus, la plupart des recherches se sont concentrées sur des langues à grande échelle, telles que l'anglais, laissant un besoin important de travaux dans d'autres langues, en particulier les langues moins répandues.

2.3. **Modèles de langue et leurs paramètres.**

Les modèles de langue constituent une composante essentielle du NLP. Ils sont conçus pour estimer la probabilité d'apparition d'une séquence de mots dans une langue donnée. Les modèles de langue peuvent être basés sur des règles, des statistiques ou des réseaux de neurones. Les paramètres d'un modèle de langue déterminent sa complexité et sa capacité à capturer les relations entre les mots.

2.4. **Importance de la taille des données d'entraînement.**

La taille des données d'entraînement est un facteur déterminant dans la performance des modèles de langue. Un ensemble de données d'entraînement plus vaste permet généralement d'obtenir des modèles de langue plus précis et plus généralisables. Cependant, l'obtention et le stockage de grandes quantités de données peuvent poser des défis logistiques et computationnels, en particulier dans des environnements avec des ressources limitées.
En comprenant ces concepts fondamentaux du NLP et des modèles de langue, nous sommes mieux équipés pour aborder notre méthodologie expérimentale et interpréter les résultats obtenus dans la section suivante.

2.5. **Glossaire.**

Ci-desosus, nous définissons les principaux termes et concepts que nous utiliserons librement dans la suite de ce rapport.

- **Attention, tête d'attention :** Mécanisme clé dans les réseaux neuronaux, notamment dans les Transformers, permettant de pondérer l'importance des différentes parties de l'entrée en fonction de leur pertinence pour une tâche donnée. Une tête d'attention est une composante d'un mécanisme d'attention, qui peut être utilisée pour capturer des relations à longue distance entre les éléments de la séquence.
- **Contexte :** Le contexte est l'ensemble des informations textuelles ou linguistiques qui entourent un mot, une phrase ou un document donné. Il s'agit des éléments qui fournissent des indices ou des informations supplémentaires pour comprendre le sens ou l'intention d'une unité de texte spécifique. La longueur de contexte d'un modèle est la longueur (en tokens) de la plage prise en compte pour comprendre ou générer une unité de texte.
- **Couche :** Une couche fait référence à une unité structurelle dans un réseau neuronal, composée de plusieurs neurones et responsable de la transformation des données d'entrée en des représentations plus abstraites et significatives. Les couches peuvent être de différents types, telles que des couches denses, des couches de convolution ou des couches récurrentes, en fonction de la tâche et de la structure du réseau.
- **Embedding (*plongement sémantique*) :** Dans le contexte du NLP, un embedding, ou plongement sémantique, fait référence à une représentation vectorielle dense de mots ou de tokens, souvent apprise à partir de grands corpus de texte. Ces embeddings permettent de capturer les relations sémantiques et syntaxiques entre les mots, et sont largement utilisés dans les modèles de langue et les tâches de NLP pour représenter le sens des mots de manière plus informatique.
- **GPT (Generative Pre-trained Transformer) :** GPT est une famille de modèles de langage développée par OpenAI, basée sur l'architecture des Transformers. Ces modèles sont pré-entraînés sur de vastes corpus de texte afin d'apprendre des représentations linguistiques générales, puis peuvent être fine-tunés sur des tâches spécifiques.
- **Loss (perte) :** Dans un modèle de machine learning, la perte est une mesure qui évalue à quel point les prédictions du modèle correspondent aux vraies valeurs des données d'entraînement. La perte est calculée en comparant les prédictions du modèle aux étiquettes (labels) réelles des données d'entraînement à l'aide d'une fonction de perte spécifique, choisie à l'avance et comportant sa part d'arbitraire.

- **Overfitting (*surapprentissage*) :** Phénomène où un modèle d'apprentissage développe une capacité excessive à s'adapter aux données d'entraînement spécifiques utilisées pour son apprentissage, au détriment de sa capacité à généraliser à de nouvelles données non vues. L'overfitting se produit souvent lorsque le modèle est trop complexe par rapport à la quantité ou à la qualité des données d'entraînement, ce qui entraîne une mémorisation des exemples spécifiques plutôt qu'une compréhension des patterns généraux dans les données.
- **Prompt :** Dans le contexte du NLP, un prompt est un texte d'entrée donné à un modèle de langue pour générer une réponse ou une prédiction. Les prompts sont souvent utilisés pour guider le modèle vers une tâche spécifique ou pour influencer le contenu de la réponse générée.
- **Token :** Un token est une unité de base dans le traitement du langage naturel, correspondant généralement à un mot ou à un symbole individuel dans un texte. Les tokens sont utilisés comme entrées pour les modèles de langue et sont souvent représentés sous forme d'embeddings dans les réseaux neuronaux.
- **Train/test/validation :** Le train, test et validation sont des ensembles de données distincts utilisés dans le processus d'apprentissage automatique pour entraîner, évaluer et valider les performances des modèles. L'ensemble d'entraînement est utilisé pour ajuster les paramètres du modèle, l'ensemble de test est utilisé pour évaluer les performances du modèle sur des données non vues auparavant, et l'ensemble de validation est utilisé pour ajuster les hyperparamètres et prévenir le surapprentissage (*overfitting*).
- **Transformer :** Le Transformer est une architecture de réseau neuronal, révolutionnaire à l'époque de sa sortie et développée par Google. Cette architecture est aujourd'hui largement utilisée dans le traitement du langage naturel. Elle est basée sur le mécanisme d'attention multi-têtes et a été utilisé comme fondement pour de nombreux modèles de pointe dans le domaine du NLP, y compris BERT, GPT et T5.
- **Warmup phase (phase d'échauffement) :** La warmup step fait référence à une période initiale pendant laquelle le taux d'apprentissage est progressivement augmenté jusqu'à sa valeur maximale.

3. Méthodologie

Dans cette section, nous détaillons notre méthodologie expérimentale, en mettant en lumière les étapes spécifiques que nous avons suivies pour collecter et prétraiter nos données d'entraînement, issues des pages Wikipédia en anglais existant au 1er mars 2022, disponibles sur le site de HuggingFace.[1]

3.1. **Collecte et prétraitement des données.**

Nous avons accédé au jeu de données des pages Wikipédia en anglais à partir du site de HuggingFace. Ce jeu de données comprend l'intégralité des pages Wikipédia en langue anglaise de l'époque, ce qui représente environ 6,5 millions de pages distinctes et 20 Go de données brutes. À titre de comparaison, le modèle GPT-2 a été entraîné sur le jeu *Webtext* constitué d'environ 8 millions de pages Web. [4]
La plupart de ces pages sont des articles, mais il y a également les pages d'homonymie. L'ensemble couvre divers sujets et domaines de connaissances. Ces articles sont déjà prétraités par HuggingFace, ce

[1] `https://huggingface.co/datasets/wikipedia`, dataset 20220301.en

qui signifie que les éléments non textuels ont été éliminés et que le texte est prêt à être utilisé pour l'entraînement de modèles de langue.
Un jeu de données sensiblement différent provenant de CNN[2], lui aussi téléchargé sur le site de HuggingFace a servi d'échantillon de test pour le calcul de la perplexité, ceci afin d'évaluer les performances du modèle sur des textes qui s'éloignent suffisamment de ses données d'entraînement.

3.2. **Choix et traitement des données.**

Bien que le prétraitement initial ait été effectué par HuggingFace, nous avons réalisé quelques ajustements supplémentaires pour répondre aux besoins spécifiques de notre étude.

3.2.1. *Échantillonnage aléatoire.*

Nous avons effectué un échantillonnage aléatoire sur le jeu de données complet le découper en sous-paquets de 10 000 pages chacun. La seed utilisée pour l'échantillonnage a été choisie de manière aléatoire, mais elle est connue et enregistrée pour assurer la reproductibilité de nos expériences.
Cette étape nous a permis de faire tourner rapidement les modèles au début de notre projet, afin de vérifier que la pipeline d'entraînement fonctionnait correctement. Une fois cette étape dépassée, nous sommes revenus à des jeux de données plus conséquents.

3.2.2. *Échantillonnage par taille décroissante.*

Partant de l'échantillonnage aléatoire précédent, nous avons reconstitué de nouveaux jeux de données en sélectionnant et regroupant les pages les plus volumineuses (en nombre de caractères) de chaque sous-paquet.
Trois jeux de données en ont résulté, constitué des 10, 100 et 500 plus longues pages sur chaque sous-paquet de 10 000 pages. Ces jeux de données pèsent respectivement environ 2, 10 et 15 Go. Un tel filtrage par taille présente l'avantage d'éliminer les pages d'homonymie, souvent très courtes et présentant un intérêt moindre pour l'entraînement.

3.2.3. *Sélection basée sur la diversité des textes.*

En plus de l'échantillonnage aléatoire, nous avons également réalisé des sélections basées sur une métrique de diversité des textes. Cette métrique nous a permis de choisir des articles qui couvrent un large éventail de sujets et de styles de rédaction, garantissant ainsi une représentation équilibrée dans notre ensemble d'entraînement.

3.2.4. *Filtrage thématique.*

Enfin, en appliquant à l'inverse la méthode utilisée pour construire des ensmebles divers, il est possible de sélectionner des sous-ensembles thématiquement homogènes (par exemple, ceux parlant de la France).

3.2.5. *Filtrage par expression rationnelle.*

Mentionnons ici un point plus technique. Afin d'améliorer les performances de notre modèle, nous avons tenté de réduire drastiquement la diversité du vocabulaire et des symboles rencontrés en effectuant un filtrage par expression rationnelle. Le principe était de ne garder que les articles contenant des caractères dans l'ensemble ASCII strict ainsi que quelques signes jugés indispensables (points, parenthèses, etc).
La combinaison des méthodes de sélection exposées ci-dessus a permis de construire suffisamment de jeux de données différents, tant dans leur construction que leur taille, pour en faire un hyperparamètre d'entraînement en soi.

[2] `https://huggingface.co/datasets/cnn_dailymail`

3.3. **Choix des modèles de langue à évaluer.**

Dans le cadre de notre étude, nous avons restreint notre sélection de modèles de langue à GPT-2 (Generative Pre-trained Transformer 2), en raison de la taille des modèles et des contraintes de ressources. Même la version GPT-2 medium s'est avérée trop volumineuse pour notre jeu de données et les capacités de calcul disponibles.
Ce compromis nous permet de concentrer notre analyse sur les performances d'un modèle de langue spécifique, GPT-2, dans des conditions où les ressources sont restreintes, offrant ainsi des informations précieuses sur son utilité et son efficacité dans de telles situations.

3.4. **Configuration des expériences.**

Dans cette sous-section, nous détaillons la configuration spécifique des expériences que nous avons menées pour évaluer les performances du modèle de langue GPT-2 dans des environnements de ressources limitées.

- **Taille du modèle :** Nous sommes partis de la version de base de GPT-2, avec environ 125 millions de paramètres, en raison de sa compatibilité avec nos ressources disponibles et de sa capacité à fonctionner dans des environnements avec des contraintes de mémoire et de puissance de calcul.
- **Paramètres d'entraînement :** Nous avons adapté les paramètres d'entraînement du modèle en fonction de notre jeu de données et de nos objectifs spécifiques. Cela inclut des hyperparamètres tels que le taux d'apprentissage, la taille du contexte ou le nombre de tête d'attention. Un principe heuristique veut, en principe, que l'on adapte la taille du jeu de données d'entraînement (en nombre de tokens) à la taille du modèle (en millions de poids). Un jeu de données trop imposant est supposé conduire à un surapprentissage, nuisant à la capacité de généralisation. Un jeu de données trop maigre est supposé conduire à un sous-apprentissage. Nous agissons ici en ignorance volontaire de ce principe afin d'explorer le plus possible l'espace des hyperparamètres, taille du jeu de données comprise.
- **Prétraitement des données :** Nous avons effectué un prétraitement minimal des données pour adapter le texte au format d'entrée requis par le modèle GPT-2. Cela inclut la tokenisation du texte en tokens individuels et la conversion en séquences numériques.
- **Matériel et infrastructure :** Nous avons utilisé la puissance de calcul du Datalab. Cette configuration reflète les ressources typiquement disponibles dans des environnements de recherche académique ou industrielle d'envergure modérée.

4. Expérimentations

4.1. **Description des expériences réalisées.**

Dans cette section, nous détaillons nos expériences visant à explorer l'impact des différents hyperparamètres sur les performances de nos modèles de langue. Nous avons systématiquement varié plusieurs paramètres-clés, que nous décrivons ici.

1. **Taille du plongement sémantique (embedding), de 128 à 768 :** L'embedding représente la dimensionnalité de l'espace dans lequel les mots sont projetés. Une taille plus grande permet de capturer des informations plus riches et complexes sur les relations entre les mots, mais nécessite également plus de mémoire et de calcul. Une augmentation de la taille de l'embedding

peut améliorer la capacité du modèle à capturer des nuances subtiles du langage, mais un embedding trop grand (comapративement à la quantité de données disponibles) augmente le risque de sous-apprentissage.

2. **Nombre de couches, de 3 à 16 :** Le nombre de couches fait référence à la profondeur du réseau, c'est-à-dire le nombre de transformations successives appliquées aux données d'entrée. Une augmentation du nombre de couches permet au modèle de capturer des représentations hiérarchiques plus complexes et abstraites, mais augmente également la complexité du modèle et le temps d'entraînement. Un nombre de couches plus élevé peut améliorer les performances du modèle, mais peut également entraîner un surapprentissage si le modèle devient trop complexe pour les données d'entraînement disponibles.
3. **Nombre de têtes d'attention, de 4 à 12 :** Les têtes d'attention permettent au modèle de se concentrer sur différentes parties de l'entrée simultanément. Concrètement, lors de l'étape d'attention, chaque tête d'attention calcule une pondération pour chaque mot dans la séquence en fonction de ses relations avec les autres mots. En ayant plusieurs têtes d'attention, le modèle peut apprendre à mettre l'accent sur différentes parties du texte et à capturer des aspects plus riches et complexes des relations sémantiques et syntaxiques entre les mots. En augmentant le nombre de têtes d'attention, le modèle dispose de plus de capacité pour capturer des motifs complexes dans les données, ce qui peut améliorer les performances du modèle.
4. **Taux d'apprentissage maximal, de 0,0006 à 0,001 :** Le taux d'apprentissage contrôle la vitesse à laquelle les poids du modèle sont mis à jour pendant l'entraînement. Un taux d'apprentissage plus élevé peut accélérer la convergence de l'entraînement, mais peut également entraîner une instabilité et des oscillations dans la descente de gradient. Un taux d'apprentissage plus bas peut permettre une convergence plus stable, mais peut nécessiter un temps d'entraînement plus long pour atteindre de bonnes performances.
5. **Ratio train set/dataset total, de 20% à 98% :** Le ratio entre l'ensemble d'entraînement et l'ensemble de données total contrôle la quantité de données utilisées pour entraîner le modèle. Un ratio plus élevé permet d'utiliser une plus grande partie des données pour l'entraînement, ce qui peut améliorer les performances du modèle en lui fournissant plus d'exemples pour apprendre. Cependant, cela peut également augmenter le risque de surapprentissage si le modèle devient trop spécialisé aux données d'entraînement.
6. **Taille ou type de dataset utilisé (petit, moyen, grand et latin) :** La taille et le type de dataset utilisé peuvent avoir un impact significatif sur les performances du modèle. Un dataset plus grand peut permettre au modèle de capturer une plus grande variété de structures et de phénomènes linguistiques, mais peut également nécessiter plus de temps et de ressources pour l'entraînement. Le filtrage du dataset pour ne retenir que les pages utilisant l'alphabet latin peut aider à limiter le bruit et à améliorer la qualité des données utilisées pour l'entraînement.
7. **Longueur de la warmup step, de 400 à 1000 :** Une warmup step plus longue peut permettre au modèle de s'ajuster progressivement aux données d'entraînement et de stabiliser l'entraînement. Cependant, une warmup step trop longue peut retarder la convergence de l'entraînement et prolonger le temps nécessaire pour atteindre de bonnes performances.

4.2. **Diversité des corpora.**

L'importance de la diversité du corpus textuel utilisé pour l'entraînement des modèles de langage ne peut être sous-estimée. Un corpus présentant une faible diversité peut conduire à un modèle très spécialisé, performant dans le domaine spécifique du corpus mais souffrant d'un fort biais. En revanche, un corpus plus diversifié offre une meilleure capacité de généralisation.
Mentionnons tout d'abord notre méthode de représentation des documents. Nous utilisons les outils de l'analayse sémantique latente (LSA, *Latent semantic analysis*), qui est une méthode couramment utilisée pour la représentation vectorielle de documents. Elle consiste à transformer les documents en matrices de termes et à les décomposer en matrices de termes latents, permettant ainsi de capturer les relations sémantiques entre les mots et les documents.
Afin d'évaluer la diversité de notre corpus, nous nous sommes appuyés sur l'article [5], qui propose un indice mesurant cette diversité comme le "nombre d'éléments différents du corpus".
Cet indice repose sur l'exponentielle de l'entropie de Shannon des valeurs propres de la matrice de cosimilarité, reflétant ainsi la similarité paire à paire des éléments du corpus. Plusieurs approches peuvent être employées pour sélectionner la combinaison optimale de n textes maximisant l'entropie parmi toutes les combinaisons possibles.
Une méthode évidente consiste à calculer la matrice de cosimilarité sur l'ensemble du corpus, à la diagonaliser et à sélectionner les textes correspondant aux valeurs propres maximisant l'indice de Vendi. Cependant, cette approche peut être coûteuse en termes de temps de calcul, nécessitant de trouver les racines d'un polynôme de degré égal à la taille du corpus.
Une alternative consiste à découper le corpus en sous-corpora sur lesquels appliquer la méthode précédente en utilisant des batchs (sous-paquets). Bien que cette approche nécessite plus d'opérations, elle est moins coûteuse en ressources computationnelles puisqu'elle opère sur des corpora de taille réduite.
Une autre approche consiste à tirer au hasard plusieurs fois un corpus de n textes et à sélectionner celui avec le score de Vendi le plus élevé, évitant ainsi de traiter tous les sous-corpora. Cette méthode permet de tirer en quelques secondes un corpus de 500 textes parmi 1000.
Il peut être compliqué de calculer le score de Vendi d'un corpus, particulièrement lorsque sa taille devient conséquente. Une méthode d'approximation qui n'a pas été exploré aurait été d'appliquer les méthodes précédentes sur des regroupements de textes plutôt qu'à des textes individuels. La dimension de notre matrice de cosimilarité s'en trouverait notablement réduite, tout comme nos temps de calculs.
Le recours à des techniques de regroupement a en retour un impact sur la pertinence d'autres méthodes, parmi lesquelles celles nécessitant de tirer des textes aléatoirement. Leur coût en temps de calcul devient prohibitif lorsque la taille des corpora que nous souhaitions sélectionner devenait trop élevée.
Cependant, ces regroupements induisent une diversité des corpora plus faibles et nous contraignent alors à inclure des textes moins riches et moins efficaces pour l'entraînement.
Il est crucial d'examiner l'impact de la diversité du corpus sur les performances du modèle, afin de comprendre comment la composition du corpus influence les résultats produits.

4.3. **Métriques d'évaluation utilisées.**

En parallèle de l'entraînement proprement dit, nous avons défini deux listes préalables de *prompts* que le modèle doit compléter avec ce qu'il estime être la suite de tokens la plus probable. Une première liste, que nous avons appelée "de prompts simples", contient uniquement des prompts courts, sur des sujets communs. Une deuxième liste, dite "de prompts complexes", contient des prompts plus longs, appelant une réponse plus développée.

Une fois chaque modèle entraîné, nous examinons l'évolution de sa fonction de perte au cours de l'entraînement, sa perplexité face aux extraits de CNN et la qualité de ses réponses à nos prompts.

4.4. **Résultats obtenus.**

4.4.1. *Caractéristiques de nos modèles.*

La plupart des modèles que nous avons entraînés sont décrits par le tableau ci-dessous.

Plongement	**Couches**	**Têtes**	**Taux appr.**	**Ratio entr.**	**Données**	**Échauff.**
768	3	8	0.001	0.2	top100	400
768	12	12	0.0006	0.98	top100	1000
512	6	8	0.001	0.98	top100	1000
512	6	8	0.001	0.5	top100	1000
384	8	8	0.001	0.5	top100	1000
384	6	6	0.002	0,9	Vendi	1000
384	6	6	0.002	0,9	Random	1000
384	6	6	0.001	0.98	top500	4000
384	6	6	0.001	0.98	top100	1000
384	6	6	0.001	0.95	top10	1000
384	6	6	0.001	0.5	top100	1000
384	6	6	0.001	0.5	latin	400
384	6	6	0.001	0.2	top100	400
384	6	6	0.001	0.25	latin	400
256	8	8	0.002	0.5	top100	1000
256	6	8	0.002	0.98	top100	1000
128	4	4	0.003	0.98	top100	1000
128	4	4	0.003	0.5	top100	1000
128	16	8	0.001	0.2	top100	400

TABLE 1. Tableau des hyperparamètres testés avec une longueur de contexte de 256.

Le dataset "Vendi" a été obtenu à partir des techniques de maximisation de la diversité au sein du corpus. Le dataset "Random" résulte d'une sélection purement aléatoire d'un grand ensemble de pages au sein du jeu de données d'origine.

Il faut ajouter à cette liste un dernier modèle un peu à part, puisqu'il s'agit du modèle GPT2-small, publiquement téléchargeable, que nous avons réentraîné sur le plus imposant de nos jeux de données. notre jeu de données intermédiaire, celui nommé "top 100".

4.4.2. *Résultats quantitatifs.*

Commençons par deux graphiques qui comparent chacun un indicateur de taille de nos modèles : le premier en gigaoctets (Go), l'autre en millions de paramètres.

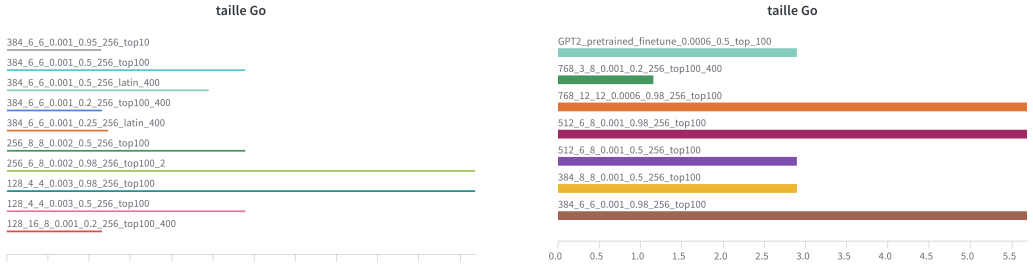

FIGURE 1. Taille (en Go).

L'échelle varie ici de 1,1 Go pour les plus petits à 5,7 Go pour les plus imposants.

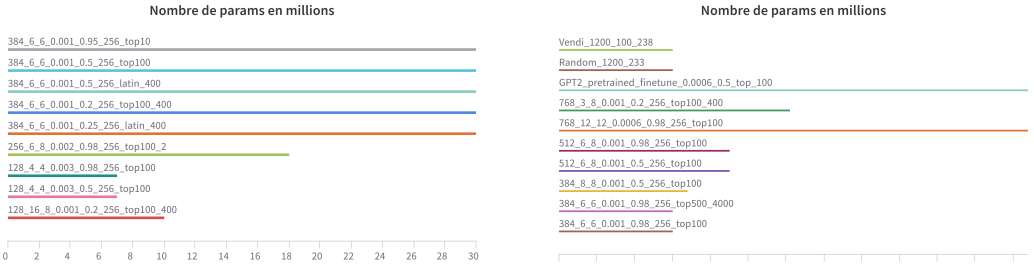

FIGURE 2. Nombre de paramètres (en millions).

Les deux plus gros modèles utilisent 124 millions de paramètres (la taille native de GPT2-small), les plus petits 7,4 millions. Les modèles de taille intermédiaire en utilisent environ 30 millions.

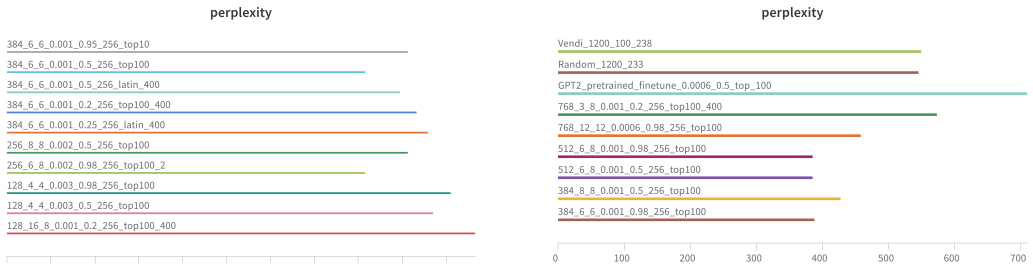

FIGURE 3. Perplexité (sans unité).

Concernant la perplexité, la plupart de nos modèles termine avec une valeur entre 400 et 500. Une exception notable : GPT2-small réentraîné a une performance bien *moindre* que les autres, avec une perplexité d'environ 700.

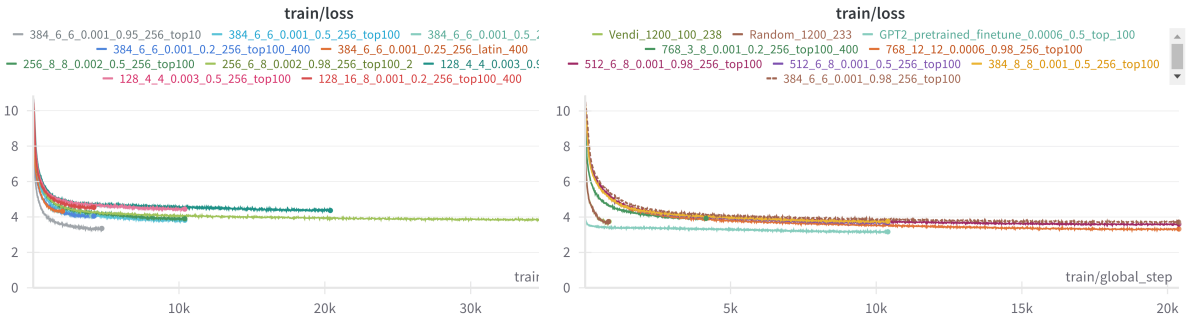

FIGURE 4. Perte (sans unité) sur le jeu de données d'entraînement.

Ce graphique représente la variation de la perte du modèle de langue sur l'ensemble de données d'entraînement. Il permet de suivre la convergence de l'entraînement et peut indiquer si le modèle apprend efficacement à partir des données fournies.

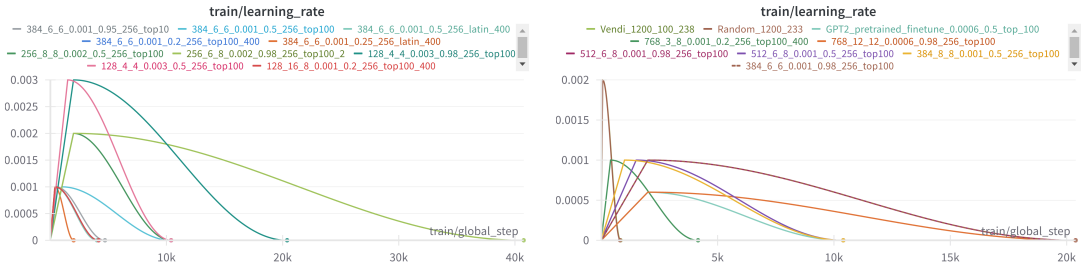

FIGURE 5. Évolution du taux d'apprentissage (sans unité).

Ce graphique illustre l'évolution du taux d'apprentissage utilisé. Cette évolution peut influencer la stabilité et la vitesse de convergence de l'entraînement.
Ces premières données brutes étant exposées, croisons-les afin d'en déduire de nouvelles informations.

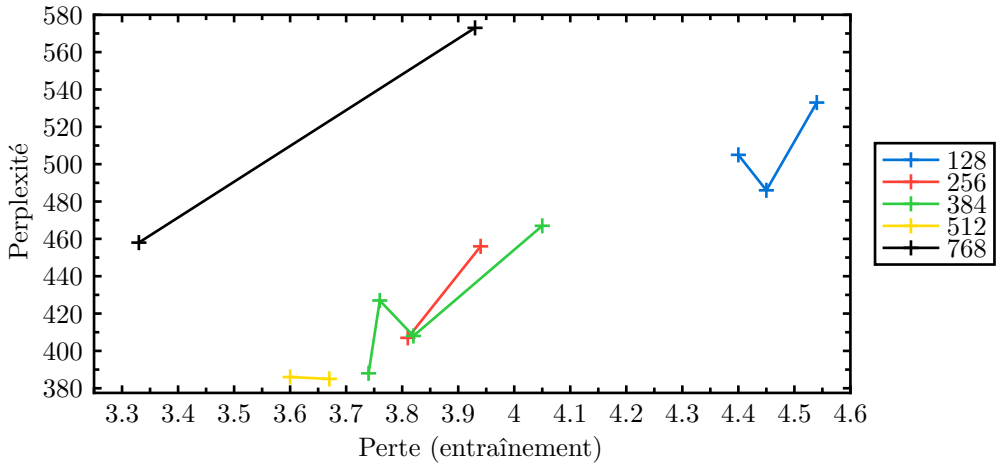

FIGURE 6. Perplexité et perte d'entraînement.

Commençons par situer nos modèles dans un plan (perte, perplexité), afin de déterminer si de bonnes performances durant l'entraînement se traduisent par de bonnes performances sur un nouveau jeu de données. On teste ainsi un aspect de la capacité du modèle à généraliser ce qu'il a appris.
Idéalement, on souhaiterait avoir des modèles qui atteignent à la fois une faible perplexité et une faible perte, ce qui indique à la fois une capacité à générer des prédictions précises et une réduction significative de l'erreur lors de l'apprentissage.
Cependant, il est important de noter que dans certains cas, une faible perte finale peut être accompagnée d'une perplexité plus élevée, situation typique de surapprentissage.
Les résultats ne sont pas notablement modifiés si l'on considère la perte d'évaluation au lieu de la perte d'entraînement.

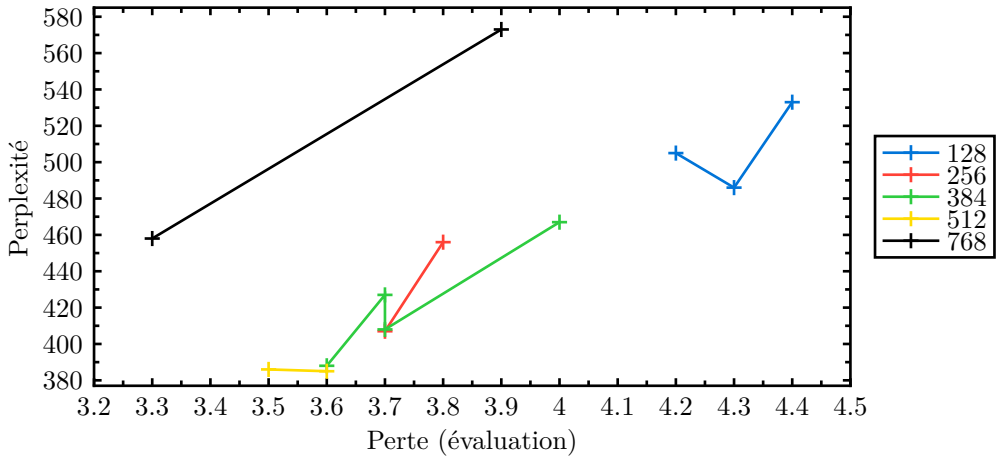

FIGURE 7. Perplexité et perte d'évaluation.

Le graphique suivant représente l'influnce du ratio d'entraînement sur la perplexité. La dernière paire de points (embedding 768) n'a volontairement pas été reliée an nom de la lisibilité. Deux remarques peuvent être faites à partir de ce tracé.
D'une part et de manière générale, un plus grand ratio d'entraînement fait voir au modèle une plus grande quantité de données, ce qui tend à faire augmenter sa performance dans un premier temps avant de la dégrader par sur-apprentissage. La convexité des courbes, particulièrement les trois premières (si l'on omet le point double pour l'embedding 384), peut être un reflet de ce phénomène.

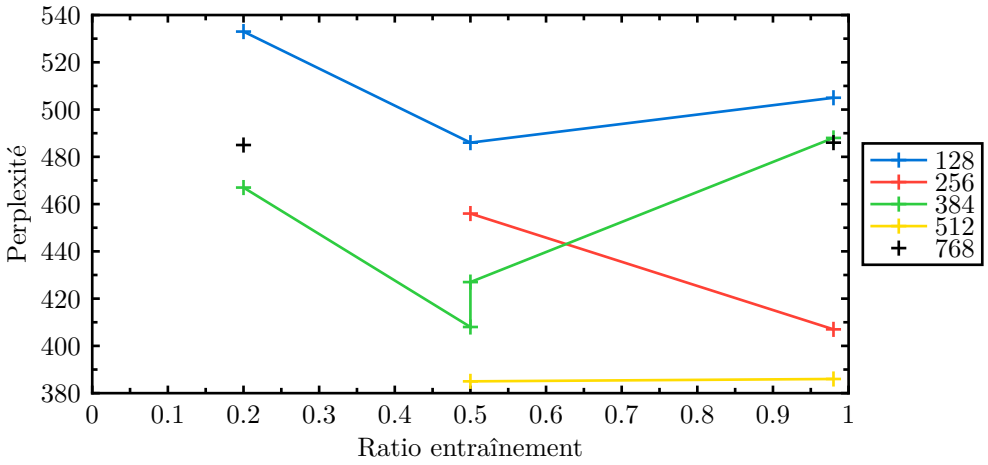

FIGURE 8. Perplexité en fonction du ratio d'entraînement, à dataset fixé (top 100), par groupes d'embedding.

D'autre part, l'aplatissement constaté pour les grands embeddings (courbe jaune et points noirs) laisse penser à une saturation de la capacité d'apprentissage du modèle

Enfin, nous souhaitons introduire une dimension énergétique à notre analyse. Il est notoire que l'entraînement des modèles de langue est un processus coûteux en temps, en matériel et en énergie électrique, compte tenu du grand nombre de cartes graphiques spécialisées à alimenter.

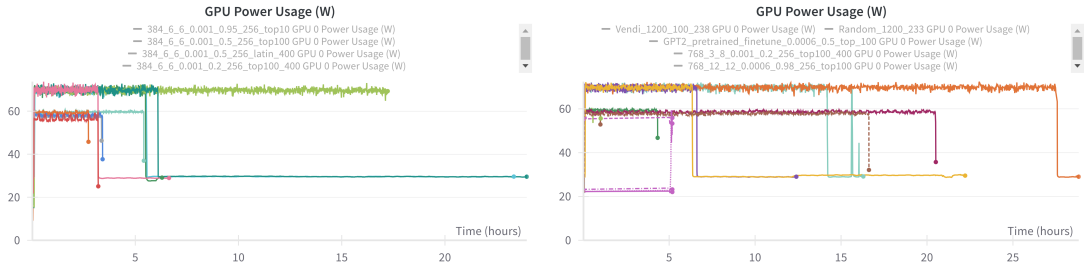

FIGURE 9. Consommation électrique (W).

Le graphique ci-dessus représente la consommation énergétique du processus d'entraînement du modèle de langue. Il permet d'évaluer l'efficacité énergétique du processus d'entraînement. On remarque que la plupart de nos modèles saturent les ressources allouées par le Datalab.
Nous choisissons d'évaluer l'efficacité énergétique de notre entraînement en comparant, pour nos modèles, la perplexité finale atteinte et l'énergie totale consommée par le GPU.

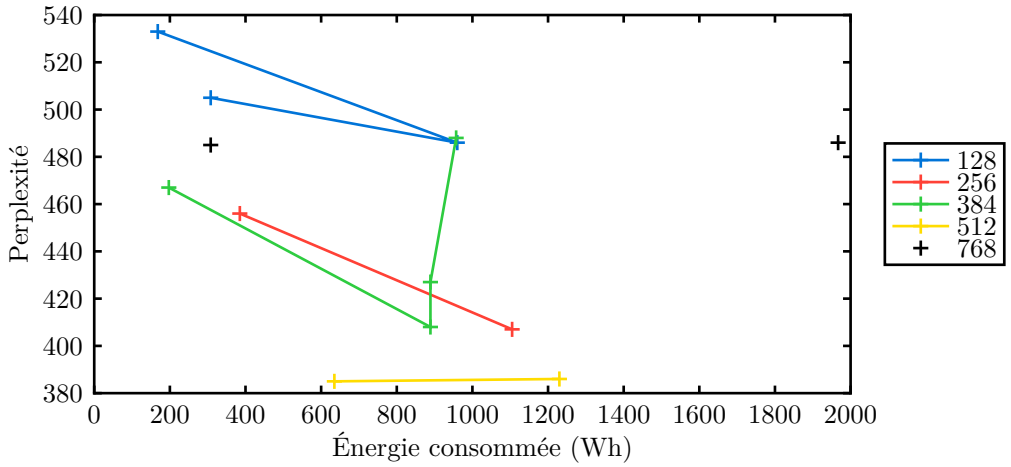

FIGURE 10. Perplexité en fonction de la consommation énergétique totale.

À ce stade de l'analyse, il semble que les modèles avec un embedding de longueur 512 sont non seulement les plus performants mais aussi les plus efficaces énergétiquement, au moins pour le jeu de données retenu.

4.4.3. *Résultats qualitatifs.*
Afin de ne pas se limiter à de simples métriques, nous souhaitons compléter et nuancer l'analyse de nos modèles par une évaluation qualitative. Nous avons à cette fin défini deux listes distinctes de prompts, chacune visant à explorer différentes facettes de la capacité des modèles à générer du texte en réponse à des stimuli donnés.
La première liste est composée de prompts à la construction simple, conçus pour évaluer la capacité des modèles à produire des réponses directes et concises à des questions ou des déclarations simples. Ces prompts sont formulés de manière à solliciter des réponses claires et facilement interprétables, ce qui permet d'évaluer la clarté et la pertinence des réponses générées par les modèles.
La seconde liste de prompts est constituée de questions plus complexes et nuancées, visant à tester la capacité des modèles à interpréter et à répondre à des stimuli plus élaborés. Ces prompts nécessitent une capacité d'analyse et de synthèse plus poussée de la part des modèles. En soumettant les modèles à ces questions plus exigeantes, nous cherchons à évaluer leur capacité à traiter des informations complexes et à générer des réponses cohérentes et informatives dans des situations plus subtiles.
À titre d'exemple et afin de ne pas encombrer plus que de raison ce rapport, nous donnons ici les réponses d'un seul modèle, supposé avoir des performances correctes si l'on se fie uniquement aux métriques usuelles.

- "In a galaxy far, far away… there are no visible differences in any of its visible stars, but no more than the luminosity of a galaxy.\"\n\n\"The luminosity of galaxies is a natural phenomenon\" (a) or \"massive\" in which galaxies are absorbed by a single atom; the \"massive black\" nuclei and stars are absorbed into stars by the two. These are only scattered by a small fraction of the stars, and the galaxies are absorbed by the Milky Way galaxy. \n\nThe universe is not generally"
- "I am but a simple man, because he is a man.\" He has always made it to be the most influential character in films. He also serves as a series writer.\n\nSee also\n In the American silent film, he appeared in many films during the first two years of the silent film trilogy, which

was played by David R. Kargis and starred as Toni Jugy. \n\nReferences\n\nBibliography\n\nBibliography\n\nExternal links\n\nWorks by Christopher R. Batson\n\n \n"

- “We all live in a yellow submarine. It is called \"Hockey House\" on the site. \n\nK\u00f6pp\nThe first recorded underwater submarine that was in operation since 1939. The first, the German submarine, was launched on 28 May 1942. It was the first of two successful submarines in the world that flew from to of the S\u00e4\u00e4ndel, where the first is in the history of the ship, but it took three weeks and a third. The first, however, was completed on 23 October"
- “A stranger is someone who is not a woman who is always a person of an independent class, but rather a woman who is not a woman who is in an independent class. It is important that someone who is not the person who is in a separate class is in the same class as that of a woman who is in the same class, who is in the same class as the woman who is in the same class. In most situations, some woman is in an independent class.\n\nIn some cases, some of the”
- “The State of California passed the California Clean Water Act of 1938. The California Water Service, enacted by law by law, provided that the water system provided for irrigation purposes, and that the water system provided for irrigation in the state was prohibited from entering service. The California Clean Water Act of 1938 authorized water companies to pay the state's taxes from the local water utility company, while state laws forbade water for sewage.\n\nThe California Water Supply Corporation was created by the California Civil Department of California in 1903 by the California”
- “In France, this is a history of the Italian-born Italian-born and French-born American journalist Jean-Baptiste Duvalier.\n\nIn the years following the birth of the Belgian child, this was the first major literary scandal, by which time there were two volumes (and the first volumes published in France since 1775).\n\nBetween 1663 and 1775, the most popular of the Italian-born Italian-born and French-born writers (such as the Italian-born”

L'évaluation des performances du modèle de langue à travers ces cinq prompts révèle des résultats variés, tant en termes de grammaticalité et de syntaxe que de cohérence avec le prompt donné.
Dans certains cas, le modèle parvient à générer du texte qui semble grammaticalement correct, tel que la réponse à “In a galaxy far, far away…” qui évoque des concepts astronomiques, bien que la cohérence avec le prompt soit discutable.
Cependant, dans d'autres cas, les réponses sont moins satisfaisantes. Par exemple, la réponse à “I am but a simple man, because” semble être une concaténation de phrases sans lien apparent avec le prompt, tandis que la réponse à “We all live in a yellow submarine” semble être une suite de phrases incohérentes sur un sous-marin allemand.
De manière similaire, les réponses à “A stranger is someone who” et “The State of California” semblent être des extraits de texte générés automatiquement sans rapport direct avec les prompts fournis.
Enfin, la réponse à “In France” mentionne un journaliste italo-français sans lien avec le prompt. Après quelques recherches, aucun journaliste historiquement connu ne semble porter ce nom; on trouve en revanche trace d'un Jean-Claude Duvalier, président haïtien ayant gourverné l'île de la mort de son père en 1971 à sa fuite du pays en 1986.

Les tests menés avec des prompts plus complexes donnent des résultats similaires, avec une grande variabilité dans la qualité des réponses et un respect de la grammaire et de la syntaxe aléatoire. Ces résultats soulignent la complexité de la tâche de génération de texte et mettent en évidence la difficulté qu'il y a à développer des modèles de langue capables de répondre de manière cohérente à une variété de prompts.

5. Analyse et Discussion

5.1. **Interprétation des résultats.**

5.1.1. *Impact de la taille de l'embedding.*

L'expérience montre que la taille de l'embedding a un impact significatif sur les performances des modèles de langue, avec des augmentations ayant un effet marqué sur le nombre de paramètres. Cette constatation souligne l'importance de cette dimension dans la capacité des modèles à capturer et à représenter les informations linguistiques.

5.1.2. *Influence du pré-entraînement sur des données spécifiques.*

Les résultats indiquent que les modèles pré-entraînés, comme GPT-2, peuvent présenter des performances considérablement différentes lorsqu'ils sont évalués sur des données pour lesquelles ils n'ont pas été spécifiquement entraînés. Cette observation met en lumière l'importance de l'adaptation et de la spécialisation des modèles pour des tâches ou des corpora spécifiques.

5.1.3. *Limites de la réduction de la perte.*

Malgré des réductions significatives de la perte lors de l'entraînement des modèles, cela ne se traduit pas nécessairement par des perplexités intéressantes. Ce constat suggère que d'autres facteurs, tels que la qualité et la diversité des données d'entraînement, peuvent jouer un rôle crucial dans les performances des modèles de langue.

5.1.4. *Complexité des hyperparamètres et taille du modèle.*

L'ajustement des hyperparamètres, notamment la taille de l'embedding, peut considérablement augmenter le nombre de paramètres du modèle. Cette observation met en évidence les compromis entre la complexité du modèle, le temps d'entraînement et les performances obtenues.

5.1.5. *Considérations sur la qualité des données.*

Une réflexion approfondie est nécessaire sur la qualité et la diversité des données d'entraînement. Les observations suggèrent que des corpora limités à un seul type de source, comme Wikipedia, peuvent biaiser l'entraînement et limiter la capacité des modèles à généraliser à d'autres types de données. Ces observations soulignent l'importance de prendre en compte plusieurs facteurs, notamment la taille de l'embedding, la spécificité des données d'entraînement et l'ajustement des hyperparamètres, dans la conception et l'évaluation des modèles de langue.

5.2. **Perspectives et améliorations possibles.**

5.2.1. *Augmentation de la diversité du corpus.*

Pour améliorer la diversité du corpus, il serait judicieux d'explorer des sources de données supplémentaires et de les intégrer à notre ensemble d'entraînement. Par exemple, en ajoutant des textes provenant de domaines variés tels que la science, la politique, la fiction littéraire, et en incluant des corpus multilingues, nous pourrions enrichir la représentation des différentes facettes du langage naturel. De plus, nous pourrions envisager d'appliquer des techniques d'augmentation de données, telles que le

remplacement de synonymes ou la modification de la structure syntaxique, pour diversifier davantage le corpus existant.

5.2.2. *Optimisation des méthodes de sélection de texte.*
Pour améliorer l'efficacité des méthodes de sélection de texte, nous pourrions explorer des approches d'échantillonnage plus intelligentes, telles que l'échantillonnage stratifié basé sur des critères de diversité linguistique ou thématique. De plus, l'utilisation d'algorithmes d'optimisation plus sophistiqués, tels que les algorithmes génétiques ou les méthodes d'optimisation par essaim, pourrait permettre de trouver plus efficacement les combinaisons optimales de textes pour maximiser l'indice de Vendi tout en réduisant la complexité computationnelle.

5.2.3. *Évaluation approfondie des performances.*
En plus des métriques classiques, nous pourrions effectuer une évaluation plus approfondie des performances des modèles sur des tâches spécifiques de NLP en utilisant des ensembles de données de référence et des benchmarks standard. Cela nous permettrait de mieux comprendre les forces et les faiblesses de nos modèles, ainsi que leur capacité à généraliser à différents types de données et de tâches. De plus, nous pourrions utiliser des techniques d'interprétabilité de modèle pour analyser les décisions prises par nos modèles et identifier les biais potentiels.

5.2.4. *Validation croisée.*
Pour évaluer la robustesse de nos modèles, nous pourrions utiliser des techniques de validation croisée parmi celles qui sont usuelles en traitement du langage. Cela nous permettrait d'estimer la variabilité des performances du modèle et d'obtenir des estimations plus fiables de ses performances réelles.
En deep learning et plus spécifiquement en NLP, les méthodes de validation les plus couramment utilisées sont les suivantes.

1. **Validation holdout :** C'est la méthode la plus simple, où le dataset est divisé en deux parties, l'une pour l'entraînement et l'autre pour la validation. Cependant, elle peut être sensible à la distribution des données et peut ne pas être représentative de la population entière.
2. **Validation croisée stratifiée :** Cette méthode est une amélioration de la validation holdout qui garantit une répartition équilibrée des classes dans chaque partition. Cela réduit le risque de biais lié à une répartition inégale des données.
3. **Validation croisée leave-one-out (LOOCV) :** Dans cette méthode, un seul échantillon est retenu comme ensemble de validation, tandis que tous les autres échantillons sont utilisés pour l'entraînement. Cela permet une utilisation maximale des données, mais peut être coûteux en termes de temps de calcul.
4. **Validation croisée k-fold :** Cette méthode divise le dataset en k partitions de taille égale. Chaque partition est utilisée une fois comme ensemble de validation, tandis que les k-1 partitions restantes sont utilisées pour l'entraînement. Cela permet d'estimer la variabilité des performances du modèle et réduit le risque de surajustement.
5. **Validation croisée bootstrapping :** Cette méthode implique de créer plusieurs échantillons bootstrap à partir du dataset d'origine et d'utiliser chaque échantillon pour l'entraînement et la validation. Cela permet également d'estimer la variabilité des performances du modèle.

En pratique, la validation holdout et la validation croisée stratifiée sont souvent privilégiées en raison de leur simplicité et de leur efficacité computationnelle, tandis que la validation croisée k-fold est utilisée lorsque des estimations plus précises des performances du modèle sont nécessaires. Cependant, son coût en temps de calcul serait prohibitif pour nos moyens, elle sera donc écartée.

5.2.5. *Régularisation.*
L'utilisation de techniques de régularisation telles que la réduction du taux d'apprentissage, la normalisation des poids ou l'ajout de termes de pénalisation dans la fonction de perte pourrait aider à prévenir le surajustement et à améliorer la généralisation des modèles.

5.2.6. *Systématisation.*
Dans le cadre de cette étude, nous avons exploré seulement une fraction des hyperparamètres disponibles pour l'entraînement des modèles de langue. La raison principale en est le manque de temps et de ressources disponibles pour entraîner en série nos modèles.
Nous avons fait le choix de nous concentrer sur certains hyperparamètres clés, comme la taille de l'embedding. À l'inverse, un jeu de données a été plus utilisé que les autres, principalement en raison de sa taille intermédiaire. Si d'autres jeux de données ont été utilisés ponctuellement, notamment dans le but de voir l'impact sur les performances des modèles, cette exploration a dû rester marginale.
Dans cette optique, des travaux futurs pourraient être entrepris pour systématiser notre approche et explorer de manière exhaustive l'ensemble des hyperparamètres pertinents. Une démarche de recherche plus exhaustive permettrait d'obtenir une compréhension plus approfondie des interactions entre les différents paramètres et de déterminer les configurations optimales pour des scénarios d'application spécifiques.

6. Implications pratiques et futures directions de recherche

Des pistes de recherche futures pourraient inclure l'exploration de techniques avancées pour augmenter la diversité du corpus, telles que l'utilisation de méthodes d'augmentation de données basées sur la génération de texte synthétique ou l'intégration de corpus multilingues.
De plus, il serait intéressant d'étudier l'impact de la diversité du corpus sur des tâches spécifiques de NLP, telles que la traduction automatique, la génération de texte créatif ou la classification de texte dans des domaines spécialisés. Par ailleurs, l'optimisation des méthodes de sélection de texte basées sur l'indice de Vendi pourrait être approfondie, en explorant des algorithmes d'optimisation plus efficaces et en évaluant leur performance sur des ensembles de données de grande taille.
Enfin, une étude approfondie de l'interaction entre la diversité du corpus, les caractéristiques des modèles de langue et les performances des modèles sur des tâches spécifiques de NLP pourrait fournir des idées intéressantes pour la conception et l'entraînement de modèles de langue plus performants et plus robustes.

Bibliographie

[1] G. Francopoulo, Pruning Texts with NLP and Expanding Queries with an Ontology: TagSearch, in: 2003: pp. 319–321. https://doi.org/10.1007/978-3-540-30222-3_30

[2] R. Tang, Y. Lu, J. Lin, Natural Language Generation for Effective Knowledge Distillation, in: 2019: pp. 202–208. https://doi.org/10.18653/v1/D19-6122

[3] G. Xu, J. Li, G. Gao, H. Lu, J. Yang, D. Yue, Lightweight Real-Time Semantic Segmentation Network With Efficient Transformer and CNN, IEEE Transactions on Intelligent Transportation Systems 24 (2023) 15897–15906. https://doi.org/10.1109/TITS.2023.3248089

[4] OpenAI, GPT-2 partial release statement, (n.d.). https://openai.com/index/better-language-models/ (accessed May 12, 2024)

[5] D. Friedman, A. B. Dieng, The Vendi Score: A Diversity Evaluation Metric for Machine Learning, Transactions on Machine Learning Research (n.d.). https://par.nsf.gov/biblio/10427561

[6] S. Biderman, et al., Pythia: a suite for analyzing large language models across training and scaling, in: Proceedings of the 40th International Conference on Machine Learning, JMLR.org, 2023

[7] A. Vaswani, et al., Attention is All you Need, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2017: p. . https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[8] A. P. Pasarkar, A. B. Dieng, Cousins Of The Vendi Score: A Family Of Similarity-Based Diversity Metrics For Science And Machine Learning, (2023)

[9] HuggingFace, Training a causal language model from scratch, (n.d.). https://huggingface.co/learn/nlp-course/en/chapter7/6 (accessed November 22, 2024)

[10] R. Mehrotra, Topic Modelling using LDA and LSA in Sklearn, (2022). https://www.kaggle.com/code/rajmehra03/topic-modelling-using-lda-and-lsa-in-sklearn (accessed February 15, 2024)

[11] Weights and biases, (n.d.). https://docs.wandb.ai/quickstart (accessed March 15, 2024)

Département de statistiques, Ensae, Palaiseau