

TOKENS
RECENT ADVANCES IN MACHINE LEARNING

Théo Lopès-Quintas

BPCE Payment Services,
Université Paris Dauphine

2023 - 2025

COMMENT PASSER DES MOTS AUX TOKENS ?

| | | |
|----------|---|-----------|
| 1 | Comment passer des mots aux tokens ? | 1 |
| 1.1 | Byte-Pair Encoding et WordPiece | 3 |
| 1.2 | SentencePiece et Unigram | 5 |
| 2 | Comment composer un corpus d'apprentissage ? | 6 |
| 2.1 | Qualité du dataset | 7 |
| 2.2 | Diversification du dataset | 16 |
| 3 | Comment choisir le prochain token ? | 18 |
| 3.1 | Échantillonnage par température | 19 |
| 3.2 | Échantillonnage top- k et <i>nucleus sampling</i> | 21 |

COMMENT PASSER DES MOTS AUX TOKENS ?

Exemple de tokenization pour un modèle Mistral :

| | | | |
|-------|--------|------|----------|
| Une | phrase | en | français |
| 16803 | 15572 | 1249 | 15067 |

Ou pour un modèle Gemma :

| | | | |
|-------|--------|-----|----------|
| Une | phrase | en | français |
| 19750 | 20911 | 659 | 24913 |

COMMENT PASSER DES MOTS AUX TOKENS ?

BYTE-PAIR ENCODING ET WORDPIECE

[Sennrich et al., 2015] adapte l'algorithme Byte-Pair Encoding (BPE) initialement construit pour la compression [Gage, 1994]. Cet algorithme a besoin du résultat d'un tokenizer : un texte tokenisé et le vocabulaire associé, mais également d'une taille de vocabulaire souhaité.

Exercice 1 (Byte-Pair Encoding)

On considère le vocabulaire suivant $V = [a, c, e, h, i, n, p, s, t]$ et les mots suivant avec leurs fréquences :

$(\text{"chat"}, 5), (\text{"chats"}, 3), (\text{"chien"}, 2), (\text{"patte"}, 5)$

1. Après avoir écrit les mots avec le vocabulaire de base, quelle est la paire la plus fréquente ?
2. Réécrire les mots avec un nouveau symbole, associé à la paire la plus fréquente.
3. Recommencer les deux premières étapes.
4. Quelle était la longueur du texte avant ? Et maintenant ?

COMMENT PASSER DES MOTS AUX TOKENS ?

BYTE-PAIR ENCODING ET WORDPIECE

WordPiece introduit initialement pour un problème de traduction Japonais-Coréen [Schuster and Nakajima, 2012] puis décrit plus en détail dans [Wu et al., 2016]. L'algorithme est très similaire à BPE et se différencie par le choix de la paire à former.

Concrètement, en reprenant l'exercice précédent, la paire la plus fréquente reste "at", mais comme les caractères "a" et "t" sont présent souvent, on obtient un score de :

$$S("at") = \frac{\text{Fréquence de "at"}}{\text{Fréquence de "a"} \times \text{Fréquence de "t"}} = \frac{13}{13 \times 18} \simeq 0.05$$

Mais si l'on regarde, la paire "ch" est la plus vraisemblable :

$$S("ch") = \frac{10}{10 \times 10} = 0.1$$

COMMENT PASSER DES MOTS AUX TOKENS ?

SENTENCEPIECE ET UNIGRAM

Si BPE et WordPiece utilisent une taille de vocabulaire faible puis l'augmente, Unigram [Kudo, 2018] fait l'inverse. A partir d'un vocabulaire contenant l'ensemble des caractères du texte et par exemple des sous-chaînes de caractères les plus commune, l'objectif d'Unigram est de supprimer des tokens pour réduire le vocabulaire à la taille souhaitée.

Unigram n'est jamais utilisé seul, mais conjointement avec SentencePiece [Kudo and Richardson, 2018]. L'algorithme a été proposé pour résoudre un problème dont les précédents souffrent : les mots ne sont pas forcément séparés par des espaces dans toutes les langues. Ainsi, SentencePiece inclut les espaces dans le jeu de caractères à utiliser.

| Modèle | Année | Algorithme | Vocabulaire |
|--------------|-------|---------------|-------------|
| Bert | 2018 | WordPiece | 30k |
| GPT2 | 2019 | BPE | 50k |
| LLaMa 1 | 2023 | SentencePiece | 32k |
| LLaMa 2 | 2023 | SentencePiece | 32k |
| Mistral 7B | 2023 | SentencePiece | 32k |
| Mixtral 8x7B | 2024 | SentencePiece | 32k |
| Gemma | 2024 | SentencePiece | 256k |

Table – Algorithme de tokenization et taille du vocabulaire

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

| | | |
|----------|---|-----------|
| 1 | Comment passer des mots aux tokens ? | 1 |
| 1.1 | Byte-Pair Encoding et WordPiece | 3 |
| 1.2 | SentencePiece et Unigram | 5 |
| 2 | Comment composer un corpus d'apprentissage ? | 6 |
| 2.1 | Qualité du dataset | 7 |
| 2.2 | Diversification du dataset | 16 |
| 3 | Comment choisir le prochain token ? | 18 |
| 3.1 | Échantillonnage par température | 19 |
| 3.2 | Échantillonnage top- k et <i>nucleus sampling</i> | 21 |

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

QUALITÉ DU DATASET

CommonCrawl est une organisation à but non-lucratif qui crée des *images* d'internet à chaque fois qu'un *crawl* est lancé. Parmi l'ensemble des données récoltés on y trouve des pages web, du texte et du code par exemple.

Dans l'optique de nettoyer les *dumps* de CommonCrawl est introduit par Google le dataset C4 [Raffel et al., 2019], pour *Colossal Cleaned Common Crawl Corpus*, pour entraîner le modèle T5. Plus tard les dataset RefinedWeb [Penedo et al., 2023] et FineWeb [Penedo et al., 2024] sont construit de la même manière pour respectivement entraîner Falcon et LLaMa 3.

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

QUALITÉ DU DATASET

Pour nettoyer les *dumps*, de nombreux filtres sont utilisés :

- ▶ **URL** : suppression de page selon une liste d'URL (4.6M de sites) et selon la présence de certains mots dans les URL
- ▶ **Langue** : ne sont conservé que les sites dont la langue identifié est l'anglais (le modèle fastText [Joulin et al., 2016] est utilisé avec un score de 0.65)
- ▶ **Qualité et répétition** : ne sont conservé que les sites qui vérifie les conditions proposées pour construire MassiveText qui a entraîné Gopher [Rae et al., 2021] :
 - La page contient entre 50 et 100 000 mots, et la longueur moyenne des mots est comprise entre 3 et 10
 - La page a un ratio entre symbole et mots inférieur à 0.1 pour les symboles dièse ou points de suspension
 - La page a moins de 90% de ses lignes qui commence par des puces et qui ont moins de 30% des lignes qui termine par des points de suspension
 - 80% des mots doivent contenir au moins une lettre
 - Au moins deux mots parmi la liste : *the, be, to, of, and, that ; have, with* doivent être présent dans le document
 - La page ne dépasse aucun des seuils de répétitions décrit dans l'article [Rae et al., 2021] section A.1.1. En un mot, plusieurs approches à plusieurs niveaux sont utilisées pour identifier les n -grams qui se répète dans une phrase, paragraphe...

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

QUALITÉ DU DATASET

[Lee et al., 2021] identifie quatre avantages d'entraîner un LLM sur un dataset dédoublonné :

1. Réduction du risque de mémorisation de certaines séquences présentent trop souvent (démontré dans [Carlini et al., 2022])
2. Réduction de l'overlap entre train et test. L'article exhibe une séquence de 61 mots qui est répété 61 036 dans C4
3. Gain en rapidité d'entraînement, donc évitement de coût d'entraînement
4. Gain en performance jusqu'à 10% grâce à du texte de meilleure qualité

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

COMMENT DÉDUPLIQUER ?

Considérons deux ensembles A et B , on définit l'indice de Jaccard comme :

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (\text{Indice de Jaccard})$$

Calculer ce coefficient avec l'ensemble de nos données est beaucoup trop coûteux. [Broder, 1997] introduit la méthode MinHash qui permet d'approcher cet indice à l'aide de fonction de hachage. En appliquant la fonction de hachage à A et B , la probabilité que la valeur minimale du hash de A et la valeur minimale du hash de B soit égale est exactement $J(A, B)$!

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

ALGORITHME MINHASH

| Texte 1 | Texte 2 | Texte 3 | Texte 4 |
|---------|---------|---------|---------|
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |

Table – Vecteurs représentant des textes

On considère les permutations :

$$P1 = [1, 3, 7, 6, 2, 5, 4], \quad P2 = [4, 2, 1, 3, 6, 7, 5], \quad P3 = [1, 4, 7, 6, 1, 2, 5]$$

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

ALGORITHME MINHASH

On obtient alors :

| T1 | T2 | T3 | T4 |
|----|----|----|----|
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 |

| T1 | T2 | T3 | T4 |
|----|----|----|----|
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |

| T1 | T2 | T3 | T4 |
|----|----|----|----|
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

QUALITÉ DU DATASET

On calcule la signature en prenant le premier indice où la valeur est 1 :

| S1 | S2 | S3 | S4 |
|----|----|----|----|
| 1 | 2 | 1 | 2 |
| 2 | 1 | 4 | 1 |
| 2 | 1 | 2 | 1 |

Avec cette table, on estime par exemple la similarité de T1 et T3 à 0.66 alors que la véritable similarité est 0.75 : c'est une bonne approximation.

FineWeb collecte l'ensemble des 5-grams de chaque document et calcule 112 fonctions de hash réparties dans 14 blocs de 8 hashes chacun. Ainsi, si deux documents ont pour similarité s , la probabilité qu'il soit effectivement identifié comme similaire est :

$$\mathbb{P}(\text{documents similaire identifié}) = (1 - (1 - s^8)^{14})$$

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

QUALITÉ DU DATASET

| Modèle | Année | Tokens d'entraînement |
|------------|---------------|-----------------------|
| Chinchilla | Mars 2022 | 1.3T |
| PaLM | Avril 2022 | 768B |
| phi-1 | Novembre 2022 | 6B |
| LLaMa | Février 2023 | 1.4T |
| PaLM 2 | Mai 2023 | 3.6T |
| Falcon | Juin 2023 | 5T |
| LLaMa 2 | Juillet 2023 | 2T |
| Gemma | Février 2024 | 6T |
| LLaMa 3 | Avril 2024 | 15T |

Table – Nombre de token d'entraînement pour quelques LLM

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

QUALITÉ DU DATASET

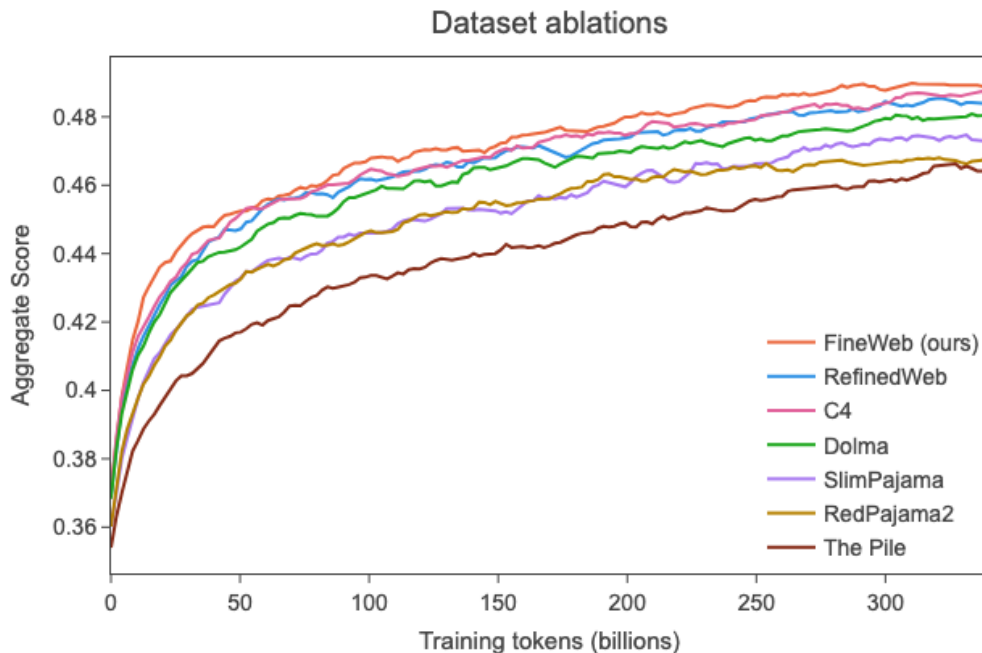


Figure – Performances d'un LLM en fonction du nombre de token d'entraînement selon le dataset

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

DIVERSIFICATION DU DATASET

| Modèle | Page Web | Code | Encyclopédique | Livres | Académique | Réseaux sociaux | Langue |
|------------|----------|------|----------------|--------|------------|-----------------|--------|
| LLaMa | 82 | 4.5 | 4.5 | 4.5 | 2.5 | 2.0 | 0 |
| GPT-3 | 60 | 22 | 3 | 15 | 0 | 0 | 0 |
| PaLM | 27 | 50 | 4 | 13 | 0 | 5 | 1 |
| Gopher | 58 | 3 | 2 | 27 | 0 | 0 | 10 |
| Chinchilla | 55 | 4 | 1 | 30 | 0 | 0 | 10 |
| Falcon | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| phi-1 | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| LaMDA | 12.5 | 50 | 12.5 | 0 | 0 | 12.5 | 12.5 |

Table – Composition du corpus d'entraînement (en %). Source : [Liu et al., 2024]

COMMENT COMPOSER UN CORPUS D'APPRENTISSAGE ?

DIVERSIFICATION DU DATASET

| Langue | Tokens dans mC4 | | Locuteurs natifs | |
|-------------------|-----------------------|----------------|----------------------|----------------|
| | Nombre (en milliards) | Proportion (%) | Nombre (en millions) | Proportion (%) |
| Anglais | 2000 | 45.5 | 379 | 5.1 |
| Russe | 250 | 5.7 | 154 | 2.1 |
| Allemand | 200 | 4.5 | 95 | 1.3 |
| Français | 170 | 3.9 | 76.8 | 1.0 |
| Espagnol | 160 | 3.6 | 460 | 6.2 |
| Chinois simplifié | 150 | 3.4 | 1300 | 17.6 |
| Portugais | 120 | 2.7 | 234 | 3.2 |
| Italien | 100 | 2.3 | 59.8 | 0.8 |
| Polonais | 90 | 2.0 | 46.6 | 0.6 |
| Japonais | 80 | 1.8 | 128 | 1.7 |

Table – 10 langues les plus représentées dans mC4, sources Ethnologue et [Raffel et al., 2019]

COMMENT CHOISIR LE PROCHAIN TOKEN ?

| | | |
|----------|---|-----------|
| 1 | Comment passer des mots aux tokens ? | 1 |
| 1.1 | Byte-Pair Encoding et WordPiece | 3 |
| 1.2 | SentencePiece et Unigram | 5 |
| 2 | Comment composer un corpus d'apprentissage ? | 6 |
| 2.1 | Qualité du dataset | 7 |
| 2.2 | Diversification du dataset | 16 |
| 3 | Comment choisir le prochain token ? | 18 |
| 3.1 | Échantillonnage par température | 19 |
| 3.2 | Échantillonnage top- k et <i>nucleus sampling</i> | 21 |

COMMENT CHOISIR LE PROCHAIN TOKEN ?

ÉCHANTILLONNAGE PAR TEMPÉRATURE

Considérons un LLM qui pour une séquence de token prédit le prochain. Alors il produit un vecteur de taille $d \in \mathbb{N}$ avec d le nombre de token dans le vocabulaire du modèle. Nous avons besoin d'avoir un vecteur de probabilité qui soit généré. Un réseau de neurones ne le fait pas naturellement mais en ajoutant la fonction d'activation softmax on peut alors former un vecteur de probabilité :

i -ème valeur initiale du vecteur

$$p_i = \frac{\exp\left(\frac{-\varepsilon_i}{\tau}\right)}{\sum_{j=1}^d \exp\left(\frac{-\varepsilon_j}{\tau}\right)}$$

Température

Le paramètre τ est appelé la température par inspiration du domaine de la thermodynamique en physique. Étudions plus en détail les possibilités de cette fonction d'activation.

COMMENT CHOISIR LE PROCHAIN TOKEN ?

ÉCHANTILLONNAGE PAR TEMPÉRATURE

Exercice 2 (Température)

On considère $\varepsilon_1, \varepsilon_2 \in \mathbb{R}$ tels que $\varepsilon_1 < \varepsilon_2$. On définit $0 < \tau_1 < \tau_2$ deux températures.

1. Pour une température $\tau > 0$ fixée, on note p_1 et p_2 les valeurs associées à la transformation softmax de ε_1 et ε_2 . A-t-on que $p_1 < p_2$?
2. Comment varie la valeur de p quand τ varie ?
3. Calculer $\lim_{\tau \rightarrow +\infty} p_i$.

COMMENT CHOISIR LE PROCHAIN TOKEN ?

ÉCHANTILLONNAGE TOP- k ET *nucleus sampling*

[Fan et al., 2018] propose la méthode *top-k sampling* et [Holtzman et al., 2019] proposent le *nucleus sampling*.

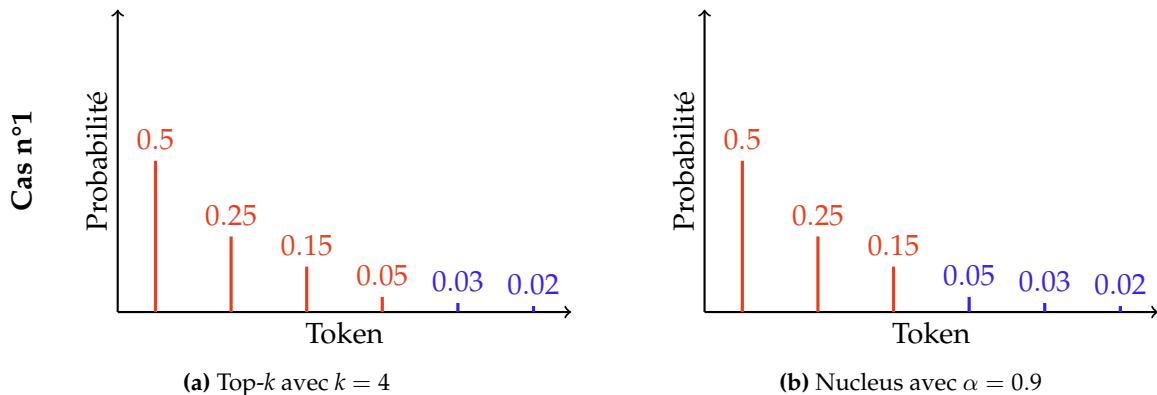


Figure – Exemple de deux stratégies de sampling pour **sélections** les tokens

COMMENT CHOISIR LE PROCHAIN TOKEN ?

ÉCHANTILLONNAGE TOP- k ET *nucleus sampling*

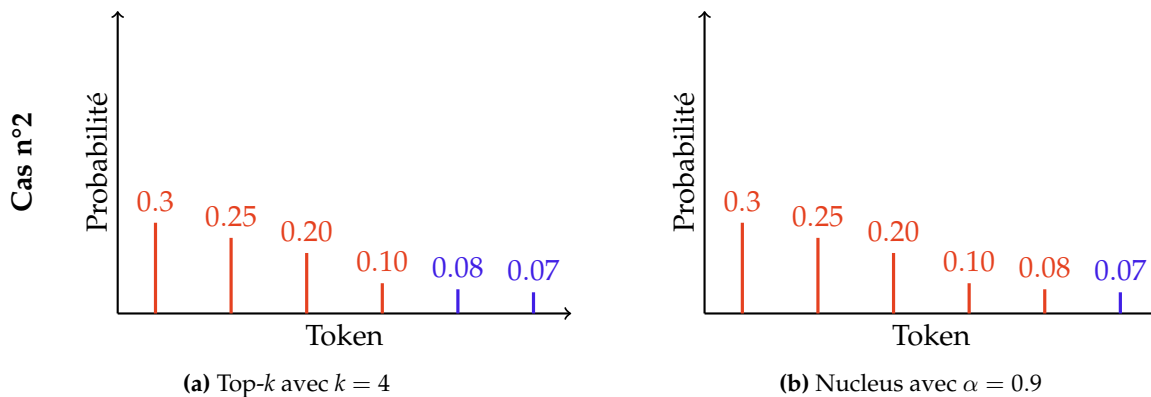













Figure – Exemple de deux stratégies de sampling pour **sélections** les tokens

Combiner une valeur bien choisie pour τ permet de mieux calibrer la valeur α du nucleus sampling. En revanche, cela n'a pas d'impact pour le top- k sampling.





BIBLIOGRAPHIE I

-  [Broder, A. Z. \(1997\).](#)
On the resemblance and containment of documents.
In Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171).
-  [Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. \(2022\).](#)
Quantifying memorization across neural language models.
arXiv preprint arXiv :2202.07646.
-  [Fan, A., Lewis, M., and Dauphin, Y. \(2018\).](#)
Hierarchical neural story generation.
arXiv preprint arXiv :1805.04833.
-  [Gage, P. \(1994\).](#)
A new algorithm for data compression.
The C Users Journal.
-  [Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. \(2019\).](#)
The curious case of neural text degeneration.
arXiv preprint arXiv :1904.09751.
-  [Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. \(2016\).](#)
Bag of tricks for efficient text classification.
arXiv preprint arXiv :1607.01759.



BIBLIOGRAPHIE II

-  Kudo, T. (2018).
Subword regularization : Improving neural network translation models with multiple subword candidates.
arXiv preprint arXiv :1804.10959.
-  Kudo, T. and Richardson, J. (2018).
Sentencepiece : A simple and language independent subword tokenizer and detokenizer for neural text processing.
arXiv preprint arXiv :1808.06226.
-  Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. (2021).
Deduplicating training data makes language models better.
arXiv preprint arXiv :2107.06499.
-  Liu, Y., Cao, J., Liu, C., Ding, K., and Jin, L. (2024).
Datasets for large language models : A comprehensive survey.
arXiv preprint arXiv :2402.18041.
-  Penedo, G., Kydlíček, H., von Werra, L., and Wolf, T. (2024).
Fineweb.

BIBLIOGRAPHIE III

-  Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E., and Launay, J. (2023).
The refinedweb dataset for falcon llm : outperforming curated corpora with web data, and web data only.
arXiv preprint arXiv :2306.01116.
-  Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., et al. (2021).
Scaling language models : Methods, analysis & insights from training gopher.
arXiv preprint arXiv :2112.11446.
-  Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019).
Exploring the limits of transfer learning with a unified text-to-text transformer.
Journal of machine learning research.
-  Schuster, M. and Nakajima, K. (2012).
Japanese and korean voice search.
In 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP).

BIBLIOGRAPHIE IV

-  Sennrich, R., Haddow, B., and Birch, A. (2015).
Neural machine translation of rare words with subword units.
arXiv preprint arXiv :1508.07909.
-  Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016).
Google's neural machine translation system : Bridging the gap between human and machine translation.
arXiv preprint arXiv :1609.08144.