

NMLM : Réseaux de neurones pré-entraînés pour la génération de texte et implications

Antoine Simoulin ^[1,2]

^[1] Quantmetry

^[2] Université de Paris, Laboratoire de Linguistique Formelle

Nantes Machine Learning Meetup, Septembre 2021





Antoine SIMOULIN
Data Scientist • Quantmetry
Doctorant LLF

antoine.simoulin@gmail.com



Le **Laboratoire de linguistique formelle** étudie tous les aspects du langage, du mot au discours et au dialogue, du signal acoustique à l'interprétation. Ses membres développent une approche formelle de ce système cognitif particulier qu'est le langage, et déclinent leurs recherches en utilisant des méthodes et des objectifs relevant de la linguistique théorique, de la linguistique expérimentale, de la linguistique computationnelle, de la linguistique de terrain et de la typologie. Animé d'un esprit ouvert et collaboratif, le Laboratoire de Linguistique Formelle est membre du Labex EFL – Empirical Foundations of Linguistics, et participe à de nombreuses collaborations nationales et internationales.



Près d'un article scientifique sur dix publié en France est issu des laboratoires de **Université de Paris**. Première université française en termes de citations par article, elle délivre 8 % des doctorats au niveau national et développe une recherche au meilleur niveau mondial, directement connectée aux enjeux et aux acteurs internationaux, avec une ambition particulière à l'échelle européenne.



Quantmetry est un cabinet de conseil de 120 personnes, spécialisé en Intelligence Artificielle, pour accompagner de bout en bout la transformation data des grands groupes français. Nous concevons des Intelligences Artificielles à l'état de l'art des dernières technologies. L'ensemble de nos consultants consacrent une partie substantielle de leur temps en Recherche & Développement. La présence d'un directeur scientifique et de directeurs techniques garantit l'excellence et l'exceptionnalité de nos sujets.

1

NLP « Révolution »

En 2019, Google introduit la technologie **BERT** (*Bidirectional Encoder Representations from Transformers*) dans son moteur de recherche. La technologie **augmente la pertinence des résultat d'une recherche sur dix**.

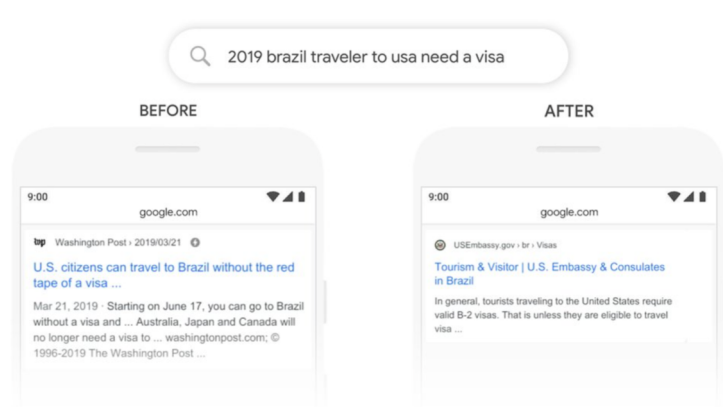


Figure : Le modèle BERT comprend qu'il s'agit des visiteurs arrivant aux U.S.A et non l'inverse en composant correctement les mots de la phrase articulés autour de l'expression « to »^[1].

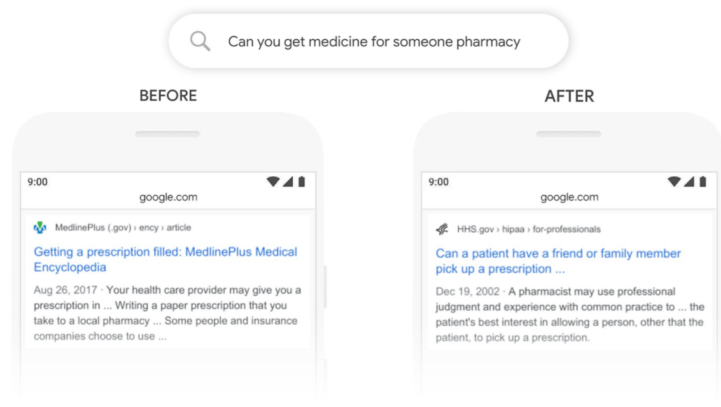


Figure : Le modèle BERT saisit la nuance « for someone » et permet d'affiner les résultats par rapport à des informations plus générales^[1].

^[1] <https://blog.google/products/search/search-language-understanding-bert/>

GLUE et SuperGLUE

Les modèles dépassent les performances humaines sur les benchmarks d'évaluation.

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
1	ERNIE Team - Baidu	ERNIE		91.1	75.5	97.8	93.9/91.8	93.0/92.6	75.2/90.9	92.3	91.7	97.3	92.6	95.9	51.7
2	AlicMind & DRL	StructBERT + CLEVER		91.0	75.3	97.7	93.9/91.9	93.5/93.1	75.6/90.8	91.7	91.5	97.4	92.5	95.2	49.1
3	DeBERTa Team - Microsoft	DeBERTa / TuringNLrv4		90.8	71.5	97.5	94.0/92.0	92.9/92.6	76.2/90.8	91.9	91.6	99.2	93.2	94.5	53.2
4	HFL-FLYTEK	MacALBERT + DKM		90.7	74.8	97.0	94.5/92.6	92.8/92.6	74.7/90.6	91.3	91.1	97.8	92.0	94.5	52.6
+	PING-AN Omni-Sinix	ALBERT + DAAF + NAS		90.6	73.5	97.2	94.0/92.0	93.0/92.4	76.1/91.0	91.6	91.3	97.5	91.7	94.5	51.2
5	langshu ge	Deberta + adv (ensemble)		90.4	72.7	97.3	92.7/90.3	93.2/92.9	75.6/90.8	91.7	91.5	96.4	92.5	95.2	35.2
7	T5 Team - Google	T5		90.3	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	92.2	91.9	96.9	92.8	94.5	53.1
8	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
+	Huawei Noah's Ark Lab	NEZHA-Large		89.8	71.7	97.3	93.3/91.0	92.4/91.9	75.2/90.7	91.5	91.3	96.2	90.3	94.5	47.9
+	Zhang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)		89.7	70.5	97.5	93.4/91.2	92.6/92.3	75.4/90.7	91.4	91.1	95.8	90.0	94.5	51.6
+	ELECTRA Team	ELECTRA-Large + Standard Tricks		89.4	71.7	97.1	93.1/90.7	92.9/92.5	75.6/90.8	91.3	90.8	95.8	89.8	91.8	50.7
+	Microsoft D365 AI & UMD	FreeB-RoBERTa (ensemble)		88.4	68.0	95.8	93.1/90.8	92.3/92.1	74.8/90.3	91.1	90.7	95.6	88.7	89.0	50.1
13	Junjie Yang	HIRE-RoBERTa		88.3	68.5	97.1	93.0/90.7	92.4/92.0	74.3/90.2	90.7	90.4	95.5	87.9	89.0	49.3
14	Facebook AI	RoBERTa		88.1	67.8	95.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	95.4	88.2	89.0	48.7
+	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8
16	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	95.3/90.8	92.7/92.5	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-
+	Adrian de Wynter	Bort (Alexa AI)		83.6	63.9	96.2	94.1/92.3	89.2/88.3	66.0/55.9	88.1	87.8	92.3	82.7	71.2	51.9
18	Lab LV	ConvBERT base		83.2	67.8	95.7	91.4/88.3	90.4/89.7	73.0/90.0	88.3	87.4	93.2	77.9	65.1	42.9

Rank	Name	Model	URL	Score	BoolQ	CB	COPA	MultIRCC	ReCoRD	RTE	WIC	WSC	AX-b	AX-g
1	ERNIE Team - Baidu	ERNIE 3.0		90.6	91.0	98.6/99.2	97.4	88.6/63.2	94.7/94.2	92.6	77.4	97.3	68.6	92.7/94.7
+	Zirui Wang	T5 + UDQ, Single Model (Google Brain)		90.4	91.4	95.8/97.6	98.0	88.3/63.0	94.2/93.5	93.0	77.9	96.6	69.1	92.7/91.9
+	DeBERTa Team - Microsoft	DeBERTa / TuringNLrv4		90.3	90.4	95.7/97.6	98.4	88.2/63.7	94.5/94.1	93.2	77.5	95.9	66.7	93.3/93.8
4	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	76.6	99.3/99.7
+	T5 Team - Google	T5		89.3	91.2	93.9/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8	65.6	92.7/91.9
+	Huawei Noah's Ark Lab	NEZHA-Plus		86.7	87.8	94.4/96.0	93.6	84.6/55.1	90.1/89.6	89.1	74.6	93.2	58.0	87.1/74.4
+	Alibaba PAI/ICBU	PAI Albert		86.1	88.1	92.4/96.4	91.8	84.6/54.7	89.0/88.3	88.8	74.1	93.2	75.6	98.3/99.2
+	Infosys : DAWN : AI Research	RoBERTa-ICETS		86.0	88.5	93.2/95.2	91.2	86.4/58.2	89.9/89.3	89.9	72.9	89.0	61.8	88.8/81.5
+	Tencent Jarvis Lab	RoBERTa (ensemble)		85.9	88.2	92.5/95.6	90.8	84.4/53.4	91.5/91.0	87.9	74.1	91.8	57.6	89.3/75.6
10	Zhuiyi Technology	RoBERTa-mtl-adv		85.7	87.1	92.4/95.6	91.2	85.1/54.3	91.7/91.3	88.1	72.1	91.8	58.5	91.0/78.1
11	Facebook AI	RoBERTa		84.6	87.1	90.5/95.2	90.6	84.4/52.5	90.6/90.0	88.2	69.9	89.0	57.9	91.0/78.1
+	Anuar Sharafudinov	AILabs Team, Transformers		82.6	88.1	91.6/94.8	86.8	85.1/54.7	82.8/79.8	88.9	74.1	78.8	100.0	100.0/100.0
13	Rakesh Radhakrishnan Menon	ADAPET (ALBERT) - few-shot		76.0	80.0	82.3/92.0	85.4	76.2/35.7	86.1/85.5	75.0	53.5	85.6	-0.4	100.0/90.0
+	Timo Schick	IPET (ALBERT) - Few-Shot (32 Examples)		75.4	81.2	79.9/88.8	90.8	74.1/31.7	85.9/85.4	70.8	49.3	88.4	36.2	97.8/57.9
15	Adrian de Wynter	Bort (Alexa AI)		74.1	83.7	81.9/86.4	89.6	83.7/54.1	49.8/49.0	81.2	70.1	65.8	48.0	96.1/61.5
16	IBM Research AI	BERT-mtl		73.5	84.8	89.6/94.0	73.8	73.2/30.5	74.6/74.0	84.1	66.2	61.0	29.6	97.8/57.3
17	Ben Mann	GPT-3 few-shot - OpenAI		71.8	76.4	52.0/75.6	92.0	75.4/30.5	91.1/90.2	69.0	49.4	80.1	21.1	90.4/55.3
18	SuperGLUE Baselines	BERT++		71.5	79.0	84.8/90.4	73.8	70.0/24.1	72.0/71.3	79.0	69.6	64.4	38.0	99.4/51.4

Figure : Les leaderboards des benchmarks GLUE et SuperGLUE. Les performances humaines sont classées respectivement 16^{ème} et 4^{ème}. Les performances des modèles sur SuperGLUE ont dépassé les performances humaines en 2020 [1, 2].



[1] <https://gluebenchmark.com/leaderboard>

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu: *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. J. Mach. Learn. Res. 21: 140:1-140:67 (2020)

[3] Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen: *Deberta: decoding-Enhanced Bert with Disentangled Attention*. ICLR 2021

En 2020, OpenAI déploie son API qui permet d'adresser n'importe quelle tâche linguistique : recherche sémantique, résumé, analyse de sentiments, génération de contenu, traduction à l'aide de quelques exemples seulement.

Prompt

English: I do not speak French.
French: Je ne parle pas français.

English: See you later!
French: À tout à l'heure!

English: Where is a good restaurant?
French: Où est un bon restaurant?

English: What rooms do you have available?
French: Quelles chambres avez-vous de disponible?

English: Where is the restroom?
French:

Sample response

Où est les toilettes?

Prompt

Write a creative ad for the following product to run on Facebook:

Airee is a line of skin-care products for young women with delicate skin.
The ingredients are all-natural.

This is the ad I wrote for Facebook aimed at teenage girls:

Sample response

Do you have sensitive skin? If so, then Airee is for you! Airee products are made with natural ingredients to help you feel good about your skin. Try out Airee for a happy, healthy you!

Q: Who is Fred Rickerson?
A: ?

Q: What is an atom?
A: An atom is a tiny particle that makes up everything.

Q: Who is Alvan Muntz?
A: ?

Q: What is Kozar-09?
A: ?

Q: How many moons does Mars have?
A: Two, Phobos and Deimos.

Q: What's a language model?
A:

Sample response

A language model is a statistical model that describes the probability of a word given a context.

Figure : Quelques exemples d'application du modèle pour la traduction automatique, la génération automatique de publicité ou un système de questions/réponses.

En 2021, Le modèle DALL•E, développé par OpenAI, s'appuie sur une version de GPT-3 avec peu de paramètres pour générer des images à partir de texte.

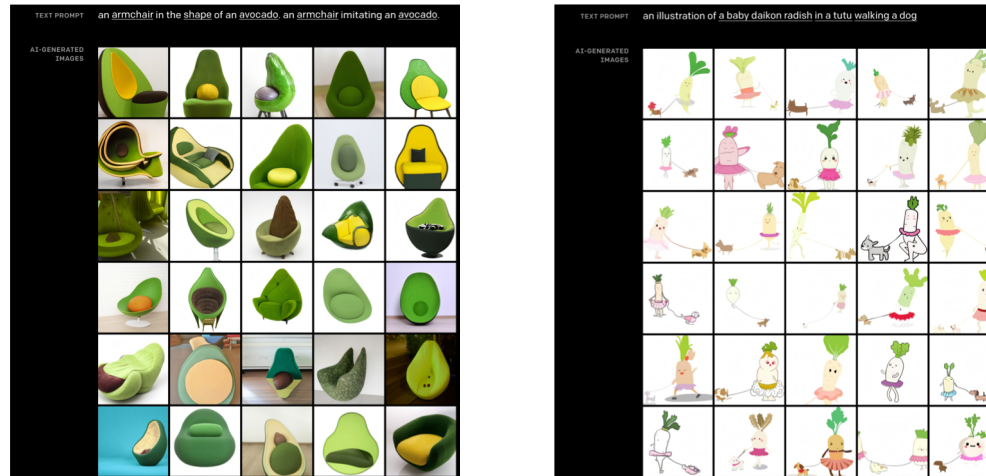


Figure : Génération d'images à partir d'un query. Ici un « fauteuil avec la forme d'un avocat » ou un « bébé radis en tutu qui promène un chien ».

En 2021, OpenAI adapte à nouveau son modèle GPT-3 pour générer du code à partir d'un query. L'IA participe à un challenge de code en ligne pour résoudre des problèmes en python et se classe parmi les 100 premiers.

92	 gggabob	18	5	81.23	11:27:28 AM
93	 a.matkurbanov	1	5	80.16	11:27:33 AM
94	 Szymon Wojdat	4	5	86.06	11:27:36 AM
95	 toonewbie	1	5	88.16	11:28:27 AM
96	 OpenAI Codex	N/A	5	88.37	11:28:44 AM
97	 Yannic Kilcher	9	5	88.56	11:29:02 AM
98	 Владимир Сотников	5	5	80.08	11:29:03 AM
99	 krsnaa	7	5	89.19	11:29:24 AM
100	 daste745	7	5	88.39	11:29:55 AM

Figure : Position de l'OpenAI Codex dans le leaderboard.

HuggingFace

En 2021, HuggingFace [1] lève **\$40 M** pour développer une librairie **Open Source**. La librairie compte près de 15,000 modèles. Le modèle bert-base-uncased compte près de **34M de téléchargements par mois**.

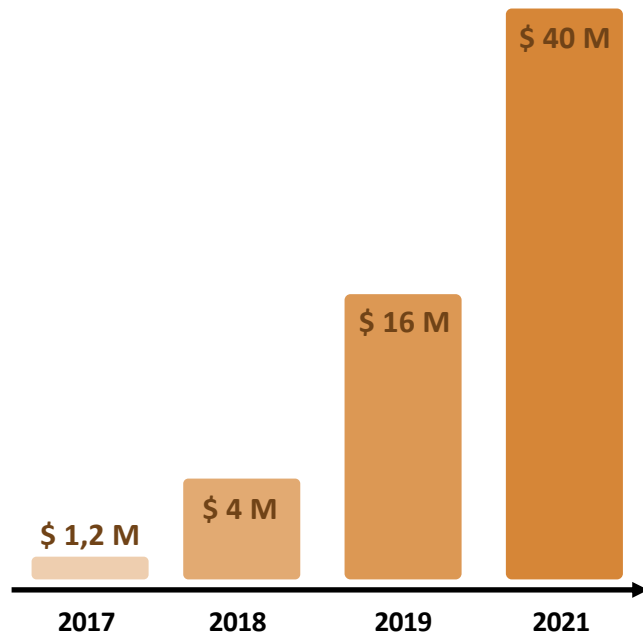


Figure : Historique des levées de fonds de HuggingFace [1].

[1] <https://huggingface.co/>



Figure : Page d'accueil de la librairie HuggingFace [1].

HuggingFace

A la conférence IO de 2021, Google introduit MUM. Un nouveau modèle qui s'appuie sur l'architecture texte-to-texte T5 [2]. Selon eux, le modèle est 1 000 fois plus puissant que BERT [1]. MUM permet **d'encoder et de générer du langage**. Il est entraîné sur **75 langues différentes** et sur **plusieurs tâches** différentes à la fois. Finalement, MUM est **multimodal**, pour l'instant à travers le texte et les images et, à l'avenir éventuellement la vidéo et l'audio.

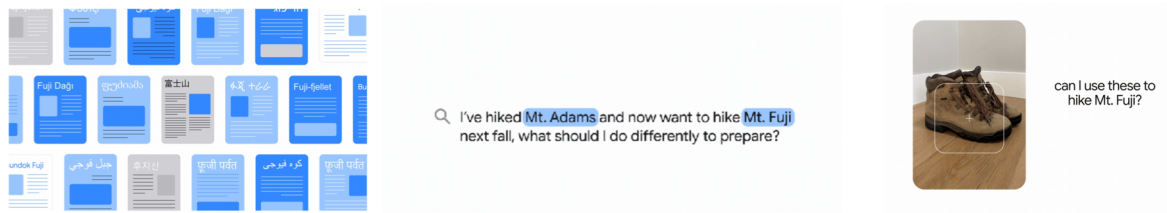


Figure : Les caractéristiques de MUM résume bien les tendances actuelles en NLP: (1) encodage et génération, (2) multilingue, (3) entraîné sur plusieurs tâches et (4) multimodal.



[1] <https://blog.google/products/search/introducing-mum/>

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu: *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. J. Mach. Learn. Res. 21: 140:1-140:67 (2020)

2

Architectures transformers

L'architecture Seq2seq

Le réseau **Seq2Seq** ^[1] (*sequence to sequence*) encode dans un premier temps le texte sous la forme d'un vecteur de contexte de taille fixe. Ce vecteur est ensuite utilisé comme entrée pour un module décodeur pour conditionner la génération de texte.

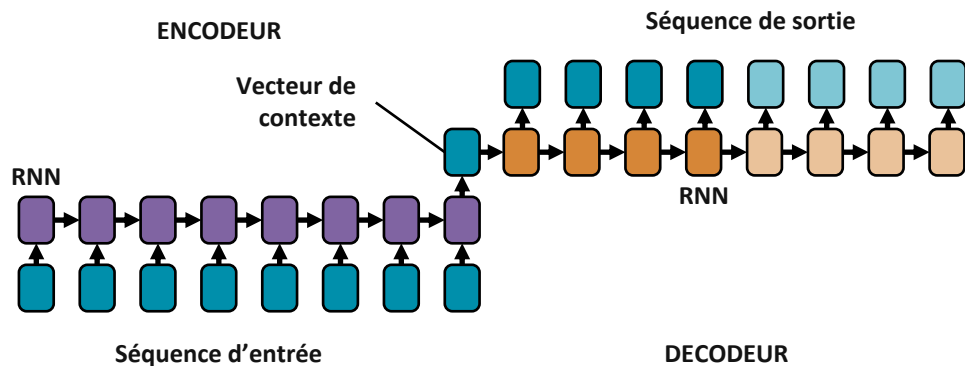


Figure : Le réseau encodeur/décodeur sans attention

[1] Ilya Sutskever, Oriol Vinyals, Quoc V. Le: *Sequence to Sequence Learning with Neural Networks*. NIPS 2014: 3104-3112

La méthode d'attention

Les réseaux **Seq2Seq** (*sequence to sequence*) ne capturent pas efficacement la sémantique de longues phrases. La méthode d'attention ^[1] permet de **calculer un vecteur de contexte, comme une somme pondérée des représentations contextuelles de chaque mot, à chaque étape du décodage**. Cette méthode est en particulier introduite pour la traduction afin de se focaliser sur les mots que l'on est en train de traduire.

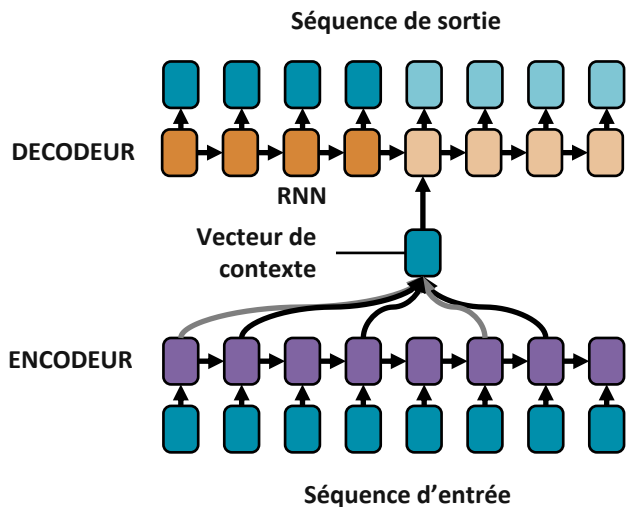


Figure : Le réseau encodeur/décodeur sans attention avec la méthode d'attention, le vecteur de contexte est recalculé à chaque pas du décodage comme une somme pondérée des sorties du réseau d'entrée.

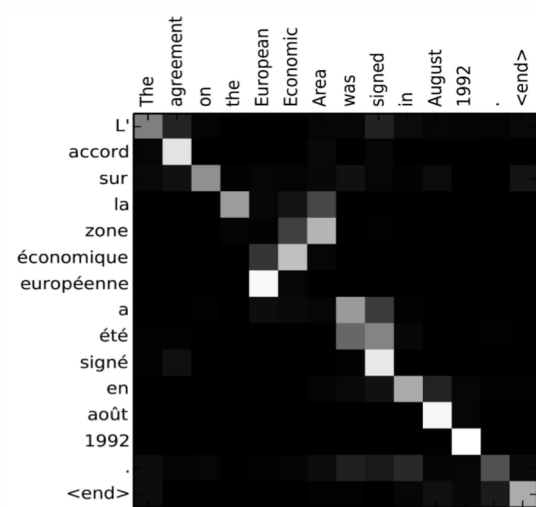


Figure : Matrice d'alignement entre "L'accord sur l'Espace économique européen a été signé en août 1992" et sa traduction en anglais "The agreement on the European Economic Area was signed in August 1992". Image extraite de ^[1].

^[1] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio: *Neural Machine Translation by Jointly Learning to Align and Translate*. ICLR 2015

La méthode d'attention

Depuis l'introduction de la méthode d'attention, de nombreuses variantes ont émergé. L'article de blog [2] donne les repères principaux, détaille les différentes variantes et leur évolution.



Figure : “A woman is throwing a frisbee in a park.” Image extraite de [1].



[1] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio: *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. ICML 2015: 2048-2057

[2] <https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html>

Les transformers et l'auto-attention

Les **transformers** [1] sont une architecture Seq2seq qui s'appuie uniquement sur l'attention (plus de module séquentiel ou récurrent). Le réseau est utilisé à l'origine pour la traduction. Il introduit un nouveau type d'attention : **l'auto-attention**.

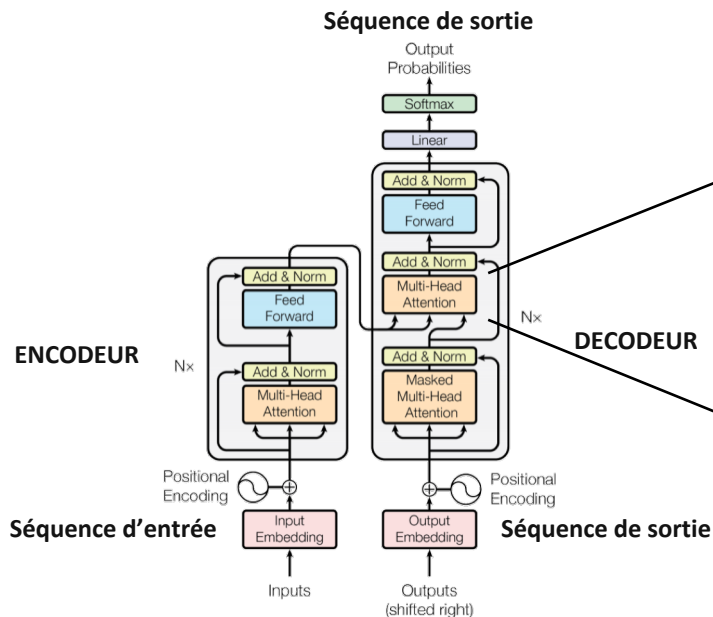


Figure : L'architecture transformers. Image extraite de [1].

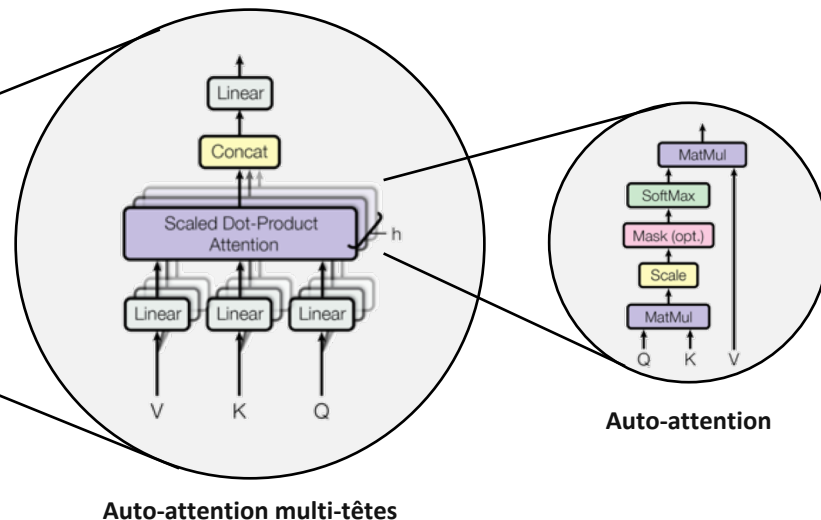


Figure : Le module d'auto-attention. Image extraite de [1].



[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin: *Attention is All you Need*. NIPS 2017: 5998-6008

Les architectures transformers

L'architecture du modèle **BERT** [1, 4] est inspirée de l'architecture du modèle **transformers**. Elle est constituée du série **d'encodeurs** qui s'appuient sur l'opération d'**auto attention**. Cette dernière est très facile à paralléliser. Le modèle compte entre 110 et 336 millions de paramètres. Ce qui est énorme par rapport aux modèles de NLP traditionnels. Ce modèle a permis une nette amélioration des performances sur les benchmarks académiques : **7,7% d'amélioration absolue sur le benchmark GLUE** [5]. L'architecture GPT [2, 3] consiste quant à elle à une succession de couche de décodeur. Les gains en performances sont un peu moindre car on ne s'appuie que sur une partie du contexte.

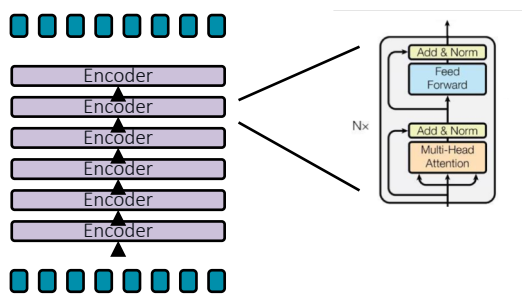


Figure : L'architecture BERT correspond à un empilement de couches d'encodeurs.

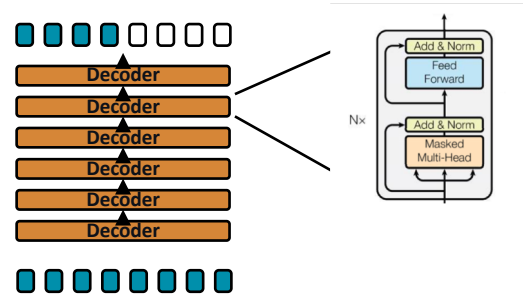


Figure : L'architecture GPT correspond à un empilement de couches de décodeurs.



[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova: **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. NAACL-HLT (1) 2019: 4171-4186

[2] Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever: **Improving Language Understanding by Generative Pre-Training**. <https://openai.com/> 2018

[3] <https://jalammar.github.io/illustrated-gpt2/>

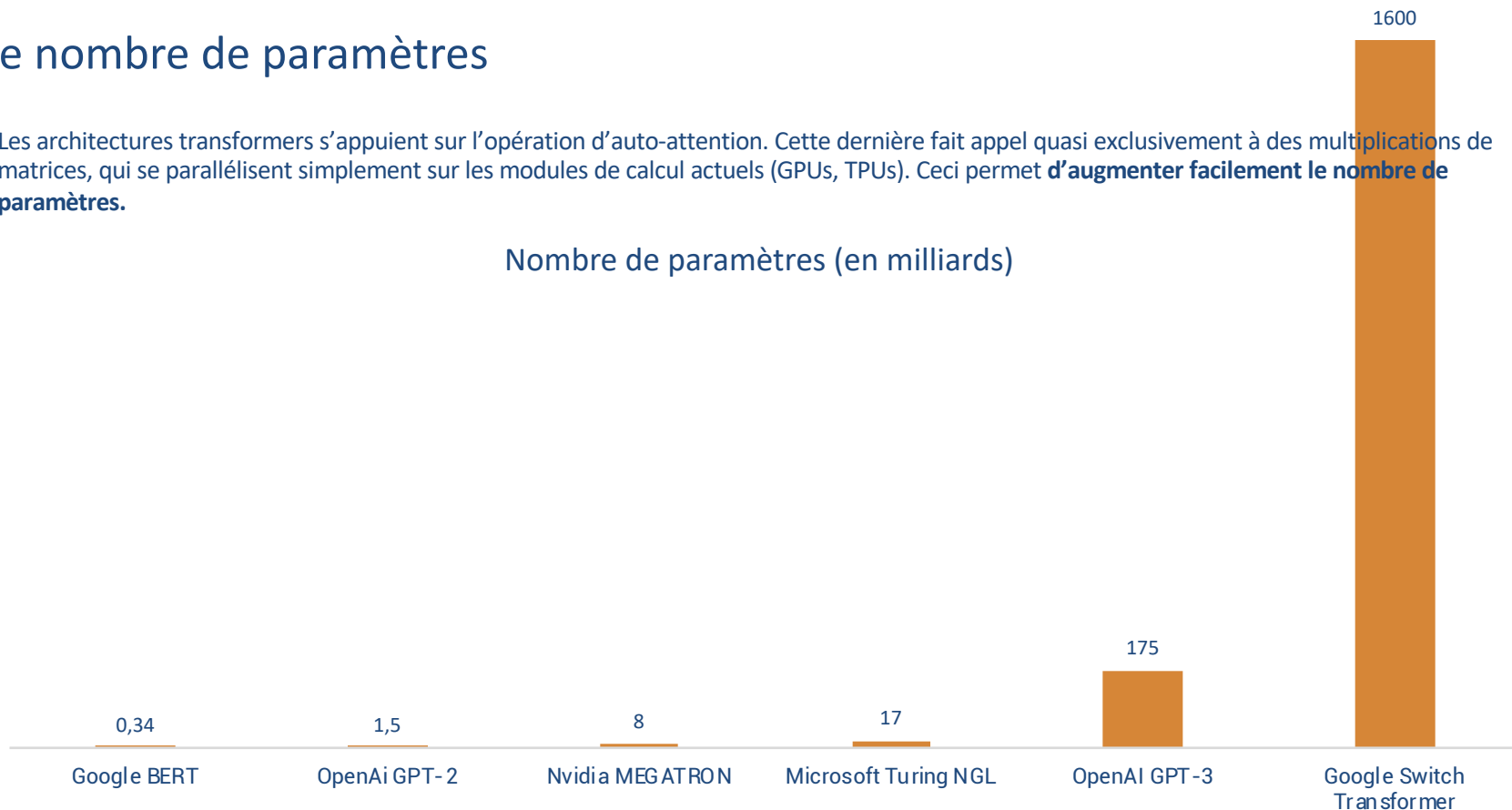
[4] <https://jalammar.github.io/illustrated-transformer/>

[5] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman: **GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding**. ICLR (Poster) 2019

Le nombre de paramètres

Les architectures transformers s'appuient sur l'opération d'auto-attention. Cette dernière fait appel quasi exclusivement à des multiplications de matrices, qui se parallélisent simplement sur les modules de calcul actuels (GPUs, TPUs). Ceci permet **d'augmenter facilement le nombre de paramètres**.

Nombre de paramètres (en milliards)



[1] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, Bryan Catanzaro: *Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism*. CoRR abs/1909.08053 (2019)

[2] Corby Rosset. 2020. Turing-NLG: *A 17-billionparameter language model by microsoft*. Microsoft Research Blog, 2:13.

[3] William Fedus, Barret Zoph, Noam Shazeer: *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity*. CoRR abs/2101.03961 (2021)

Les modèles pré-entraînés

L'utilisation des modèles pré-entraînés et des architectures à base de transformers ^[1] a significativement amélioré les performances des modèles de traitement automatique du langage (TAL).

Le pré-entraînement est une **tâche auto-supervisée comme un modèle de langue masqué**. Cette étape est effectuée une unique fois sur un corpus très large. **Le modèle peut ensuite être adapté sur différentes tâches** en mettant à jour les paramètres et en adaptant la dernière couche du réseau. Cette adaptation, bien que beaucoup plus simple que le pré-entraînement nécessite une modification du modèle spécifique à chaque tâche.

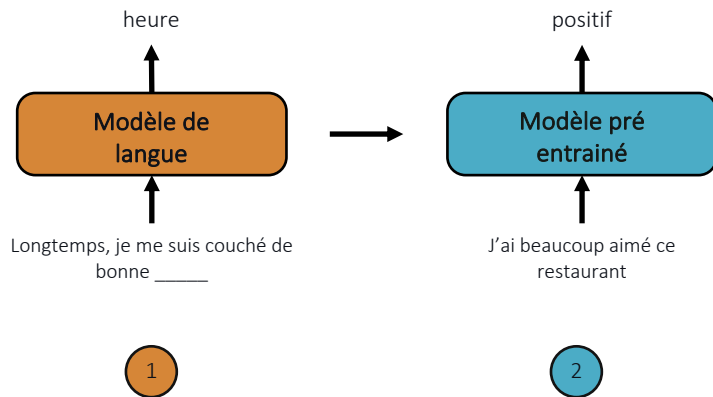


Figure : Par pré-entraînement puis apprentissage incrémental (fine-tuning)



[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin: *Attention is All you Need*. NIPS 2017: 5998-6008

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. NAACL-HLT (1) 2019: 4171-4186

Les tâches de pré-entraînement

Les modèles GPT [1, 2, 3] et BERT sont pré-entraînés sur des tâches de modèle de langue classique ou masqué.

GPT

Forward

Longtemps je me suis couché de bonne _____

BERT

Masked

Longtemps je me suis _____ de bonne heure

Figure : Tâche utilisées pour le pré-entraînement pour les modèles BERT et GPT.



[1] Radford, Alec and Wu, Jeff and Child, Rewon and Luan, David and Amodei, Dario and Sutskever, Ilya: *Language Models are Unsupervised Multitask Learners*. <https://openai.com/> 2019

[2] Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever: *Improving Language Understanding by Generative Pre-Training*. <https://openai.com/> 2018

[3] 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, Dario Amodei: *Language Models are Few-Shot Learners*. NeurIPS 2020

Les architectures transformers

Bilan

La « révolution NLP »

Les architectures transformers s'appuient sur l'**opération d'auto-attention**. Cette dernière nécessite presque exclusivement que des opérations de multiplications de matrices. Ce type d'opération se **parallélise facilement sur les modules de calcul récent (GPUs, TPUs)**.

Les transformers intègrent un **grand nombre de paramètres** qui semblent leur permettre d'obtenir de meilleures propriétés de généralisation

Ils sont **pré-entraînés sur de très larges corpus**. Cette procédure semble également augmenter leur capacité de généralisation. Les modèles peuvent ensuite être spécialisés sur des tâches spécifiques. Le **fine-tuning** correspond à un ajustement incrémental des poids du modèles pour répondre à la tâche. Le « **few-shot learning** » ne requiert pas de mise à jour des poids du modèle, on va juste conditionner la sortie du modèle à partir de la description de la tâche , d'un nombre réduit d'exemples et de celui pour lequel on cherche à inférer la réponse. Le « **zero-shot learning** » propose de conditionner la sortie du modèle simplement à partir de la description de la tâche et de l'exemple pour lequel on cherche à inférer la réponse.

Les limites des transformers

Les modèles transformers sont difficiles à interpréter

Ils reproduisent les biais des corpus de pré-entraînement trop volumineux pour être bien prétraités

Ils ont beaucoup de paramètres, ce qui les rend difficiles à déployer en pratique

3

Modèles génératifs

Les modèles génératifs

Les modèles GPT ne s'appuient que sur les mots précédents pour construire la représentation du mot courant. On peut donc les utiliser pour générer la suite d'un texte.

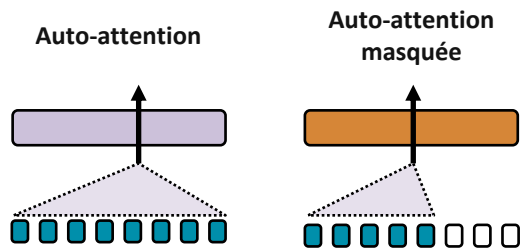
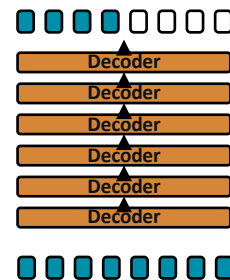


Figure : Comparaison entre l'auto attention et l'auto attention masquée

Longtemps, je me suis couché **de bonne heure.**



Longtemps, je me suis couché

Figure : Utilisation de GPT pour la génération de texte.

Formaliser les tâches pour les génératifs

Il est possible de formaliser les tâches pour les adapter aux propriétés génératives du modèle. En particulier, on va **formaliser chaque tâche sous la forme d'un problème de modèle de langue pour lequel il faudra générer la réponse**. Par exemple ici pour la classification.

Phrases	Labels	
J'ai adoré ce film	Positive	} TRAIN
J'ai détesté ce film	Négatif	
Je me suis ennuyé au cinéma	Négatif	
Incroyable !	Positive	
...		
	Positive	} TEST
Pas terrible.	?	

Figure : Jeu de donnée original

Phrases	
J'ai adoré ce film => Positif	} TRAIN
J'ai détesté ce film => Négatif	
Je me suis ennuyé au cinéma => Négatif	
Incroyable ! => Positif	
...	
Juste génial, je recommande ! => Positif	} TEST
Pas terrible. =>	

Figure : Jeu de donnée formalisé comme une tâche de modèle de langue

Conditionner la génération de texte

Il est même **possible de ne pas « apprendre » sur un jeu d'entraînement et d'utiliser le modèle pré-entraîné directement**. On inclue pour cela directement un certain nombre d'exemples dans le texte en entrée du modèle.

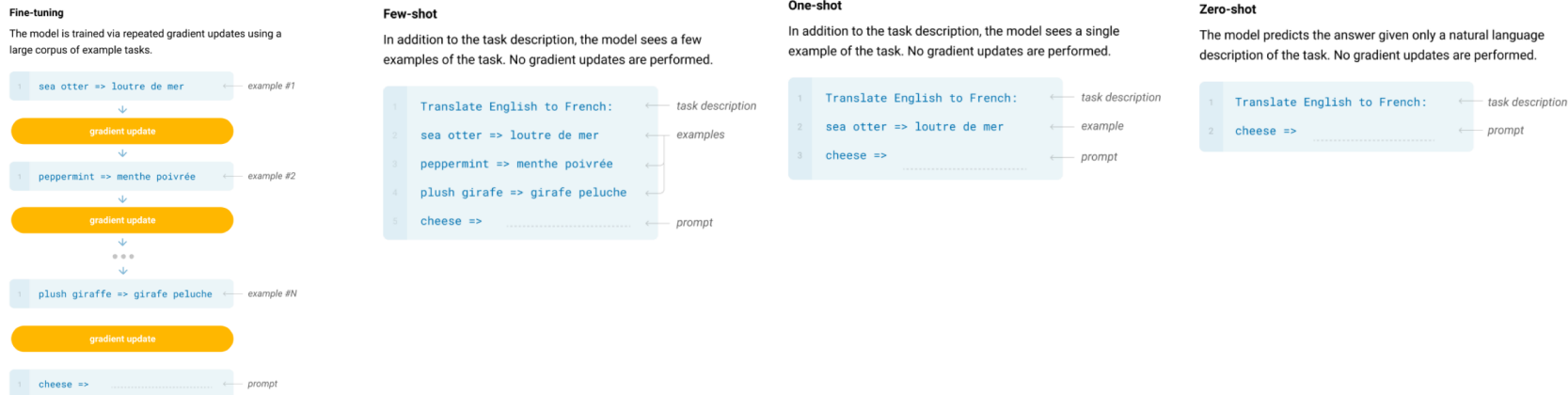


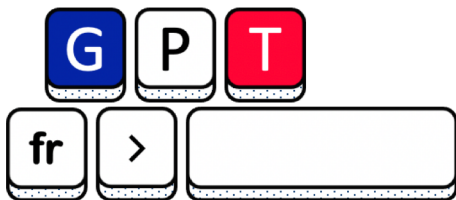
Figure : Comparaison entre : Zero-shot, one-shot et few-shot learning avec le fine-tuning traditionnel. Image extraite de [3].



[1] Radford, Alec and Wu, Jeff and Child, Rewon and Luan, David and Amodei, Dario and Sutskever, Ilya: *Language Models are Unsupervised Multitask Learners*. <https://openai.com/> 2019

[2] Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever: *Improving Language Understanding by Generative Pre-Training*. <https://openai.com/> 2018

[3] 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, Dario Amodei: *Language Models are Few-Shot Learners*. NeurIPS 2020



*Un modèle Transformer Génératif
Pré-entraîné pour le _____
français*

Introduction

Les modèles pré-entraînés en français

Des modèles de type *encodeur* comme **Flaubert** [3, 4] et **Camembert** [5] ou *encodeur/décodeur* comme **BARThez** [1] ont été développés pour le français. Des données sont également proposées spécifiquement pour le français : le benchmark **FLUE** [3, 4], ou le jeu de données **FQuAD** [6], **PIAF** [2]. Ces ressources permettent d'adresser et d'évaluer de nombreux cas d'usages en français, comme les systèmes de réponse ou résumé automatique, classification de texte ...

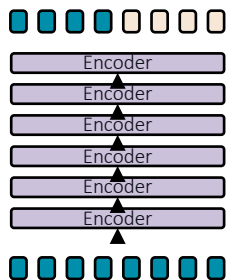


Figure : Architecture encodeur. Le modèle encode le texte en calculant un vecteur de représentation contextuel pour chaque token (FlauBERT, Camembert)

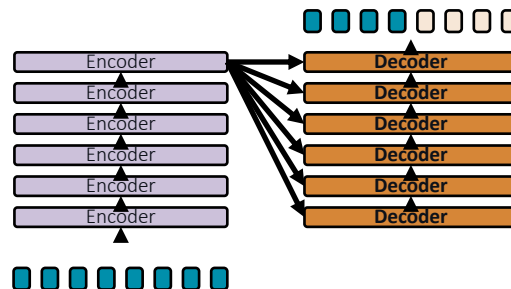


Figure : Architecture encodeur-décodeur autorégressifs. Le modèle encode le texte puis génère le texte correspondant (Barthez)



[1] Moussa Kamal Eddine, Antoine J.-P. Tixier, Michalis Vazirgiannis: **BARThez: a Skilled Pretrained French Sequence-to-Sequence Model**. CoRR abs/2010.12321 (2020)

[2] Rachel Keraron, Guillaume Lancrenon, Mathilde Bras, Frédéric Allary, Gilles Moyse, Thomas Scialom, Edmundo-Pavel Soriano-Morales, Jacopo Staiano: **Project PIAF: Building a Native French Question-Answering Dataset**. LREC 2020: 5481-5490

[3] Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, Didier Schwab: **FlauBERT: Unsupervised Language Model Pre-training for French**. LREC 2020: 2479-2490

[4] Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, Didier Schwab: **FlauBERT : des modèles de langue contextualisés pré-entraînés pour le français (FlauBERT : Unsupervised Language Model Pre-training for French)**. JEP-TALN-RECITAL (2) 2020: 268-278

[5] Louis Martin, Benjamin Müller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, Benoît Sagot: **CamemBERT: a Tasty French Language Model**. ACL 2020: 7203-7219

[6] Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, Maxime Vidal: **FQuAD: French Question Answering Dataset**. EMNLP (Findings) 2020: 1193-1208

Construction des corpus

Constitution du corpus d'entraînement (1/2)

Le pré-entraînement des modèles GPT nécessite un **important corpus**. Nous comparons ci-dessous les corpus que nous avons utilisés avec ceux utilisés pour l'entraînement des modèles anglais.

Modèles	 OpenAI GPT	 OpenAI GPT-2	 124M	 1B
# Documents ($\times 10^6$)	2,26 [†]	8,00	1,66	7,36
# Tokens ($\times 10^9$)	1,16 [†]	4,68 [†]	1,60	3,11
Moy. # tokens par document	512 [†]	585 [†]	965	422

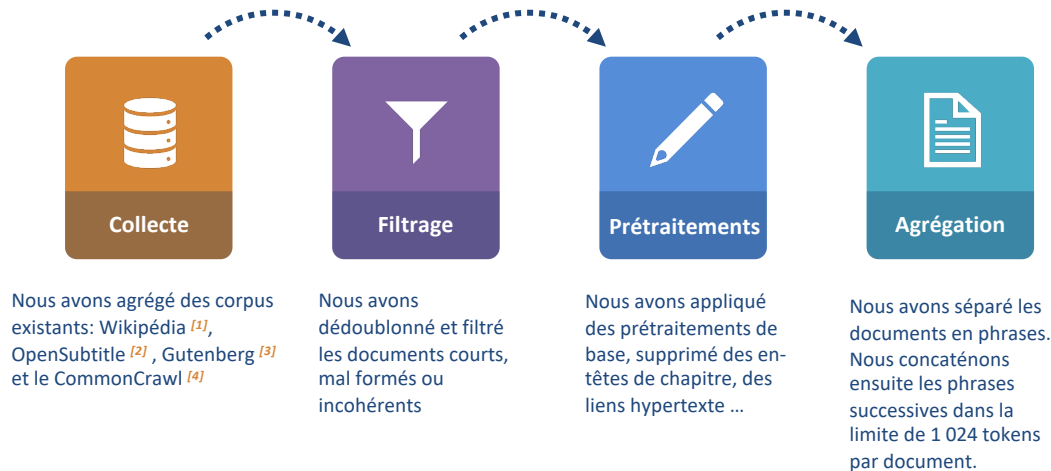
Table : Caractéristiques des corpus utilisés pour le pré-entraînement des modèles. Les données marquées [†] sont estimées à partir des caractéristiques disponibles. En particulier, pour le modèle OpenAI GPT on confond le nombre de tokens par document et la taille du contexte. Les caractéristiques du modèle OpenAI GPT-2 sont estimées à partir de l'échantillon proposé en accès libre.

Construction des corpus

Constitution du corpus d'entraînement (2/2)

Le pré-entraînement des modèles GPT nécessite un **important corpus constitué de longs documents**. Les corpus à l'échelle de la phrase et ne conservant pas l'ordre ne peuvent pas être utilisés directement. Nous avons donc agrégé des corpus spécifiques.

Pour l'entraînement de GPT_{fr}-124M, **nous avons agrégé des corpus existants : Wikipédia [1], OpenSubtitle [2] et Gutenberg [3]**. Les documents sont séparés en phrases. Les phrases successives sont ensuite concaténées dans la limite de 1 024 *tokens* par document. Le second corpus, utilisé pour l'entraînement de GPT_{fr}-1B **augmente le premier avec des données extraites du Common Crawl [4] en français**. Nous avons utilisé le modèle GPT_{fr}-124M pour filtrer les documents présentant une perplexité trop élevée.



[1] <https://dumps.wikimedia.org/frwiki/>

[2] <http://opus.nlpl.eu/download.php?f=OpenSubtitles/v2016/mono/>

[3] <http://www.gutenberg.org>

[4] <http://data.statmt.org/ngrams/deduped2017/>

Construction des corpus

Corpus pour l'évaluation des modèles de langues

Nous avons constitué **deux corpus d'évaluation d'un modèle de langue à partir de l'encyclopédie en ligne Wikipédia**. Pour cela nous avons collecté le texte des articles en français, labélisés « articles de qualité » et « bons articles ». Nous avons divisé aléatoirement les articles de qualité en des corpus d'entraînement/validation/test. **Ces articles ont été spécifiquement exclus du corpus utilisé pour le pré-entraînement de nos modèles.**

	WikiText-EN				WikiText-FR 			
	Valid	Test	Train-2	Train-103	Valid	Test	Train-35	Train-72
# Documents	60	60	600	28 475	60	60	2 126	5 902
# Tokens ($\times 10^6$)	218	246	2 089	103 227	896	987	35 166	72 961
Vocabulaire			33 278	267 735			137 589	205 403
Hors du vocabulaire			2,6%	0,4%			0,8%	1,2%

Table : Statistiques descriptives des corpus WikiText-FR. La taille du vocabulaire est évaluée en utilisant le tokenizer MOSES. La proportion de mots hors du vocabulaire correspond au nombre de tokens apparaissant moins de trois fois.

Modèles

Configuration du modèle à l'inférence

Apprentissage incrémental (fine-tuning)

Pour spécialiser le modèle, on peut **ajouter une couche et ajuster l'ensemble des paramètres de manière incrémentale** (en anglais, fine-tuning) en fonction des exemples $x_1 \dots x_m$ et des labels correspondants y d'une tâche spécifique.

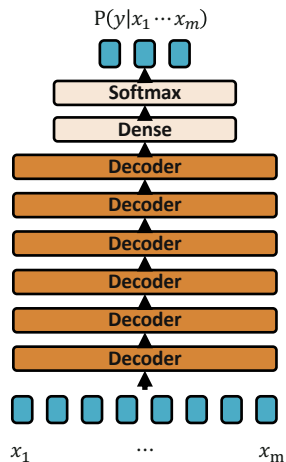


Figure : Architecture pour apprentissage incrémental

Apprentissage avec peu ou sans exemples

On peut également formaliser les tâches comme un modèle de langue : il n'est alors **pas nécessaire de modifier l'architecture du modèle**. Le modèle doit "générer" le label y comme la suite de la séquence $x_1 \dots x_m[SEP]$. Il est également **possible de résoudre la tâche sans mise à jour des poids du modèle** (apprentissage sans exemple).

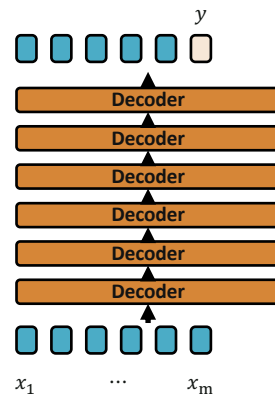


Figure : Architecture pour apprentissage avec peu ou sans exemples (few-shots, zero-shot learning en anglais)

Nous avons entraîné deux modèles. Nous avons proposé une architecture permettant de ne **pas utiliser de parallélisation du modèle** ^[1]. Les modèles utilisent un **vocabulaire de type bytewise encoding (BPE)** avec 50 000 unités ^[2] entraîné sur l'ensemble du corpus utilisé pour l'entraînement de GPT_{fr}-124M.

Modèles	 OpenAI GPT	 OpenAI GPT-2	 124M	 1B
Taille du contexte	512	1 024	1 024	1 024
# Couches	12	48	12	24
# Têtes d'attentions	12	25	12	14
Dimension des embeddings	768	1 600	768	1 792
# Paramètres ($\times 10^6$)	117	1 558	124	1 017

Table : Caractéristiques des architectures et comparaison avec les modèles OpenAI.



[1] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, Bryan Catanzaro: **Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism**. CoRR abs/1909.08053 (2019)

[2] Rico Sennrich, Barry Haddow, Alexandra Birch: **Neural Machine Translation of Rare Words with Subword Units**. ACL (1) 2016

L'entraînement de GPT_{fr}-124M a été effectué sur **un TPU v2-8** à partir de l'interface Google Colab. L'entraînement du modèle GPT_{fr}-1B a été mené en utilisant **le supercalculateur français Jean Zay**. Un cumul de 140 heures de calcul a été effectué sur du matériel de type Tesla V100 (TDP de 300W). Les émissions totales sont estimées à 580.61 kgCO₂eq. Ces travaux ont bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources 2020-AD011011823 attribuée par GENCI.

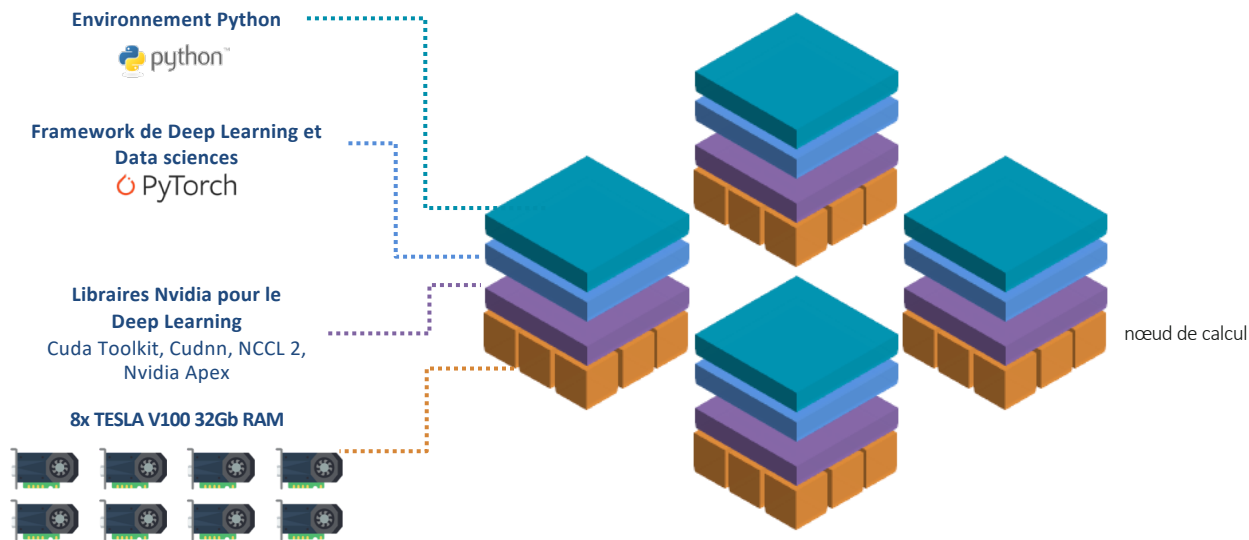


Figure : Infrastructure pour l'entraînement de GPT_{fr}-1B. L'entraînement a été distribué sur 4 nœuds de calcul de 8 GPUs. Nous avons utilisé de la parallélisation de données afin de diviser chaque micro batch sur les unités de calcul.

Evaluation

Modèle de langue

Le premier intérêt du modèle est de générer du texte cohérent. Pour évaluer la perplexité de nos modèles, nous avons utilisé les corpus d'évaluation wikitext-35 et 102 sans effectuer d'entraînement supplémentaire. Nous avons considéré un modèle de langue de type 5-grams avec lissage de kneser-ney ^[1] comme référence.

Modèles	Modèle 5-grams	 124M	 1B
WikiText-35-FR (ppl)	166,7	109,2	12,9
WikiText-72-FR (ppl)	99,1		

Table : Perplexité de nos modèles. Nous n'avons pas mis à jour les modèles sur le jeu d'entraînement et la perplexité est directement mesurée sur le jeu de test qui sont identiques pour deux benchmarks. Le modèle n-gram est entraîné sur les corpus d'entraînements correspondants. **Les approches ne sont pas directement comparables car la tokenization est différente et notre modèle est entraîné sur un volume de données beaucoup plus conséquent. Les résultats sont donc donnés à titre illustratif** mais soulignent la performance de notre modèle GPT_{fr}-1B.



[1] Hermann Ney, Ute Essen, Reinhard Kneser: *On structuring probabilistic dependences in stochastic language modelling*. *Comput. Speech Lang.* 8(1): 1-38 (1994) a service of Schloss Dagstuhl - Leibniz Center for Informatics

Evaluation

Benchmark FLUE

Les modèles génératifs ne permettent pas d'atteindre les mêmes performances qu'avec un modèle prenant en compte l'ensemble du contexte.
Nous avons tout de même comparé notre modèle sur le *benchmark* FLUE.

Modèles	CLS			PAWS-X	XNLI	Moy.
	Livres	DVD	Musique			
mBERT†	86,2	86,9	86,7	89,3	76,9	85,2
CamemBERT†	92,3	93,0	94,9	90,1	81,2	90,3
FlauBERT-base†	93,1	92,5	94,1	89,5	80,6	90,0
FlauBERT-large†	95,0	94,1	95,9	89,3	83,4	91,5
Nos modèles						
 124M	88,3	86,9	86,9	83,3	75,6	84,7
 1B	91,6	91,4	91,4	86,3	77,9	88,0

Table : Scores de précisions pour les tâches discriminatives du benchmark FLUE. Le symbole † désigne les scores rapportés de [1].



[1] Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, Didier Schwab: *FlauBERT: Unsupervised Language Model Pre-training for French*. LREC 2020: 2479-2490

Evaluation

Résumé automatique

Nous considérons la tâche de résumé automatique en utilisant la **configuration sans ajustement de l'architecture**. Nous rajoutons simplement le motif **"Pour résumer :"** après le texte original pour encourager le modèle à générer un texte qui résume les articles. Les poids du modèle ne sont donc pas mis à jour et aucune donnée d'entraînement n'est utilisée. Nous avons considéré le jeu de données OrangeSum^[1] pour le résumé abstratif. Nous comparons notre modèle à la référence qui consiste à considérer la première phrase du texte comme résumé. Nous comparons les métriques ROUGES. Dans cette configuration complexe, **nos modèles parviennent tout juste à s'approcher de la référence proposée**.

Modèles	Synthèse			Titre		
	R1	R2	RL	R1	R2	RL
Première phrase	22,1	7,1	15,3	18,6	7,7	15,0
 124M	17,5	3,1	12,1	13,9	2,3	9,7
 1B	16,6	3,4	11,5	10,2	2,6	8,4

Table : Comparaison des résumés générés avec le titre de l'article ou la synthèse proposée. Nous utilisons le score ROUGE et le corpus OrangeSum^[1]. Nos modèles sont utilisés en apprentissage sans exemple et donc sans mise à jour des paramètres sur le jeu d'entraînement



[1] Moussa Kamal Eddine, Antoine J.-P. Tixier, Michalis Vazirgiannis: *BARThez: a Skilled Pretrained French Sequence-to-Sequence Model*. CoRR abs/2010.12321 (2020)

Entrée : Alors que les cheminots entament ce dimanche 8 avril leur troisième jour d'une grève perlée, qui pourrait aller "au-delà" du mois de juin, le gouvernement monte au créneau. Alors que le Premier ministre Édouard Philippe réaffirme au Parisien que l'exécutif ira "jusqu'au bout" dans ses projets de réforme de la SNCF, Nicolas Hulot explique à travers une tribune publiée dans le Journal du Dimanche pourquoi la modernisation de l'entreprise ferroviaire sera bénéfique aux usagers et à l'environnement. Le ministre de la Transition écologique, ministre de tutelle des Transports, ne s'était pas encore publiquement exprimé sur cette réforme contestée. "Le train c'est écologique. C'est l'un des piliers de la mobilité du XXI^e siècle, et c'est parce qu'on aime le train qu'il faut réformer la SNCF", assure-t-il. Selon lui, le le gouvernement a "le devoir" de remettre la SNCF "sur des rails soutenables". "On ne peut pas rester indifférents face à la dégradation de ce service public, de ce bien commun", écrit le ministre. "Malgré les investissements, malgré les travaux, notre réseau est en mauvais état, la qualité du service se dégrade, les usagers en subissent les conséquences, alors qu'ils contribuent chaque année pour la SNCF, en dehors du prix des billets, à un effort de 14 milliards d'euros, bien plus que pour d'autres services publics", poursuit-il. "FAIRE MIEUX AVEC L'ARGENT QUE NOUS CONSACRONS AU TRAIN" Cette situation "donne le droit de poser des questions et nous donne le devoir de remettre l'entreprise sur des rails soutenables", plaide l'écologiste, soulignant que "l'objectif de cette réforme, c'est de faire mieux avec l'argent que nous consacrons au train". "C'est bien parce qu'on aime le train qu'il faut que ces évolutions aient lieu. Ne pas agir, ce serait trahir une histoire vieille de 100 ans. (...) On ne peut pas préparer l'avenir, réussir la transition écologique et répondre aux défis de la mobilité du quotidien avec 46 milliards de dette", estime Nicolas Hulot. "Il faut maintenant retrouver le chemin du dialogue. C'est ce que fait (la ministre des Transports, ndlr) Elisabeth Borne, à mes côtés, parce que c'est avec le train que nous construirons ensemble, une mobilité durable, sans pollution, pour tous", affirme le ministre. **Pour résumer :**

Sortie générée par le modèle : "La SNCF n'a pas vocation à se substituer aux autres modes de transport, elle doit être au service des usagers, des territoires et des générations futures". Par Europe1.

Référence: TRIBUNE. Dans une tribune publiée dans Le Journal du Dimanche, le ministre de la Transition écologique assure que le gouvernement a "le devoir" de remettre la SNCF "sur des rails soutenables".

Open Source et utilisation

Le modèle peut être utilisé simplement en utilisant la librairie 🤗 **Transformers** [1].

```
from transformers import GPT2Tokenizer, GPT2LMHeadModel

# Load pretrained model and tokenizer
model = GPT2LMHeadModel.from_pretrained("asi/gpt-fr-cased-base")
tokenizer = GPT2Tokenizer.from_pretrained("asi/gpt-fr-cased-base")

# Generate a sample of text
model.eval()
input_sentence = "Longtemps je me suis couché de bonne heure."
input_ids = tokenizer.encode(input_sentence, return_tensors='pt')

beam_outputs = model.generate(
    input_ids, max_length=100, do_sample=True, top_k=50, top_p=0.95,
    num_return_sequences=1 )

print("Output:\n" + 100 * '-')
print(tokenizer.decode(beam_outputs[0], skip_special_tokens=True))
```



[1] <https://huggingface.co/asi/gpt-fr-cased-base>

Entrée : Longtemps je me suis couché de bonne heure.

Sortie générée par le modèle : La nuit se faisait assez obscure, les étoiles semblaient dormir de temps à autre, cependant le peu de clarté qui me permit de marcher la nuit dans la chambre éveilla en moi une profonde pitié de ce que je faisais là, et cette peur de l'avenir me devint plus vive et plus aiguë. Je me réveillai en sursaut, comme un homme ivre qui s'endort en repassant sa bière. Je passai rapidement le seuil de la porte, et un homme entra, vêtu d'un long manteau d'homme et portant un paquet à la main. Je lui demandai la permission d'entrer. Il me répondit qu'il viendrait demain, et qu'il ne pouvait pas attendre, à cause des travaux de la nuit. Il était habillé simplement, et l'homme me dit qu'il ne voulait pas me donner d'argent; seulement, il voulait que je m'habillasse, pour aller dans le même train

Entrée : La presse française dézingue les Bleus, l'Europe ironise sur le penalty manqué par Mbappé ou encore l'Espagne en folie après sa qualification pour les quarts,

Sortie générée par le modèle : elle juge "scandaleux" le but du Ghana d'Antoine Griezmann à l'Euro-2008. Une avalanche de commentaires. Mais les images ont un effet ravageur sur le cœur du public français. En France, c'est déjà un record: selon Médiamétrie, le pic d'audience depuis la défaite en finale de Coupe du monde du football français à l'Euro-2008 a été battu, plus de 40% de l'audience n'a pas franchi la barre des 10 000 fans. Une chute impressionnante que les joueurs ne se sont pas privés de fustiger. Certains ont même trouvé que l'équipe de France avait perdu son âme. >> Lire aussi: Griezmann, Benzema et Griezmann au Mondial-2010: le plus petit écart en matière de nombre de fans France-Brésil-Argentine: les Bleus

Conclusion et travaux futurs

Nous proposons une **version française du modèle GPT**. S'il n'égale pas les performances brutes de BERT, ses propriétés génératives permettent de l'utiliser dans des configurations remarquablement flexibles. Comme illustré dans nos expérimentations pour le résumé automatique, l'utilisation d'une configuration sans apprentissage reste très difficile pour le modèle. Cette configuration ouvre néanmoins des perspectives différentes d'un apprentissage traditionnel.

Nous espérons que les **performances de modèles de langues obtenues favoriseront son utilisation pour des cas d'usages correspondants**.

5

Applications et Démo !

Annexes

L'architecture de GPT s'appuie sur **plusieurs couches successives de décodeurs** telles que définies dans l'architecture transformers. Le modèle applique une **attention multi-têtes uniquement sur les tokens qui précèdent la position considérée**.

$$h_0 = UW_e + W_p$$

$$h_i = \text{transformeur_decodeur}(h_i) \quad \forall i \in [1, n]$$

$$P(u|U) = \text{softmax}(h_n W_e^T)$$

Le modèle est **pré-entraîné selon une tâche de modèle de langue**. Étant donné un corpus d'entraînement décrit comme une séquence de *tokens* $U = \{u_1 \cdots u_n\}$, on optimise les paramètres Θ du modèle pour maximiser la log-probabilité suivante :

$$\mathcal{L}(U) = \sum_i \log P(u_i | u_{i-k} \cdots u_{i-1}; \Theta)$$

Avec k la taille de la fenêtre de contexte, $U = \{u_{-k} \cdots u_{-1}\}$ le vecteur d'embeddings des tokens du contexte, n le nombre de couches, W_e la matrice d'embeddings et W_p la matrice d'embeddings positionnels.

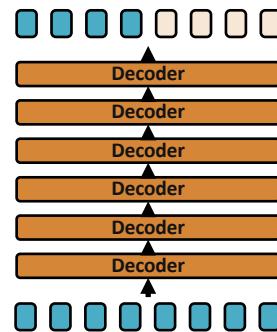


Figure : Architecture décodeur autorégressif. Le modèle génère la suite d'un texte.

Nous avons **gardé le même paramétrage pour les deux modèles**. Le taux d'apprentissage est fixé à $1.5e^{-4}$ avec une phase d'échauffement (en anglais, *warm up*) de 2 000 itérations puis une décroissance (en anglais, *decay*) selon une fonction cosinus. Nous avons effectué **125 000 itérations** en utilisant une taille de batch de 128 documents et la précision mixte. Nous avons conservé 6,080 documents pour constituer un jeu de validation. On peut suivre l'évolution de la perplexité sur ce jeu de validation.

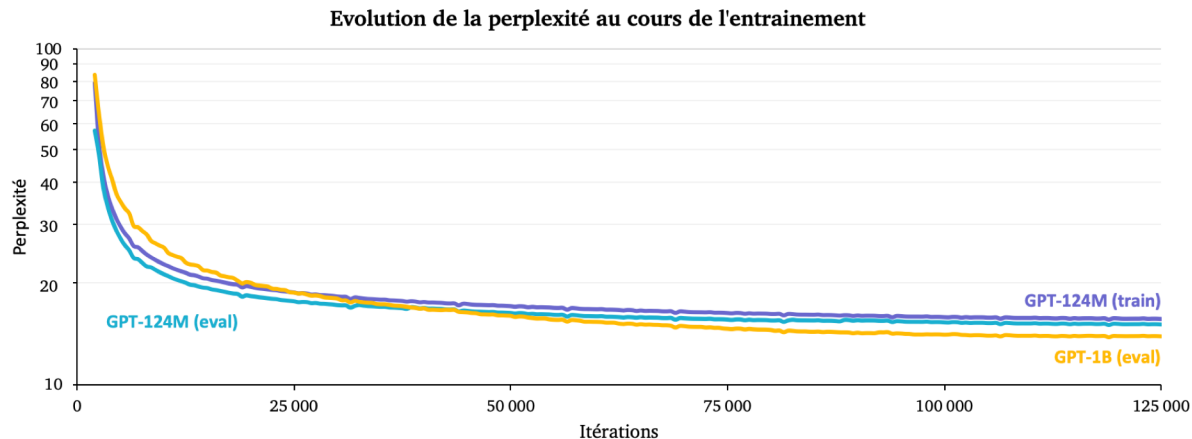


Figure : Evolution de la perplexité lors du pré-entraînement du modèle. Nous mesurons la perplexité sur un jeu de validation commun pour les deux modèles.

Construction des corpus

Constitution du corpus d'entraînement

Distribution de la longueur des documents.

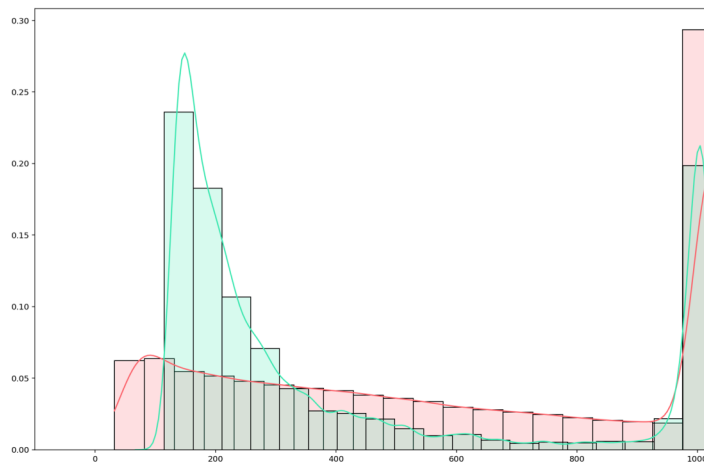


Figure : Distribution de la longueur des documents. Comparaison avec le corpus de OpenAI GPT-2 (en rouge) et notre corpus (en vert).