Improving language models by retrieving from trillions of tokens

Séminaire TALia du 18/02/2022

Source

Computer Science > Computation and Language

[Submitted on 8 Dec 2021 (v1), last revised 7 Feb 2022 (this version, v3)]

Improving language models by retrieving from trillions of tokens

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, Laurent Sifre

Comment découpler l'apprentissage de la linguistique des informations relatives à la connaissance du monde ?

The Dune film was released in *1965*

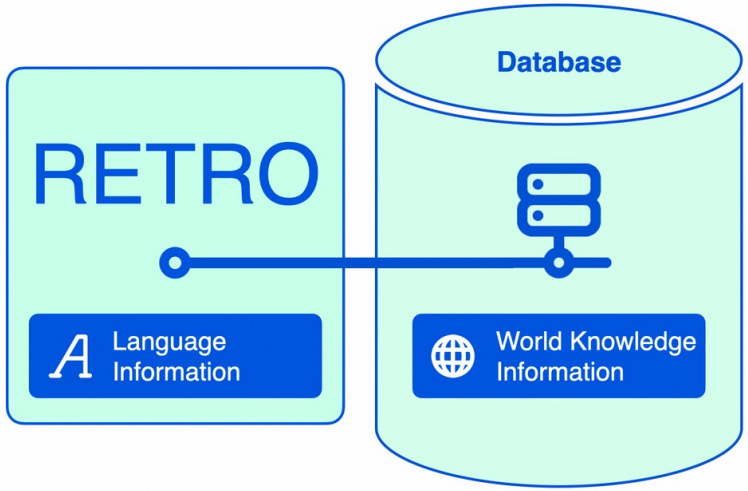
Information factuelle

its popularity spread by word-of-mouth to allow Herbert to start working full *time*

Information linguistique

Différences entre RETRO et GPT-3

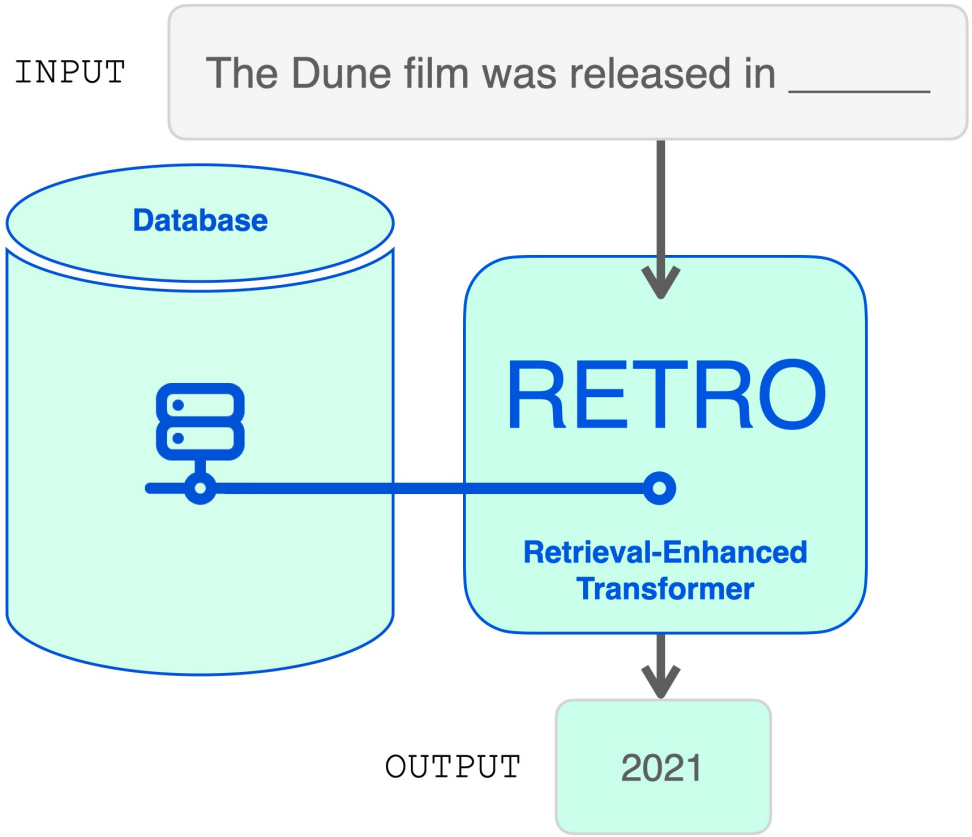
- L'architecture RETRO est un *transformer* classique (*encoder-décoder*) dont la séquence d'entrée (les mots à l'input du décodeur) est enrichie grâce à une base de données textuelles.
- L'architecture de GPT-3 est un modèle de langue du type *transformer* avec seulement une partie *decoder*.
- GPT-3 possède 175 milliards de paramètres contre 7,5 milliards pour RETRO (à performance égale).
- Cette diminution du nombre de paramètres est expliquée par l'information *factuelle* extraite de la base de données par RETRO.



RETRO à un haut niveau

3 étapes :

- On détermine les séquences de mots les plus proches de l'input dans la base de données (L_2 distance entre les *embeddings* de l'*input* / bdd).
- On encode ses plus proches voisins pour obtenir les matrices de *Key*/*Value* dans la partie *encoder*.
- On utilise les *Keys*/*Values* de l'*encoder* à certaines couches particulière du *decoder* (*Chunk Cross Attention)* pour ajouter de l'information.

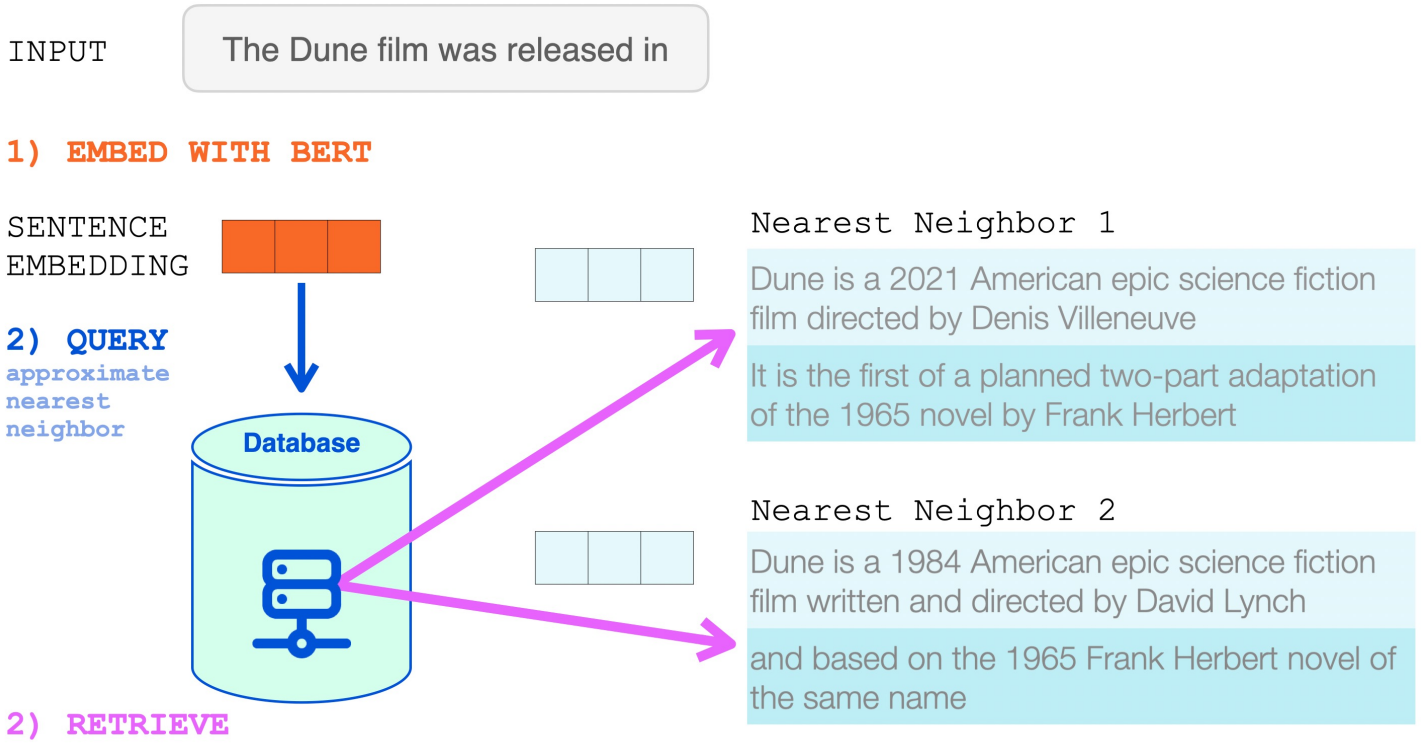


Zoom sur la partie base de données

- Base de données = dictionnaire contenant $2 * 10^{12}$ mots extraits du jeu d'entraînement.
- Les valeurs du dictionnaire sont composées de deux tronçons : le **voisin** et la suite de la phrase du voisin (**complétion**).
- On calcule grâce à BERT l'*embedding* du **voisin** (*sentence embedding*) qui sera la clé du dictionnaire.

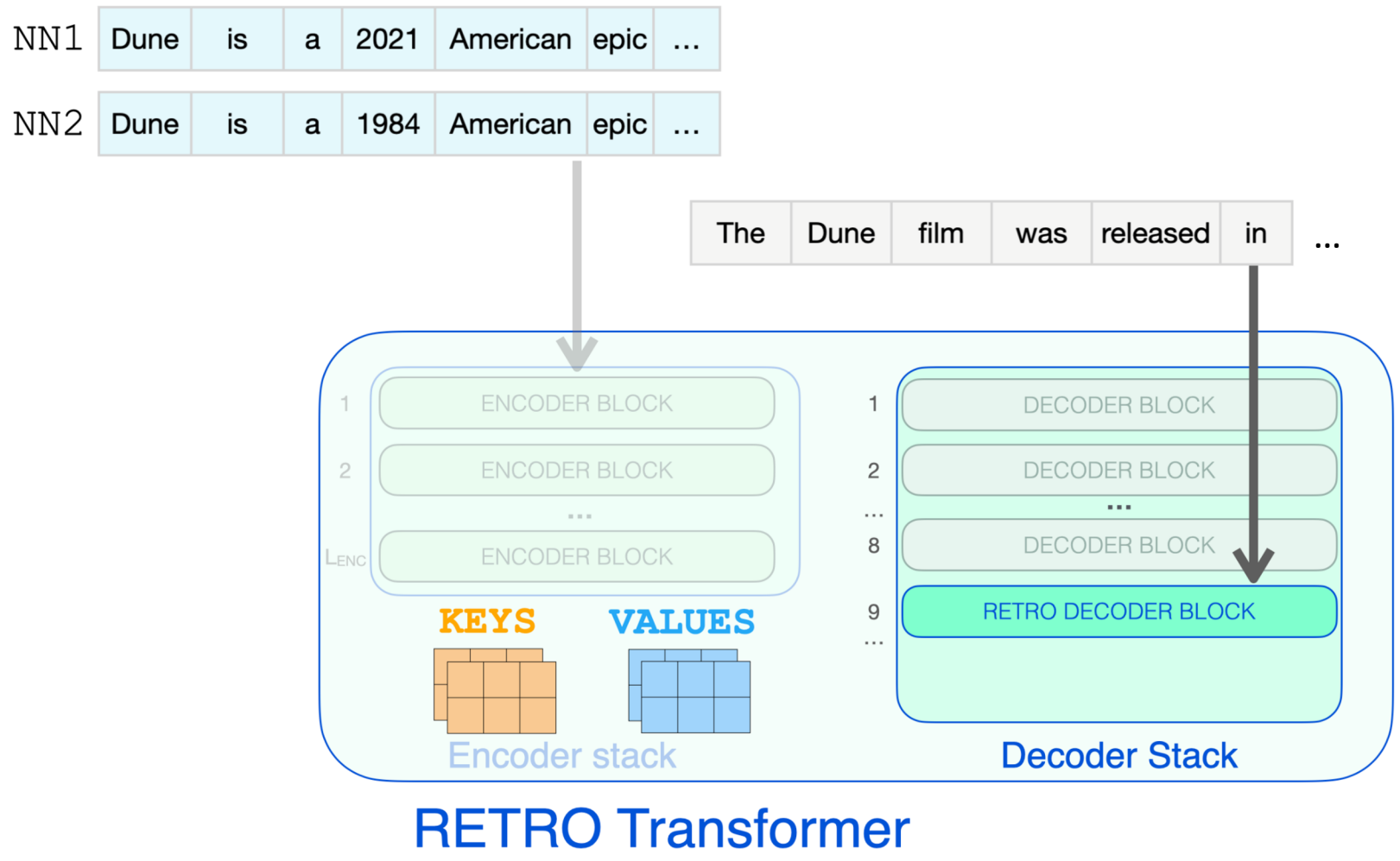
Key (BERT sentence embedding)	Value (text. neighbor and completion chunks. Each up to 64 tokens in length)	
	Dune is a 2021 American epic science fiction film directed by Denis Villeneuve	NEIGHBOR
	It is the first of a planned two-part adaptation of the 1965 novel by Frank Herbert	COMPLETION
	Dune is a 1965 science fiction novel by American author Frank Herbert	NEIGHBOR
	originally published as two separate serials in Analog magazine	COMPLETION
...	...	

Zoom sur la partie base de données



- On utilise BERT dont les paramètres sont figés pour ne pas ralentir la prédiction.
- On calcule $d(C, N) = ||\textsc{Bert}(C) - \textsc{Bert}(N)||_2^2$ où *C* est l'entrée et *N* le voisin.
- On prend les *k* plus proches voisins avec leur complétion qui enrichiront l'input.

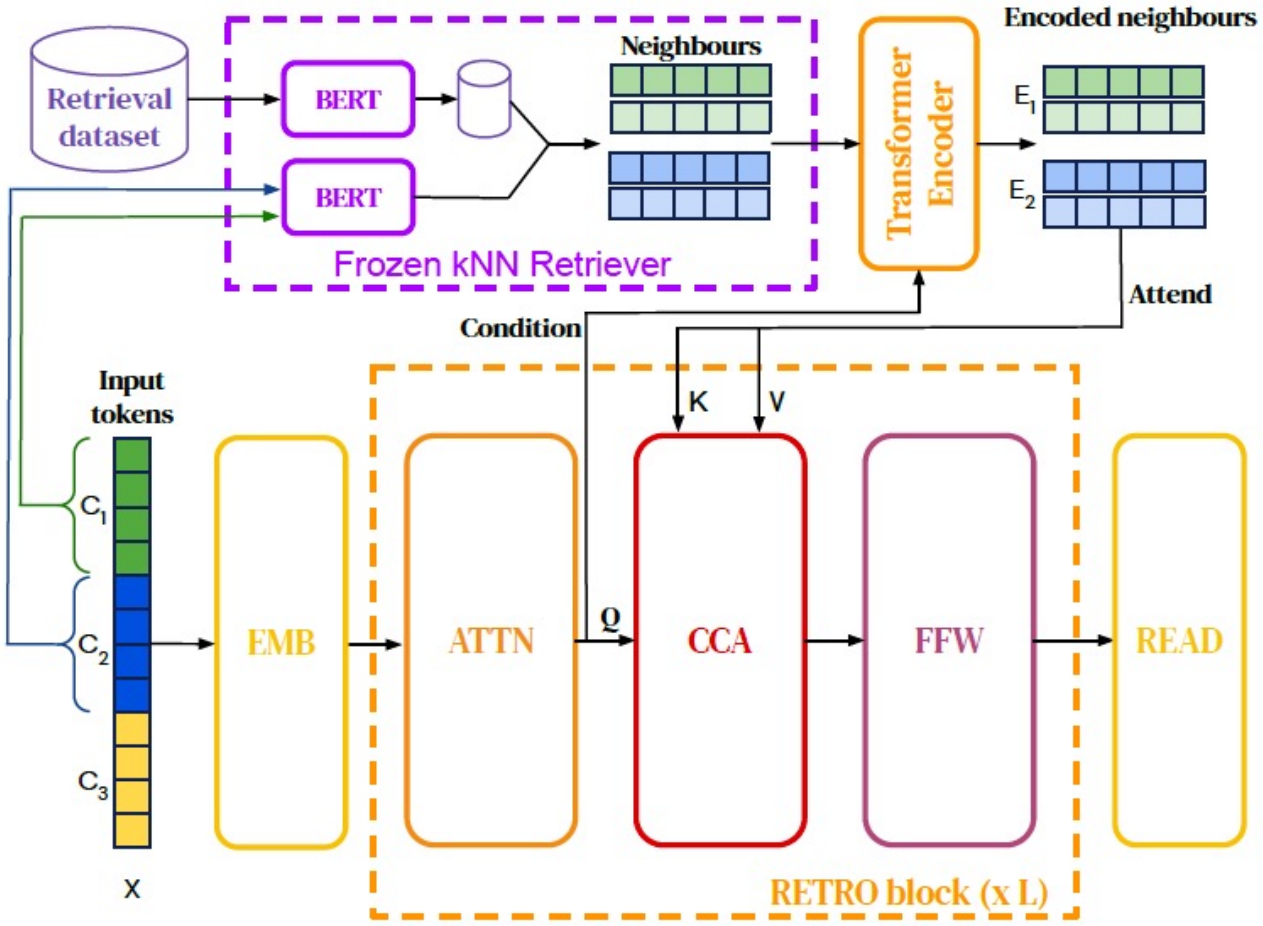
Zoom sur la partie transformer



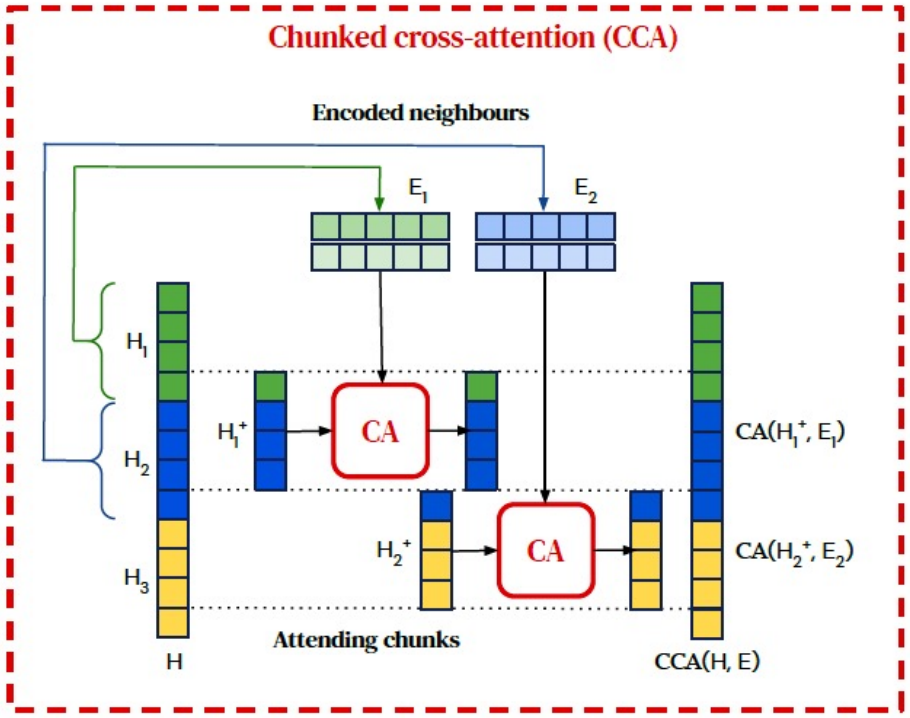
- La partie *decoder* est composée de couches classiques (*masked self-attention* + FFNN) mais aussi de couches particulières pour intégrer l'information des voisins.

La partie *encoder* est classique : plusieurs têtes d'attention (*Self Attention* + FFNN) pour construire les matrices de *Keys* et *Values* des voisins.

RETRO decoder block

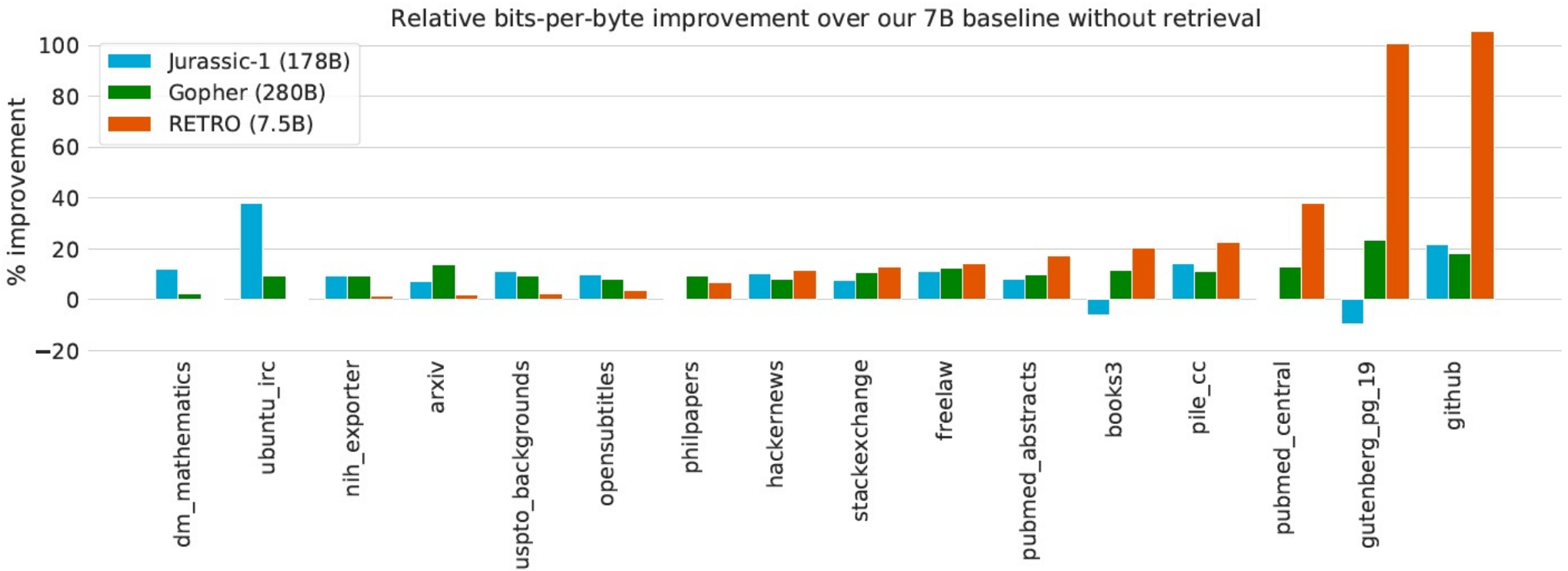


Chunked cross-attention

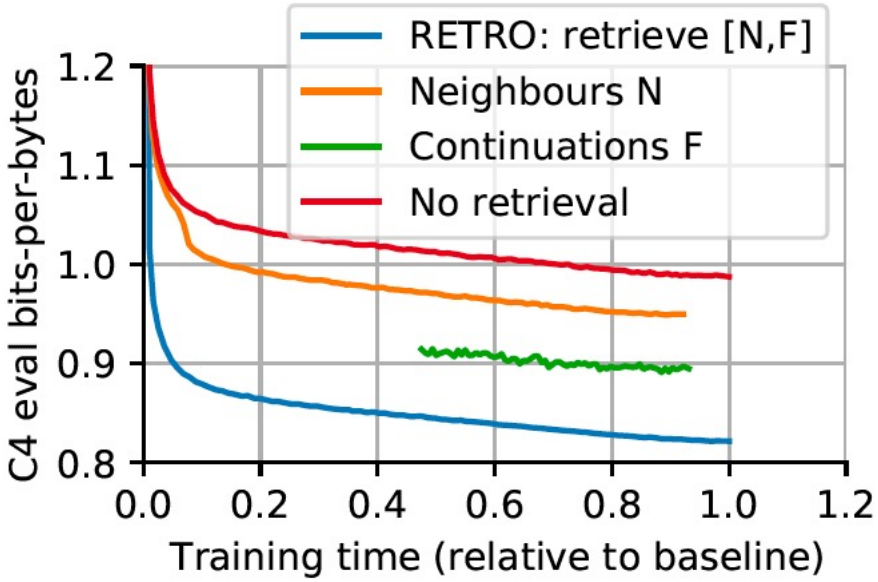


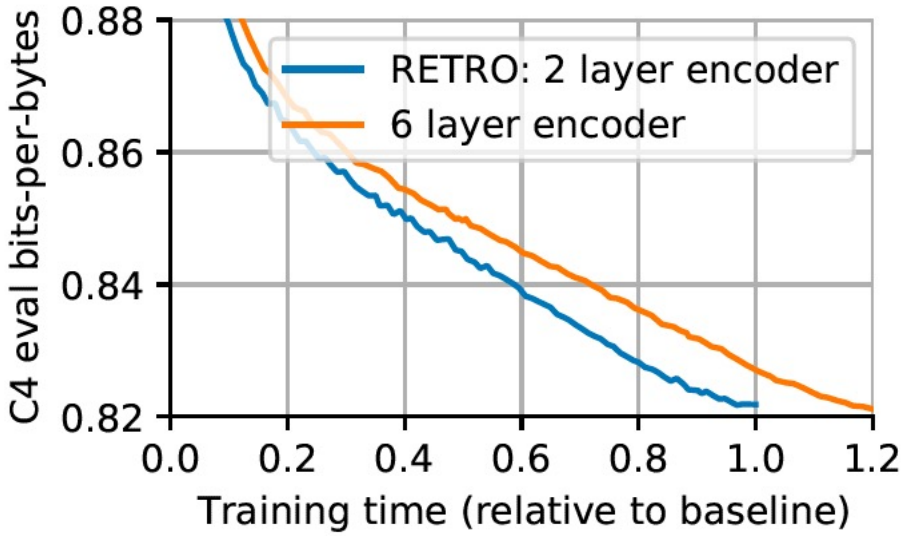
- On découpe l'*input* en tronçons pour déterminer les voisins.
- On calcule la *cross-attention* entre les tronçons décalés (vecteurs résultants des couches intermédiaires de chaque mots du decoder) et les voisins + complétions.
- On préserve bien l'auto-régressivité car la *cross-attention* n'est calculée qu'avec les mots précédents.

Résultats de RETRO



Utilité des **voisins/complétions** sur le modèle





Conclusion

- Les chercheurs de DeepMind réussissent de fait à entraîner un modèle 10 fois plus petit avec les mêmes résultats que les *Large Language Models*. C'est une voie prometteuse pour travailler avec de plus petits modèles qui restent performant.

- Est-ce qu'ils arrivent vraiment à découpler l'information factuelle de la linguistique ? Je ne suis pas sûr et il n'y a pas de discussion poussée des cas où le modèle différemment (ils se sont surtout attachés à éviter le reproche de data leakage).