Vers une optimisation de RAG en français : conception d'un reranker open source, fine-tuning et évaluation

Ying ZHANG, Matthieu PETIT GUILLAUME, Aurélien KRAUTH

Leviatan

APIA 2025 - Dijon
01/07/2025

LEVIATAN

LEVIATAN

LEVIATAN

Introduction - Contexte et Problématique

- Système RAG (Retrieval-Augmented Generation) minimal
  - **Retriever** (bi-encoder [14] + similarité) : interroge une base de connaissances pour identifier les passages pertinents.
  - **Générateur** : (ex. : GPT [12], LLaMA [10]) produit la réponse à partir des documents sélectionnés.

LEVIATAN

Introduction - Contexte et Problématique

- Système RAG (Retrieval-Augmented Generation) minimal
  - **Retriever** (bi-encoder [14] + similarité) : interroge une base de connaissances pour identifier les passages pertinents.
  - **Générateur** : (ex. : GPT [12], LLaMA [10]) produit la réponse à partir des documents sélectionnés.
- Limites du bi-encoder
  - Sensibilité aux biais du modèle : certaines nuances de la requête ne sont pas captées [6]
  - Scores de pertinence approximatifs, en particulier pour des requêtes complexes [6]
    $\Longrightarrow$ Extraction de nombreux passages $\Longrightarrow$ Coût élevé

LEVIATAN

Introduction - Contexte et Problématique

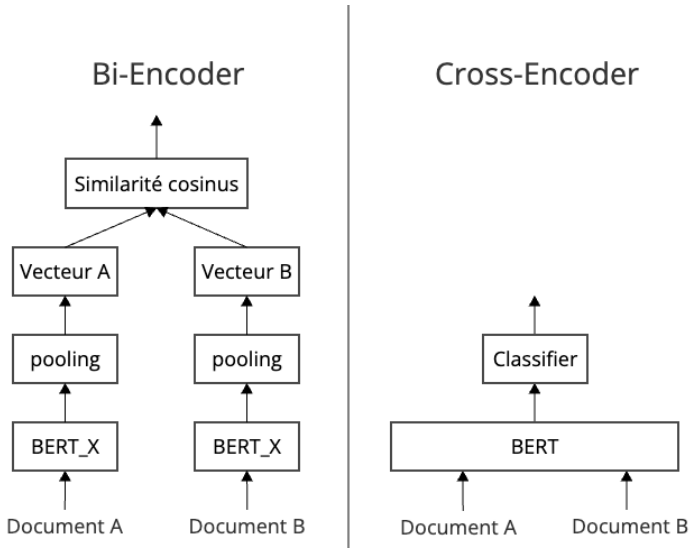
- Système RAG (Retrieval-Augmented Generation) minimal
  - **Retriever** (bi-encoder [14] + similarité) : interroge une base de connaissances pour identifier les passages pertinents.
  - **Générateur** : (ex. : GPT [12], LLaMA [10]) produit la réponse à partir des documents sélectionnés.
- Limites du bi-encoder
  - Sensibilité aux biais du modèle : certaines nuances de la requête ne sont pas captées [6]
  - Scores de pertinence approximatifs, en particulier pour des requêtes complexes [6]
    $\Longrightarrow$ Extraction de nombreux passages $\Longrightarrow$ Coût élevé
- Solution proposée : introduction d'un **Reranker** [11, 14]

LEVIATAN

Introduction - Cross-Encoder et objectif du projet

Notre choix : Cross-Encoder [11]

- Capture fine des interactions sémantiques
- Exécutable sur CPU
- Récupération en deux étapes
  - Réordonne les passages extraits
  - Améliore la pertinence
  - Réduit la charge computationnelle par le générateur



Bi-Encoder vs Cross-Encoder [14]

LEVIATAN

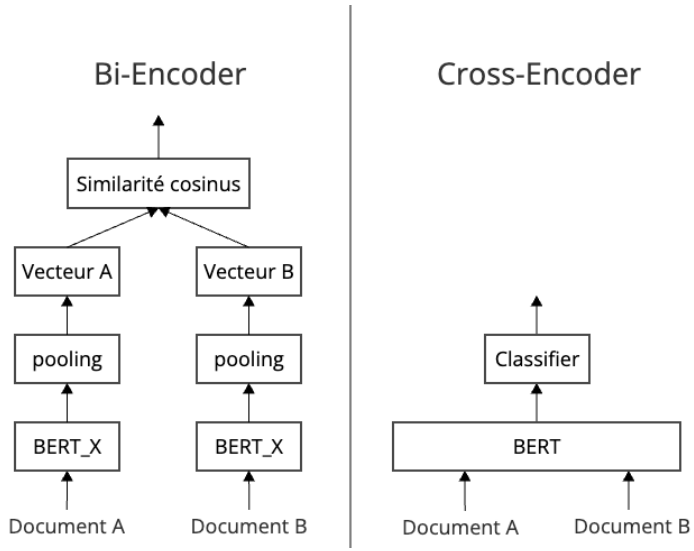
Introduction - Cross-Encoder et objectif du projet

Notre choix : Cross-Encoder [11]

- Capture fine des interactions sémantiques
- Exécutable sur CPU
- Récupération en deux étapes
  - Réordonne les passages extraits
  - Améliore la pertinence
  - Réduit la charge computationnelle par le générateur

Objectif du projet

- Construire un reranker français
- Rivaliser avec des solutions commerciales
- Rendre le code, les données et les modèles accessibles



Bi-Encoder vs Cross-Encoder [14]

LEVIATAN

LEVIATAN

Composition du jeu de données

Deux types de données utilisées pour l'entraînement :

- **1. Données de similarité sémantique**
  - `STS Benchmark (STSB)` [1] - corpus multilingue pour évaluer la similarité entre des paires de phrases.
- **2. Données de questions-réponses en français**
  - `PIAF` [7] - Projet d'IA francophone pour les Q/R
  - `FQuAD` [9] - French Question Answering Dataset
  - `SQuAD-Fr` [5] - version traduite de SQuAD
  - `pandora-rag-fr` [13] - corpus de Q/R traduit automatiquement, filtré pour le français

LEVIATAN

Préparation des jeux de données

Deux pipelines de traitement distincts :

1. STS Benchmark (STSB) - Similarité sémantique
   - Extraction de la portion en langue française
   - Normalisation des scores dans l'intervalle `[0, 1]`
2. Données de questions-réponses (PIAF, FQuAD, etc.)
   1. **Filtrage linguistique** : uniquement les exemples en français
   2. **Paires positives (`label = 1`)** : `(contexte, question)` contenant la réponse
   3. **Paires négatives (`label = 0`)** : question provenant d'un autre contexte
   4. **Fusion** : structuration unifiée avec colonnes - `contexte`, `question`, `label`, `source`
   5. **Split train/test** : respect du découpage original (sauf PIAF : 70%/30%)

LEVIATAN

Fine-tuning - Combinaison données / modèles

Quatre configurations testées :

Expérience	Modèle	Jeu de données
Exp. 1	`BERT-base-uncased [4]`	`STSB-FR`
Exp. 2	`distilroberta-base [15]`	`PIAF, FQuAD, SQuAD-Fr`
Exp. 3	`distilroberta-base`	`PIAF, FQuAD, SQuAD-Fr, pandora-rag-fr`
Exp. 4	`distilroberta-base`	`PIAF, FQuAD, SQuAD-Fr, pandora-rag-fr, STSB-FR`

Objectif : évaluer l'impact du modèle de base et de la diversité des données sur la performance du reranker.

LEVIATAN

Fine-tuning - Configuration

Infrastructure : GPU `NVIDIA T4` pour le fine-tuning
Inférence et évaluation effectuées sur CPU

Paramètres d'entraînement :

- `EPOCHS` : 4
- `TRAIN_BATCH_SIZE` : 16
- `SAVE_BEST_MODEL` : True
- `WARMUP_STEPS` : 10% des données d'entraînement
- Autres hyperparamètres : *valeurs par défaut*

LEVIATAN

LEVIATAN

Évaluation des apprentissages et choix de l'évaluateur

Objectif : Vérifier si le modèle a bien appris à capturer la similarité sémantique, selon la nature des données.

Choix de l'évaluateur :

- **Données avec scores continus** (ex. STSB-FR) :
  - On mesure la qualité des représentations par la **corrélation** avec les scores humains
  - Métriques : `Pearson`, `Spearman` → **CorrelationEvaluator** [16]
- **Données binaires** (questions-réponses) :
  - On vérifie si le modèle distingue bien des paires pertinentes ou non
  - Métriques : `Accuracy`, `F1 score` → **BinaryClassificationEvaluator** [16]

LEVIATAN

Analyse des résultats : corrélation basée sur STSB-FR

Exp.	STSB-FR Dev		STSB-FR Test	
	Pearson	Spearman	Pearson	Spearman
Exp. 1	0,8722	0,8692	**0,8362**	**0,8245**
Exp. 2	0,4710	0,6119	0,3492	0,4673
Exp. 3	0,5258	0,6990	0,4225	0,5908
Exp. 4	**0,9219**	**0,9187**	0,7565	0,7460

- **Exp. 1** : BERT-base + STSB-FR Train → bons résultats sur les deux sets (Dev et Test).
- **Exp. 2 à 4** : DistilRoBERTa-base + Q/R (+ STSB-FR pour Exp. 4)
- L'ajout de STSB-FR dans **Exp. 4** améliore fortement la corrélation sur le Dev Set, mais les résultats sont moins bons sur le Test Set que ceux d'Exp. 1.

LEVIATAN

Analyse des résultats : Précision sur les jeux de questions-réponses

Jeu de test combiné : PIAF, FQuAD, SQuAD-Fr

Exp.	Acc.	F1	Prec.	Rec.	Avg. Prec.
Exp. 1	0,9527	0,9529	0,9455	0,9603	0,9889
Exp. 2	**0,9765**	**0,9765**	**0,9785**	0,9745	0,9931
Exp. 3	0,9763	0,9762	0,9769	0,9756	**0,9955**
Exp. 4	0,9753	0,9754	0,9720	**0,9788**	0,9954

- Tous les modèles donnent de bons résultats.
- Les expériences 2 à 4 donnent des performances très proches.
- Exp.1 est moins performante car entraînée uniquement sur STS-B (similarité), alors que le test est 100
- **Prochaine étape :** utiliser ces modèles comme rerankers dans un contexte réaliste.

LEVIATAN

LEVIATAN

Benchmark - Configuration

- **Objectif :** comparer nos rerankers (Exp. 1 à 4) avec :
  - **Cohere Reranker** (modèle commercial) [2]
  - **D.V. CrossEncoder CamemBERT Large** (open source) [3]

LEVIATAN

Benchmark - Configuration

- **Objectif :** comparer nos rerankers (Exp. 1 à 4) avec :
  - **Cohere Reranker** (modèle commercial) [2]
  - **D.V. CrossEncoder CamemBERT Large** (open source) [3]
- **Jeux de test utilisés :**
  - **FQuAD Test** - 3188 questions
  - **PIAF Test** - 1151 questions (découpé 70% / 30% car pas de split fourni)

LEVIATAN

Benchmark - Configuration

- **Objectif :** comparer nos rerankers (Exp. 1 à 4) avec :
  - **Cohere Reranker** (modèle commercial) [2]
  - **D.V. CrossEncoder CamemBERT Large** (open source) [3]
- **Jeux de test utilisés :**
  - **FQuAD Test** - 3188 questions
  - **PIAF Test** - 1151 questions (découpé 70% / 30% car pas de split fourni)
- **Pipeline de benchmark :**
  - **Étape 1 - Retriever (bi-encoder) :**
    - `intfloat/multilingual-e5-large` (Microsoft) [17]
    - Mesure de similarité : cosinus
    - 30 premiers passages sélectionnés
  - **Étape 2 - Reranker :**
    - Classement des 30 passages candidats
    - Métrique : **Recall@k (Rappel@k)** (k = 5, 7, 10)

LEVIATAN

Benchmark - Résultats obtenus

FQuAD Test - Recall@k

Modèle	Recall@5	Recall@7	Recall@10
Cohere Reranker	**92,50%**	**93,48%**	**94,26%**
D.V. CrossEncoder	52,23%	62,33%	71,46%
Exp. 1	**84,54%**	**87,92%**	**90,90%**
Exp. 2	30,14%	38,33%	49,53%
Exp. 3	39,12%	47,33%	56,81%
Exp. 4	**72,49%**	**78,67%**	**84,07%**

LEVIATAN

Benchmark - Résultats obtenus

FQuAD Test - Recall@k

Modèle	Recall@5	Recall@7	Recall@10
Cohere Reranker	**92,50%**	**93,48%**	**94,26%**
D.V. CrossEncoder	52,23%	62,33%	71,46%
Exp. 1	**84,54%**	**87,92%**	**90,90%**
Exp. 2	30,14%	38,33%	49,53%
Exp. 3	39,12%	47,33%	56,81%
Exp. 4	**72,49%**	**78,67%**	**84,07%**

PIAF Test - Recall@k

Modèle	Recall@5	Recall@7	Recall@10
Cohere Reranker	**95,57%**	**96,35%**	**97,22%**
D.V. CrossEncoder	61,77%	69,24%	78,63%
Exp. 1	**94,87%**	**96,00%**	**96,96%**
Exp. 2	38,66%	48,05%	58,99%
Exp. 3	56,73%	64,90%	73,59%
Exp. 4	**90,70%**	**93,83%**	**95,83%**

LEVIATAN

Benchmark - Analyse des résultats

- **Benchmark externes** :
  - Les performances des expériences **1 et 4** sont nettement supérieures à celles du modèle open source **D.V. CrossEncoder**.
  - Le modèle de l'expérience 1 se rapproche fortement du reranker commercial **Cohere**.
- **Benchmark internes** :
  - Les expériences **1 et 4** présentent des performances nettement supérieures aux expériences 2 et 3.
  - L'intégration des données Q/R a **dégradé** les performances de reranking.

LEVIATAN

Benchmark - Analyse des erreurs

Observation

- **Exemple de question** :
  Quelle dynastie accélère la croissance de Babylone ? - FQuAD
- **Problème observé** :
  - Tous les passages candidats proviennent du même article sur Babylone.
  - Le modèle leur attribue des **scores très élevés** ($> 0{,}98$), même si la réponse n'est pas présente.

LEVIATAN

Benchmark - Analyse des erreurs

Observation

- **Exemple de question** :
  Quelle dynastie accélère la croissance de Babylone ? - FQuAD
- **Problème observé** :
  - Tous les passages candidats proviennent du même article sur Babylone.
  - Le modèle leur attribue des **scores très élevés** ($> 0{,}98$), même si la réponse n'est pas présente.

Analyse

- **Cause** :
  - Les exemples négatifs ont été générés aléatoirement, souvent à partir de documents sur des sujets très différents.
  - Le modèle apprend à détecter des différences de thème, mais pas à distinguer la présence ou l'absence de réponse dans un contexte similaire.
- **Conséquence** :
  - Incapacité à discriminer des passages **pertinents** parmi des documents thématiquement proches.
  - Mauvaise performance de reranking dans les cas où tous les candidats partagent un même contexte global.

LEVIATAN

LEVIATAN

Perspectives

- **Amélioration des données négatives** :
  - Éviter la sélection aléatoire de questions sans lien thématique.
  - Intégrer des segments issus du même article, mais **ne contenant pas la réponse**, pour générer des exemples négatifs plus réalistes.
- **Reprise complète des expérimentations** :
  - Réaliser un nouveau cycle complet : **fine-tuning, évaluation, benchmarks** sur le jeu de données corrigé.
  - Explorer l'intégration d'autres modèles pré-entraînés (ex. Microsoft MiniLM [18], Meta RoBERTa [8], etc.).
- **Objectif à long terme** :
  - Concevoir un **reranker multilingue open source**, optimisé pour une exécution sur CPU.
  - Atteindre des performances comparables aux solutions commerciales, tout en garantissant l'accessibilité et la transparence.

LEVIATAN

LEVIATAN

Ressources ouvertes

- **Code source** : https://github.com/LeviatanAI/reranker-cross-encoder
- **Jeu de données et modèles entraînés** : https://huggingface.co/LeviatanAIResearch

LEVIATAN

LEVIATAN

Références I

[1] Daniel Cer et al. "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Ed. by Steven Bethard et al. https://aclanthology.org/S17-2001/. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–14. DOI: 10.18653/v1/S17-2001.

[2] Cohere. *Improve search performance with a single line of code*. https://cohere.com/rerank. Accessed: 2025-03-05. 2025.

[3] Van Tuan DANG. *dangvantuan/CrossEncoder-camembert-large*. https://huggingface.co/dangvantuan/CrossEncoder-camembert-large. Accessed: 2025-03-05. 2022.

[4] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). http://arxiv.org/abs/1810.04805. arXiv: 1810.04805.

[5] Ali Kabbadj. *French-SQuAD : French Machine Reading for Question Answering*. https://github.com/Alikabbadj/French-SQuAD. Accessed: 2025-03-05. 2019.

[6] Vladimir Karpukhin et al. *Dense Passage Retrieval for Open-Domain Question Answering*. 2020. arXiv: 2004.04906 [cs.CL]. URL: https://arxiv.org/abs/2004.04906.

[7] Rachel Keraron et al. *Project PIAF: Building a Native French Question-Answering Dataset*. https://arxiv.org/abs/2007.00968. 2020. arXiv: 2007.00968 [cs.CL].

[8] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv preprint arXiv:1907.11692* (2019). https://arxiv.org/abs/1907.11692.

[9] d'Hoffschmidt Martin et al. "FQuAD: French Question Answering Dataset". In: *arXiv e-prints*, arXiv:2002.06071 (Feb. 2020). https://arxiv.org/abs/2002.06071, arXiv:2002.06071. arXiv: 2002.06071 [cs.CL].

[10] Meta. *Llama 3.2: Revolutionizing Edge AI and Vision with Open, Customizable Models*. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/. Accessed: 2025-02-21. Sept. 2024.

[11] Rodrigo Nogueira and Kyunghyun Cho. "Passage Re-ranking with BERT". In: *CoRR* abs/1901.04085 (2019). http://arxiv.org/abs/1901.04085. arXiv: 1901.04085.

[12] OpenAI. *OpenAI Platform Models*. https://platform.openai.com/docs/models. Accessed: 2025-02-21. 2025.

LEVIATAN

Références II

[13] pandora-s. *pandora-s / neural-bridge-rag-dataset-12000-google-translated*. https://huggingface.co/datasets/pandora-s/neural-bridge-rag-dataset-12000-google-translated. Accessed: 2025-03-05. 2024.

[14] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. https://arxiv.org/abs/1908.10084. Association for Computational Linguistics, Nov. 2019.

[15] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *ArXiv* abs/1910.01108 (2019). https://arxiv.org/abs/1910.01108.

[16] SBERT.net. *Evaluation*. https://sbert.net/docs/package_reference/cross_encoder/evaluation.html. Accessed: 2025-03-05. 2025.

[17] Liang Wang et al. *Multilingual E5 Text Embeddings: A Technical Report*. https://arxiv.org/abs/2402.05672. 2024. arXiv: 2402.05672 [cs.CL].

[18] Wenhui Wang et al. "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://arxiv.org/abs/2002.10957. 2020, pp. 1469–1481.

LEVIATAN

Merci!

Des questions?

LEVIATAN