Les Modèles de Langue pour la Recherche d'Information

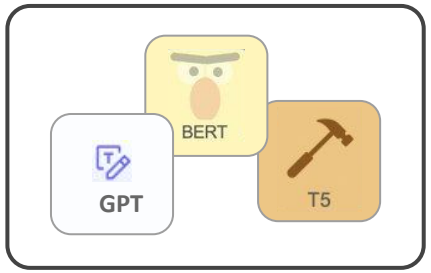
Le Rôle des Connaissances du Domaine dans l'ère des Transformers

Lila Boualili

07 Septembre 2023

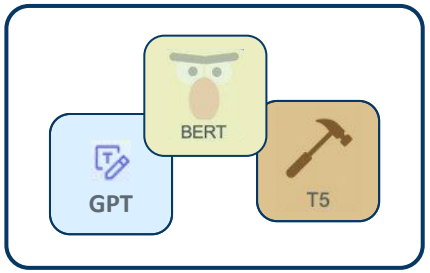
Du NLP à la RI

NLP



Adaptation

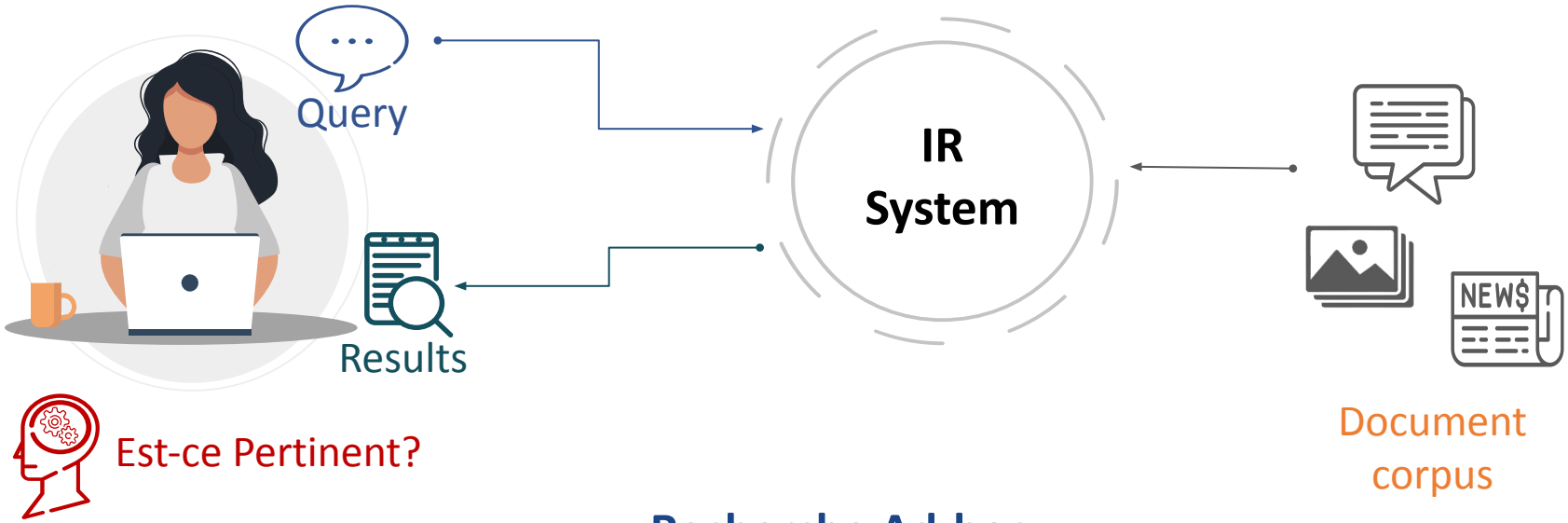
RI



La Recherche d'Information (RI)

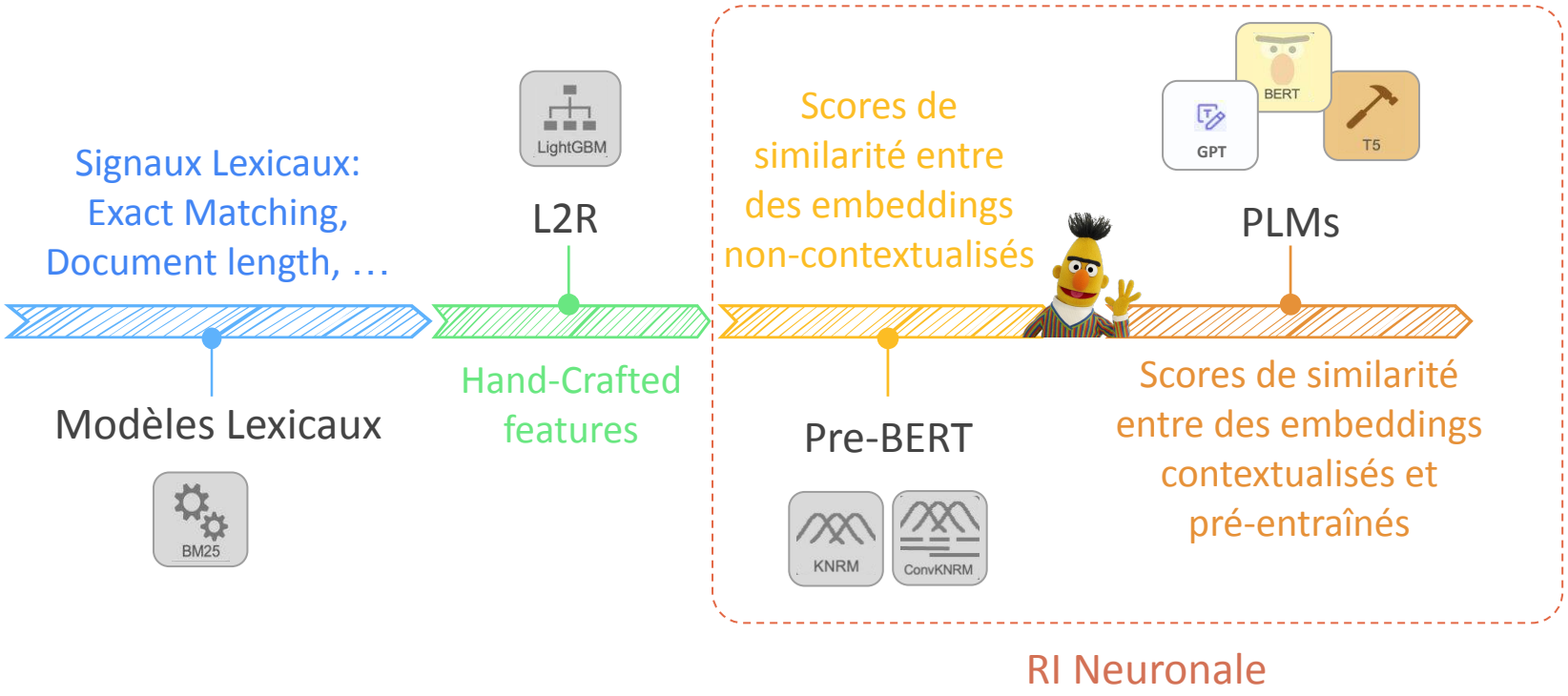
La Recherche d'Information (RI)

"**Information retrieval** is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information." – (Salton, 1968)



Recherche Ad-hoc

Estimer la pertinence

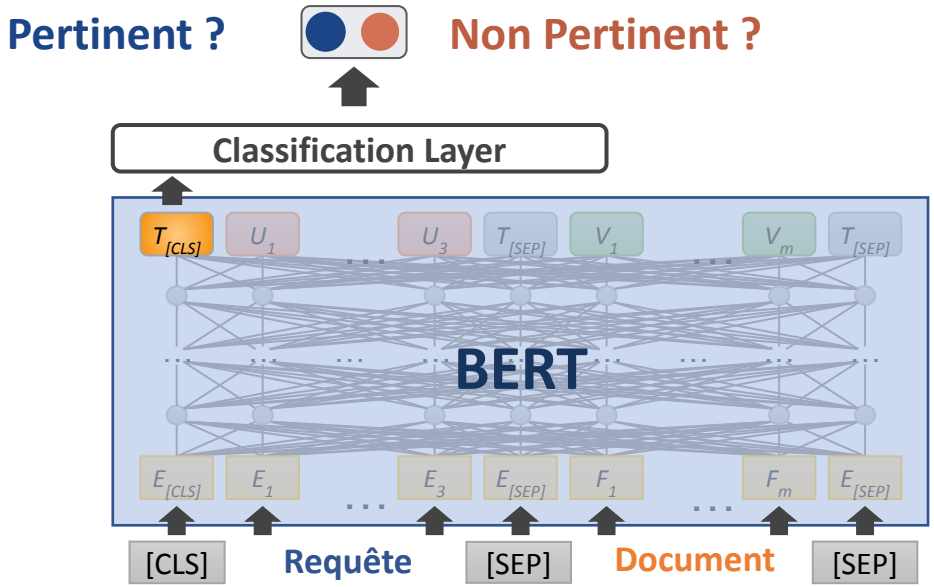


BERT Revolution (Nogueira et al., 2019)

	Method	MS MARCO Passage Development MRR@10	MS MARCO Passage Test MRR@10
Lexicale	BM25 (Microsoft Baseline)	0.167	0.165
Neuronales PRE-BERT	KNRM [251]	0.218	0.198
	Conv-KNRM [54]	0.290	0.271
	IRNet [174]	0.278	0.281
	BERT_{large}	0.365	0.358

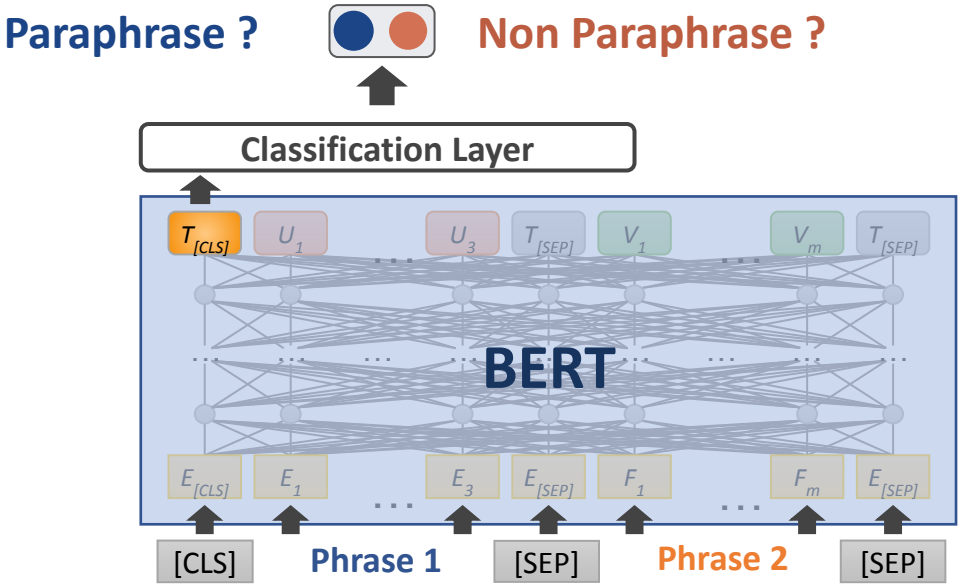
+ 27 %

MonoBERT (Nogueira et al., 2019)



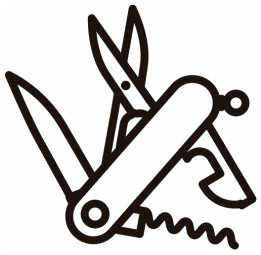
Relevance Matching (asymmetric)

BERT(Devlin et al., 2019)



Semantic Matching (symmetric)

Polyvalence des PLMs



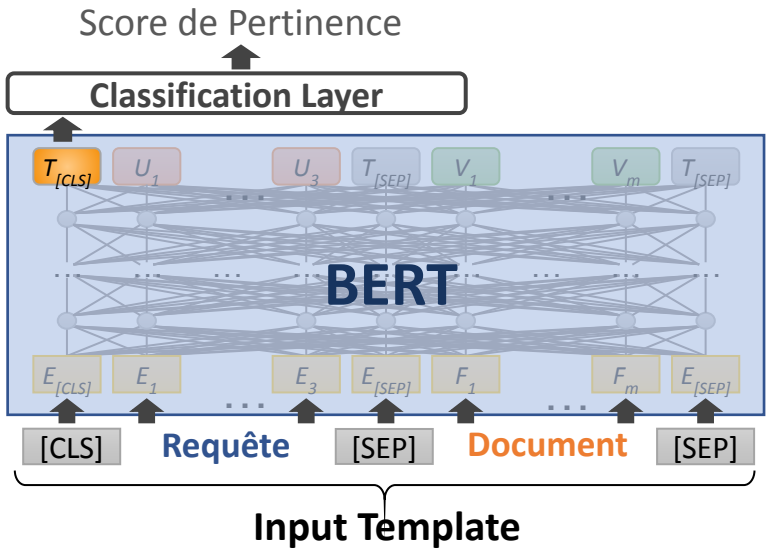
- **La même architecture importée des tâches de NLP s'adapte bien à la recherche ad-hoc en RI grâce au fine-tuning**
- **Le même mechanism d'attention homogène peut estimer la pertinence sans spécialisation**

Applications des PLMs en RI et TAL

Paradigmes d'Application

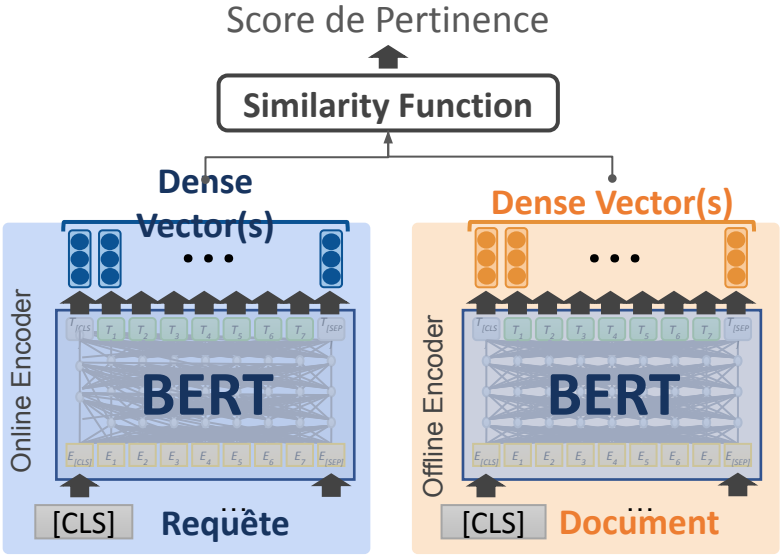
Cross-encoder

- Re-Ranking
- Performance



Bi-encoder

- Retrieval
- Efficience



Les connaissances du domaine dans tout ça !!!

Semantic Matching vs. Relevance Matching

Constat: Les frontières entre le "Matching sémantique" en NLP et le "Matching de pertinence" en RI s'estompent

- Pertinence ⇒ Un modèle de RI robuste doit (Guo et al., 2017; Luan et al., 2021):
  1. Vérifier le **Matching Exacte** des termes

     E.g., "What are Parastratiosphecomyia stratiosphecomyioides ?"
  2. Estimer la **Similarité Sémantique** entre les concepts proches

     E.g., "Movies similar to 'The Imitation Game' ?"
- **BERT n'intègre pas de manière explicite le matching exacte des termes**

Matching Sémantique =/= Matching de Pertinence

Query Causes of **left** ventricular hypertrophy?

Results

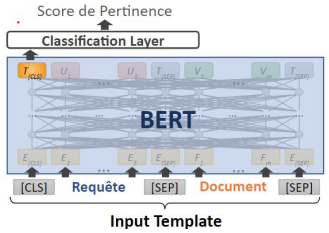
1. Causes of **right** ventricular hypertrophy. There are four usual causes of **right** ventricular hypertrophy …
2. The last common cause of **right** ventricular hypertrophy is the ventricular septal defect …
3. The most common causes of **right** ventricular hypertrophy (RVH) are diseases that damage the lung …

Sémantiquement: **left** ≈ **right**

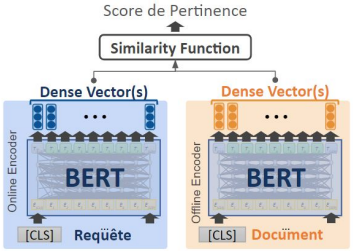
Le Matching Sémantique entre **left** et **right** biaise les prédictions

Exploiter les Connaissances du Domaine

Exact Matching



- Combine Lexical Scores
- Explicit Exact Match Marking



- Exact Match Filtering
- Sparse Retrieval
- Hybrid Sparse-Dense Retrieval

Exact Matching in Re-Ranking

Traditional Cues to complement Neural Capabilities

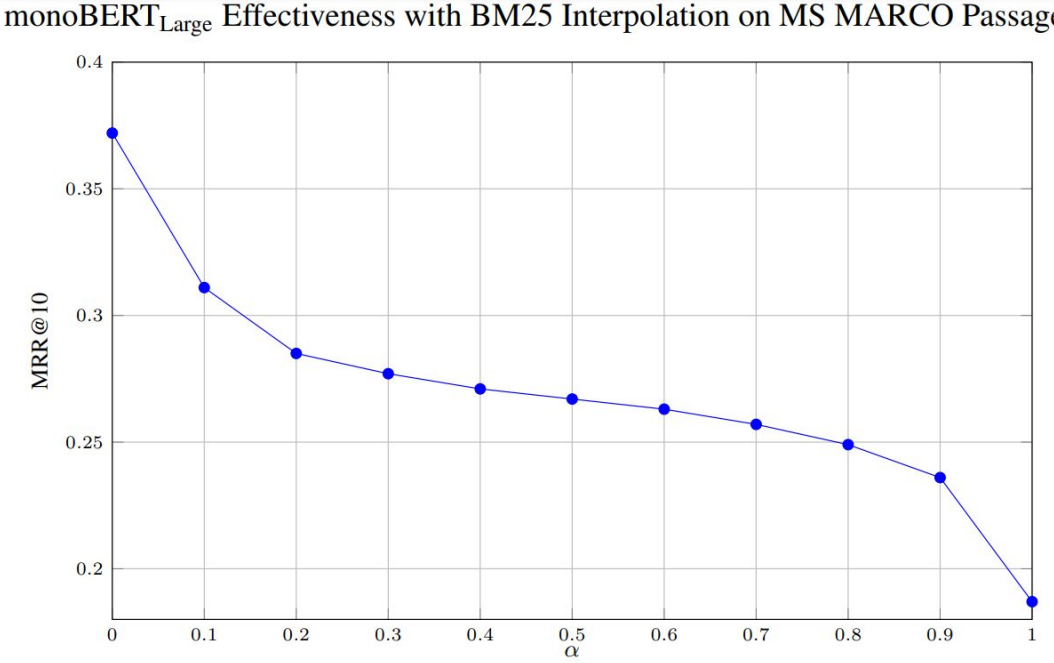
Intégration de Scores Lexicaux

Combinaison linéaire des scores :

- **Sémantique:** produit par BERT
- **Lexical:** produit par un modèle traditionnel BM25

$$s_i \triangleq \alpha \cdot \hat{s}_{\text{BM25}} + (1 - \alpha) \cdot s_{\text{BERT}}$$

Intégration de Scores Lexicaux: In domain



(Li et al., 2021)

Figure 10: The effectiveness of monoBERT$_{\text{Large}}$ on the development set of the MS MARCO passage ranking test collection varying the interpolation weight of BM25 scores: $\alpha = 0.0$ means that only the monoBERT scores are used and $\alpha = 1.0$ means that only the BM25 scores are used. BM25 scores do not appear to improve end-to-end effectiveness using this score fusion technique.

Intégration de Scores Lexicaux: Generalization

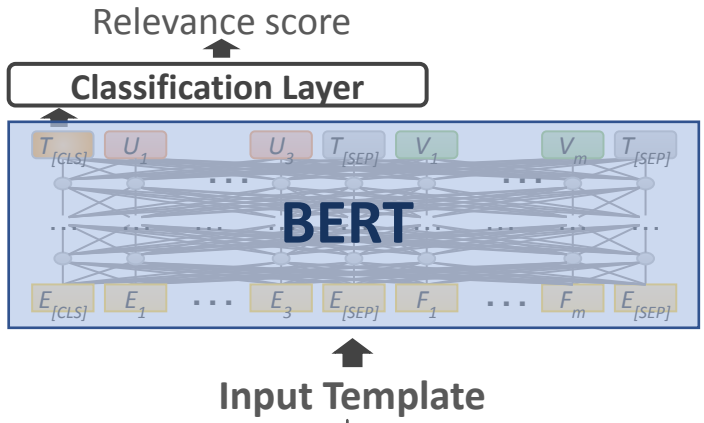
Robusto4	Title run				Description run			
Model	nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–
Vanilla $_{\text{BERT}}$	0.4652	–	0.4046	–	0.4510	–	0.3851	–
+ BM25	0.4932	▲6.0%	0.4255	▲5.2%	0.4856	▲7.7%	0.4062	▲5.5%
GOV2	Title run				Description run			
Model	nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–
Vanilla $_{\text{BERT}}$	0.4533	–	0.5272	–	0.4696	–	0.5248	–
+ BM25	0.5320	▲17.%	0.5987	▲13.	0.5166	▲10.%	0.5742	▲9.4%

⇒ La simple combinaison de scores produits par BERT et BM25 réalise des gains immenses en généralisation

⇒ Les scores BM25 fournissent des signaux de pertinence supplémentaires que BERT ne capture pas efficacement (Yang et al., 2019, Akkalyoncu et al., 2019, MacAvaney et al., 2019, Karpukhin et al., 2020, Boualili et al., 2022)

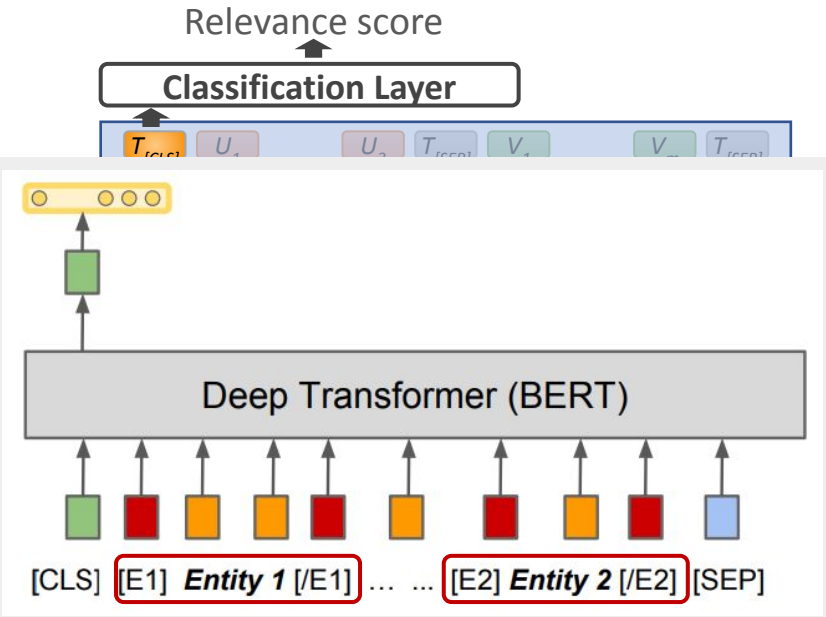
Exact Match Marking (Boualili et al., 2022)

Boualili et al., 2020 propose des **Tokens Spéciaux de Balisage** dans le **Input Template** pour souligner explicitement les **exact terms matches**



Exact Match Marking (Boualili et al., 2022)

Boualili et al., 2020 propose des **Tokens Spéciaux de Balisage** dans le **Input Template** pour souligner explicitement les **exact terms matches**



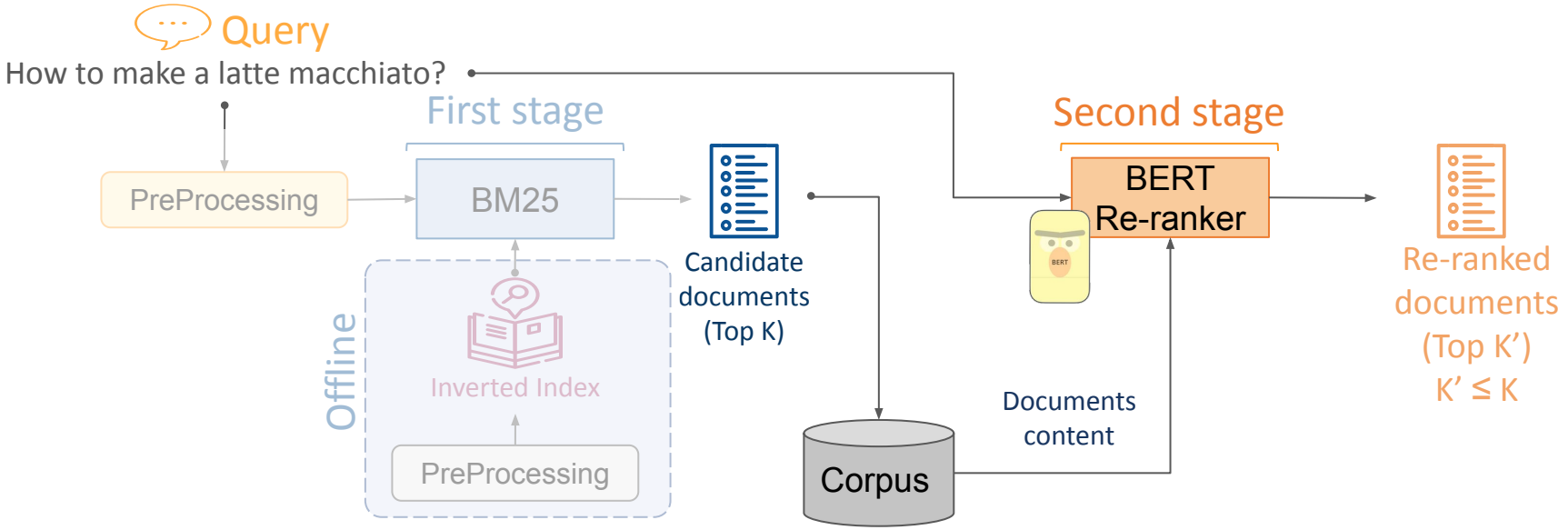
NLP [Entity marking] (Soares et al., 2019)

Exact Match Marking

Définir une stratégie de marquage pour mettre en évidence les termes qui match exactement dans le Input Template en introduisant des tokens marqueurs.

- **Marker Token**: Deux types de tokens
  1. Simple: un token unique “#”
  2. Precise: une paire de tokens “$[e_k]q_k[\backslash e_k]$” qui identifie chaque terme de la requête q_k
- **Marking level**:
  1. Document: utiliser le marquage dans le Document Segment
  2. Pair: utiliser le marquage dans le Query et le Document Segment

Re-ranking + cross-encoder



Scénarios Expérimentaux

1. Evaluer l'impact de l'Exact Match Marking sur les collections in-domain
   - TREC Deep Learning Document Ranking Tasks 2019/2020
2. Evaluer l'impact de l'Exact Match Marking sur la généralisation vers des collections out-of-domain
   - TREC Robust04 and TREC Gov2

Evaluation In-Domain

Impact of exact match marking vs. Vanilla baseline

Re-ranking the top-1000 documents retrieved by BM25+RM3

TREC DL Doc	DL 2019				DL 2020			
Model	nDCG@10		MAP@100		nDCG@10		MAP@100	
BM25	0.5176	–	0.2434	–	0.5286	–	0.3793	–
BM25+RM3	0.5169	–	0.2772	–	0.5248	–	0.4006	–
Vanilla BERT	0.6726	–	0.3006	–	0.6340	–	0.4523	–
Sim-Doc BERT	0.6858	▲2.0%	0.3038	▲1.1%	0.6340	▲0.0%	0.4414	▽2.4%
Sim-Pair BERT	0.6798	▲1.1%	0.3057	▲1.7%	0.6495	▲2.4%	0.4505	▽0.4%
Pre-Doc BERT	0.6777	▲0.8%	**0.3061**	▲1.8%	0.6368	▲0.4%	**0.4513**	▽0.2%
Pre-Pair BERT	0.7025†	▲4.4%	0.3018	▲1.8%	0.6498	▲2.5%	0.4497	▽0.6%

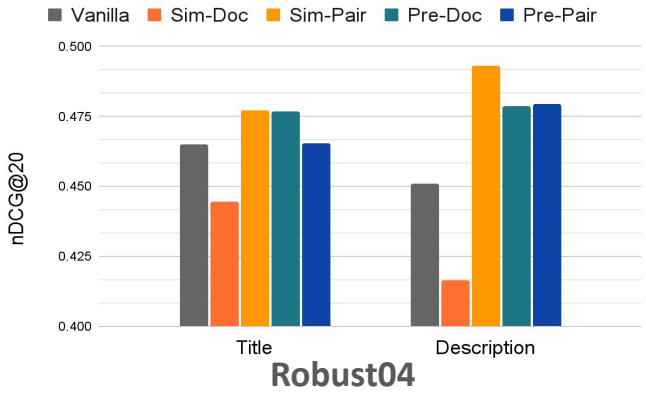
1 Gains de performance

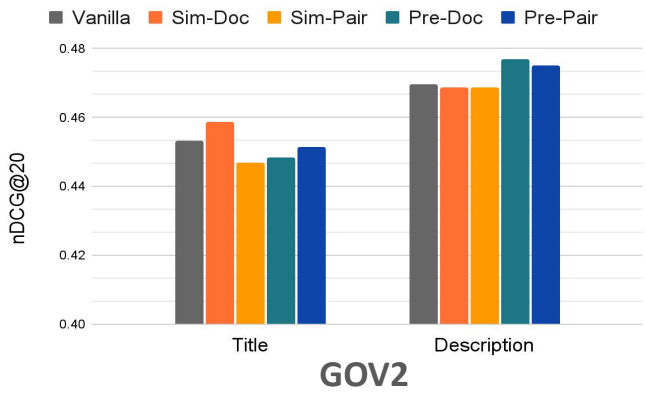
Exact Match Marking améliore la performance

2 Pair Marking

Pair marking level donne les meilleures performances

Evaluation Out-Domain (Zero-shot Transfer)



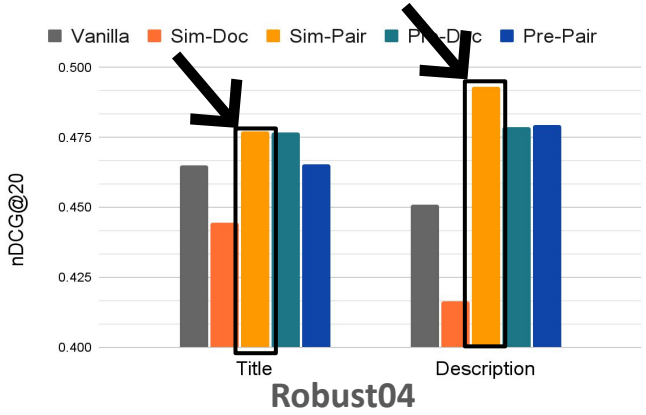


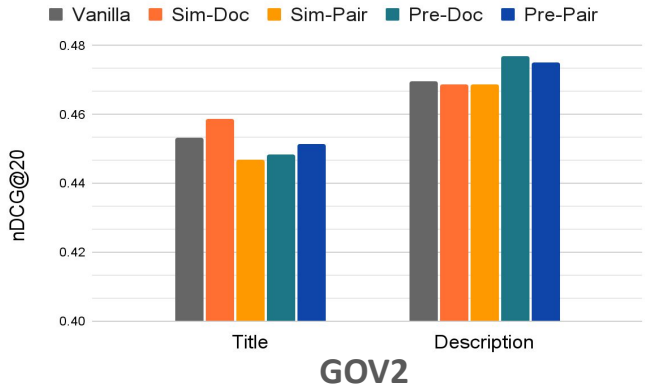
TREC Topics comprennent différentes formulations de la même requête

Title = Poliomyelitis and Post-Polio

Description = Is the disease of Poliomyelitis (polio) under control in the world?

Evaluation Out-Domain (Zero-shot Transfer)





1 Marking Strategy

Sim-Pair (#) réalise les meilleurs gain de performance

⇒ On sélectionne cette stratégie pour le restant de l'analyse

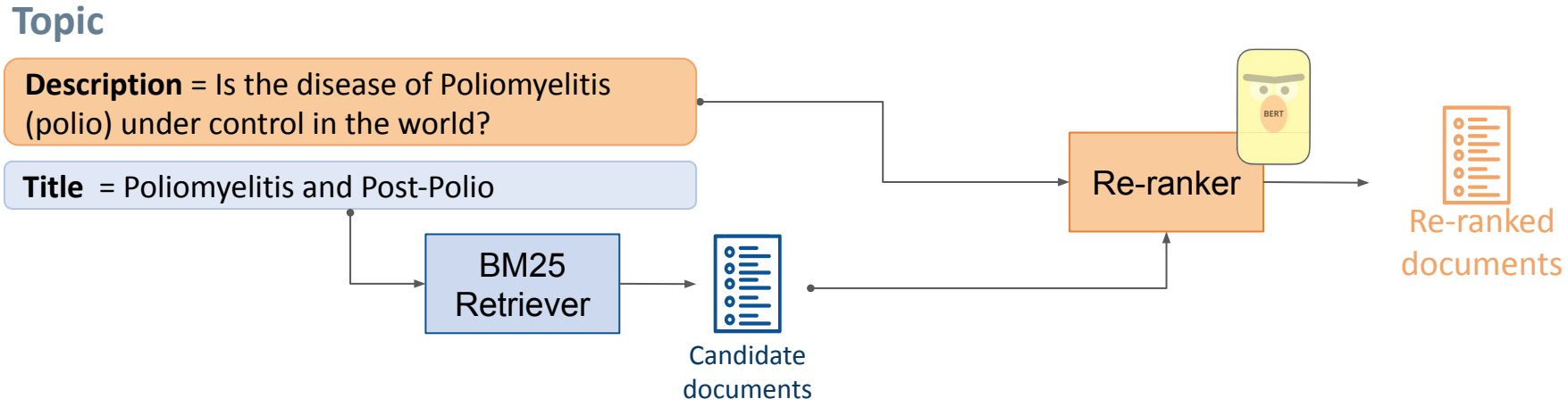
2 Title vs. Description

La performance avec les descriptions **surpassent** la performance avec titles

First-stage Retriever

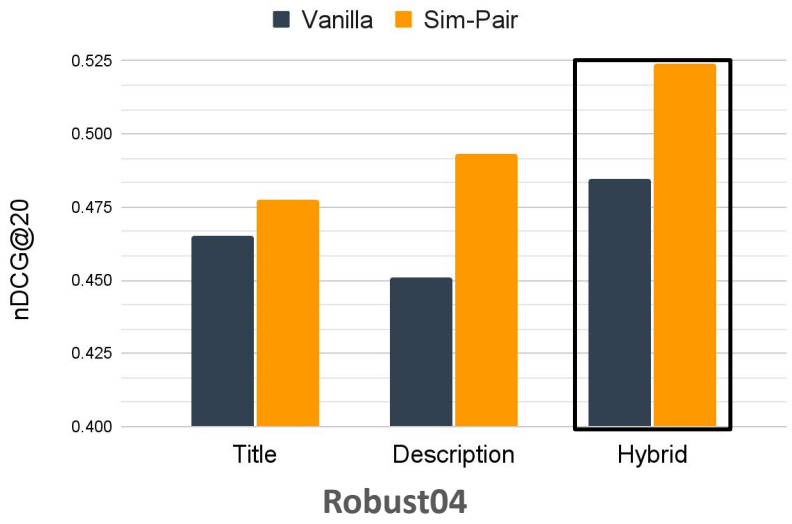
- **BM25 a un meilleur Recall sur les requêtes courtes par mots-clés (i.e., Title field)**
- BERT préfère les requêtes longues en langage naturel (i.e., Description field) (Dai et al., 2019)

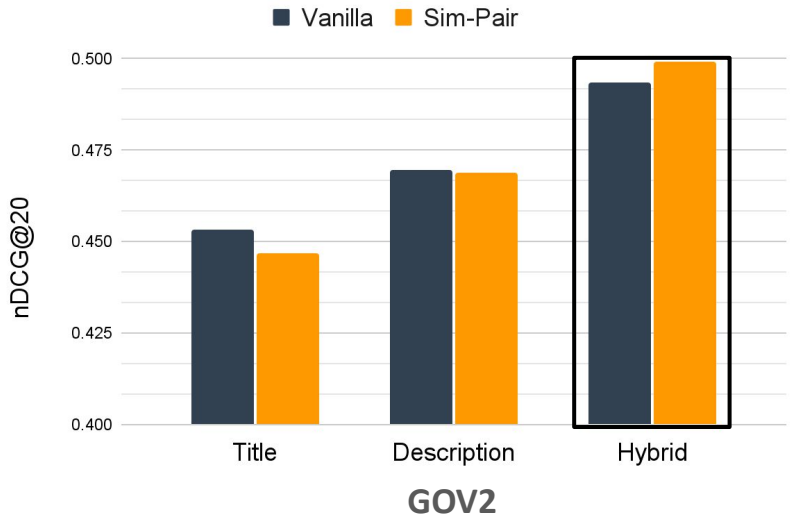
A Hybrid Retrieve-then-Re-rank Pipeline



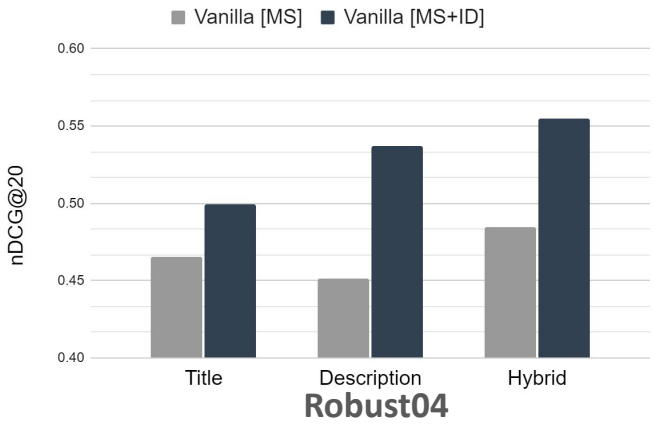
A Hybrid Retrieve-then-Re-rank Pipeline

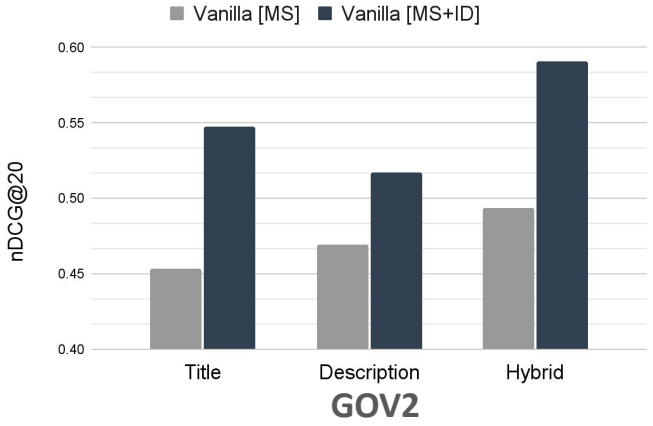
De meilleurs candidats avec les titres ⇒ Meilleure performance de re-ranking avec les descriptions





Multi-Phase Fine-Tuning





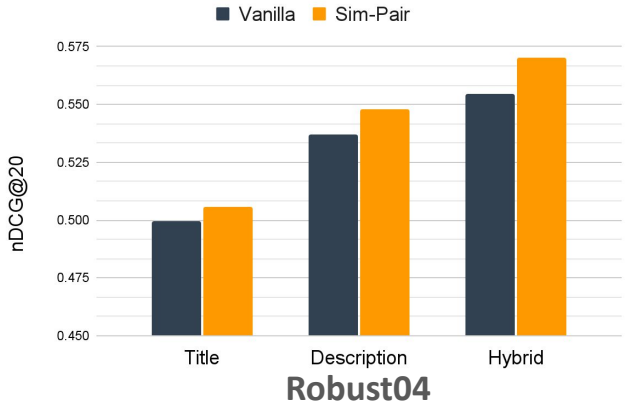
Multi-Phase Fine-Tuning [MS+ID]:

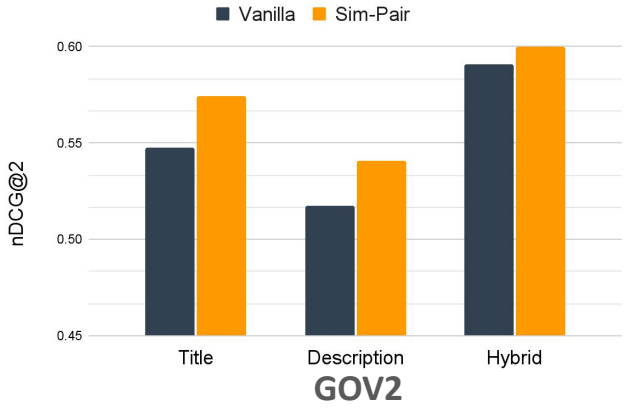
1. MS MARCO fine-tuning [MS]
2. In-Domain fine-tuning [ID]

MS+ID fine-tuning >> MS fine-tuning

Fine-tuning additionnel sur la collection cible
⇒ **Immense gains en performance**

Multi-Phase Fine-Tuning





Exact Match Marking

Sim-Pair marking ⇒ Gains significatifs en performances sur Robust04 et **GOV2**

Impact sur des PLMs plus robustes - Limites

ELECTRA (Clark et al., 2020) est une variante de BERT avec une tâche de pre-training plus robuste “Replaced Token Detection”

Zero-Shot Transfer

		Title	Desc	Hybrid
BERT	Vanilla	0.4652	0.4510	0.4845
	Sim-Pair	0.4773 Δ	**0.4931▲**	**0.5239▲**
ELECTRA	Vanilla	0.4416	0.4482	0.4782
	Sim-Pair	**0.4717▲**	0.4597 Δ	**0.5043▲**

⇒ Exact Match Marking réalise des **gains significatifs en performance ▲**

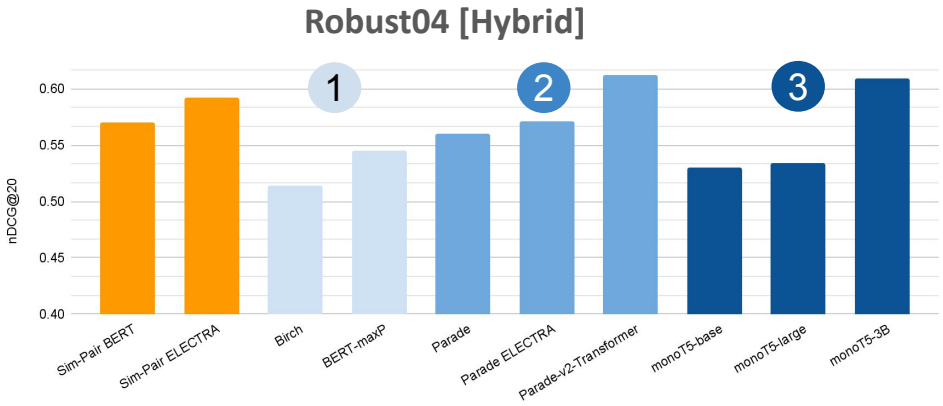
Multi-Phase Fine-Tuning

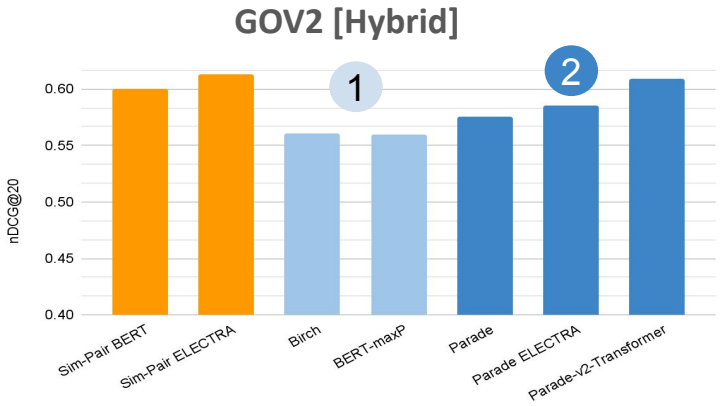
		Title	Desc	Hybrid
BERT	Vanilla	0.4995	0.5368	0.5546
	Sim-Pair	0.5058 Δ	**0.5479▲**	**0.5701▲**
ELECTRA	Vanilla	0.5375	0.5676	0.5901
	Sim-Pair	0.5380 Δ	0.5686 Δ	0.5927 Δ

⇒ ELECTRA surpasse BERT

⇒ Exact Match Marking réalise de **faibles gains en performance Δ**

Comparaison avec l'état de l'art (cross-encoders)





1. Sim-Pair marking surpasse les méthodes basées sur l'agrégation de scores
2. Des performances comparables à celles du modèle d'agrégation de représentations PARADE
3. La qualité se rapproche de celle de monoT5-3B sur Robust04 avec moins de **4% de ses paramétres**

Discussion

Les signaux d'exact matching transmis par les tokens de marquage:

✓ Peuvent compléter les capacités sémantiques inhérentes aux PLMs lorsque les données du domaine cible ne sont pas disponibles **(zero-shot)**

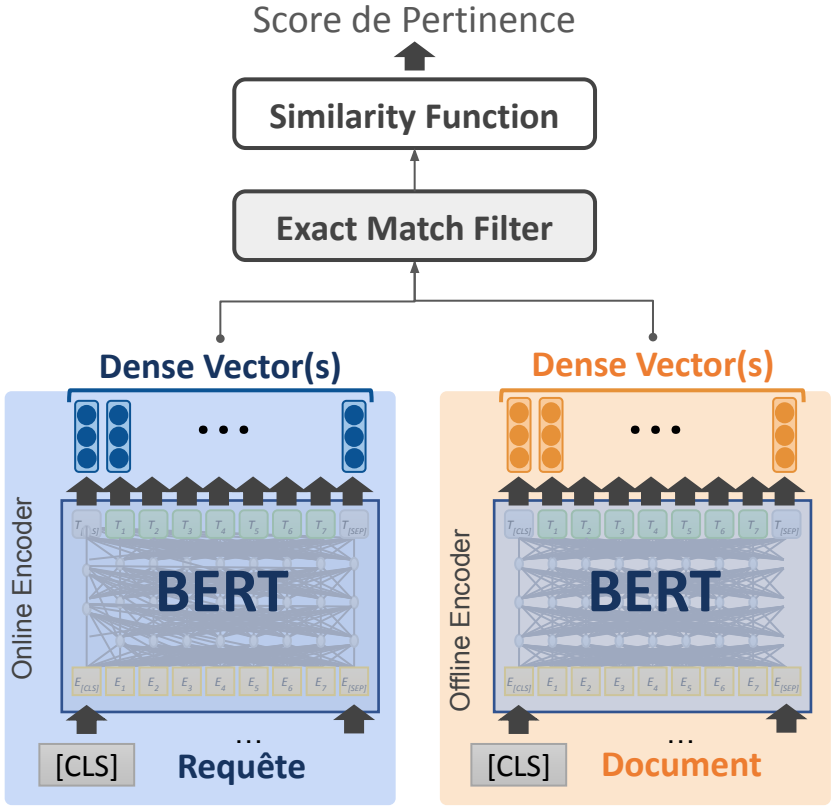
X Sont moins utiles lorsque le modèle a une meilleure compréhension de la tâche sur la distribution des données **(in-domain | modèles plus robutes/ larges)**

Exact Matching in Retrieval

Traditional Efficiency with Neural Capabilities

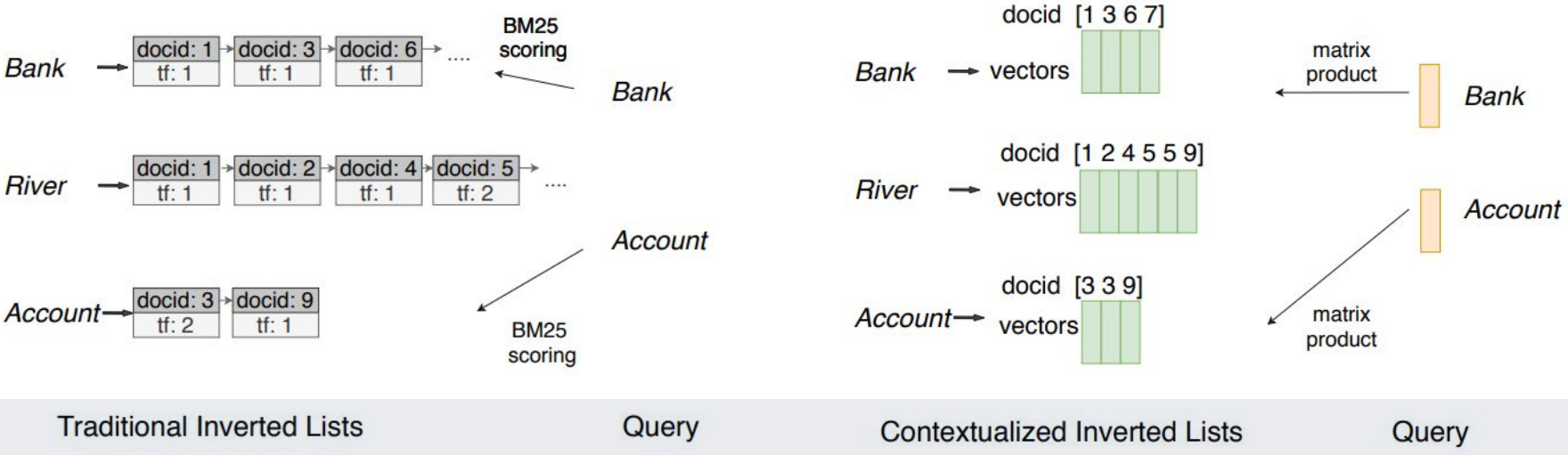
Filtre de Matching exact: COIL (Gao et al., 2020)

- Calculer la pertinence en se basant les mots qui match exactement (Exact Lexical Match) entre la requête et le document



Filtre de Matching exact: COIL (Gao et al., 2020)

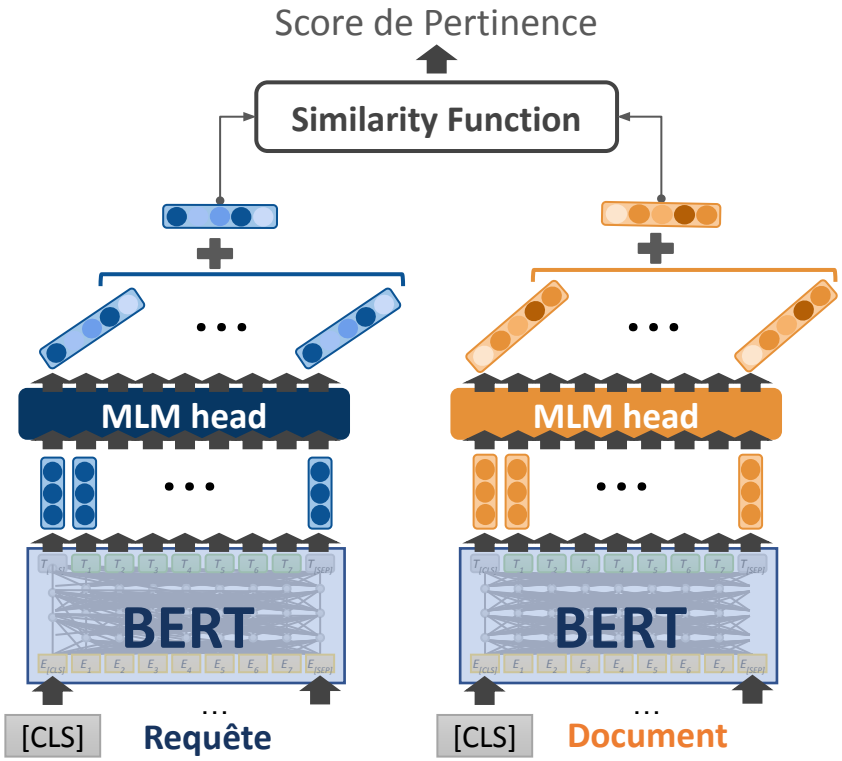
- Contextualized Inverted Lists



(Gao et al., 2020)

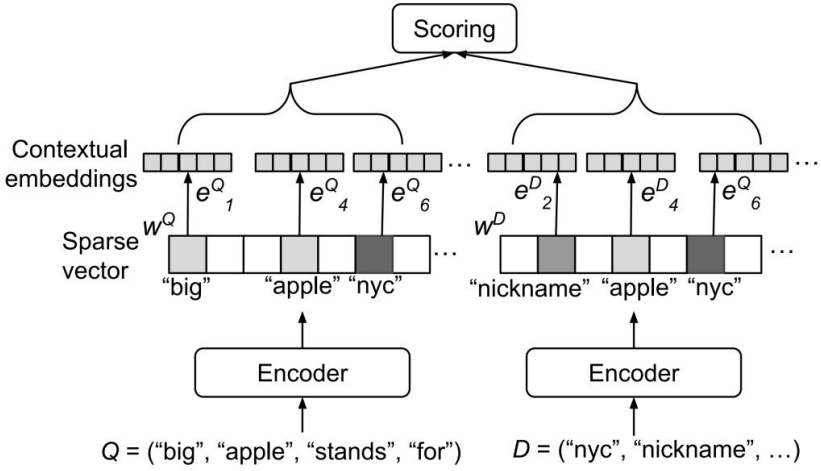
Sparse Retrieval: SPLADE (Formal et al., 2021)

- Construit des vecteurs sparses en utilisant les logits produit par le MLM head à partir des vecteurs denses
- Expansion implicite de la requête et du document
- Utilise des indexes inversés traditionnels
- Problème de mismatch sémantique, e.g., la polysémie



Hybrid sparse-dense: SparseEmbed (Kong et al., 2023)

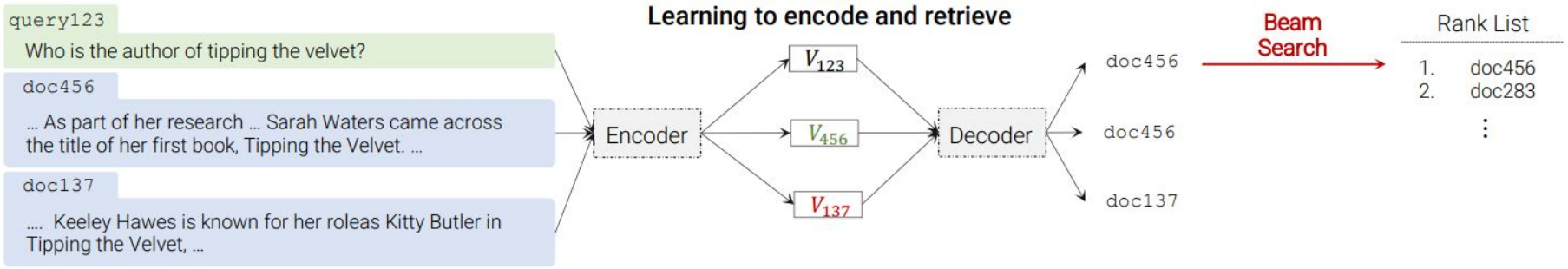
- Utilise un backbone SPLADE pour construire des représentations sparses
- Contextualiser les top-k termes du vocabulaire pour surmonter le problème de mismatch sémantique de SPLADE
- Utilise l'index inversé avec des vecteurs denses de COIL



(Kong et al., 2023)

Un Paradigme Emergeant: DSI (Tay et al., 2022)

- Un index differentiable (LLM weights)
- Encoder la requête et décoder les identifiants des documents pertinents



(Tay et al., 2022)

Conclusion

- Construire des modèles toujours plus larges et plus polyvalents est la direction naturellement adoptée ces dernières années avec les LLMs
- Malgré ces succès, l'amélioration des performances par l'augmentation du nombre de paramètres du modèle peut entraîner des coûts importants et limiter l'accès à une poignée d'organisations disposant des ressources nécessaires à l'entraînement
- Axer le développement de modèles sur le nombre de n'est ni extensible ni viable à long terme
- **Les connaissances spécifiques au domaine d'application peuvent interagir avec les modèles de langues pour mieux les adapter aux caractéristiques du domaine**
- **Ces connaissances peuvent produire des modèles plus efficients et plus interprétables**

Merci!