

RITAL @ DAC

Recherche d'Information Neuronale

Benjamin Piwowarski

Cours 4

Version PDF



Machine Learning &
Deep Learning for
Information Access



LA RI ET L'APPRENTISSAGE

PROBLÉMATIQUE

✍ Avec le temps, on a défini beaucoup d'indicateurs pour la pertinence...

❓ Comment les combiner ?

- ✍ Chaque modèle (BM25, PageRank) mesure une certaine forme de pertinence
- ✍ Comment les combiner ?

ID	Feature Description	Category
1	$\sum_{q_i \in q \cap d} c(q_i, d)$ in body	Q-D
2	$\sum_{q_i \in q \cap d} c(q_i, d)$ in anchor	Q-D
3	$\sum_{q_i \in q \cap d} c(q_i, d)$ in title	Q-D
4	$\sum_{q_i \in q \cap d} c(q_i, d)$ in URL	Q-D
5	$\sum_{q_i \in q \cap d} c(q_i, d)$ in whole document	Q-D
6	$\sum_{q_i \in q} idf(q_i)$ in body	Q
7	$\sum_{q_i \in q} idf(q_i)$ in anchor	Q
8	$\sum_{q_i \in q} idf(q_i)$ in title	Q
9	$\sum_{q_i \in q} idf(q_i)$ in URL	Q
10	$\sum_{q_i \in q} idf(q_i)$ in whole document	Q
11	$\sum_{q_i \in q \cap d} c(q_i, d) \cdot idf(q_i)$ in body	Q-D
12	$\sum_{q_i \in q \cap d} c(q_i, d) \cdot idf(q_i)$ in anchor	Q-D
13	$\sum_{q_i \in q \cap d} c(q_i, d) \cdot idf(q_i)$ in title	Q-D
14	$\sum_{q_i \in q \cap d} c(q_i, d) \cdot idf(q_i)$ in URL	Q-D
15	$\sum_{q_i \in q \cap d} c(q_i, d) \cdot idf(q_i)$ in whole document	Q-D
16	$ d $ of body	D
17	$ d $ of anchor	D
18	$ d $ of title	D
19	$ d $ of URL	D
20	$ d $ of whole document	D

21	BM25 of body	Q-D
22	BM25 of anchor	Q-D
23	BM25 of title	Q-D
24	BM25 of URL	Q-D
25	BM25 of whole document	Q-D
26	LMIR.ABS of body	Q-D
27	LMIR.ABS of anchor	Q-D
28	LMIR.ABS of title	Q-D
29	LMIR.ABS of URL	Q-D
30	LMIR.ABS of whole document	Q-D
31	LMIR.DIR of body	Q-D
32	LMIR.DIR of anchor	Q-D
33	LMIR.DIR of title	Q-D
34	LMIR.DIR of URL	Q-D
35	LMIR.DIR of whole document	Q-D
36	LMIR.JM of body	Q-D
37	LMIR.JM of anchor	Q-D
38	LMIR.JM of title	Q-D
39	LMIR.JM of URL	Q-D
40	LMIR.JM of whole document	Q-D
41	Sitemap based term propagation	Q-D
42	Sitemap based score propagation	Q-D
43	Hyperlink based score propagation: weighted in-link	Q-D
44	Hyperlink based score propagation: weighted out-link	Q-D
45	Hyperlink based score propagation: uniform out-link	Q-D
46	Hyperlink based propagation: weighted in-link	Q-D
47	Hyperlink based feature propagation: weighted out-link	Q-D
48	Hyperlink based feature propagation: uniform out-link	Q-D
49	HITS authority	Q-D
50	HITS hub	Q-D
51	PageRank	D

APPRENTISSAGE D'ORDONNANCEMENT : LES INGRÉDIENTS

✍ Une fonction $\varphi(q, d)$

💡 Premier article

💻 Fuhr, N. and Buckley, C. 1991. A Probabilistic Learning Approach for Document Indexing.

$$\varphi(q, d) = \log P(R|q, d) = 1 - \exp(\Phi(q, d) \cdot \theta)$$

✍ Un jeu de données en RI

- des questions q
- un ensemble de documents D
- un ensemble de documents pertinents $D_q^+ \subseteq D$
- un ensemble de documents non pertinents $D_q^- \subseteq D$

✍ On suppose (en général) que les documents $D \setminus D_q^+$ sont **non pertinents**

⚠ Cela n'est pas toujours vrai !

ORDONNANCEMENT ET APPRENTISSAGE

?

φ est coûteux en temps de calcul

φ a été appris... mais c'est lent !

Comment peut-on faire pour aller plus vite ?

💡 Ordonnancement en deux étapes (Two-Stage Ranking)

1. On cherche le top- K (ex. $K = 1000$) avec BM25 ou un autre modèle **rapide**
2. On ré-ordonne les K documents avec φ

ORDONNANCEMENT EN DEUX ÉTAPE : CONSÉQUENCES

(1) ON VEUT RÉ-ORDONNER

- 👉 Utilisez les exemples après BM25 (ou autre)

(2) ON VEUT ORDONNER COMPLÈTEMENT

- 👉 Échantillonnage dans l'ensemble des documents D
- 👉 Dans les deux cas, on échantillonne toujours des exemples positifs pour entraîner le modèle, mais **Ordonnancement complet**

On échantillonne les négatifs en utilisant... d'autres modèles de RI

 Hofstätter, S. et al. 2021. *Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation*. Technical Report #arXiv:2010.02666.

Ré-ordonnancement

On échantillonne les document à ré-ordonner en utilisant BM25

QUESTION FONDAMENTALE



Comment apprendre un modèle qui maximise une métrique de RI?

- 👉 Pas possible en général
- 👉 On utilise des relaxations du problème

- 👉 Trois façon d'aborder le problème

Pointwise

Régression ou classification

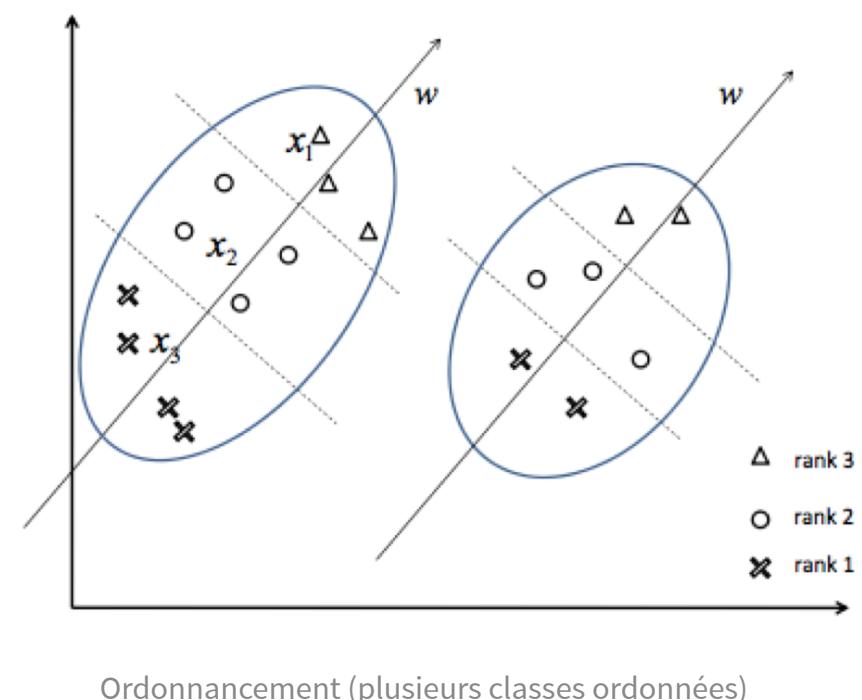
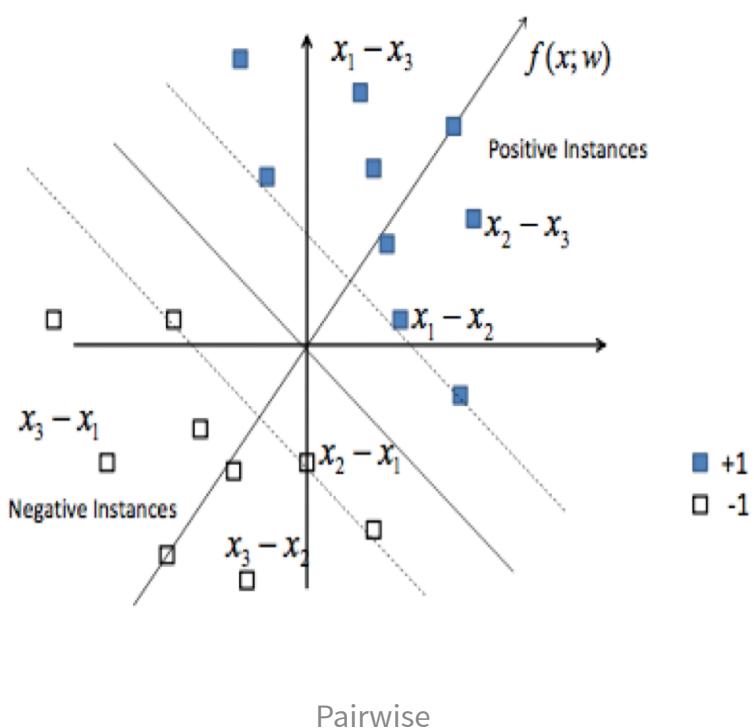
Pair-wise

Classification

List-wise

Approximation de métriques RI

PAIR-WISE VS LIST-WISE



Coûts : Pointwise Cross-Entropy (Classification)

↗ PCE

$$\Delta_{PCE}(p_\theta(q, d, y)) = y \log p_\theta(q, d) + (1 - y) \log 1 - p_\theta(q, d)$$

- ➡ Facile à mettre en place
- ➡ Peut être instable lors de l'apprentissage

Coûts : Max-Margin (PAIRWISE, CONTRASTIF)

Max-Margin

max-margin loss

$$\Delta(f_\theta(q, d_+), f_\theta(q, d_-)) = \max \{0, 1 - (f_\theta(q, d_+) - f_\theta(q, d_-))\}$$

Le minimum 0 est atteint si

$$f_\theta(q, d_+) > f_\theta(q, d_-) + 1$$

Coûts : CROSS-ENTROPIE APPROXIMATIVE (LISTWISE, CONTRASTIF)

InfoNCE

Coût défini pour une liste de documents \approx mesure de RI

La plus utilisée actuellement en LETOR / RI

$$\Delta(f, q, \tilde{D}) = \sum_{d_+ \in \tilde{D} \cap D_q^+} R(q, d_j) \log \frac{\exp(f(q, d_+))}{\sum_{d \in \tilde{D}} \exp(f(q, d))}$$

où R est 1 si d_j est pertinent pour q (0 sinon)

- 👉 Très utilisé en pairwise (\tilde{D} avec deux documents) pour le ré-ordonnancement
- 👉 Pour l'ordonnancement, on utilise l'ensemble / un sous-ensemble des documents du batch
- Enjeu = il faut trouver le meilleur \tilde{D}
- 📘 *Lindgren, E. et al. 2021. Efficient Training of Retrieval Models using Negative Cache.*
- 👉 Essaie de trouver l'ensemble \tilde{D} tel que le gradient soit le plus proche possible du cas où $\tilde{D} = D$

Coûts : SOFTRANK (LISTWISE, CONTRASTIF)

 Taylor, M. et al. SoftRank: Optimizing Non-Smooth Rank Metrics. WSDM.

 On peut approximer la mesure nDCG pour SoftNDCG

$$\mathbb{E} [nDCG] \approx \sum_{q \in Q} \frac{g(d|q)}{CG_{\max}(q)} \sum_{r \geq 1} d(r) p(\mathbf{R} = r | \mathbf{q}, \mathbf{d})$$

Deux questions

1. Pourquoi \approx ?
2. Comment estimer $p(\mathbf{R} = r | \mathbf{q}, \mathbf{d})$?

 On utilise une distribution normale pour calculer $p(\mathbf{R} = r | \mathbf{q}, \mathbf{d})$

Soit π_{ij} la probabilité que d_i soit avant d_j

En supposant que $S_i \sim \mathcal{N}(\mu_i, \sigma^2)$ où μ_i est le score donné par le SRI

 On a $\pi_{ij} \Leftrightarrow S_i - S_j > 0 \Leftrightarrow \mathcal{N}(\mu_i - \mu_j, 2\sigma^2) > 0$ ce qui correspond à la fonction cumulative de distribution (qui est facilement dérivable)

SUPERVISION FAIBLE

💡 Beaucoup de données mais plus bruitées

Actions des utilisateurs

- Reformulation
- Clics
- ...

💡 Peu de données supervisées mais des modèles

On a des questions - logs ou générées (ex. ancrés)

... mais pas d'indication de pertinence

👉 Utilisons les "vieux" modèles !

📘 Dehghani, M. et al. 2017. Neural Ranking Models with Weak Supervision.

FAUX NÉGATIFS

! Documents pertinents mais estimés non pertinents

Collection avec peu de labels (clics, jugements humains très peu nombreux, ...)

👉 On peut traiter le problème de deux façons :

1. Essayer d'estimer si c'est un faux négatif avec un autre modèle

 *RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering.*

2. Utiliser des techniques de distillation

👉 On minimise la distance entre les prédictions des modèles

 *Gao, L. et al. 2020. Understanding BERT Rankers Under Distillation.*

DISTILLATION

Une autre technique d'apprentissage est basée sur la distillation



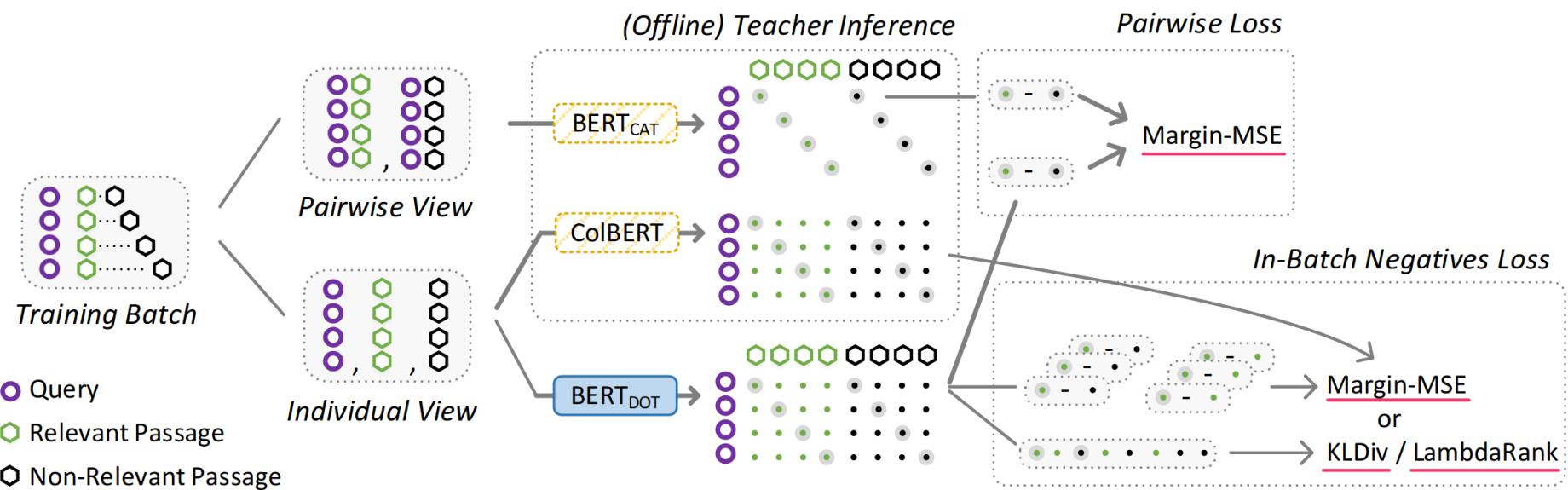
Idée originale = ré-entraîner un modèle (student-teacher) marche bien

 *Furlanello, T. et al. 2018. Born-Again Neural Networks.*

En RI, deux raisons pour lesquelles la distillation est essentielle

1. Les cross-encoder BERT sont faciles à entraîner... mais **très** gourmands en resources
2. Il y a beaucoup de faux négatifs

DISTILLATION (2)



[PDF] Hofstätter, S. et al. 2021. *Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling*.

Exemple (Margin-MSE)

$$\text{Margin-MSE}(t^+, t^-, s^+, s^-) = |(t^+ - t^-) - (s^+ - s^-)|^2$$

PRÉ-ENTRAÎNEMENT

Un meilleur pré-entraînement permet d'améliorer les performances

 Gao, L. and Callan, J. 2021. Condenser: a Pre-training Architecture for Dense Retrieval.

- 👉 On force le CLS à encoder de l'information sémantique

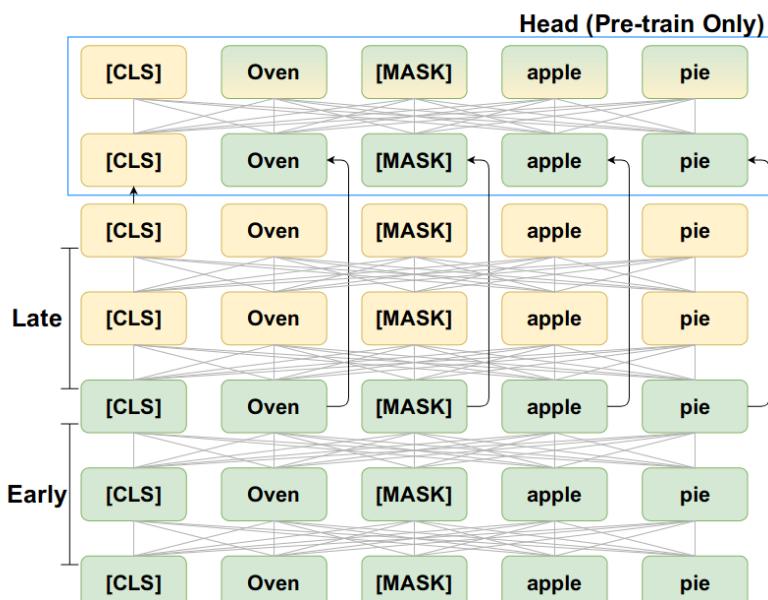


Figure 1: Condenser: Shown are 2 early and 2 late backbone layers. Our experiments each have 6 layers. Condenser Head is dropped during fine-tuning.

- 👉 On utilise des bouts de documents comme des questions

then form a training batch of coCondenser. Write a span s_{ij} 's corresponding late CLS representation h_{ij} , its corpus-aware contrastive loss is defined over the batch,

$$\mathcal{L}_{ij}^{co} = -\log \frac{\exp(\langle h_{i1}, h_{i2} \rangle)}{\sum_{k=1}^n \sum_{l=1}^2 \mathbb{I}_{ij \neq kl} \exp(\langle h_{ij}, h_{kl} \rangle)} \quad (6)$$

Familiar readers may recognize this as the contrastive loss from SimCLR (Chen et al., 2020), for which we use random span sampling as augmentation. Others may see a connection to noise contrastive estimation (NCE). Here we provide an NCE narrative. Following the spirit of the distribu-

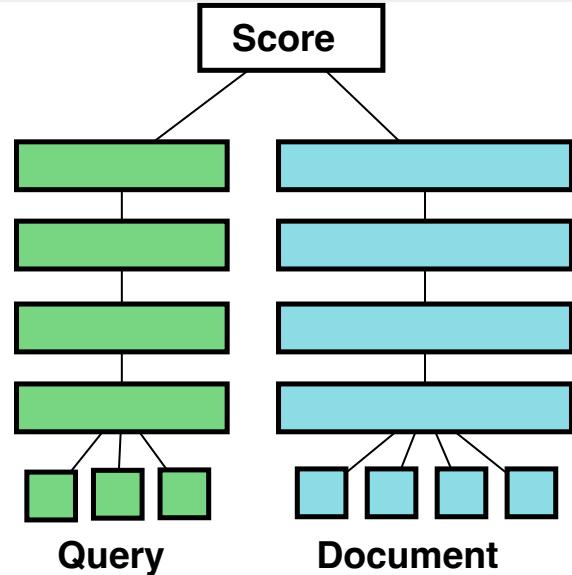
CONCLUSION

- 👉 Attention aux échantillons (ordonnancement ou ré-ordonnancement ?)
- 👉 Possibilité d'utilisation d'une supervision faible (interaction utilisateur et/ou modèles classiques)
- 👉 La distillation est **très efficace**
- 👉 Attention au problème du transfert
 - 📘 Thakur, N. et al. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models.
 - 📘 Zhan, J. et al. 2022. Evaluating Extrapolation Performance of Dense Retrieval.
- 👉 Possibilité d'apprendre en indexant (Product Quantization)
 - 📘 Zhan, J. et al. 2021. Learning Discrete Representations via Constrained Clustering for Effective and Efficient Dense Retrieval.

PLAN DU COURS

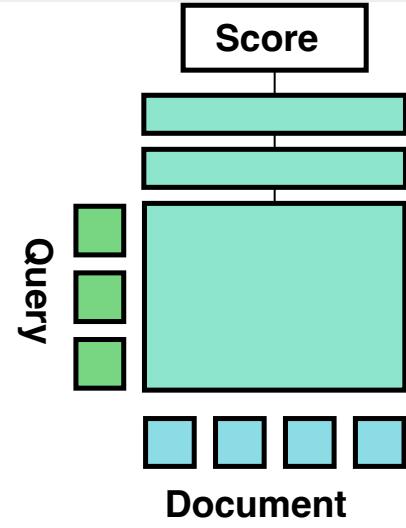
- Comment apprendre les paramètres ?
- Réseaux de neurones en RI
 - Modèles denses (représentation)
 - Modèles d'interaction (interaction faible et forte)
 - Modèles sparses (représentation)

COMPARAISON RAPIDE



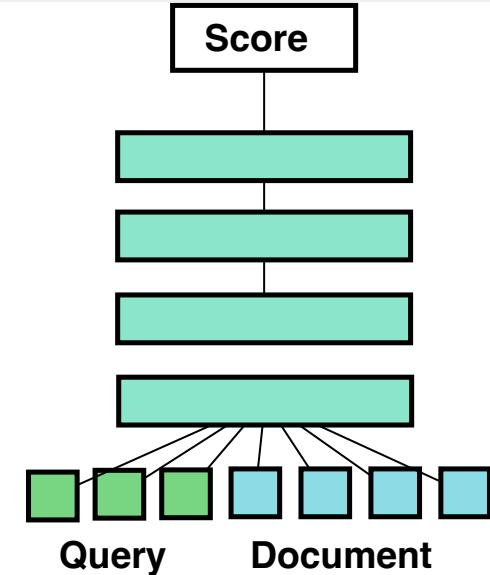
Modèle de représentation

- + Rapide
- Moins robuste
- ANCE, TAS-Balanced, SparTerm, SPLADE, ...



Modèle d'interaction faible

- + Plus robuste (à partir de ColBERT)
- Un peu plus lent
- DRMM, ColBERT



Modèle d'interaction forte

- + Meilleures performances
- Très lent
- monoBERT

MODÈLES DENSES

PLONGEMENTS DE TEXTES

💡 Espace sémantique

Tout texte t est représenté par $f_\theta(t) \in \mathbb{R}^n$

- Modèle symétrique $s(t_1, t_2) = g(f_\theta(t_1), f_\theta(t_2))$
- Modèle asymétrique $s(t_1, t_2) = g(f_\theta^{(l)}(t_1), f_\theta^{(r)}(t_2))$

❗ Problème

- Le texte est une séquence (de mots, de caractères)
- Les réseaux de neurones traitent des entrées de taille fixe
- Comment faire ???

REPRÉSENTATION DE TAILLE FIXE

❓ Comment faire ?

- Un bon vieux Latent Semantic Analysis (= SVD)...
- Réseaux de neurones récurrents (mais pas vraiment utilisé)
 - 👉 On utilise s_T et/ou $s_1^{(r)}$
- Réseaux Convolutionnels
 - 👉 Représentation de taille fixe après un pooling
- Transformers
 - 👉 On utilise le [CLS] ou la moyenne

DSSM (2013)

Un des premiers modèles neuronaux

Posterior probability
computed by softmax

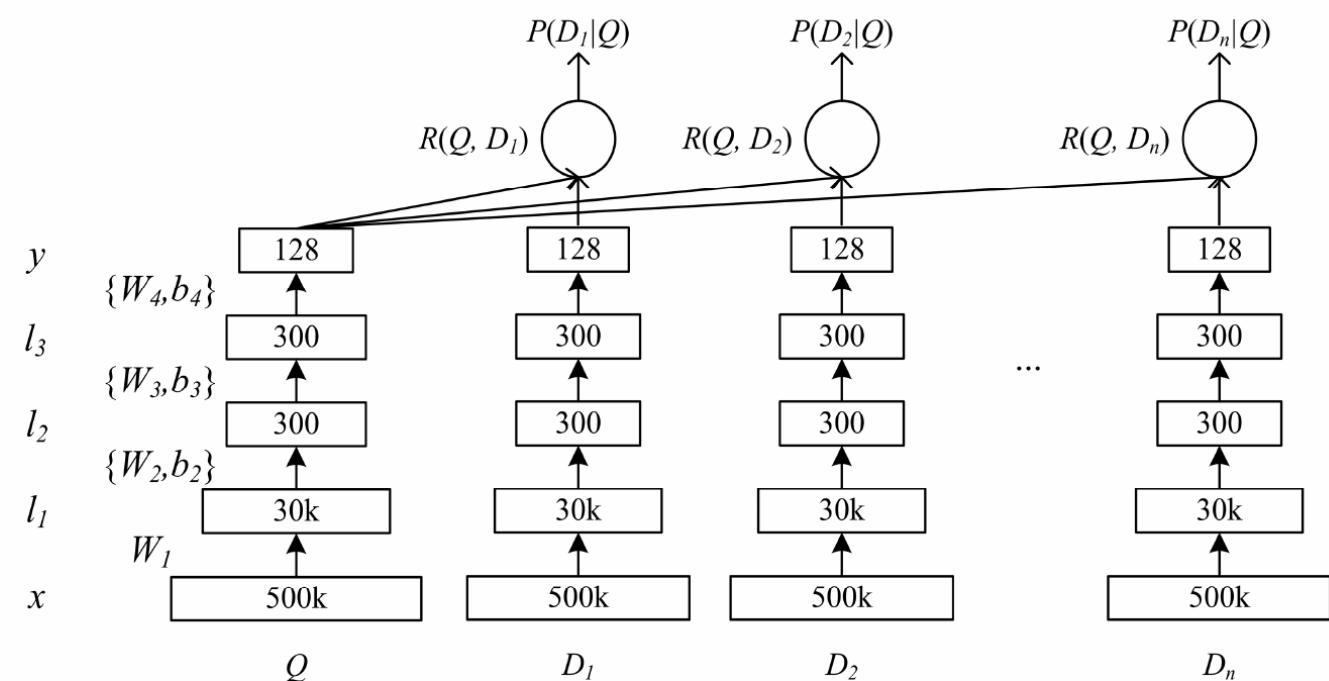
Relevance measured
by cosine similarity

Semantic feature

Multi-layer non-linear projection

Word Hashing

Term Vector



 Huang, P. et al. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data.

MODÈLES DENSES MODERNES

👉 Ils sont tous basés sur le même modèle

Un bon vieux modèle vectoriel basé sur un modèle transformer :

$$\hat{q} = BERT_{\text{CLS}}([\text{CLS}] \ q_1 \dots q_n [\text{SEP}])$$

et

$$\hat{d} = BERT_{\text{CLS}}([\text{CLS}] \ d_1 \dots d_m [\text{SEP}])$$

Le score est donné par

$$s(q, d) = \hat{q} \cdot \hat{d}$$

APPRENTISSAGE D'UN MODÈLE DENSE : TAS-BALANCED (2021)

 Hofstätter, S. et al. 2021. *Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling*.

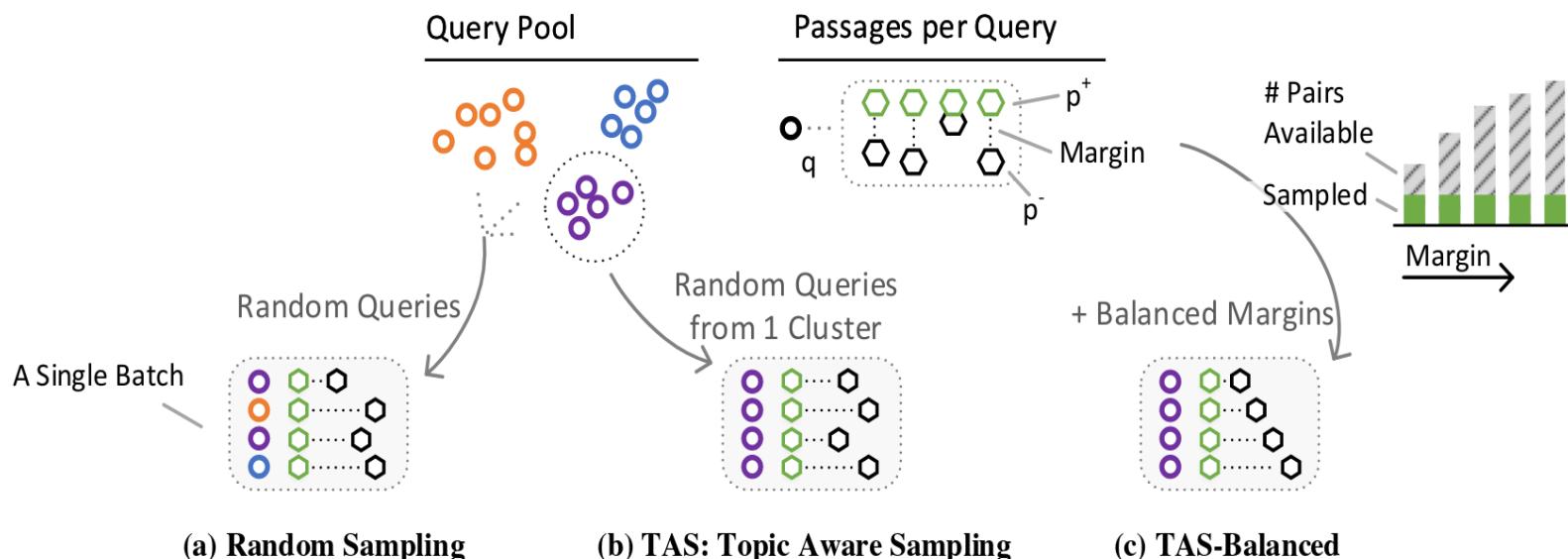


Figure 2: Comparison of batch sampling strategies. Each strategy has access to a pool of (clustered) queries; where each query has a set of relevant and non-relevant passage pairs with $BERT_{CAT}$ score margins.

- 👉 Utilisation de la représentation de [CLS] pour le document / la question
- 👉 Utilisation de la distillation (voir plus loin)
- 👉 Échantillonnage des questions intéressantes (mini-batch, voir plus loin)

INDICES DENSES

! Recherches dans un espace dense

Le calcul de produit scalaires est coûteux... Comment peut-on accélérer les recherches ?

CLUSTERING: CELL-PROBE METHODS (IVF, HNSW)

- 👉 On groupe les documents dans des clusters (éventuellement hiérarchique)
- 👉 enjeu = trouver le plus petit ensemble de clusters qui couvrent les "bons" documents

 *Malkov, Y.A. and Yashunin, D.A. 2018. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs.*

PRODUCT QUANTIZATION (PQ)

- 👉 Nous découpons le vecteur en k et utilisons un dictionnaire

$$\mathbf{x} = \hat{d}_{\hat{x}_1}^{(1)} \oplus \cdots \oplus \hat{d}_{\hat{x}_k}^{(k)}$$

où d est commun à l'ensemble des vecteurs et \hat{x}_i est un entier

- 👉 Le produit scalaire entre \mathbf{x} et \mathbf{y} est approximé par

$$\sum_{i=1}^k d_{\hat{x}_i}^{(i)} \cdot d_{\hat{y}_i}^{(i)}$$

 *Jegou, H. et al. 2011. Product quantization for nearest neighbor search.*

CONCLUSION

 Représentation compacte et sémantique

 Perte de précision

-  On combine avec un modèle standard (ex. BM25)
-  ... ou procédure d'apprentissage complexe

 Adaptation à d'autres domaines moins performante que d'autres modèles (interaction/parcimonieux)

 Temps de recherche long

-  Utilisation de bibliothèques permettant d'indexer des vecteurs ( FAISS) et en particulier les  consignes pour le choix du type d'index
-  Pas d'études du coût espace/temps à l'heure actuelle

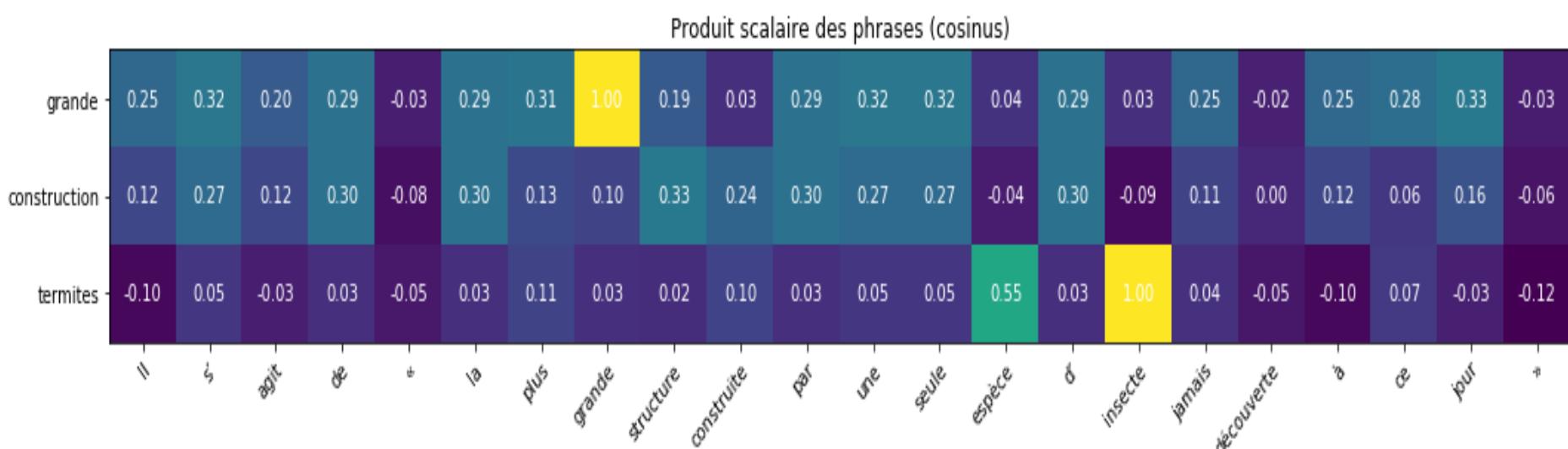
MODÈLES D'INTERACTION

MOTIVATION

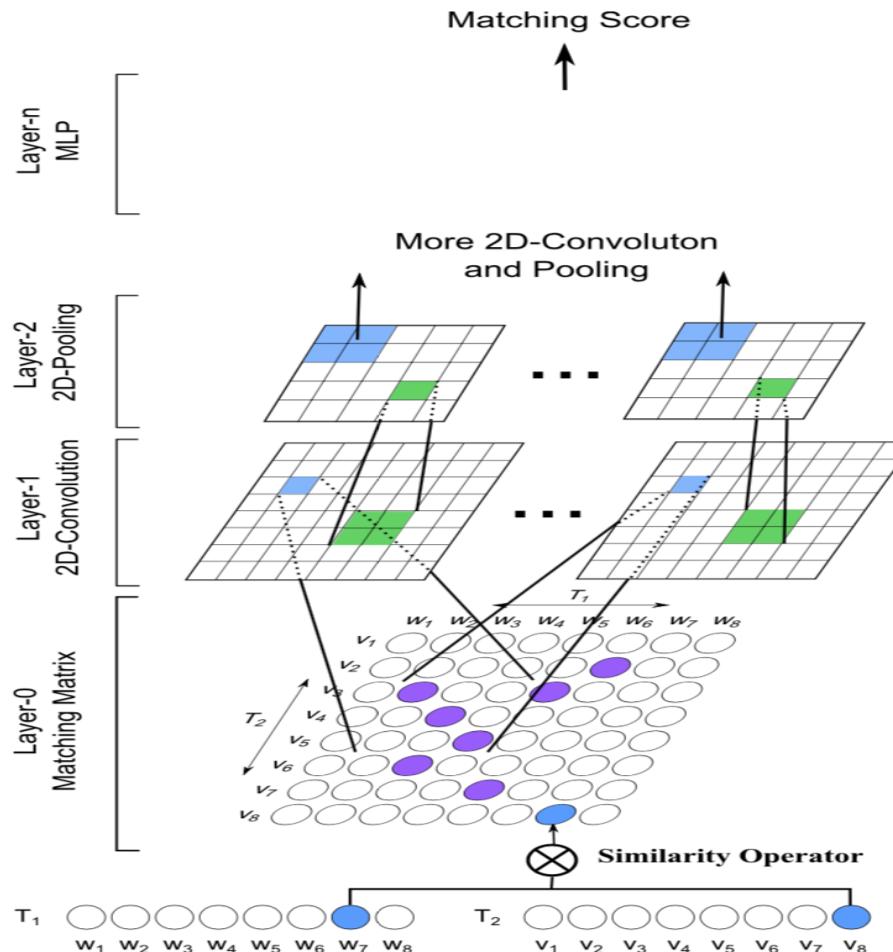
Construction d'une carte d'attention à partir des deux textes

Intérêt = identification des termes communs (exact match) et proches (semantic match)

Utilisation de techniques d'aggrégation 2D (convolution et/ou pooling)



MATCH PYRAMID

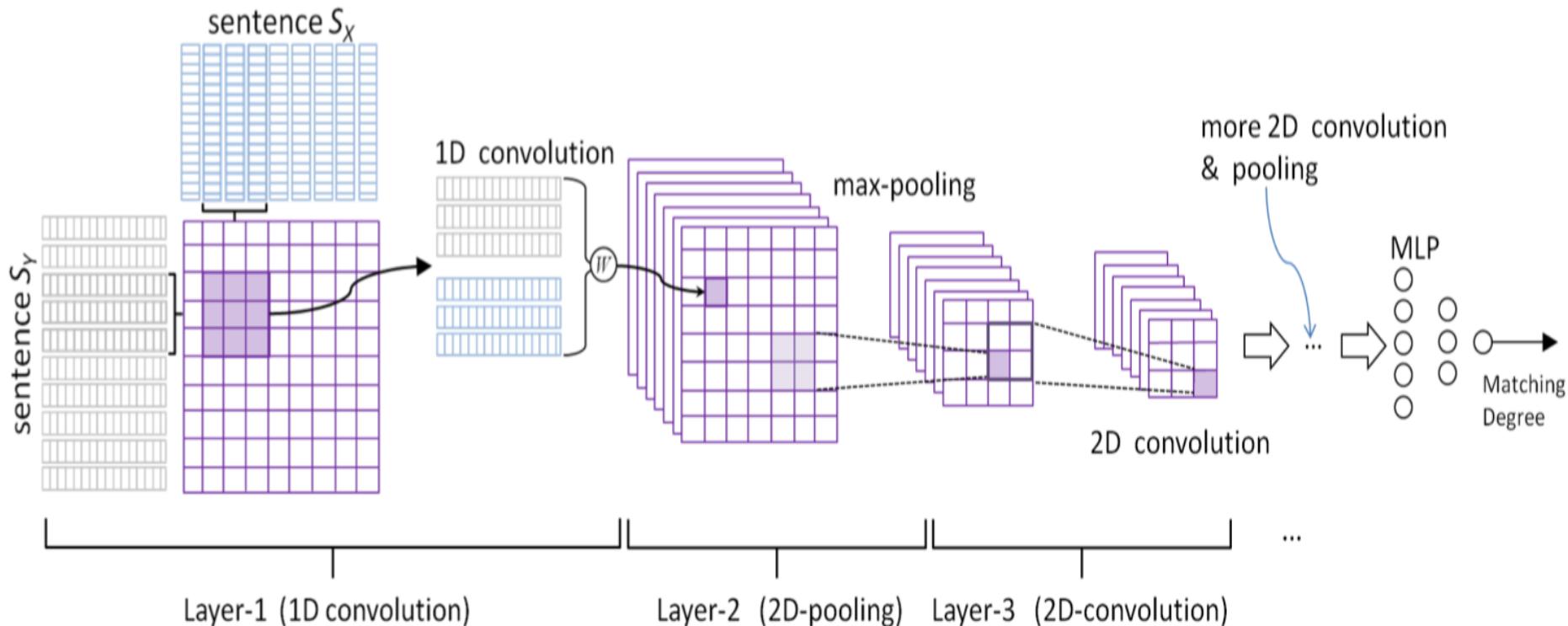


Utiliser les techniques de détection d'objets

- Matrice d'interaction
- Modèle de convolution 2D

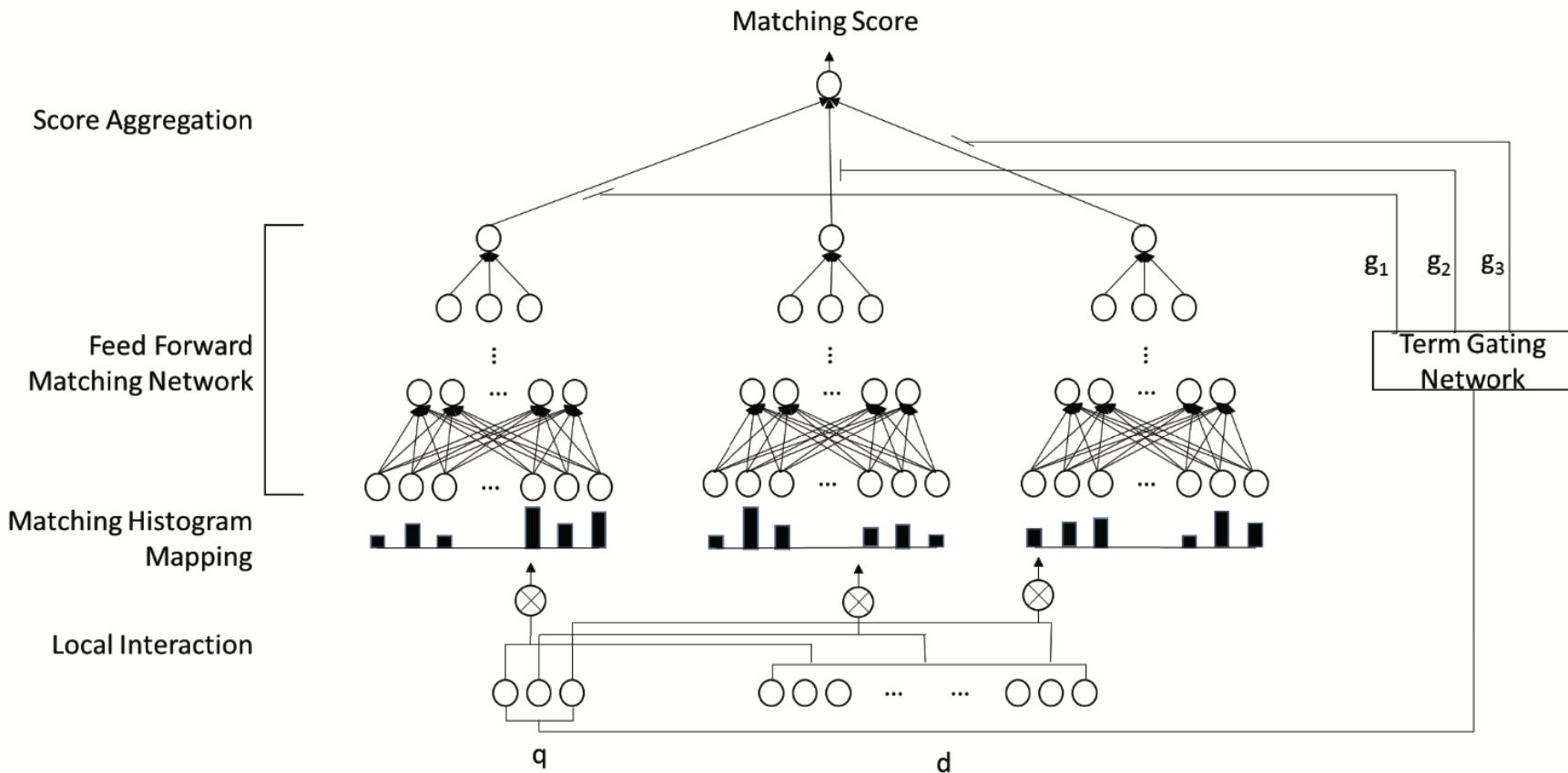
[] *Hu, B. et al. Convolutional Neural Network Architectures for Matching Natural Language Sentences. Advances in Neural Information Processing Systems 27.*

ARC-II

Matrice d'interaction *paramétrique*

 Hu, B. et al. Convolutional Neural Network Architectures for Matching Natural Language Sentences. *Advances in Neural Information Processing Systems* 27.

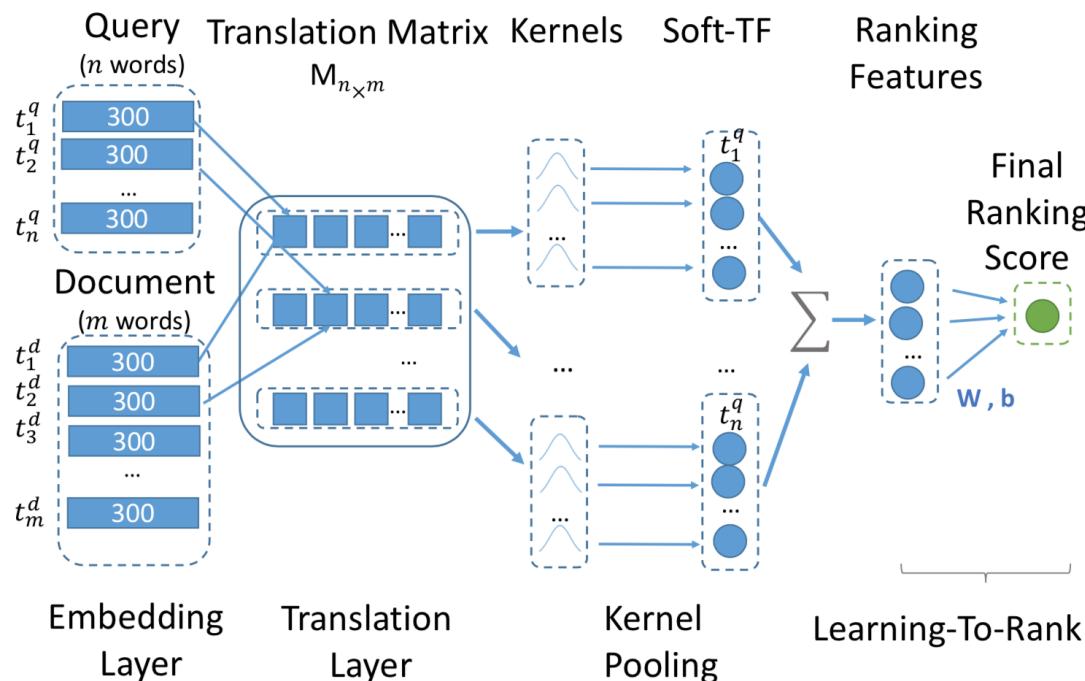
A DEEP RELEVANCE MATCHING MODEL (DRMM)



 Guo, J. et al. A Deep Relevance Matching Model for Ad-Hoc Retrieval.

KERNEL NEURAL RELEVANCE MODEL (KNRM)

👍 Un des meilleurs modèles d'interaction (avant BERT...)



Généralisation de DRMM

$$K_k(q_i \cdot d_j) = \exp\left(\frac{(I_{ij} - \mu_k)^2}{2\sigma_k^2}\right)$$

I_{ij}

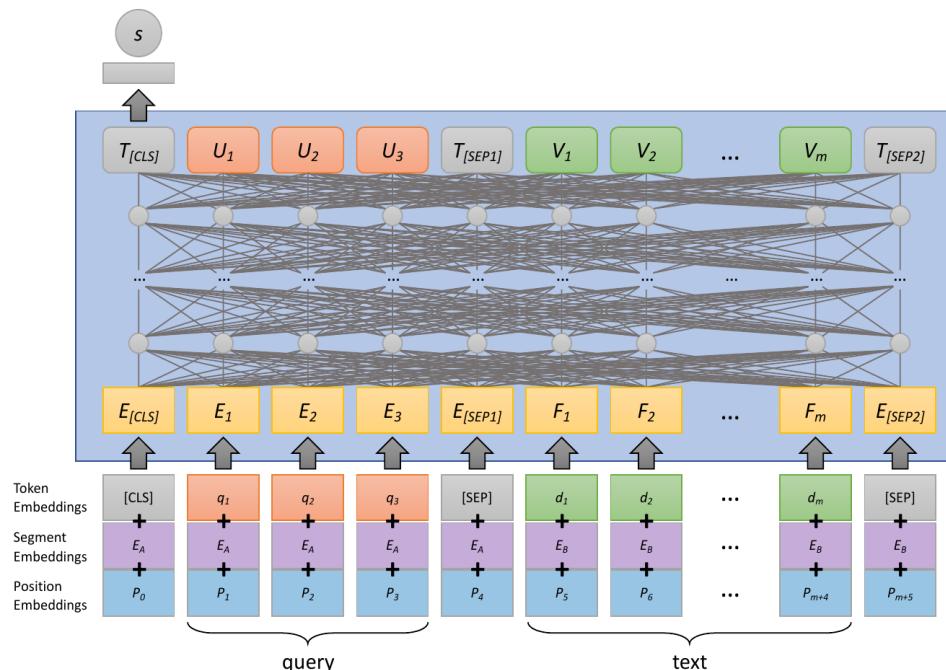
avec q_i la représentation du i ème terme de la question et d_j du j ème du document

 Xiong, C. et al. End-to-End Neural Ad-Hoc Ranking with Kernel Pooling.

TRANSFORMERS : CROSS-ENCODER (2019)

👍 On utilise BERT en concaténant question et document cible

$$rsv(q, d) = x^{[\text{CLS}]}([\text{CLS}] \text{ q } [\text{SEP}] \text{ d})$$



📘 Nogueira, R. and Cho, K. 2019. Passage Re-ranking with BERT.

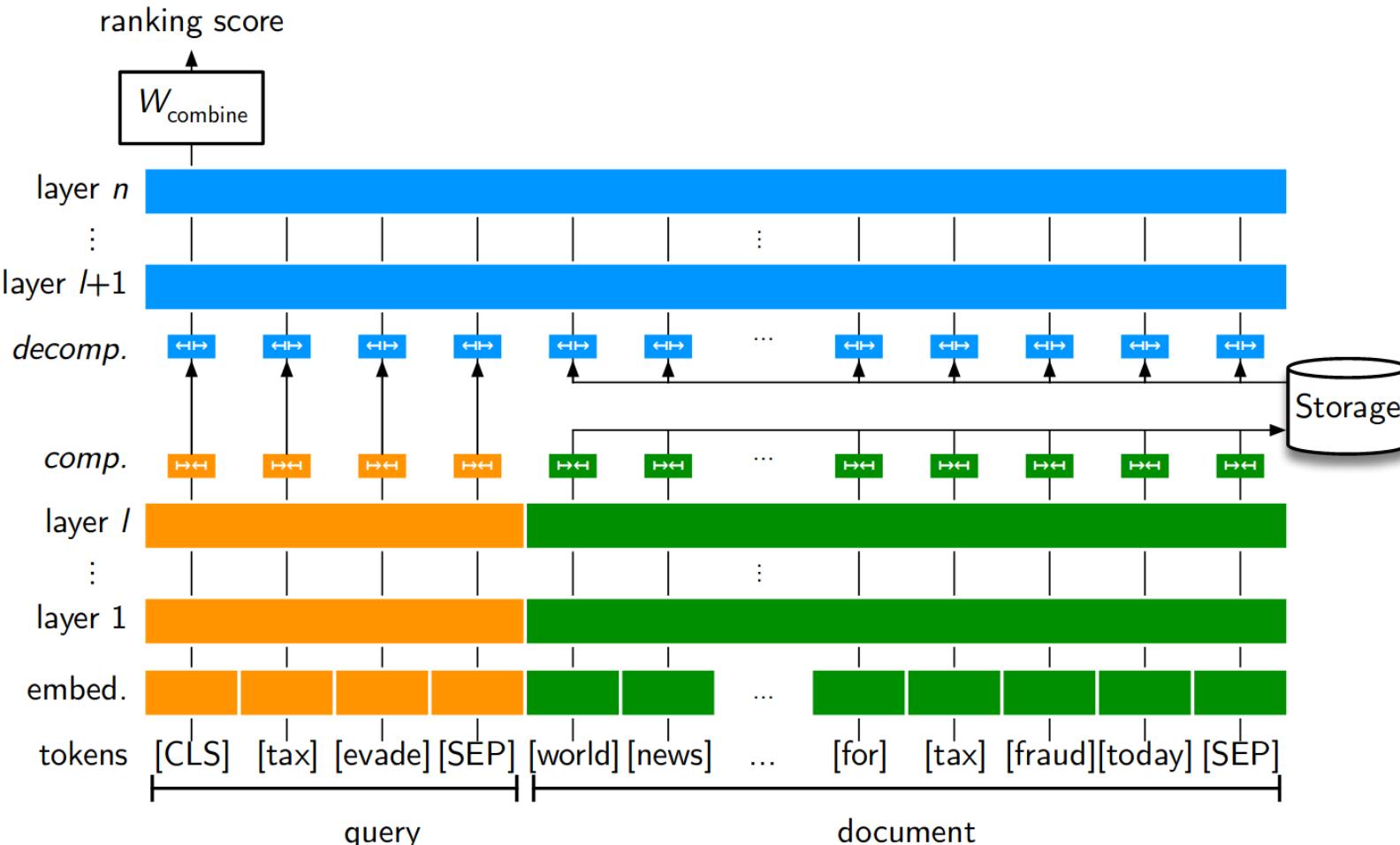
TRANSFORMERS : CROSS-ENCODER (2019) (2)

Method	TREC 2019 DL Passage		
	nDCG@10	MAP	Recall@1k
(3a) BM25 (Anserini, $k = 1000$)	0.5058	0.3773	0.7389
(3b) + monoBERT _{Large}	0.7383	0.5058	0.7389
(4a) BM25 + RM3 (Anserini, $k = 1000$)	0.5180	0.4270	0.7882
(4b) + monoBERT _{Large}	0.7421	0.5291	0.7882

 De très bon résultats...

 Mais très gourmand : des travaux récents permettent de réduire le temps de traitement substantiellement

CROSS-ENCODER - LATE INTERACTION (PRETTR)



CROSS-ENCODER - LATE INTERACTION (PRETTR) (2)

TREC WebTrack 2012

Compression	P@20					ERR@20				
	$l = 7$	$l = 8$	$l = 9$	$l = 10$	$l = 11$	$l = 7$	$l = 8$	$l = 9$	$l = 10$	* $l = 11$
(none)	0.3180	0.3140	0.3130	0.3360	0.3380	0.2255	0.2344	0.2297	0.2295	0.1940
$e = 384$ (50%)	0.3430	0.3260	0.2980	0.3360	0.3090	0.2086	0.2338	0.1685	0.2233	0.2231
$e = 256$ (67%)	0.3380	0.3120	↑ 0.3440	0.3260	0.3250	↑ 0.2716	0.2034	↑ 0.2918	0.1909	0.2189
$e = 128$ (83%)	0.3100	0.3210	0.3320	0.3220	0.3370	0.2114	0.2234	0.2519	0.2239	0.2130

TREC WebTrack 2012

Ranker	TREC WebTrack 2012					Robust04
	Total	Speedup	Query	Decom.	Combine	Total
Base	1.941s	(1.0×)	-	-	-	2.437s
$l = 1$	1.768s	(1.1×)	2ms	10ms	1.756s	2.222s
$l = 2$	1.598s	(1.2×)	3ms	10ms	1.585s	2.008s
$l = 3$	1.423s	(1.4×)	5ms	10ms	1.409s	1.792s
$l = 4$	1.253s	(1.5×)	6ms	10ms	1.238s	1.575s
$l = 5$	1.080s	(1.8×)	7ms	10ms	1.063s	1.356s
$l = 6$	0.906s	(2.1×)	9ms	10ms	0.887s	1.138s
$l = 7$	0.735s	(2.6×)	10ms	10ms	0.715s	0.922s
$l = 8$	0.562s	(3.5×)	11ms	10ms	0.541s	0.704s
$l = 9$	0.391s	(5.0×)	12ms	10ms	0.368s	0.479s
$l = 10$	0.218s	(8.9×)	14ms	10ms	0.194s	0.266s
$l = 11$	0.046s	(42.2×)	15ms	10ms	0.021s	0.053s

Performance et temps de la latence sur la collection TREC WebTrack 2012 (ré-ordonnancement de 100 documents)

 MacAvaney, S. et al. 2020. Efficient Document Re-Ranking for Transformers by Precomputing Term Representations. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.

CROSS-ENCODER - VARIATIONS (CEDR)

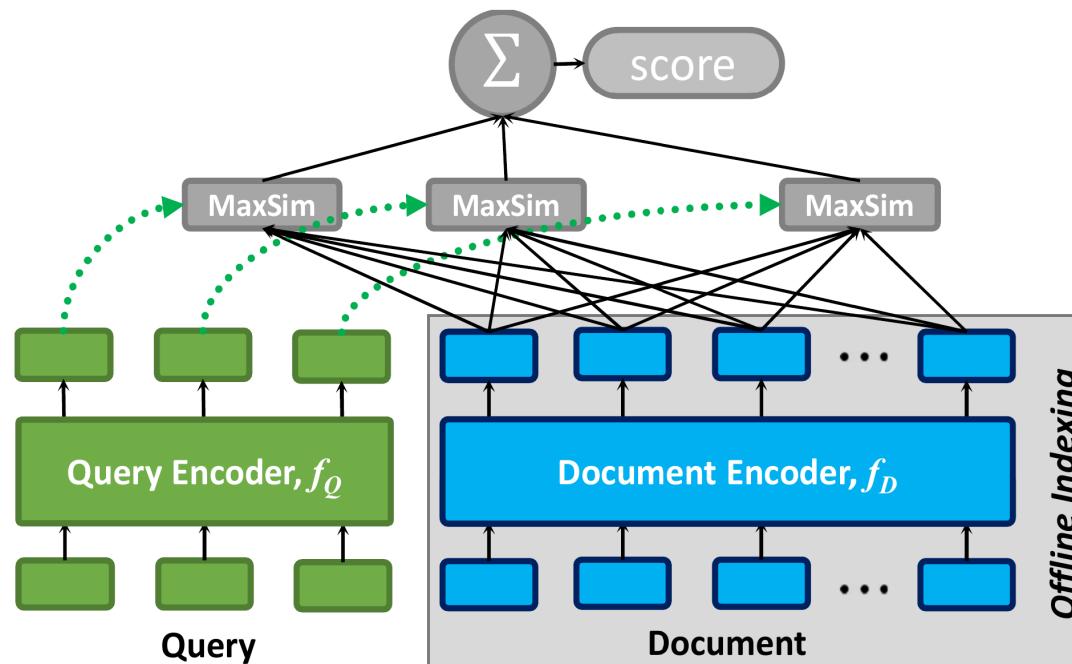
👉 Autre possibilité (marche moins bien) = Un modèle d'interaction... mais avec des embeddings contextualisés

Ranker	Input Representation	Robust04		WebTrack 2012–14	
		P@20	nDCG@20	nDCG@20	ERR@20
BM25	n/a	0.3123	0.4140	0.1970	0.1472
SDM [13]	n/a	0.3749	0.4353	-	-
TREC-Best	n/a	0.4386	0.5030	0.2855	0.2530
ConvKNRM	GloVe	0.3349	0.3806	[B] 0.2547	[B] 0.1833
Vanilla BERT	BERT (fine-tuned)	[BC] 0.4042	[BC] 0.4541	[BC] 0.2895	[BC] 0.2218
PACRR	GloVe	0.3535	[C] 0.4043	0.2101	0.1608
PACRR	ELMo	[C] 0.3554	[C] 0.4101	[BG] 0.2324	[BG] 0.1885
PACRR	BERT	[C] 0.3650	[C] 0.4200	0.2225	0.1817
PACRR	BERT (fine-tuned)	[BCVG] 0.4492	[BCVG] 0.5135	[BCG] 0.3080	[BCG] 0.2334
CEDR-PACRR	BERT (fine-tuned)	[BCVG] 0.4559	[BCVG] 0.5150	[BCVGN] 0.3373	[BCVGN] 0.2656
KNRM	GloVe	0.3408	0.3871	[B] 0.2448	0.1755
KNRM	ELMo	[C] 0.3517	[CG] 0.4089	0.2227	0.1689
KNRM	BERT	[BCG] 0.3817	[CG] 0.4318	[B] 0.2525	[B] 0.1944
KNRM	BERT (fine-tuned)	[BCG] 0.4221	[BCVG] 0.4858	[BCVG] 0.3287	[BCVG] 0.2557
CEDR-KNRM	BERT (fine-tuned)	[BCVGN] 0.4667	[BCVGN] 0.5381	[BCVG] 0.3469	[BCVG] 0.2772
DRMM	GloVe	0.2892	0.3040	0.2215	0.1603
DRMM	ELMo	0.2867	0.3137	[B] 0.2271	0.1762
DRMM	BERT	0.2878	0.3194	[BG] 0.2459	[BG] 0.1977
DRMM	BERT (fine-tuned)	[CG] 0.3641	[CG] 0.4135	[BG] 0.2598	[B] 0.1856
CEDR-DRMM	BERT (fine-tuned)	[BCVGN] 0.4587	[BCVGN] 0.5259	[BCVGN] 0.3497	[BCVGN] 0.2621

📘 MacAvaney, S. et al. 2019. CEDR: Contextualized Embeddings for Document Ranking. Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval.

COLBERT

- 👎 Encore plus simple = Modèle d'interaction basé sur une aggrégation (maximum)
- 👍 permet de construire un index !



[Khattab, O. and Zaharia, M. 2020. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*.

COLBERTv2 ET PLAID

Technique de compression

$$\hat{x}_{di} = \underbrace{z_{j(x_{di})}}_{\text{cluster}} + \underbrace{\tilde{x}_{di}}_{\text{résidu}}$$

 Santhanam, K. et al. 2021. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction.

PLAID = 3 filtrages de plus en plus précis, puis calcul complet (avec résidus)

System	MRR@10	R@100	R@1k	Latency (ms)		
				1-CPU	8-CPU	GPU
BM25 (PISA [34]; $k = 1000$)	18.7*	-	-	8.3*	-	-
SPLADEv2 (PISA; $k = 1000$)	36.8*	-	97.9*	220.3*	-	-
ColBERTv1	36.1	87.3	95.2	-	-	54.3
Vanilla ColBERTv2 ($p=2, c=2^{13}$)	39.7	90.4	96.6	3485.1	921.8	53.4
Vanilla ColBERTv2 ($p=4, c=2^{16}$)	39.7	91.4	98.3	-	4568.5	259.6
PLAID ColBERTv2 ($k = 10$)	39.4	-	-	185.5	31.5	11.5
PLAID ColBERTv2 ($k = 100$)	39.8	90.6	-	222.3	52.9	20.2
PLAID ColBERTv2 ($k = 1000$)	39.8	91.3	97.5	352.3	101.3	38.4

Table 3: End-to-end in-domain evaluation on the MS MARCO v1 benchmark. Numbers marked with an asterisk are copied from Formal et al. [11] for SPLADEv2 quality and Mackenzie et al. [30] for latencies.

System	Success@5	Success@100	Latency (ms)	
			CPU (8)	GPU
BM25	47.8*	77.6*	-	-
SPLADEv2	67.0*	89.0*	-	-
Vanilla ColBERTv2 ($p=2, c=2^{13}$)	69.3	90.3	1508.4	66.9
PLAID ColBERTv2 ($k = 10$)	69.1	-	35.5	9.2
PLAID ColBERTv2 ($k = 100$)	69.4	89.9	64.8	17.4
PLAID ColBERTv2 ($k = 1000$)	69.6	90.5	163.1	27.3

Table 5: End-to-end out-of-domain evaluation on the (dev) pooled dataset of the LoTTE benchmark. Numbers marked with an asterisk were taken from Santhanam et al. [42].

 Santhanam, K. et al. 2022. PLAID: An Efficient Engine for Late Interaction Retrieval. Technical Report #arXiv:2205.09707.

CONCLUSION

- + Les modèles d'interaction sont les plus performants
- C'est lourd - même si les travaux montrent qu'on peut beaucoup simplifier les modèles tout en gardant de bons résultats
- ⚠ (en général, sauf ColBERT) Mécanisme en deux temps : on sélectionne e.g. 1000 documents qu'on ré-ordonne

MODÈLES PARCIMONIEUX

PROBLÉMATIQUE



Les indices c'est efficace !

- 👉 Pourquoi ne pas utiliser des représentations parcimonieuses ?
- 👉 On peut ensuite ordonner l'**ensemble des documents**

DEEPCT, HDCT

- 👉 Idée = prédire le poids d'un terme
- 👉 On se base sur la représentation contextualisée + modèle (linéaire)

Table 5: Visualization of DeepCT-Index passage term weights. Red shades reflect the normalized term weights – the percentage of total passage term weights applied to the term. White indicates zero weight. Query terms are bold.

Percentage of weights a term takes in the passage: 0 10% 20% 30% 40% >50%	
Query	who is susan boyle
On-Topic	Amateur vocalist Susan Boyle became an overnight sensation after appearing on the first round of 2009's popular U.K. reality show Britain's Got Talent.
Off-Topic	Best Answer: a troll is generally someone who tries to get attention by posting things everyone will disagree, like going to a susan boyle fan page and writing susan boyle is ugly on the wall. they are usually 14-16 year olds who crave attention.
Query	what values do zoos serve
On-Topic	Zoos serve several purposes depending on who you ask. 1) Park/Garden: Some zoos are similar to a botanical garden or city park. They give people living in crowded, noisy cities a place to walk through a beautiful, well maintained outdoor area. The animal exhibits create interesting scenery and make for a fun excursion.
Off-topic	There are NO purebred Bengal tigers in the U.S. The only purebred tigers in the U.S. are in AZA zoos and include 133 Amur (AKA Siberian), 73 Sumatran and 50 Malayan tigers in the Species Survival Plan. All other U.S. captive tigers are inbred and cross bred and do not serve any conservation value .
Query	do atoms make up dna
On-Topic	DNA only has 5 different atoms - carbon, hydrogen, oxygen, nitrogen and phosphorous. According to one estimation, there are about 204 billion atoms in each DNA .
Off-Topic	Genomics in Theory and Practice. What is Genomics . Genomics is a study of the genomes of organisms. Its main task is to determine the entire sequence of DNA or the composition of the atoms that make up the DNA and the chemical bonds between the DNA atoms .

📘 Dai, Z. and Callan, J. 2019. Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval.

☒ Les mots absents ne sont pas ajoutés

DEEPCT (DÉTAILS)

- 👉 Le poids d'un terme est estimé par une transformation linéaire (poids w et biais b) de la représentation contextuelle $x^{(t)}$ donnée par BERT (pour la question q et le document d)

$$y_{t,d} = x^{(t)} \cdot w_1 + b_1$$

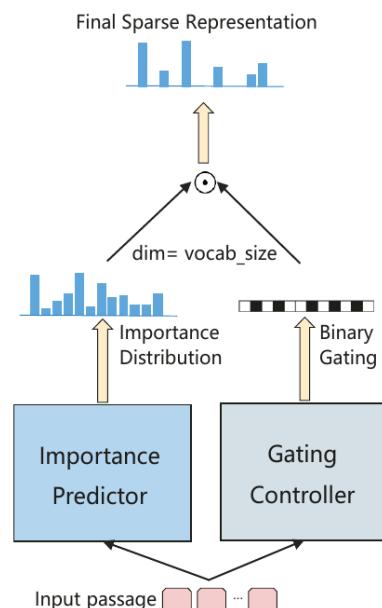
$$y_{t,q} = x^{(t)} \cdot w_2 + b_2$$

Ensuite $rsv(q, d) = \sum_{t \in q} y_{t,q} y_{t,d}$

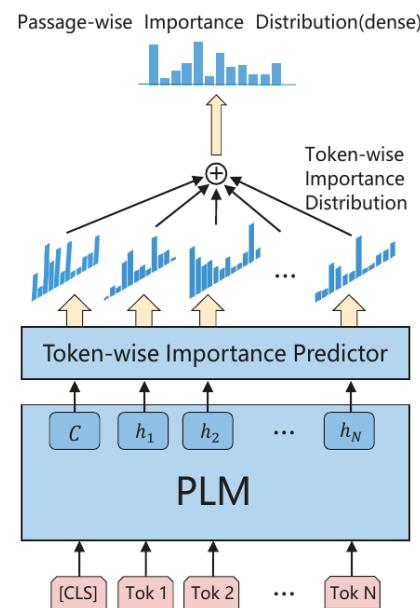
- 👉 Les valeurs cible y sont données par des estimations de pertinence sur un jeu de données d'entraînement

SPARTA, SPARTERM ET SPLADE

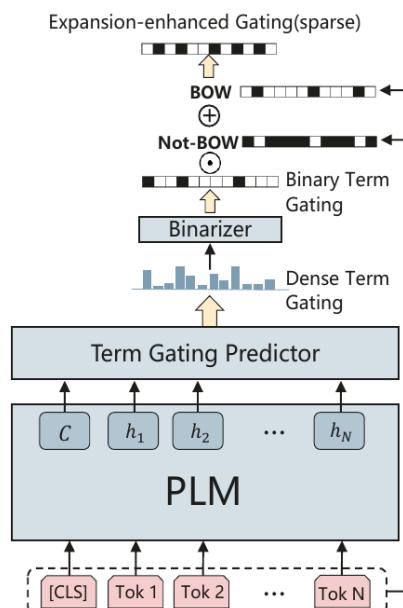
- 👉 On prédit des poids dans le vocabulaire pour chaque terme
- 👉 Mécanisme de pooling (= 1 poids par mot à la fin)



(a) SparTerm Model



(b) Importance Predictor



(c) Gating Controller

✉ Bai, Y. et al. 2020. SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval.

SPARTA, SPARTERM ET SPLADE - INTUITITON

👉 La tâche de MLM induit naturellement une forme d'expansion du vocabulaire

>> distilbert-base-uncased [Sanh et al. 2019]
prostate → (prostate, ., the)
cancer → (cancer, tumor, ##mour)
detection → (detection, identification, screening)
treatment → (., ;, treatment)

>> google/electra-base-generator [Clark et al. 2020]
prostate → (prostate, ##iate, mal)
cancer → (cancer, cancers, tumor)
detection → (detection, detecting, screening)
treatment → (treatment, therapy, treatments)

>> naver/splade-cocondenser-ensembledistil [Formal et al. 2021c]
prostate → (prostate, bp, ur)
cancer → (cancer, tumor, disease)
detection → (detection, detect, test)
treatment → (treatment, therapy, treated)

Table 3.1: MLM prediction for the Robust04 query “*prostate cancer detection treatment*”. For each token, we show the top-3 predicted tokens, i.e., with the highest MLM logits. Note that these models use the WordPiece vocabulary (Section 2.4.2). Also, note that only the generator part of ELECTRA has been pre-trained with MLM. We also include for comparison the prediction for one of our SPLADE model.

SPLADE (DÉTAILS)

On estime le poids w_{td} d'un terme t dans un document d avec

$$w_{td} = \sum_{i \in d} \text{ReLU}(y_i^{(t)})$$

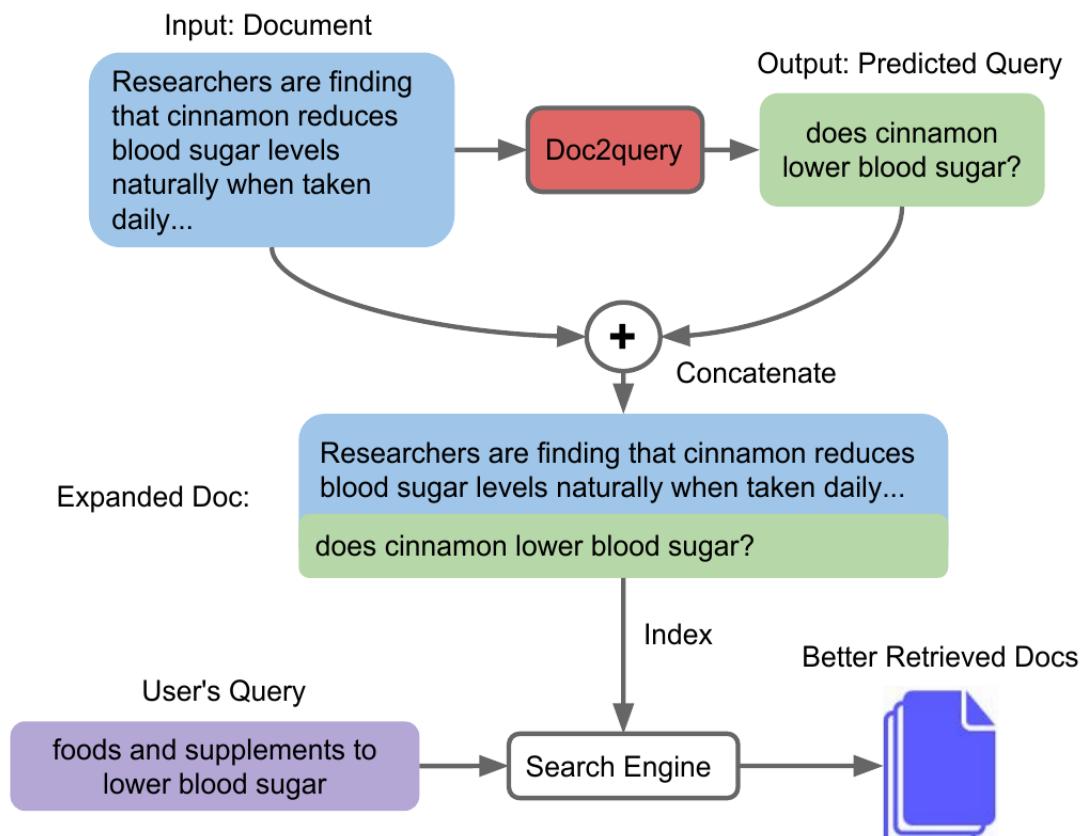
avec

$$y_i^{(t)} = \text{BERT}_i(d) \cdot \theta_t + b_t$$

où $\text{BERT}_i(d)$ est l'embedding contextualisé (BERT) du i ème token du document d

DOC2QUERY ET DOC2TTTTT

👉 Génération des questions auxquelles le document répond



📘 Nogueira, R. et al. 2019. Document Expansion by Query Prediction.
(Doc2TTTTT c'est juste avec T5 plutôt que BERT)

DEEPIMPACT



DocT5 (expansion du vocabulaire) + DeepCT (poids des termes)

Table 2: Effectiveness metrics and mean response time (MRT, in ms) for first-stage methods, on MSMARCO Dev Queries, TREC 2019 queries, and TREC 2020 queries. The symbol ∇ denotes a significant difference viz. DeepImpact

Strategy	NDCG@10	MRR@10	MAP	MRT
MSMARCO Dev Queries				
BM25	0.235 ∇	0.188 ∇	0.196 ∇	13.24
DeepCT	0.298 ∇	0.244 ∇	0.252 ∇	10.91
DocT5Query	0.338 ∇	0.278 ∇	0.286 ∇	12.62
DeepImpact	0.385	0.326	0.332	58.64
TREC 2019				
BM25	0.497 ∇	0.683	0.290 ∇	10.27
DeepCT	0.578 ∇	0.714	0.329 ∇	11.02
DocT5Query	0.648	0.799	0.405	11.76
DeepImpact	0.695	0.863	0.456	51.23
TREC 2020				
BM25	0.483 ∇	0.659 ∇	0.286 ∇	14.67
DeepCT	0.550 ∇	0.705	0.349 ∇	12.00
DocT5Query	0.619	0.742	0.408	15.51
DeepImpact	0.651	0.820	0.426	58.00

Table 3: Effectiveness metrics and mean response time (MRT, in ms) using several re-ranking techniques on MSMARCO Dev Queries, TREC 2019 queries, and TREC 2020 queries. ∇ denotes a significant difference viz. ColBERT E2E

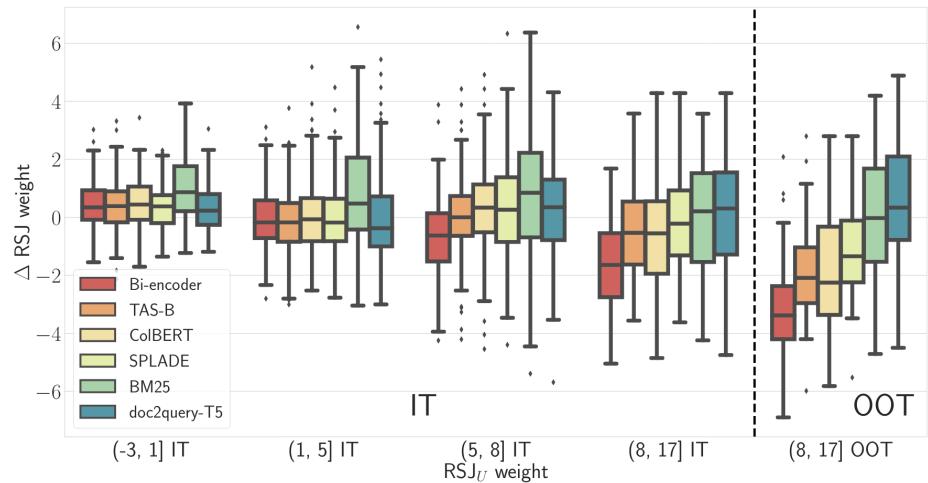
Strategy	NDCG@10	MRR@10	MRT
MSMARCO Dev Queries			
DeepImpact + EPIC	0.367 ∇	0.303 ∇	194.64
DeepImpact + ColBERT	0.425	0.362	81.00
ColBERT E2E	0.424	0.361	380.97
TREC 2019			
DeepImpact + EPIC	0.711	0.880	191.23
DeepImpact + ColBERT	0.722	0.826	73.29
ColBERT E2E	0.694	0.826	370.98
TREC 2020			
DeepImpact + EPIC	0.646	0.773	196.00
DeepImpact + ColBERT	0.691	0.781	79.84
ColBERT E2E	0.676	0.776	364.82

Mallia, A. et al. 2021. Learning Passage Impacts for Inverted Indexes.

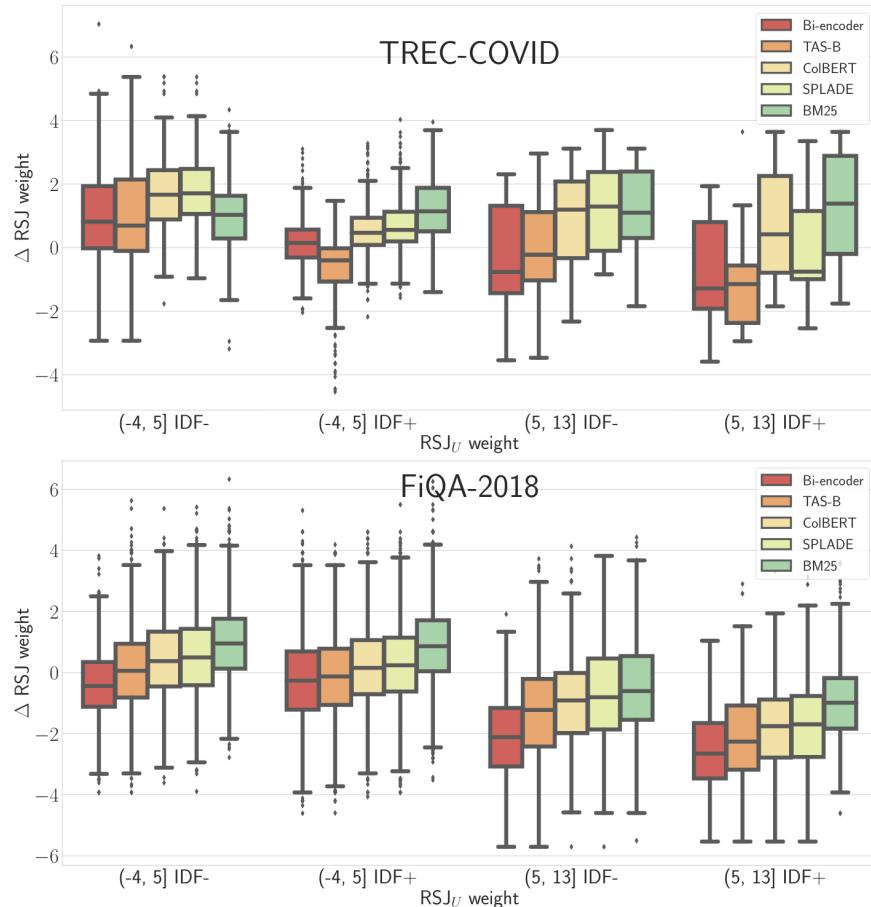
CONCLUSION

- ⊕ On arrive à de très bons résultats (bien meilleurs que BM25)
- ⊕ Modèles faciles à apprendre (SparTerm, SPLADE) ou à utiliser (DocTTTTT)
- ⊖ Pour arriver à l'état de l'art, il faut quand même ré-ordonner (e.g. avec monoBERT)

LE COMPROMIS SÉMANTIQUE / LEXICAL



👍 Delta du RSJ système vs modèle



📘 Formal, T. et al. 2022. Match your words! A study of lexical matching in neural information retrieval. 44th European Conference on Information Retrieval (ECIR)

MODÈLE COLBERTER

PARTIE SÉMANTIQUE

Utilisation d'un modèle dense

PARTIE LEXICALE

Calcul d'une représentation par mot

Projection éventuelle (dimension 1 = uni-ColBERTer)

Parcimonie : Masque avec régularisation (L_1)

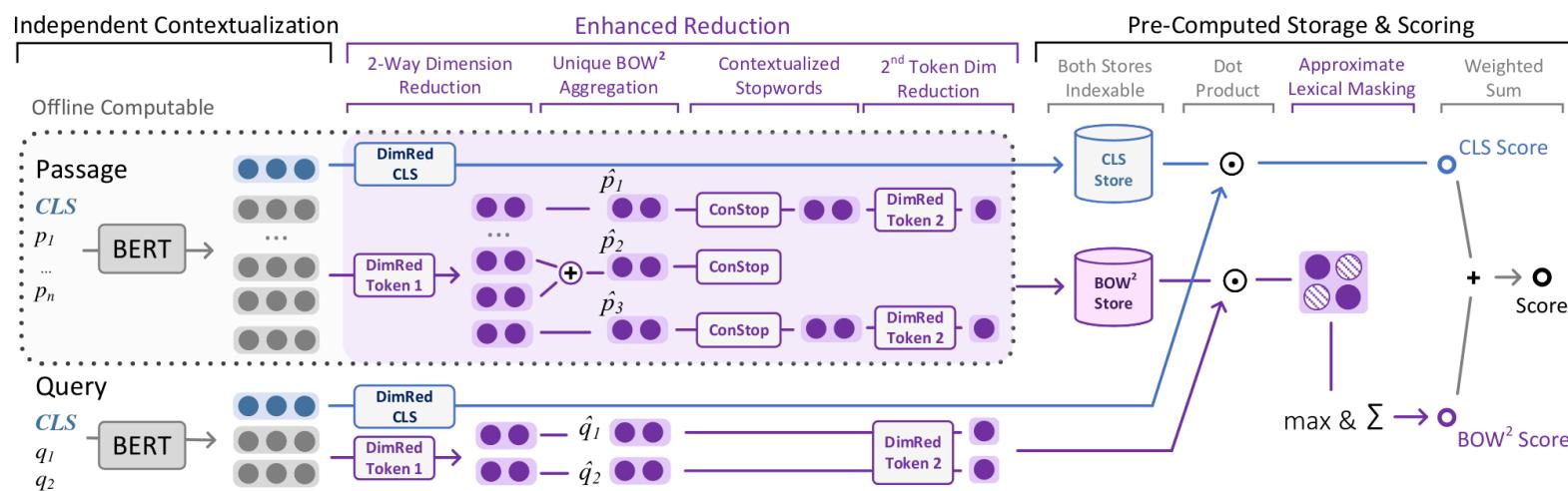


Figure 2: The ColBERTer encoding architecture, followed by the query-time workflow. The passage representations (both the single CLS and token vectors) are pre-computed during indexing time. The enhanced reductions with the 2-way dimension reduction, the unique BOW² aggregation, contextualized stopwords and token dimensionality reduction are applied symmetrically to passages and queries (except for the stopword removal).

Hofstätter, S. et al. 2022. Introducing Neural Bag of Whole-Words with ColBERTer: Contextualized Late Interactions using Enhanced Reduction

CONCLUSION

RÉSUMÉ

- 👍 Domaine très dynamique en ce moment, porté par BERT

ÉTAT DE L'ART = APPROCHES EN (DEUX) TEMPS

1. Filtrage
 1. BM25 (ou autre)...
 2. Méthodes neuronales denses + FAISS :
 3. Méthodes neuronales parcimonieuses : SparTerm, SPLADE, DocT5, ...
2. Ré-ordonnancement = Modèles d'interaction de type monoBERT optimisé (e.g. EPIC)

Autre option = approche multi-stage comme PLAID-ColBERT

- ⚠️ Documents longs : approche standard = on découpe puis maximum

APPRENTISSAGE

- 👍 Beaucoup de coûts possibles (mais la loss contrastive marche bien en général)
- 👍 Distillation / pré-entraînement

RÉFÉRENCES

- 👉 Sur la Recherche d'Information Neuronale (en particulier les loss)
📘 Mitra, B. and Craswell, N. 2017. *An Introduction to Neural Information Retrieval.*

- 👉 Sur la RI et les transformers (très complet... en 2020)
📘 Lin, J. et al. 2020. *Pretrained Transformers for Text Ranking: BERT and Beyond.*

