Vers un élagage de tokens sans coût dans les modèles de récupération à interaction tardive.

Yuxuan Zong[1] Benjamin Piwowarski[1]
(1) Sorbonne Université, CNRS, ISIR
yuxuan.zong@isir.upmc.fr
benjamin.piwowarski@cnrs.fr

RÉSUMÉ

Les modèles de RI neuronaux à interaction tardive comme ColBERT offrent un compromis compétitif entre efficacité et efficience sur de nombreuses bases de référence. Cependant, ils nécessitent un espace mémoire considérable pour stocker les représentations contextuelles de tous les tokens des documents. Certains travaux ont proposé d'utiliser soit des heuristiques, soit des techniques basées sur les statistiques pour élaguer des tokens dans chaque document. Cependant, cela ne garantit pas que les tokens supprimés n'aient aucun impact sur le score de récupération. Notre travail utilise une approche méthodique pour définir comment élaguer des tokens sans affecter le score entre un document et une question. Nous introduisons trois coûts de régularisation, qui induisent une solution avec des taux d'élagage élevés, ainsi que deux stratégies d'élagage. Nous les étudions expérimentalement (en domaine interne et externe), démontrant que nous pouvons préserver les performances de ColBERT tout en n'utilisant que 30% des tokens.

ABSTRACT

Towards Lossless Token Pruning in Late-Interaction Retrieval Models.

Late interaction neural IR models like ColBERT offer a competitive effectiveness-efficiency trade-off across many benchmarks. However, they require a huge memory space to store the contextual representation for all the document tokens. Some works have proposed using either heuristics or statistical-based techniques to prune tokens from each document. This however does not guarantee that the removed tokens have no impact on the retrieval score. Our work uses a principled approach to define how to prune tokens without impacting the score between a document and a query. We introduce three regularization losses, that induce a solution with high pruning ratios, as well as two pruning strategies. We study them experimentally (in and out-domain), showing that we can preserve ColBERT's performance while using only 30% of the tokens.

MOTS-CLÉS : Recherche d'information, Recherche dense, Recherche multi-vecteur, Compromis entre efficience et efficacité.

KEYWORDS: Information Retrieval, Dense Retrieval, Multi-vector Retrieval, Efficiency-Effectiveness Trade-off,.

ARTICLE : **Accepté à** SIGIR 2025.

1 Introduction et Travaux connexes

Les modèles de langage pré-entraînés basés sur l'architecture Transformer, comme BERT, ont révolutionné la recherche d'information (RI). Les bi-encodeurs (denses (Karpukhin *et al.*, 2020) ou parcimonieux (Formal *et al.*, 2021; Gao *et al.*, 2021)) et les modèles à interaction forte (e.g., MonoBERT (Nogueira & Cho, 2020)) représentent respectivement les extrêmes en termes d'efficacité et d'efficience. Les modèles dense *multi-vecteur* (MVDR) comme ColBERT (Khattab & Zaharia, 2020; Santhanam *et al.*, 2022b) utilisent les plongements contextualisés des tokens calculés séparément pour la question et le document, avant de calculer un score issu de leurs produits scalaires (Éq. (1)), offrant un compromis précision-vitesse via des index denses (Johnson *et al.*, 2021; Karpukhin *et al.*, 2020). Cependant, cette granularité entraîne un surcoût de stockage important car il faut stocker l'ensemble des représentations de *tous* les tokens contenus dans les documents 142 GiB (Khattab & Zaharia, 2020; Lassance *et al.*, 2022) pour MS MARCO (Bajaj *et al.*, 2018), dépassant largement les 13 GiB des encodeurs denses. La complexité numérique lors de l'inférence augmente aussi linéairement avec la longueur des documents, suggérant l'utilisation de stratégies de compression.

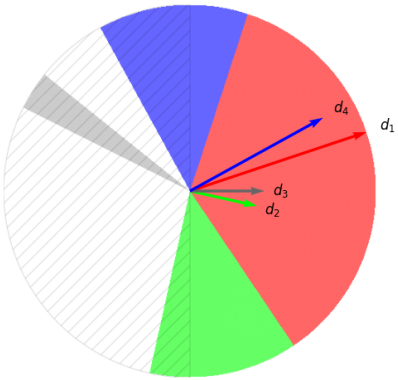

FIGURE 1 – Illustration du concept de dominance. Dans cet exemple, $\mathbf{d}_3$ est dominé par $\mathbf{d}_1$ et $\mathbf{d}_2$. $\mathbf{d}_3$ peut être supprimé de la représentation du document sans modifier une fonction de score ColBERT modifiée (voir Section 2.2). Malgré sa norme faible, $\mathbf{d}_2$ est conservé car il apporte une nouvelle information sur la pertinence du document. La zone hachurée correspond à la partie de l'espace où le produit scalaire de tout vecteur avec $\mathbf{d}_3$ est négatif.

Les travaux antérieurs sur la réduction des index MVDR peuvent être classifiés selon deux axes. Premièrement, les méthodes par *partitionnement des données* (*clustering*) comme PLAID (Santhanam *et al.*, 2022a) et EMVB (Nardini *et al.*, 2024) utilisent des centroïdes afin d'accélérer la recherche de documents pertinents, mais requièrent une recherche en plusieurs étapes. Deuxièmement, l'*élagage de tokens* supprime des tokens *avant indexation* via des approches statiques ou dynamiques. Les méthodes statiques utilisent des heuristiques basées sur l'IDF (Acquavia *et al.*, 2023), les scores d'attention (Lassance *et al.*, 2022), ou des critères positionnels (Liu *et al.*, 2024). Les performances se dégradent toutefois pour des taux d'élagage élevés (>50%). Les méthodes dynamiques, comme AligneR (Qian *et al.*, 2022) et ColBERTer (Hofstätter *et al.*, 2022) (agrégation de sous-mots), apprennent des représentations à élaguer mais ne formalisent pas la notion d'*élagage sans coût* — i.e. supprimer des tokens sans altérer les scores — comme nous le faisons dans cet article.

Nous montrons d'abord que l'élagage sans coût équivaut à un problème de Programmation Linéaire (Section 2.2), identifiant les tokens « dominants » dont la suppression n'affecte pas les scores (Section 2). Nous proposons ensuite des régularisations lors de l'entraînement et des stratégies d'élagage (Section 3), qui nous permettent d'atteindre jusqu'à 70% de tokens en moins avec un coût de performance minime (1,5% lorsque le modèle est entraîné sur le même jeu de documents que celui utilisé lors de l'apprentissage). Nous analysons aussi la généralisation de notre modèle(Section 5.3). Le code utilisé pour les expériences présentées dans cet article est accessible en ligne [1].

Questions de recherche et contributions **RQ1 :** *Peut-on formaliser l'élagage sans coût pour les modèles à interaction tardive ?* Nous définissons le concept de *dominance*(Section 2) et montrons qu'il s'agit d'un problème de Programmation Linéaire (PL) en Section 2.2). **RQ2 :** *Comment les régularisations et stratégies d'élagage affectent-elles le compromis efficacité-efficience ?* Nous proposons trois régularisations et deux stratégies (Section 3), avec des résultats compétitifs (Sections 5.1–5.2). **RQ3 :** *La généralisation hors domaine est-elle préservée ?* Nos méthodes surpassent les modèles de référence sur BEIR et LoTTE (Section 5.3).

2 Théorie de l'élagage pour ColBERT

2.1 Adaptation de ColBERT pour l'élagage

ColBERT calcule la pertinence d'un document pour une question via le score :

$$\mathrm{ColBERT}(\mathbf{Q}, \mathbf{D}) = \sum_{\mathbf{q} \in \mathbf{Q}} \max_{\mathbf{d} \in \mathbf{D}} \mathbf{q} \cdot \mathbf{d} \tag{1}$$

où $\mathbf{q}$ et $\mathbf{d}$ sont des vecteurs unitaires. Pour permettre l'élagage sans coût, on doit trouver un ensemble $\mathbf{D}^+ = \mathbf{D} \setminus \mathbf{D}^- \subseteq \mathbf{D}$ tel que

$$\forall \mathbf{q} \in \mathbb{R}^d, \max_{\mathbf{d} \in \mathbf{D}} \mathbf{q} \cdot \mathbf{d} = \max_{\mathbf{d} \in \mathbf{D}^+} \mathbf{q} \cdot \mathbf{d}$$

Pour permettre la suppression d'un plus grand nombre de tokens, nous proposons une version modifiée en utilisant une projection π_θ et un opérateur $[x]_+ = \max(x, 0)$:

$$\mathrm{ColBERT}_{\mathrm{P}}(\mathbf{Q}, \mathbf{D}) = \sum_{\mathbf{q} \in \mathbf{Q}} \max_{\mathbf{d} \in \mathbf{D}} [\pi_\theta(\mathbf{q}) \cdot \pi_\theta(\mathbf{d})]_+ \tag{2}$$

La projection π permet d'obtenir des vecteurs de norme inférieure ou égale à 1 (nécessaire pour élaguer) et $[x]_+$ élimine les interactions négatives, permettant de retirer des tokens sans changer le score ColBERT (Fig. 1). Contrairement aux approches utilisant des classifieurs binaires (Qian *et al.*, 2022; Hofstätter *et al.*, 2022), notre méthode identifie des tokens redondants même avec des normes non nulles.

1. https://github.com/yzong12138/MVDR_pruning

2.2 Dominance et Programmation Linéaire

Nous proposons la définition et le lemme suivants pour introduire la notion de dominance :

Definition 1 (Dominance Locale) *Un vecteur* $\boldsymbol{d}^- \in \mathbb{R}^d$ *est* dominée *par un ensemble de vecteurs* $\boldsymbol{D}$*, noté* $\boldsymbol{d}^- \prec \boldsymbol{D}$*, si et seulement si pour tout* $\boldsymbol{q} \in \mathbb{R}^d$*,*

$$\textit{soit } \boldsymbol{q} \cdot \boldsymbol{d}^- \leq 0, \textit{ soit } \exists \boldsymbol{d}^+ \in \boldsymbol{D} \textit{ tel que } \boldsymbol{q} \cdot \boldsymbol{d}^+ > \boldsymbol{q} \cdot \boldsymbol{d}^- \tag{3}$$

Selon cette définition, on peut partitionner un ensemble de vecteurs $\boldsymbol{D}$ en $\boldsymbol{D}^+$ et $\boldsymbol{D}^-$, en notant $\mathbf{D}^-$ l'ensemble de tous les vecteurs $\mathbf{d}$ dominés par $\boldsymbol{D}$.

Lemma 1 (Dominance globale) *Soit* $\mathbf{D}^+$ *et* $\mathbf{D}^-$ *une partition de l'ensemble des documents* $\mathbf{D} \subset \mathbb{R}^d$ *telle que tout vecteur* $\mathbf{d} \in \mathbf{D}^-$ *est dominé par* $\mathbf{D}$*. Formellement,*

$$\mathbf{D}^- = \{\mathbf{d} \in \mathbf{D} | \mathbf{d} \prec \mathbf{D}\} \textit{ and } \mathbf{D}^+ = \mathbf{D} \setminus \mathbf{D}^-$$

Alors, pour tout document $\mathbf{d}^- \in \mathbf{D}^-$*,* $\mathbf{d}^-$ *est dominé par* $\mathbf{D}^+$*.*

D'après ce lemme (preuve non incluse dans cet article), il suffit de tester la *dominance locale* pour définir l'ensemble des vecteurs qui peuvent être élagués sans impact sur le score de ColBERT$_P$.

Équivalence avec la programmation linéaire (PL) Pour identifier les tokens dominés, nous utilisons le lemme de Farkas (Garg & Mermin, 1984). Précisément, nour avons :

Lemma 2 (Lemme de Farkas) *Soit* $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ *et* $\mathbf{b} \in \mathbb{R}^m$*, alors exactement une des affirmations suivantes est vraie (les inégalités vectorielles signifient qu'elles doivent être vérifiées composante par composante) :*

1. *Il existe* $\boldsymbol{x} \in \mathbb{R}^n$ *tel que* $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *et* $\mathbf{x} \geq 0$*.*
2. *Il existe* $\mathbf{y} \in \mathbb{R}^d$ *tel que* $\boldsymbol{A}^\top \mathbf{y} \geq 0$ *et* $\boldsymbol{b}^\top \mathbf{y} < 0$*.*

Pour un token $\mathbf{d}$, on construit la matrice $\boldsymbol{A} = \begin{pmatrix} \mathbf{d} - \mathbf{d}_1 & \dots & \mathbf{d} - \mathbf{d}_n \end{pmatrix}$ et $\boldsymbol{b} = -\mathbf{d}$, où $\mathbf{d}$ est le vecteur que nous étudions son dominance et $\mathbf{d}_i \subset \mathbf{D}$. D'après le lemme de Farkas, l'existence d'un $\mathbf{x} \geq 0$ (chaque composante est supérieure ou égale à zéro) tel que

$$\begin{pmatrix} \mathbf{d} - \mathbf{d}_1 & \dots & \mathbf{d} - \mathbf{d}_n \end{pmatrix}^\top \mathbf{x} = -\mathbf{d} \tag{4}$$

est une condition nécessaire et suffisante pour que $\mathbf{d}$ soit dominé par $\mathbf{D}$ (Éq.3). Autrement dit, ce problème de Programmation Linéaire (PL) garantit que le token $\mathbf{d}$ ne contribue jamais au score maximum. Malheureusement, résoudre ce problème est coûteux en temps de calcul. Pour réduire ce coût, nous utilisons :

— Une suppression itérative des tokens dominés (via le Lemme 1),
— Une pré-sélection basée sur les similarités internes (20-30% des tokens dominants peuvent être identifiés sans PL).

Cette formalisation permet un élagage théoriquement garanti tout en restant praticable à grande échelle.

3 Méthodologie

Dans cette section, nous proposons une approche pratique pour entraîner et réduire le modèle ColBERT$_P$ en utilisant la théorie de dominance.

3.1 Élagage Approximative

La méthode d'élagage basée sur la programmation linéaire (LP) (Section 2.2) est exacte mais présente deux limitations :

— Certains vecteurs contiennent de petites composantes empêchant leur dominance.

— La résolution du problème est coûteuse.

Pour y remédier, nous utilisons une réduction de dimension et une heuristique simple basée sur la norme des vecteurs pour approximer la solution de LP.

3.1.1 Élagage Basée sur la Programmation Linéaire avec Réduction de Dimensions

Nous appliquons une Décomposition en Valeurs Singulières (SVD) pour projeter les vecteurs dans un espace de plus faible dimension, conservant les composantes communes et supprimant les résidus qui empêchent la dominance. Avec $\mathbf{D} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, nous réécrivons $\mathbf{d}_i$ sous la forme $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{v}_i$ (où $\boldsymbol{v}_i$ est la $i$ème ligne de $\boldsymbol{V}$). Reformuler la dominance dans cet espace améliore le conditionnement du problème LP (Cheung *et al.*, 2008; Freund & Vera, 2009; Cheung *et al.*, 2003) et accélère l'élimination des vecteurs :

$$\bar{\mathbf{D}} = \boldsymbol{\Sigma} \begin{pmatrix} \boldsymbol{v}_1 & \dots & \boldsymbol{v}_n \end{pmatrix} \tag{5}$$

Nous approximons ensuite la dominance en tronquant à $k < m$ dimensions, supprimant les moins utilisées :

$$\bar{\mathbf{D}}^{(k)} = \boldsymbol{\Sigma}^{(k)} \begin{pmatrix} \boldsymbol{v}_1^{(k)} & \dots & \boldsymbol{v}_n^{(k)} \end{pmatrix} \tag{6}$$

où k est choisi pour conserver la majorité des valeurs singulières, avec la condition :

$$\frac{tr(\boldsymbol{\Sigma}) - tr(\boldsymbol{\Sigma}^{(k)})}{tr(\boldsymbol{\Sigma})} \geq \theta_{LP}$$

On note que la dimension réduction est uniquement appliquée pour l'élagage : les vecteurs initiaux sont utilisés pour représenter un document.

3.1.2 Élagage Basé sur la Norme

Une alternative plus simple à l'élagage basé sur la LP est l'élagage utilisant une norme L_1, utilisé dans AligneR (Qian *et al.*, 2022) et ColBERTer (Hofstätter *et al.*, 2022), justifiée par le fait que les vecteurs dominés ont souvent une norme plus faible. Ce type d'élagage est rapide, et son impact sur le score est directement lié à la norme des vecteurs. Nous définissons θ_N comme seuil, et supprimons les tokens ayant une norme inférieure. Cependant, cette méthode ignore les relations contextuelles entre vecteurs. Comme illustré en Figure 1, $\mathbf{d}_3$ est conservé alors qu'il ne devrait pas, et la suppression de $\mathbf{d}_2$ peut altérer le score.

3.2 Régularisation

Le simple élagage (méthode heuristique ou statistique) d'un modèle ColBERT pré-entraîné peut être inefficace, car il n'est pas optimisé pour des taux d'élagage élevés. Comme l'élagage basé sur LP n'est pas facilement différentiable, nous introduisons trois techniques de régularisation pour bien pré-conditionner le problème.

3.2.1 Régularisation par Norme Nucléaire

Pour favoriser la dominance des tokens, nous encourageons les vecteurs à appartenir à un sous-espace commun en minimisant la norme nucléaire de $\mathbf{D}$, définie comme la norme L_1 de ses valeurs singulières :

$$\mathcal{L}_r^{(\text{nuc})} = \frac{1}{\min(n, d)} \|\mathbf{D}\|_{nuc} \tag{7}$$

Cela réduit la dimension du sous-espace et la norme des vecteurs, augmentant ainsi le taux d'élagage.

3.2.2 Régularisation par Similarité des Tokens

La régularisation par norme nucléaire étant instable et coûteuse, nous proposons une alternative basée sur la maximisation du produit scalaire entre les vecteurs – plus ce produit scalaire entre $\mathbf{d}$ et les vecteurs dans $\mathbf{D}$ est grand, plus il y a de chances que $\mathbf{d}$ soit dominé :

$$\mathcal{L}_r^{(\text{sim})} = -\frac{1}{n(n-1)} \sum_{\mathbf{d} \in \mathbf{D}} (1 - \|\mathbf{d}\|_2) \sum_{\mathbf{d}' \in \mathbf{D} \setminus \{\mathbf{d}\}} \frac{[\mathbf{d} \cdot \mathbf{d}']_+}{\|\mathbf{d}\|_2 + \varepsilon} \tag{8}$$

où ε prévient l'instabilité numérique ($\varepsilon = 0.01$ dans nos expériences), et où

- nous normalisons par la norme du vecteur $\|\mathbf{d}\|_2$ pour éviter que la norme de $\mathbf{d}$ augmente (ce qui empêcherait la dominance),
- nous considérons uniquement les produits scalaires positifs,
- nous faisons en sorte de donner plus d'importance dans le calcul de cette régularisation aux vecteurs de faible norme $(1 - \|\mathbf{d}\|_2)$.

3.2.3 Régularisation par Norme L1

Inspirée de ColBERTer (Hofstätter *et al.*, 2022), la régularisation par norme L1 favorise la parcimonie, facilitant l'élagage puisque les vecteurs de norme nulle sont automatiquement dominés :

$$\mathcal{L}_r^{(L1)} = \frac{1}{n} \sum_{\mathbf{d} \in \mathbf{D}} \|\mathbf{d}\|_1 \tag{9}$$

3.3 Entraînement

Notre fonction de perte principale est celle de ColBERTv2 (Santhanam *et al.*, 2022b), combinant distillation et infoNCE. Nous utilisons des exemples négatifs difficiles issus d'un mono-encodeur [2], formant des triplets avec une question q, un passage positif p^+, un négatif difficile p^- et des négatifs faciles du batch P^-_{easy}.

Suivant (Santhanam *et al.*, 2022b; Ren *et al.*, 2021; Hofstätter *et al.*, 2021; Li *et al.*, 2023; Lin *et al.*, 2021; Hinton *et al.*, 2015), nous utilisons la divergence KL pour distiller le score du modèle enseignant dans le modèle étudiant. La probabilité que p_+ soit un passage positif est donnée par :

$$p(p_+|q,N) = \frac{e^{s(q,p_+)}}{e^{s(q,p_+)} + \sum_{p \in N} e^{s(q,N)}},$$

ce qui définit le coût RI comme :

$$\mathcal{L}_{IR} = KL(p_{ColBERT_P}(.|q,N_h)||p_{teacher}(.|q,N_h)) + \log p_{ColBERT_P}(p_+|q,N_a) \quad (10)$$

où N_a regroupe tous les types de documents et N_h contient les paires positives et négatives difficiles.

La fonction de perte finale combine une fonction de perte RI ainsi que les régularisations proposées plus haut :

$$\mathcal{L}_{final} = \mathcal{L}_{IR} + \alpha \mathcal{L}_r \quad (11)$$

où α est le coefficient de régularisation et $\mathcal{L}_r$ est l'une des fonctions définies en Section 3.2.

4 Expérience

4.1 Configuration

Données Nous entraînons le modèle sur MS MARCO v1 (Bajaj *et al.*, 2018) (8,8 M passages) et l'évaluons sur MS MARCO dev (6 980 questions), TREC DL 2019 (43 questions) et TREC DL 2020 (54 questions). Pour étudier la performance en transfert de domaine, nous utilisons des sous-ensembles de BEIR (Thakur *et al.*, 2021) et LoTTE (Santhanam *et al.*, 2022b).

Pipeline Nous utilisons une approche de ré-ordonnancement en utilisant SPLADEv2 (Formal *et al.*, 2021) pour pré-sélectionner un ensemble de documents. Nous n'utilisons pas l'approche de ColBERT, car SPLADE est un moteur de recherche efficace et les interactions par centroïde de PLAID/EMVB ne conviennent pas aux tokens non normalisés. Cela réduit aussi le coût expérimental (indexation) qui aurait été nécessaire pour évaluer différentes stratégies de pruning.

Pour l'évaluation, nous ré-ordonnons les 100 premiers passages sur MS MARCO dev, BEIR et LoTTE, et les 1000 premiers pour TREC DL.

2. Nous utilisons le même jeu de données que ColBERTv2.

Implémentation Les expériences sont réalisées sur un NVIDIA TESLA V100 avec PyTorch 2.0.1, HuggingFace transformers 4.37 et XPM-IR 1.3.1 (Zong & Piwowarski, 2023). Nous entraînons pendant 320k étapes avec AdamW (lr=1e-5, batch=16), en validant tous les 8k étapes sur 1000 questions provenant de MS MARCO. Le meilleur checkpoint est sélectionné en utilisant une moyenne harmonique pondérée, i.e. F0.5, F1 et F2, entre MRR@10 et un ratio de pruning estimé sur 1024 documents avec pruning LP basé sur une proportion de valeurs singulières cumulative de 0.7.

L'entraînement dure environ 5 jours pour la régularisation L1/similarité et environ 8 jour pour la régularisation par norme nucléaire (sur un seul GPU).

Pour la projection π_θ, au lieu d'une activation scalaire comme dans (Qian *et al.*, 2022; Hofstätter *et al.*, 2022), nous projetons la dernière couche de BERT ($\mathbf{d}^{BERT}$) dans un espace de dimension supérieure à d, la normalisons et conservons les d premières dimensions :

$$\pi_\theta(\mathbf{d}^{BERT}) = \text{normalize}\left(\begin{pmatrix} \boldsymbol{W}_1 \\ \boldsymbol{W}_2 \end{pmatrix} \mathbf{d}^{BERT}\right)_{1\ldots d}$$

où $\boldsymbol{W}_1$ est initialisé depuis ColBERTv2 et $\boldsymbol{W}_2$ (dim=32) est initialisé aléatoirement.

Référence Nous les modèles de base suivants :

- BM25 (Robertson *et al.*, 1994), Une référence RI standard,
- SPLADEv2 (Formal *et al.*, 2021), notre moteur de recherche de premier niveau [3]
- ColBERTv2 (Santhanam *et al.*, 2022b), en utilisant le processus complet de ColBERTv2 (i.e. avec indexation) ;
- ColBERTv2 (Santhanam *et al.*, 2022b) avec ré-ordonnancement (des documents retournés par SPLADEv2) ;

Notre second ensemble de modèles de référence concerne les modèles interaction tardive qui effectuent de l'élagage sur les vecteurs de documents – notez que toutes les méthodes statiques de la partie élagage sont basées sur ColBERTv2 (Santhanam *et al.*, 2022b) :

- AligneR (Qian *et al.*, 2022), une méthode qui utilise un coût régularisé par l'entropie pour supprimer des tokens. Nous utilisons uniquement leurs résultats correspondant à l'utilisant de BERT-base (Devlin *et al.*, 2019) pour garantir la comparabilité avec notre approche [4] ;
- Élagage *IDF* (Liu *et al.*, 2024), qui supprime les tokens ayant l'IDF (basé sur les tokens) le plus faible. Nous reprenons leurs résultats pour un taux d'élagage de 50% ;
- *First-p* (Liu *et al.*, 2024), qui conserve uniquement les p premiers tokens. Là encore, nous reprenons leurs résultats pour un taux de l'élagage de 50% ;
- Élagage *Stopwords* (Acquavia *et al.*, 2023), qui supprime les tokens issus d'une liste prédéfinie de mots (stop-words) ;
- Élagage basée sur l'*Attention* (Liu *et al.*, 2024; Lassance *et al.*, 2022) (voir Section 1) ;
- *ColBERTer* (Hofstätter *et al.*, 2022), un modèle ColBERT basé sur DistilBERT, où les tokens sont d'abord agrégés en mots avant d'être supprimés.

Parmi tous les résultats rapportés dans ces articles, nous avons sélectionné soit la configuration recommandée par les auteurs (AligneR, ColBERTer), soit le meilleur compromis entre taux d'élagage et performance (pour IDF, First-p, Stopwords).

3. Nous n'avons pas utilisé SPLADEv3 (Lassance *et al.*, 2024) car sa stratégie d'entraînement est plus complexe que celle utilisée dans cet article.

4. AligneR est entraîné sans distillation.

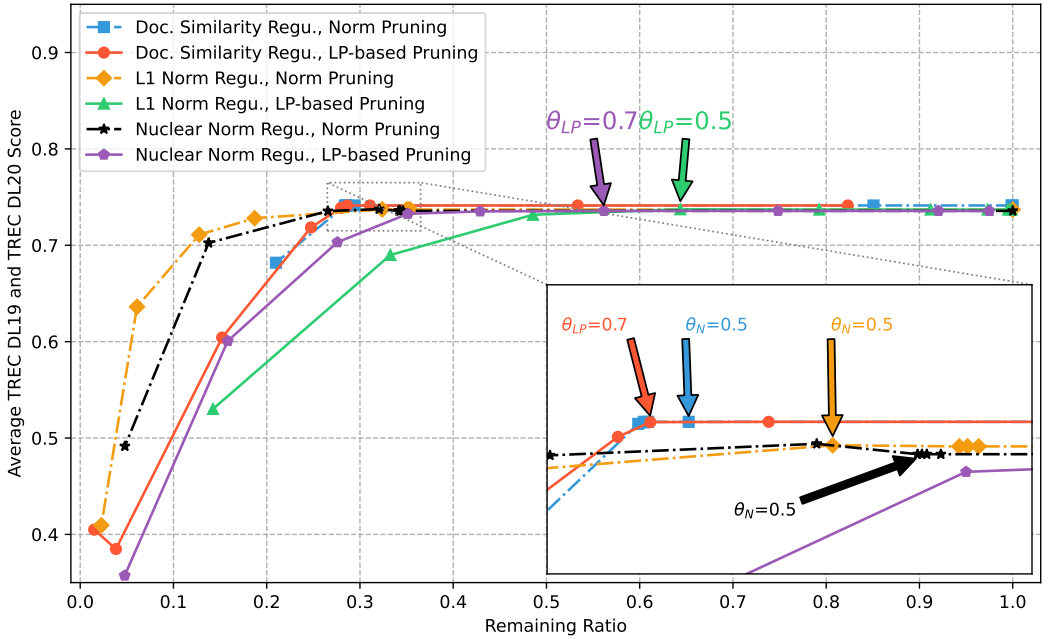

FIGURE 2 – Moyenne de TREC DL (2019 et 2020) nDCG@10 pour différentes régularisations et ratios d'élagage.

5 Résultats et Analyse

5.1 Analyse du ratio d'élagage (en domaine)

Nous analysons tout d'abord l'impact des seuils d'élagage sur le ratio d'élagage et les résultats d'évaluation, en utilisant nos stratégies norme L1 (seuil entre 0 à .999) et LP (seuil de .2 à .95). Les poids de régularisation sont constants pour chaque stratégie ($\alpha = 0.01$ pour la norme L1, $\alpha = 0.8$ pour la Similarité de Document, et $\alpha = 0.1$ pour la norme nucléaire), assurant une performance similaire sans élagage.

Dans la figure 2, la performance reste stable jusqu'à 30-40% de tokens restants, après quoi une chute de performance est observée. Les meilleurs résultats sont obtenus avec la régularisation basée sur la similarité de document et l'élagage basé sur LP ou basé sur la norme. Pour un élagage plus important, les régularisations norme L1 ou norme nucléaire, avec élagage basé sur la norme, sont les plus performantes. La régularisation $\mathcal{L}_r^{(sim)}$ est proche de l'élagage LP, tandis que $\mathcal{L}_r^{(nuc)}$ et $\mathcal{L}_r^{(L1)}$ montrent une différence plus grande, indiquant qu'elles s'éloignent de notre approche basée sur LP.

5.2 Recherche en Domaine

Dans le tableau 1, nous présentons les résultats d'évaluation en domaine, en comparant nos méthodes avec les modèles de référence. Nous utilisons $\theta_{LP} = 0.7$ et $\theta_N = 0.5$ sur la base des résultats

précédents. Pour chaque modèle, nous sélectionnons le checkpoint avec la meilleure valeur F2, équilibrant performance d'évaluation et ratio d'élagage.

Notre meilleur modèle (coût de similarité $\mathcal{L}_r^{(sim)}$ avec $\alpha = 0.8$ et élagage LP avec $\theta_{LP} = 0.7$) est comparable au modèle ColBERTv2 sur les ensembles dev-small (MRR@10 de 39.7 vs. 40.0) et DL19 (nDCG@10 de 73.8 vs. 74.4) tout en conservant seulement 40% des tokens des documents. Sur DL20, la performance baisse légèrement (nDCG@10 de 73.3 vs. 75.6), mais reste au-dessus de SPLADEv2 (68.7).

Nos modèles surpassent les méthodes d'élagage statiques (par ex. IDF, Attention Score) et dynamiques (par ex. AligneR), tant en efficacité qu'en ratio d'élagage.

Parmi nos modèles, le type de régularisation impacte directement l'élagage : augmenter α mène à un ratio d'élagage plus important. L'élagage norme L1 fonctionne bien avec la régularisation nucléaire et norme L1, tandis que l'élagage LP est plus avantageux avec la régularisation basée sur la similarité, suggérant que cette dernière est plus lié à la notion de dominance que les autres.

5.3 Évaluation Hors Domaine

Nous évaluons nos modèles sur des ensembles de données hors domaine (BEIR et LoTTE) pour tester la généralisation. Les meilleures configurations sont (1) la régularisation norme L1 ($\alpha = 0.01$) avec $\theta_N = 0.5$ et (2) la régularisation de similarité de document ($\alpha = 0.8$) avec $\theta_{LP} = 0.7$. Nous incluons également la régularisation de similarité de document ($\alpha = 0.1$) avec $\theta_{LP} = 0.7$, qui donne les meilleurs résultats (performance RI) sur MS MARCO dev-small.

Les résultats sont présentés dans les tableaux 2 et 3. AligneR (Qian *et al.*, 2022) n'est pas inclus car il ne rapporte que des résultats sans élagage pour BEIR et LoTTE. Nos modèles sont légèrement moins performants que la version avec ré-ordonnancement de ColBERTv2 (nDCG@10 45.9 vs. 47.0 sur BEIR et Success@5 71.3 vs. 72.0 sur LoTTE), mais restent compétitifs avec 60% des tokens conservés.

De plus, l'élagage est moins important hors domaine par rapport aux tâches en domaine (par exemple, 39% sur BEIR vs. 33% sur MS MARCO). Cela est attendu car la régularisation affecte la distribution des tokens, et le modèle apprend à prioriser les tokens en domaine. Néanmoins, la performance reste bien meilleure que pour le modèle avant ré-ordonnancement (SPLADEv2).

5.4 Analyse de l'interpretabilité

Dans le tableau 4, nous montrons un exemple d'élagage pour un passage de MS MARCO v1. Nous observons que les tokens restants sont principalement des noms et des verbes clés, ce qui est cohérent avec l'idée que ces tokens jouent un rôle important dans l'appariement sémantique. Pour la régularisation basée sur la similarité, l'élagage basé sur LP supprime la deuxième occurrence de "Population", tandis que l'élagage basé sur la norme ne le fait pas, ce qui met en évidence l'avantage de l'élagage basé sur LP. De même, "city" est éliminé avec l'élagage basé sur LP, car sa signification est déjà couverte par "New Delhi".

TABLE 1 – Les résultats de l'évaluation en domaine (MRR@10 pour dev small, nDCG@10 pour TREC DL19 et DL20) et le ratio de tokens restant (pour les modèles d'élagage). Nous mettons en évidence les modèles et la stratégie d'élagage correspondante utilisés dans nos expériences ultérieures. † : AUCUNE baisse de performance statistiquement significative ($p \leq 0.05$) par rapport au ré-ordonnancement ColBERTv2 selon le t-test de Student bilatéral.

	model	**Remain**	**Dev set**	**DL 19**	**DL 20**
sans élagage	BM25	-	18.7	50.6	47.5
	SPLADEv2	-	35.8	70.6	68.7
	ColBERTv2 original	-	39.7	74.5	-
	ColBERTv2 ré-ordonnancement	-	40.0	74.4	75.6
avec élagage	First p	50%	37.7	72.3	-
	IDF	50%	32.6	70.2	-
	Stopwords	67%	-	70.9	67.8
	Att. Score	50%	36.0	72.0	-
	AligneR$_{base}$	40%	38.1	-	-
Élagage LP, seuil 0.7					
Colbert$_P$ $\alpha\mathcal{L}_r^{(L1)}$	$\alpha = 0.1$	25%	37.1	72.8†	70.6
	$\alpha = 0.05$	46%	37.9	72.6†	71.4
	$\alpha = 0.01$	83%	39.2	73.6†	72.1
Colbert$_P$ $\alpha\mathcal{L}_r^{(sim)}$	$\alpha = 2$	24%	38.4	72.5†	72.6
	$\alpha = 1.25$	29%	39.3	73.7†	72.8
	$\alpha = 0.8$	**32%**	**39.7**†	**73.8**†	**73.3**
	$\alpha = 0.5$	37%	39.5	73.9†	73.7
	$\alpha = 0.1$	**52%**	**39.9**†	**74.2**†	**73.8**
Colbert$_P$ $\alpha\mathcal{L}_r^{(nuc)}$	$\alpha = 0.3$	38%	38.5	71.9†	73.9†
	$\alpha = 0.2$	40%	39.2	73.9†	73.1
	$\alpha = 0.1$	56%	39.4	73.9†	73.2
Élagage avec norme L1, seuil 0.5					
Colbert$_P$ $\alpha\mathcal{L}_r^{(L1)}$	$\alpha = 0.1$	9%	37.5	73.0†	70.6
	$\alpha = 0.05$	14%	37.9	72.6†	71.4
	$\alpha = 0.01$	**31%**	**39.2**	**73.6**†	**72.1**
Colbert$_P$ $\alpha\mathcal{L}_r^{(sim)}$	$\alpha = 2$	25%	38.7	72.8†	72.6
	$\alpha = 1.25$	29%	39.4	73.1†	72.9
	$\alpha = 0.8$	33%	39.7†	73.8†	73.2
	$\alpha = 0.5$	37%	39.5	73.9†	73.7
	$\alpha = 0.1$	51%	39.9†	74.2†	73.8
Colbert$_P$ $\alpha\mathcal{L}_r^{(nuc)}$	$\alpha = 0.3$	18%	38.5	71.9†	73.9†
	$\alpha = 0.2$	25%	39.2	73.9†	73.1
	$\alpha = 0.1$	34%	39.4	73.9†	73.2

TABLE 2 – Les résultats BEIR en nDCG@10 et le ratio restant de nos méthodes pour chaque sous-dataset. † : AUCUNE baisse de performance statistiquement significative ($p \leq 0.05$) par rapport aux résultats de la version ré-ordonnancement de ColBERTv2 selon le t-test de Student bilatéral.

model		**Remain**	**CF**	**DB**	**FQ**	**NF**	**NQ**	**QU**	**SD**	**SF**	**TC**	**TO**	**Avg.**
w/o. Pruning	BM25	-	21.3	31.3	23.6	32.5	32.9	78.9	15.8	66.5	65.6	36.7	40.5
	SPLADEv2	-	21.3	42.0	31.6	32.8	50.8	81.1	14.0	65.9	65.5	25.5	43.0
	ColBERTv2 end-to-end	-	17.6	44.6	35.6	33.8	56.2	85.2	15.4	69.3	73.8	26.3	45.8
	ColBERTv2 reranking	-	20.3	45.9	35.7	34.3	56.8	85.7	14.4	68.5	75.5	33.7	47.0
w/. Pruning	First p	50%	15.5	44.0	32.4	32.4	53.2	78.7	15.2	61.6	72.1	26.1	43.1
	IDF	50%	17.0	45.1	34.0	32.6	55.0	85.3	15.5	64.8	71.4	26.2	44.7
	Att. Score	50%	16.7	44.4	33.6	32.6	54.7	84.4	15.5	64.0	70.0	26.2	44.2
Colbert$_P$ $\alpha\mathcal{L}_r^{(L1)}$	$\alpha = 0.01$	-	18.6	44.3	34.3	33.8†	54.0	84.5	14.2†	67.0†	71.6	31.8	45.4
	Remain %		34%	39%	30%	29%	35%	76%	32%	30%	32%	28%	39%
Colbert$_P$ $\alpha\mathcal{L}_r^{(sim)}$	$\alpha = 0.8$	-	19.6	42.9	35.0†	33.8†	54.7	82.6	13.9	66.9†	74.3†	33.1†	45.7
	Remain %		36%	34%	36%	38%	36%	40%	36%	38%	37%	40%	37%
	$\alpha = 0.1$	-	20.2†	44.7	35.0†	34.4†	55.5	84.4	14.0	67.4†	71.1	32.8†	45.9
	Remain %		60%	58%	57%	64%	57%	58%	58%	69%	65%	65%	61%

TABLE 3 – Les résultats de recherche LoTTE en Success@5 et le ratio restant de nos méthodes pour chaque sous-dataset. † : AUCUNE baisse de performance statistiquement significative ($p \leq 0.05$) par rapport aux résultats de la version ré-ordonnancement de ColBERTv2 selon le t-test de Student bilatéral.

model		**Wrt.**	**Rcr.**	**Sci.**	**Tch.**	**LS.**	**Avg.**
w/o. Pruning	BM25	60.3	56.5	32.7	41.8	63.8	51.0
	SPLADEv2	72.1	67.8	53.2	60.1	80.0	66.6
	ColBERTv2 end-to-end	80.1	72.3	56.7	66.1	84.7	72.0
	ColBERTv2 reranking	80.0	73.0	56.9	66.4	84.3	72.1
Colbert$_P$ $\alpha\mathcal{L}_r^{(L1)}$	$\alpha = 0.01$	77.5	72.5†	57.2†	65.4†	84.1†	71.4
	Remain %	29%	29%	26%	31%	26%	29%
Colbert$_P$ $\alpha\mathcal{L}_r^{(sim)}$	$\alpha = 0.8$	77.4	72.1†	56.9†	65.9†	84.0†	71.3
	Remain %	34%	35%	35%	37%	34%	35%
	$\alpha = 0.1$	78.9†	72.0†	57.2†	65.4†	84.1†	71.3
	Remain %	57%	58%	70%	59%	59%	61%

6 Conclusion, limites et travaux futurs

Dans cet article, nous avons introduit un cadre pour l'élagage de vecteurs du document pour les modèles d'interaction tardive, basé sur le concept de dominance (vectorielle), qui définit quels tokens peuvent être retirés des vecteurs de documents sans affecter la fonction de score. Nous avons adapté le modèle ColBERT et proposé différentes stratégies d'élagage ainsi que les régularisations associées.

TABLE 4 – Une analyse qualitative de nos deux stratégies d'élagage proposées, basée sur le meilleur modèle entraîné avec nos régularisations. Les tokens restants sont en **gras**.

Example passage	Norm pruning : $\theta = 0.5$	LP pruning : $\tau = 0.7$
Doc-Sim Regularization	What is the **Population** of **New Delhi**. According to **estimated** figures from **Census** of India, **Population** of New Delhi in **2016** is 18.6 million. Delhi's is witnessing a huge **growth** in its population every **year**. **Population** of Delhi **city** is estimated to cross 25 million in **2020**.	What is the **Population** of **New Delhi**. According to **estimated figures** from **Census** of India, **Population** of New Delhi in **2016** is 18.6 million. Delhi's is witnessing a huge **growth** in its population every **year**. Population of Delhi city is estimated to cross 25 million in **2020**.
norme L1 Regularization	What is the **Population** of **New Delhi**. According to **estimated figures** from **Census** of India, **Population** of New Delhi in **2016** is 18.6 million. Delhi's is witnessing a huge **growth** in its population every **year**. **Population** of Delhi **city** is estimated to cross 25 **million** in **2020**.	What is the **Population** of **New Delhi**. According **to estimated figures** from **Census of India**, **Population of New Delhi in 2016** is **18.6 million**. Delhi's is witnessing a **huge growth in** its **population** every **year**. **Population** of **Delhi city** is **estimated** to **cross** 25 **million** in **2020**.

Nos expériences montrent que nos méthodes peuvent supprimer environ 70% des tokens tout en maintenant de bonnes performances à la fois en domaine et hors domaine. Cependant, il y a certaines limitations :

1. **Pipeline de ré-ordonnancement** : Notre méthode repose actuellement sur le ré-ordonnancement SPLADEv2, ce qui nécessite plus de stockage d'index que l'approche de bout en bout. L'adaptation de techniques comme (Nardini *et al.*, 2024) pour une meilleure efficacité pourrait améliorer cela.
2. **Coût Computationnel** : Bien que nous ayons réduit le coût de la procédure LP, il reste trop élevé. Exploiter les symétries dans le problème LP pourrait conduire à un élagage plus efficace, avec des approximations contrôlées pendant l'entraînement.
3. **Stratégies de Régularisation** : Les méthodes de régularisation sont seulement faiblement liées aux ratios de dominance. Concevoir de meilleures régularisations et adapter l'architecture Transformer pour produire plus de vecteurs dominés pourrait améliorer les performances. De plus, utiliser des méthodes de pré-entraînement comme celles de (Gao & Callan, 2021; Ma *et al.*, 2023) pourrait encore renforcer les résultats.

Références

ACQUAVIA A., MACDONALD C. & TONELLOTTO N. (2023). Static pruning for multi-representation dense retrieval. In *Proceedings of the ACM Symposium on Document Engineering*

2023, p. 1–10, Limerick Ireland : ACM. DOI : 10.1145/3573128.3604896.

BAJAJ P., CAMPOS D., CRASWELL N., DENG L., GAO J., LIU X., MAJUMDER R., MCNAMARA A., MITRA B., NGUYEN T., ROSENBERG M., SONG X., STOICA A., TIWARY S. & WANG T. (2018). Ms marco : A human generated machine reading comprehension dataset. (arXiv :1611.09268). arXiv :1611.09268 [cs].

CHEUNG D., CUCKER F. & PENA J. (2008). A condition number for multifold conic systems. *SIAM Journal on Optimization*, **19**, 261–280. DOI : 10.1137/060665427.

CHEUNG D., CUCKER F. & PEÑA J. (2003). Unifying condition numbers for linear programming. *Mathematics of Operations Research*, **28**(4), 609–624.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In J. BURSTEIN, C. DORAN & T. SOLORIO, Éds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

FORMAL T., LASSANCE C., PIWOWARSKI B. & CLINCHANT S. (2021). Splade v2 : Sparse lexical and expansion model for information retrieval. (arXiv :2109.10086). arXiv :2109.10086 [cs].

FREUND R. M. & VERA J. R. (2009). Equivalence of convex problem geometry and computational complexity in the separation oracle model. *Mathematics of Operations Research*, **34**(4), 869–879. DOI : 10.1287/moor.1090.0408.

GAO L. & CALLAN J. (2021). Condenser : a pre-training architecture for dense retrieval. In M.-F. MOENS, X. HUANG, L. SPECIA & S. W.-T. YIH, Éds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 981–993, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.75.

GAO L., DAI Z. & CALLAN J. (2021). COIL : Revisit exact lexical match in information retrieval with contextualized inverted list. In K. TOUTANOVA, A. RUMSHISKY, L. ZETTLEMOYER, D. HAKKANI-TUR, I. BELTAGY, S. BETHARD, R. COTTERELL, T. CHAKRABORTY & Y. ZHOU, Éds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 3030–3042, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.naacl-main.241.

GARG A. & MERMIN N. D. (1984). Farkas's lemma and the nature of reality : Statistical implications of quantum correlations. *Foundations of Physics*, **14**(1), 1–39. DOI : 10.1007/BF00741645.

HINTON G. E., VINYALS O. & DEAN J. (2015). Distilling the knowledge in a neural network. *ArXiv*, **abs/1503.02531**.

HOFSTÄTTER S., KHATTAB O., ALTHAMMER S., SERTKAN M. & HANBURY A. (2022). Introducing neural bag of whole-words with colberter : Contextualized late interactions using enhanced reduction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, p. 737–747, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3511808.3557367.

HOFSTÄTTER S., LIN S.-C., YANG J.-H., LIN J. & HANBURY A. (2021). Efficiently teaching an effective dense retriever with balanced topic aware sampling. (arXiv :2104.06967). arXiv :2104.06967 [cs].

JOHNSON J., DOUZE M. & JÉGOU H. (2021). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, **7**(3), 535–547. DOI : 10.1109/TBDATA.2019.2921572.

KARPUKHIN V., OGUZ B., MIN S., LEWIS P., WU L., EDUNOV S., CHEN D. & YIH W.-T. (2020). Dense passage retrieval for open-domain question answering. In B. WEBBER, T. COHN, Y.

HE & Y. LIU, Éds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 6769–6781, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.550.

KHATTAB O. & ZAHARIA M. (2020). Colbert : Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, p. 39–48, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3397271.3401075.

LASSANCE C., DÉJEAN H., FORMAL T. & CLINCHANT S. (2024). Splade-v3 : New baselines for splade. (arXiv :2403.06789). arXiv :2403.06789 [cs].

LASSANCE C., MAACHOU M., PARK J. & CLINCHANT S. (2022). Learned token pruning in contextualized late interaction over bert (colbert). In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, p. 2232–2236, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3477495.3531835.

LI M., LIN S.-C., OGUZ B., GHOSHAL A., LIN J., MEHDAD Y., YIH W.-T. & CHEN X. (2023). CITADEL : Conditional token interaction via dynamic lexical routing for efficient and effective multi-vector retrieval. In A. ROGERS, J. BOYD-GRABER & N. OKAZAKI, Éds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 11891–11907, Toronto, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/2023.acl-long.663.

LIN S.-C., YANG J.-H. & LIN J. (2021). In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In A. ROGERS, I. CALIXTO, I. VULIĆ, N. SAPHRA, N. KASSNER, O.-M. CAMBURU, T. BANSAL & V. SHWARTZ, Éds., *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, p. 163–173, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.repl4nlp-1.17.

LIU Q., GUO G., MAO J., DOU Z., WEN J.-R., JIANG H., ZHANG X. & CAO Z. (2024). An analysis on matching mechanisms and token pruning for late-interaction models. *ACM Transactions on Information Systems*, **42**(5), 1–28. DOI : 10.1145/3639818.

MA X., FUN H., YIN X., MALLIA A. & LIN J. (2023). Enhancing sparse retrieval via unsupervised learning. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP '23, p. 150–157, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3624918.3625334.

NARDINI F. M., RULLI C. & VENTURINI R. (2024). Efficient multi-vector dense retrieval with bit vectors. In N. GOHARIAN, N. TONELLOTTO, Y. HE, A. LIPANI, G. MCDONALD, C. MACDONALD & I. OUNIS, Éds., *Advances in Information Retrieval*, p. 3–17, Cham : Springer Nature Switzerland.

NOGUEIRA R. & CHO K. (2020). Passage re-ranking with bert. (arXiv :1901.04085). arXiv :1901.04085 [cs].

QIAN Y., LEE J., DUDDU S. M. K., DAI Z., BRAHMA S., NAIM I., LEI T. & ZHAO V. Y. (2022). Multi-vector retrieval as sparse alignment. (arXiv :2211.01267). arXiv :2211.01267 [cs].

REN R., QU Y., LIU J., ZHAO W. X., SHE Q., WU H., WANG H. & WEN J.-R. (2021). RocketQAv2 : A joint training method for dense passage retrieval and passage re-ranking. In M.-F. MOENS, X. HUANG, L. SPECIA & S. W.-T. YIH, Éds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 2825–2835, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.224.

ROBERTSON S., WALKER S., JONES S., HANCOCK-BEAULIEU M. & GATFORD M. (1994). Okapi at trec-3. p. 0–.

SANTHANAM K., KHATTAB O., POTTS C. & ZAHARIA M. (2022a). Plaid : An efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, p. 1747–1756, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3511808.3557325.

SANTHANAM K., KHATTAB O., SAAD-FALCON J., POTTS C. & ZAHARIA M. (2022b). ColBERTv2 : Effective and efficient retrieval via lightweight late interaction. In M. CARPUAT, M.-C. DE MARNEFFE & I. V. MEZA RUIZ, Éds., *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 3715–3734, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.naacl-main.272.

THAKUR N., REIMERS N., RÜCKLÉ A., SRIVASTAVA A. & GUREVYCH I. (2021). Beir : A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In J. VANSCHOREN & S. YEUNG, Éds., *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

ZONG Y. & PIWOWARSKI B. (2023). Xpmir : A modular library for learning to rank and neural ir experiments. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, p. 3185–3189, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3539618.3591818.