MarkedDPR : Un bi-encodeur exploitant la correspondance lexicale

Smail Oussaidene[1,2,3,*], Lynda Said Lhadj[1,3] and Mohand Boughanem[2,4]

[1]*Ecole Superieure d'Informatique d'Alger (ESI)*
[2]*Institut de Recherche en Informatique de Toulouse (IRIT)*
[3]*Laboratoire de la Communication dans les Systèmes Informatiques (LCSI)*
[4]*Université Toulouse III - Paul Sabatier (UPS)*

Abstract

Les modèles récents de RI neuronale tels que le modèle dense DPR estiment la pertinence à travers l'appariement sémantique entre la requête et le document en ignorant l'appariement exact. Nous présentons dans cet article MarkedDPR, une extension du modèle DPR, conçue pour prendre en compte explicitement les signaux de correspondances exactes entre la requête et le document. MarkedDPR est principalement basé sur une méthode de marquage qui met en évidence les correspondances exactes pour chaque paire requête-document afin de guider le modèle lors de l'apprentissage. Les évaluations empiriques que nous avons réalisées ont montré l'intérêt du marquage aussi bien sur les données issues du même domaine (*in-domain*) que sur celles qui sont hors domaine (*out-domain*) et ont affiché des améliorations significatives vis-à-vis des modèles de référence (*baselines*) que nous avons considérés.

Keywords

Recherche d'information dense, Correspandance exacte, Transférabilité

1. Introduction

Le problème de l'inadéquation du vocabulaire dans la recherche d'information (RI), communément nommé en anglais *vocabulary mismatch*, se produit lorsque les termes de la requête ne correspondent pas au vocabulaire utilisé par les auteurs des documents. Il en résulte que des documents pertinents qui ne comportent aucun terme de la requête ne soient jamais sélectionnés. Les modèles récents de la RI dense tels que DPR [1], ANCE [2], ColBERT [3] ont apporté une réponse efficace à ce problème comparé aux modèles traditionnels tels que BM25 [4]. En effet, ce succès est principalement dû à l'utilisation de modèles de langage pré-entraînés, tels que BERT (Bidirectional Encoder Representations from Transformers) [5], qui, grâce à leur pré-entraînement intensif sur de vastes corpus textuels, permettent la construction de représentations sémantiques des mots et ainsi produisent des représentations de requête et de document plus riches en information que les modèles de la RI classiques [6].

Bien que ces modèles excellent dans la capture d'informations contextuelles et l'amélioration de la précision de la recherche dans leur domaine d'apprentissage, ils peuvent avoir du mal à se

. *CORIA 2024 : Conférence en Recherche d'Information et Applications, La Rochelle, France*
*. Corresponding author.
. is_oussaidene@esi.dz (S. Oussaidene); l_said_lhadj@esi.dz (L. S. Lhadj); bougha@irit.fr (M. Boughanem)
. 0000-0001-7747-0658 (S. Oussaidene); 0000-0001-7004-0807 (M. Boughanem)

généraliser efficacement à des données nouvelles et inconnues. Ce problème dit de transférabilité constitue un nouveau défi en recherche d'information dense. De nombreux travaux tels que [7, 8, 9, 10, 11, 12, 13] se sont penchés sur cette question selon plusieurs approches.

Nous présentons dans cet article une nouvelle approche de recherche dense qui met l'accent sur l'importance des signaux de correspondance exacte. À cette fin, nous introduisons des marqueurs spéciaux, comme dans [14], dans la représentation des requêtes et des documents qui favorisent l'appariement exact entre la requête et le document. Nous partons de l'hypothèse que les signaux de correspondance exacte restent essentiels pour une correspondance de pertinence précise. Par conséquent, notre approche intègre l'intuition traditionnelle de la correspondance exacte dans des modèles de langage pré-entraînés (MLP) de pointe, dans le but de combiner entre la pertinence sémantique et la pertinence lexicale.

Nos expérimentations ont démontré l'efficacité de notre approche sur la collection MS MARCO [15] et le benchmark BEIR [16], dans un contexte de reclassement par rapport au modèle dense purement sémantique DPR.

2. Travaux connexes

Plusieurs modèles denses ont été proposés depuis la sortie de BERT en 2018. L'un des premiers modèles ayant montré des performances significatives est DPR (Dense Passage Retriever) [1] qui exploite les représentations vectorielles denses des passages et des requêtes pour réaliser un appariement sémantique, particulièrement dans des tâches telles que la réponse à des questions et la recherche de passages. Techniquement, le modèle DPR utilise une architecture bi-encodeur, où les requêtes et les passages sont encodés séparément, ce qui permet une recherche efficace grâce au calcul de la similarité des vecteurs denses.

ANCE (Approximate Nearest Neighbor Contrastive Estimation) [2] sélectionne efficacement des échantillons négatifs de haute qualité pour l'apprentissage contrastif dans la recherche dense. En s'appuyant sur une stratégie d'échantillonnage négatif adaptative basée sur les plus proches voisins approximatifs, ANCE atteint des performances supérieures en termes de précision de recherche par rapport aux méthodes traditionnelles.

ColBERT (Contextualized Late Interaction over BERT) [3] s'inscrit également dans la recherche dense en incorporant des mécanismes d'interaction tardive avec des embeddings contextualisés. En affinant les encodeurs basés sur BERT sur des collections de données à grande échelle, ColBERT atteint des performances de pointe dans les tâches de recherche dense, surpassant les modèles précédents en termes de précision et d'efficacité.

CLEAR (Complementary Retrieval Model) [12] est un modèle de recherche hybride dense-éparse qui intègre à la fois les signaux lexicaux (BM25) et sémantiques (denses) pour améliorer la précision de la recherche. En utilisant une architecture bi-encodeur, CLEAR capture les nuances sémantiques absentes des modèles lexicaux traditionnels. Pendant l'entraînement, il ajuste dynamiquement le paramètre de marge de la perte de marge maximale par paire sur la base des scores BM25, optimisant ainsi l'équilibre entre les signaux lexicaux et sémantiques.

MarkedBERT [14] est un cross-encodeur basé sur BERT qui est spécifiquement conçu pour la recherche sémantique lexicale en apprenant à BERT à prendre en considération la correspondance exacte lors du calcul de la pertinence. Pour ce faire, la séquence d'entrée du modèle est

enrichie de termes spéciaux qui marquent les termes correspondant exactement à la requête et au document. Ces termes marqués servent d'indicateurs ou d'indices permettant au modèle de prêter attention à certains aspects du texte pendant l'entraînement et l'inférence.

Dans cet article, nous adapterons cette stratégie de marquage à l'architecture bi-encodeur dans le but de tenter d'apprendre aux modèles denses l'appariement lexical et par la suite améliorer la capacité de transférabilité du modèle.

3. MarkedDPR

3.1. Modèle de base

Dans le modèle DPR, les requêtes et les documents sont encodés séparément. Étant donné une requête $Q = \{q_1, q_2, \ldots, q_{|Q|}\}$ et un document $D = \{d_1, d_2, \ldots, d_{|D|}\}$, les séquences d'entrée des deux encodeurs E_Q et E_D sont formulées comme suit :

$$E_Q = [E_{[CLS]}, E_{[Q]}, E_{q_1}, \ldots, E_{q_{|Q|}}, E_{[SEP]}] \tag{1}$$

$$E_D = [E_{[CLS]}, E_{[D]}, E_{d_1}, \ldots, E_{d_{|D|}}, E_{[SEP]}] \tag{2}$$

où E_x représente l'embedding du terme x. $E_{[Q]}$ et $E_{[D]}$ sont ajoutés pour distinguer entre les requêtes et les documents.

Les représentations de sortie des termes $[CLS]$ standard générées par les deux encodeurs sont ensuite utilisées dans l'estimation de la pertinence du document candidat par rapport à la requête se basant sur une fonction de similarité vectorielle, à savoir le produit scalaire. Les sorties des autres termes peuvent également être utilisées, comme étudié dans [1, 3].

Le modèle est entraîné pour maximiser la similarité entre les requêtes et leurs documents pertinents et minimiser la similarité entre les requêtes et les documents non pertinents en utilisant la fonction de perte d'apprentissage contrastive pour distinguer les paires requête-passage pertinentes et non pertinentes comme suit :

$$\mathcal{L}(q_i, d_i^+, D_i^-) = -log \frac{e^{\phi(\eta_q(q_i), \eta_d(d_i^+))}}{e^{\phi(\eta_q(q_i), \eta_d(d_i^+))} + \sum_{d_i^- \in D_i^-} e^{\phi(\eta_q(q_i), \eta_d(d_i^-))}} \tag{3}$$

Où q_i est une requête, d_i^+ est le document pertinent qui lui est associé et D_i^- est un ensemble de documents non pertinents. η_q et η_d sont respectivement les fonctions d'encodage des requêtes et des documents. ϕ est la fonction de comparaison (le produit scalaire).

3.2. Le modèle proposé

MarkedDPR, le modèle que nous proposons, consiste à étendre le modèle DPR en y intégrant le même principe de marquage que dans [14]. Cette approche propose l'exploitation de la notion de marquage pour construire une nouvelle représentation des documents et des requêtes. Les entrées de MarkedDPR sont ainsi augmentées avant d'être encodées séparément par les encodeurs avec un terme spécial pour marquer les termes qui correspondent exactement dans

le passage et la requête, introduisant le terme # autour de chaque terme apparaissant à la fois dans le passage et dans la requête.

Considérons une requête $Q = \{q_1, q_2, \ldots, q_{|Q|}\}$ et un passage $P = \{p_1, p_2, \ldots, p_{|P|}\}$, si le terme q_i est identique à la fois à p_m et à p_n, et si le terme q_j est identique à p_l avec $i < j$ et $n < m < l$, alors les entrées sont augmentées de la manière suivante :

$$Q = \{[CLS], q_1, \ldots, \#, q_i, \#, \ldots, \#, q_j, \#, \ldots, q_{|Q|}, [SEP]\} \tag{4}$$

$$P = \{[CLS], p_1, \ldots, \#, p_n, \#, \ldots, \#, p_m, \#, \ldots, \#, p_l, \#, \ldots, p_{|P|}, [SEP]\} \tag{5}$$

Il est important à préciser que ce marquage s'applique sur les données textuelles avant d'être transformé en token par le tokenizer (comme WordPiece [17]). Ceci implique que si un terme est découpé en plusieurs token, la séquence des tokens est entourée de # et non chaque token. De plus, le fait d'utiliser le caractère # ne rentre pas en conflit avec les délimitations de fin de mots du tokenizer de BERT, vu que terme marqueur et le terme marqué sont encodés séparément.

Ainsi, le processus de marquage dans le contexte de MarkedDPR, comme exemplifié dans le tableau 1, consiste à identifier et à marquer les termes qui apparaissent simultanément dans une requête et dans un passage. Cela met l'accent sur les correspondances exactes, en orientant l'attention du modèle, tel que BERT, vers ces termes spécifiques. L'objectif est d'aider le modèle à mieux comprendre et à utiliser ces indices de correspondance exacte.

Le processus de ce modèle est présenté sur la figure 1.

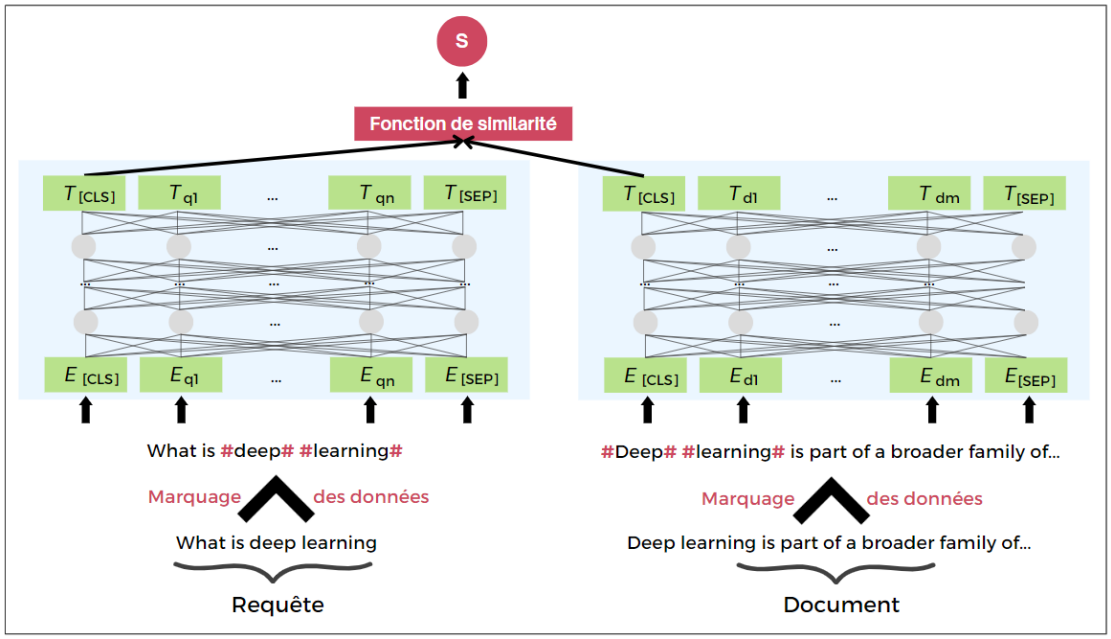

Figure 1: L'architecture du modèle MarkedDPR

Table 1
Exemple de la stratégie de marquage proposée appliquée à une paire de requête-document

	Requête	Document
Sans marquage	what is deep learning	Deep learning is part of a broader family of machine learning methods...
Avec marquage	what is #deep# #learning#	#Deep# #learning# is part of a broader family of machine #learning# methods...

4. Cadre expérimental

Dans cette section, nous décrivons les questions de recherche qui ont guidé nos expérimentations, les données utilisées pour les expérimentations, ainsi que la configuration de base.

4.1. Objectifs

Nos expérimentations menées visent à répondre à trois questions principales :

1. **Q1** : Le marquage des correspondances exactes est-il bénéfique pour DPR dans un environnement in-domain ?
2. **Q2** : Le marquage des correspondances exactes est-il bénéfique pour DPR dans un environnement out-domain ?
3. **Q3** : Dans quelle mesure les scores des correspondances exactes issus d'une recherche initiale (avec BM25, par exemple) contribuent-ils à l'efficacité globale ?

4.2. Collections de tests

MarkedDPR est d'abord évalué dans un contexte "*in-domain*" (c'est-à-dire que les données de tests sont issues du même domaine que celles utilisées lors de l'optimisation (*fine-tuning*) sur MS MARCO (Microsoft MAchine Reading COmprehension dataset) [15], qui est une collection de données tirée d'un échantillon représentatif de plus d'un million de requêtes à partir du moteur de recherche Bing. Des éditeurs humains ont attribué divers degrés de pertinence à chaque requête. Nous nous intéressons principalement à la collection de données sur le classement des passages, qui compte 8,8 millions de passages. Le tableau 2 reprend les statistiques de ces collections de tests.

Ensuite, nous évaluons MarkedDPR dans un contexte "*out-domain*" (c'est-à-dire que les données de tests sont issues d'un domaine différent que celles utilisées lors de l'optimisation sur le benchmark BEIR [16], qui est conçu pour évaluer les capacités de généralisation des modèles de recherche d'information. Ce benchmark comprend 18 collections de données de recherche d'information, couvrant un large éventail de tâches et de domaines, ce qui permet une comparaison et une évaluation complètes des performances des modèles [18]. Le tableau 3 reprend les statistiques de ces collections de tests.

Table 2
Résumé des statistiques de la collection de données MS MARCO

Collection de données	Statistiques
Collection d'entraînement	
- Nombre de requêtes	502 939
- Nombre de passages	8 841 823
- Nombre de paire requête-passage	532 761
- Nombre de passages pertinents par requête	1,04
Collection de développement	
- Nombre de requêtes	101 093
- Nombre de passages pertinents par requête	1,04

Table 3
Résumé des statistiques des collections de BEIR

Statistique	trec-covid	fiqa	scifact	scidocs	nfCorpus	arguana
Requêtes	30	116	65	100	1500	300
Passages	202 337	28 498	27 340	9 220	99 590	6 712
assages pertinents par requêtes	6 744,57	245,48	420,62	92,20	66,39	22,37
Termes par Requête	27,27	31,48	44,14	28,09	21,97	24,52
Termes par Passage	196,21	218,14	251,27	181,69	156,69	215,75
Passages Perts par Rq	1,03	0,87	0,88	1,00	0,78	0,93

4.3. Métrique d'évaluation

Nous avons utilisé les mesures d'évaluation officielles désignées pour chaque collection de données, comme montré dans le tableau 4.

Table 4
Mesures d'évaluation officielles pour chaque collection de données et leur rôle respectif.

collection de données	Mesure d'évaluation	Rôle de la mesure
MS MARCO	Mean Reciprocal Rank (**MRR**)	Évalue l'efficacité du classement des résultats récupérés.
BEIR	Normalized Discounted Cumulative Gain (**nDCG**)	Évalue la qualité du classement en tenant compte de la pertinence et de la position.

4.4. Modèles de références

Nous comparons MarkedDPR avec deux modèles de référence :
Modèle de recherche lexicale

— BM25 [4], nous utilisons l'implémentation de pyterrier [1] avec les paramètres par défaut. Ce modèle non supervisé sert à la fois de modèle de référence et de première étape de recherche dans toutes nos expérimentations.

Modèle de recherche dense

— DPR [1], nous utilisons notre propre implémentation du modèle qui est optimisée avec MS MARCO.

4.5. Optimisation du bi-encodeur

Nous utilisons la version distillée de BERT (6 couches, 768 tailles cachées, 12 têtes, 66 millions de paramètres) en raison des limitations matérielles. Le même encodeur est utilisé pour encoder les requêtes et les documents. Nous effectuons une optimisation sur la collection MS MARCO avec une taille de batch de 10 et une longueur de séquence maximale de 512 termes ($10\ sequences * 512\ termes = 5120\ termes/batch$). Nous utilisons l'optimiseur Adam avec un taux d'apprentissage initial fixé à $3 * 10^{-6}$. Le taux d'abandon est fixé à 0,1 pour toutes nos expérimentations. Nous rappelons que l'optimiseur utilise la fonction dans l'équation 3. Nous utilisons une implémentation open source de BERT par Hugging Face [2].

Pour cette phase d'optimisation, nous avons marqué au préalable toutes les requêtes avec leurs documents pertinents et non pertinents. MarkedDPR utilise la méthode in-batch negative au cours de son entraînement. Cela implique que pour un batch de n paires de requête-document $\{(Q_i, D_i)\}_{i=1}^{i=n}$, chaque paire de requête et document associé (Q_i, D_i) est marquée et encodée par le même encodeur. De plus, chaque paire de requête et document associé à une autre requête (Q_i, D_j) $(i \neq j)$ est marquée et encodée. Cela augmente la taille du batch de $2n$ vers $2n^2$ et par conséquent alourdit le processus de l'optimisation.

Lors du processus de marquage, nous veillons à ce que seuls les mots significatifs (hors mots vides) soient marqués. Nous avons utilisé la liste UDel Stop Words, conçue par l'Université du Delaware (UDel) [3] De plus, dans notre processus de marquage, nous considérons deux mots ayant le même radical comme des correspondances exactes. Pour ce faire, nous avons utilisé le stemmer de Porter [4].

4.6. Inférence

Nous avons utilisé deux scénarios principaux pour évaluer les performances de MarkedDPR :

— Recherche initiale : MarkedDPR est évalué en tant que modèle de recherche initial. L'intégralité du corpus de tests est encodé par l'encodeur en *offline* sans avoir marqué les documents. Ensuite, au moment de l'inférence, la requête non-marquée est encodée par l'encodeur et nous récupérons une liste ordonnée de documents pertinents en utilisant la bibliotheque FAISS[19]. L'encodeur est entraîné sur des données marquées, ce qui nous différencie du modèle DPR.

1. https ://github.com/terrier-org/pyterrier
2. https ://huggingface.co
3. https ://github.com/igorbrigadir/stopwords/edit/master/en/galago_inquery.txt
4. http ://terrier.org/docs/v4.0/javadoc/org/terrier/terms/PorterStemmer.html

— *Reranker* : MarkedDPR réordonne une liste de 1 000 documents préalablement retournée par une recherche initiale (ici, par BM25). Dans ce scénario, la requête ainsi que les documents sélectionnés sont marqués, la requête d'un côté et le document de l'autre, chacun d'entre eux est encodé par le même encodeur, puis la similarité entre les [CLS] produits par l'encodeur est calculée. Dans ces expérimentations, BM25»MarkedDPR désigne une recherche initiale par BM25 suivie par un *reranking* avec MarkedDPR.

Il est essentiel de noter que MarkedDPR a été entraîné pour évaluer la pertinence en utilisant des données marquées. Cependant, lorsqu'il est évalué en tant que premier modèle de recherche, MarkedDPR doit identifier les passages pertinents sans avoir préalablement marqué les passages ni les requêtes. Cette nécessité découle du fait qu'il serait peu pratique de marquer l'intégralité du corpus pour chaque requête individuelle. Par conséquent, une disparité significative se présente entre les données utilisées pour l'entraînement et celles utilisées pour l'évaluation. D'où l'intérêt d'évaluer MarkedDPR en tant que *reranker*, vu que comme le modèle ne considère qu'un nombre réduit de documents, il est possible de les marquer suivant la requête en question.

5. Résultats et discussion

5.1. Le marquage des correspondances exactes est-il bénéfique pour DPR dans un environnement *in-domain* ?

Comme MarkedDPR a été optimisé avec la collection d'entraînement de MS MARCO il sera évalué sur la sous-collection de développement de MS MARCO.

Dans le premier scénario, MarkedDPR récupère les k passages les plus pertinents pour une requête donnée à partir du corpus intégral. k est un hyper-paramètre statique égal à 1000.

Table 5
L'efficacité de MarkedDPR sur la collection du développement de classement de passages MS MARCO. Les meilleures performances des modèles évalués sont indiquées en gras. Pour chaque mesure, le taux d'amélioration de MarkedDPR par rapport à la baseline DPR est indiqué (%). ▽ indique une amélioration négative.

		MS MARCO Passage (Dev)			
Méthode		MRR@10		Recall@1k	
(1)	BM25[4]	0.186		0.878	
(2)	DPR[1]	**0.247**		**0.900**	
(3)	MarkedDPR	0.145	▽ 41.29%	0.793	▽11.99%

Nous présentons les performances de MarkedDPR en recherche initiale sur la collection de développement MS MARCO dans le tableau 5. Sur cette première expérimentation, MarkedDPR est visiblement moins performant que son prédécesseur DPR, avec une différence de 10 points et une différence de 4 points avec BM25. Ce qui indique que l'absence d'indicateurs de pertinence (marquage) dans les données d'évaluation handicape fortement MarkedDPR entraînant une baisse de 41.29% comparé à DPR.

Dans le second scénario, MarkedDPR reclasse les documents déjà retournés par BM25.

Nous présentons les performances sur la collection de développement MS MARCO dans le

Table 6
L'efficacité de MarkedDPR en reranking sur la collection du développement de classement de passages MS MARCO. Les meilleures performances des modèles évalués sont indiquées en gras. Le taux d'amélioration de BM25 » MarkedDPR par rapport à la baseline BM25 » DPR est indiqué (%). ▲ indique une amélioration positive.

		MS MARCO Passage (Dev)		
Méthode		MRR@10		Recall@1k
(1)	BM25	0.186		0.878
(2)	BM25 » DPR	0.230		0.878
(3)	BM25 » MarkedDPR	**0.325**	▲41.30%	0.878

tableau 6. L'ajout du marquage des correspondances exactes permet d'obtenir de meilleures performances que le modèle de référence DPR (dans les deux cas de figure, DPR comme recherche initiale (ligne (2) du tableau 5) et comme *reranker* (la ligne (2) du tableau 6)). En effet, le score MMR@10 s'est amélioré avec un taux de 41.30%, ce qui indique que MarkedDPR a efficacement promu les documents les plus pertinents aux premières places. Le score Recall@1k reste constant dans les trois modèles, car le reranking n'induit aucun changement dans le rappel à ce stade.

5.2. Le marquage des correspondances exactes est-il bénéfique pour DPR dans un environnement *out-domain* ?

Dans cette évaluation, nous étudions la transférabilité de notre approche à des collections out-domaine. Nous utilisons les modèles optimisés sur les passages MS MARCO pour l'évaluation sur les collections de test du benchmark BEIR. Nous n'entraînons pas les modèles sur ces collections de test, nous utilisons toutes leurs requêtes et tous leurs jugements de pertinence comme une collection de test retenu. Cette évaluation est donc un exemple de transfert à zéro.

Table 7
L'efficacité du reranking dans le cadre d'un transfert en *zero-shot* des différents modèles sur les collections du benchmark BEIR en terme de nDCG@10. Les meilleures performances des modèles en reranking évalués sont indiquées en gras. Les meilleures performances globales sont soulignées. Le taux d'amélioration de BM25 » MarkedDPR par rapport à la baseline BM25 » DPR est indiqué (%). ▲ indique une amélioration positive.

Méthode	**BM25**	**DPR**	**BM25 » DPR**	**BM25 » MarkedDPR**	
(1) Covid	<u>0.625</u>	0.393	0.443	**0.461**	▲4.06%
(2) nfCorpus	<u>0.322</u>	0.193	0.274	**0.295**	▲7.66%
(3) Arguana	<u>0.342</u>	0.260	0.270	**0.280**	▲3.70%
(4) Scifact	<u>0.672</u>	0.376	0.381	**0.546**	▲43.33%
(5) Scidocs	<u>0.147</u>	0.091	0.087	**0.122**	▲40.22%
(6) Fiqa	<u>0.252</u>	0.158	0.225	**0.271**	▲20.44%
(÷) Moyenne					▲19.90%

Le tableau 7 liste les performances de différents modèles et modèles de référence sur les 1000 premiers documents candidats extraits par BM25 de six collections du benchmark BEIR en termes

de NDCG@10. La colonne (6) montre le taux d'amélioration de BM25 » MarkedDPR par rapport à BM25 » DPR. Nous constatons que BM25 est plus performant que DPR et MarkedDPR sur toutes les collections de données (Covid, nfcCorpus, Arguana, Scifact, Scidocs et fiqa). Les résultats montrent que MarkedDPR dépasse DPR sur les collections de données out-domaine (Covid, nfcCorpus, Arguana, Scifact, Scidocs et fiqa) avec un gain moyen de 19.90%. En incorporant des informations lexicales dans le modèle MarkedDPR, comparé à DPR, s'adapte mieux aux variations de style linguistique et aux termes spécifiques au domaine, ce qui permet une recherche plus précise dans diverses collections de données. L'amélioration observée des capacités de transfert correspond à notre hypothèse initiale et valide l'importance de la pertinence lexicale dans la recherche d'information inter-domaines.

5.3. Dans quelle mesure les scores des correspondances exactes issus d'une recherche initiale contribuent-ils à l'efficacité globale ?

Lors de nos expérimentations, les scores générés par le premier modèle de recherche (BM25) ne sont pas pris en compte lors de la détermination du classement final des documents. En s'inspirant de la technique d'interpolation des scores utilisée dans Birch [20], nous examinons une combinaison linéaire directe des scores générés par BM25 et le modèle dense. Plus formellement, le score final s_f d'une paire de requête-document est calculé comme suit :

$$s_f = \alpha * s_{BM25} + (1 - \alpha) * s_d \tag{6}$$

où s_{BM25} et s_d sont les scores retournés par BM25 et le modèle dense respectivement et α est un hyperparamètre qui contrôle la contribution de BM25 dans le score final.

Table 8
Efficacité du reranking dans le cadre d'un transfert en *zero-shot* des différents modèles sur les collections du benchmark BEIR en terme de nDCG@10. Les meilleures performances des modèles évalués sont indiquées en gras. Les taux d'amélioration de BM25⊕MarkedDPR et de BM25⊕DPR sont indiqués (%). ▲ indique une amélioration positive.

Méthode	BM25	BM25»DPR	BM25⊕DPR		BM25»MarkedDPR	BM25⊕MarkedDPR	
(1) Covid	0.625	0.443	0.599	▲ 35.21 %	0.461	**0.639**	▲ 38.61 %
(2) nfCorpus	0.322	0.274	**0.334**	▲ 21.89 %	0.289	0.331	▲ 14.53 %
(3) Scifact	0.672	0.381	0.683	▲ 79.26 %	0.546	**0.689**	▲ 26.19 %
(4) Fiqa	0.252	0.225	0.282	▲ 25.33 %	0.271	**0.293**	▲ 8.11 %
(÷) Moyenne				▲ 40.42 %			▲ 21.86 %

Le tableau 8 compare le score nDCG@10 des différents modèles et modèles de référence, avec et sans la contribution des scores de la première étape, sur les 1000 premiers documents candidats extraits par BM25 de quatre collections du benchmark BEIR (les modèles n'ont pas été évalués sur les collections Arguana et Scidocs par manque de ressources matérielles et de temps). Le symbole ⊕ signifie une combinaison des scores de pertinence. Les scores de BM25⊕DPR et BM25⊕MarkedDPR sont accompagnés des taux d'amélioration par rapport à BM25»DPR et BM25»MarkedDPR respectivement. Il est intéressant de noter que la combinaison des scores a

permis d'améliorer les performances de manière significative, en dépassant BM25 dans toutes les collections de données.

DPR et MarkedDPR combiné avec BM25 se sont améliorés sur toutes les collections évaluées avec un taux moyen de 40.42% et 21.86% respectivement. De plus, MarkedDPR dépasse DPR sur trois des quatre collections de données (Covid, Scifact et Fiqa).

Il est à noter que les valeurs optimales de α avec DPR et MarkedDPR se situent entre 0,5 et 0,6 et entre 0,3 et 0,4 respectivement. Ceci indique que MarkedDPR est moins dépendant de BM25 comparé à DPR et ainsi renforce notre hypothèse selon laquelle MarkedDPR a appris la correspondance lexicale au cours de son processus d'entraînement.

6. Limites du modèle

La première limite du modèle MarkedDPR comparativement aux modèles denses réside dans la nécessité de marquer les documents en fonction de la requête au moment de l'inférence. Cette étape est absente (inutile) dans le modèle DPR. Ceci contraint le modèle à être utilisé que comme *reranker*, ou le marquage des documents pour chaque requête est faisable. Ensuite, sur le plan efficience, de part ce marquage, MarkedDPR est de toute évidence moins performant en termes de temps d'exécution d'une requête comparé à DPR.

7. Conclusion

Dans cet article, nous avons proposé MarkedDPR, une extension du modèle dense DPR, qui utilise des termes marqueurs spéciaux pour signaler les correspondances exactes entre les termes de la requête et ceux d'un document. En mettant en évidence des termes importants pour l'estimation de la pertinence, MarkedDPR capitalise sur les avantages de BERT tout en amplifiant sa capacité à capturer des correspondances exactes, essentielles dans la recherche d'information.

Suite aux expérimentations menées, nous avons montré que la prise en compte du marquage des documents et des requêtes améliore les performances du modèle en tant que *reranker* sur les données *in-domain* et les données *out-domain*.

Cette étude encourage les travaux futurs sur (a) l'adaptation de l'architecture du modèle MarkedDPR comme un premier modèle de recherche, (b) l'étude visant à réduire la complexité et le temps de réponse du modèle et (c) l'exploration des termes marqueurs pour transmettre aux transformers d'autres signaux de RI que les correspondances exactes.

Références

[1] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense Passage Retrieval for Open-Domain Question Answering, in : B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6769–6781. doi :10.18653/v1/2020.emnlp-main.550.

[2] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. Bennett, J. Ahmed, A. Overwijk, Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval, 2020. doi :10.48550/arXiv.2007.00808. arXiv:2007.00808.

[3] O. Khattab, M. Zaharia, 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, Association for Computing Machinery, New York, NY, USA, 2020, pp. 39–48. doi :10.1145/3397271.3401075.

[4] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at TREC-3., 1994.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding, in : J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi :10.18653/v1/N19-1423.

[6] J. Lin, R. Nogueira, A. Yates, Pretrained Transformers for Text Ranking : BERT and Beyond, 2021. doi :10.48550/arXiv.2010.06467. arXiv:2010.06467.

[7] L. Gao, J. Callan, Condenser : A Pre-training Architecture for Dense Retrieval, in : M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 981–993. doi :10.18653/v1/2021.emnlp-main.75.

[8] L. Gao, J. Callan, Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval, in : S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2843–2853. doi :10.18653/v1/2022.acl-long.203.

[9] S. Xiao, Z. Liu, Y. Shao, Z. Cao, RetroMAE : Pre-Training Retrieval-oriented Language Models Via Masked Auto-Encoder, in : Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 538–548. doi :10.18653/v1/2022.emnlp-main.35.

[10] C. Xu, D. Guo, N. Duan, J. McAuley, LaPraDoR : Unsupervised Pretrained Dense Retriever for Zero-Shot Text Retrieval, in : S. Muresan, P. Nakov, A. Villavicencio (Eds.), Findings of the Association for Computational Linguistics : ACL 2022, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3557–3569. doi :10.18653/v1/2022.findings-acl.281.

[11] Z. Dai, V. Y. Zhao, J. Ma, Y. Luan, J. Ni, J. Lu, A. Bakalov, K. Guu, K. B. Hall, M.-W. Chang, Promptagator : Few-shot Dense Retrieval From 8 Examples, 2022. arXiv:2209.11755.

[12] L. Gao, Z. Dai, T. Chen, Z. Fan, B. Van Durme, J. Callan, Complement Lexical Retrieval Model with Semantic Residual Embeddings, in : D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, F. Sebastiani (Eds.), Advances in Information Retrieval, Lecture

Notes in Computer Science, Springer International Publishing, Cham, 2021, pp. 146–160. doi :10.1007/978-3-030-72113-8_10.

[13] J. Ni, C. Qu, J. Lu, Z. Dai, G. Hernandez Abrego, J. Ma, V. Zhao, Y. Luan, K. Hall, M.-W. Chang, Y. Yang, Large Dual Encoders Are Generalizable Retrievers, in : Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 9844–9855. doi :10.18653/v1/2022.emnlp-main.669.

[14] L. Boualili, J. G. Moreno, M. Boughanem, MarkedBERT : Integrating Traditional IR Cues in Pre-trained Language Models for Passage Retrieval, in : Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1977–1980. doi :10.1145/3397271.3401194.

[15] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, T. Wang, MS MARCO : A Human Generated MAchine Reading COmprehension Dataset, 2018. doi :10.48550/arXiv.1611.09268. arXiv:1611.09268.

[16] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, I. Gurevych, BEIR : A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models, 2021. doi :10.48550/arXiv.2104.08663. arXiv:2104.08663.

[17] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google's Neural Machine Translation System : Bridging the Gap between Human and Machine Translation, 2016. doi :10.48550/arXiv.1609.08144. arXiv:1609.08144.

[18] W. X. Zhao, J. Liu, R. Ren, J.-R. Wen, Dense Text Retrieval based on Pretrained Language Models : A Survey, ACM Transactions on Information Systems (2023). doi :10.1145/3637870.

[19] J. Johnson, M. Douze, H. Jegou, Billion-Scale Similarity Search with GPUs, IEEE Transactions on Big Data 7 (2021) 535–547. doi :10.1109/TBDATA.2019.2921572.

[20] Z. Akkalyoncu Yilmaz, S. Wang, W. Yang, H. Zhang, J. Lin, Applying BERT to Document Retrieval with Birch, in : S. Padó, R. Huang (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) : System Demonstrations, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 19–24. doi :10.18653/v1/D19-3004.