Impact des collections sur les performances des Systèmes de Recherche d'Information

Thi Hoang Thi Pham, Petra Galuščáková, Philippe Mulhem, Gabriela Gonzalez-Saez, Lorraine Goeuriot

Impact des collections sur les performances des Systèmes de Recherche d'Information

Thi Hoang Thi Pham[†], Petra Galuščáková[‡], Philippe Mulhem[§], Gabriela Nicole González Sáez[§] and Lorraine Goeuriot[§]

[1]*LS2N, Université de Nantes*
[2]*University of Stavanger, Norvège*
[3]*Univ. Grenoble Alpes, CNRS, Grenoble INP*[*0], LIG, 38000 Grenoble, France*

Abstract

Cet article est une étude préliminaire sur les évolutions des corpus et leur impact sur les performances des systèmes de recherche d'information. Nous proposons une approche pour créer des corpus intermédiaires entre deux existants, puis de mesurer leurs différences suivant plusieurs caractéristiques. Nous étudions ensuite les corrélations entre les différences entre les caractéristiques et les évaluations d'un certain nombre de systèmes de recherche d'information, et nous montrons que les représentations des requêtes sont des indicateurs de différences entre collections bien corrélé aux performances de plusieurs variants de systèmes de recherches d'informations.

Keywords

Evaluation, Evolution de corpus, Corrélation

1. Introduction

L'évaluation des performances des systèmes de recherche d'information (SRI) est cruciale pour connaître leurs forces et leurs faiblesses, et donc les améliorer. Les performances des SRI sont généralement évaluées à l'aide de collections de tests standard [1] et de mesures d'évaluation. Cependant, chaque collection de tests présente des caractéristiques différentes (comme le type de documents, le domaine cible considéré, etc.) et les performances des systèmes de recherche d'information en dépendent.

Le travail de recherche présenté ici a pour objectif de mieux comprendre comment les performances des systèmes IR dépendent des caractéristiques des collections de tests, ou plus exactement comment les différences entre les états successifs d'une collection de test qui évolue (le cas d'une collections de documents sur le Web par exemple) sont corrélées aux différences de performances de systèmes de recherche d'information. Nous nous concentrons particulièrement sur : la création de collections adaptées à l'étude de ces effets, sur l'extraction d'un certain nombre de caractéristiques de ces collections, ainsi que sur les calculs de différences entre ces caractéristiques. Notre processus repose pour partie sur l'utilisation de techniques de traitement

[0]Institute of Engineering Univ. Grenoble Alpes
CORIA 2024, La Rochelle
hoangthi.phamthi@gmail.com (T. H. T. Pham); galuscakova@gmail.com (P. Galuščáková); Philippe.Mulhem@imag.fr (P. Mulhem); gabriela.gonzalezsaez@gmail.com (G. N. G. Sáez); lorraine.goeuriot@imag.fr (L. Goeuriot)


de la langue naturelle neuronales pour extraire des plongements de documents et de requêtes. Le choix de l'utilisation tels plongements est lié à leurs succès dans le traitement des langues et de la recherche d'information depuis l'avènement des modèles à base de *transformers* et utilisant des mécanismes d'attention, comme BERT [2]. Ces plongements nous servent de base pour caractériser des collections de tests, avec l'hypothèse que les collections sur lesquelles nous nous focalisons sont évolutives, comme TREC-COVID [3], et ont donc de grandes similarités. Nous pouvons alors calculer la corrélation entre la dissimilarité des collections de tests suivant ces caractéristiques et différences des performances des systèmes. Nos résultats peuvent servir de point de départ pour prédire les performances des systèmes de RI sur des collections inconnues, même si dans cet article nous n'allons pas jusque là.

Le plan suivi dans cet article est le suivant. Dans le section 2, nous dressons un état de l'art des approches connexes à nos travaux. Nous détaillons notre approche en trois étapes, respectivement dans les parties 3 pour la construction des représentations des collections intermédiaires, 4 pour leur caractérisation des collections et 5 pour la détermination de leur similarité. Le résultat de nos expérimentations en section 6. Nous concluons et indiquons le futur de nos travaux en section 7.

2. État de l'art

L'objectif que nous visons dans cet article est de trouver s'il est possible de déterminer des corrélations entre les résultats de systèmes recherche d'information et les changements dans une collection de test (composée d'un corpus de documents, d'un ensemble de requêtes et d'annotations de pertinences des documents du corpus pour chaque requête) qui évolue. A notre connaissance, cette question n'est pas exactement traitée dans l'état de l'art.

Ce cadre est connexe aux travaux sur la prédiction des performances des requêtes (QPP). La QPP [4, 5, 6], vise à prédire les performances des requêtes pour un système, de manière à adapter le processus de recherche en fonction des performances prédites. Par exemple, si une catégorie de requêtes est prédite comme étant mal traitée, alors il est possible de modifier les traitements sur cette catégorie, par exemple en intégrant de l'expansion de requêtes. Notre positionnement diffère cependant du QPP dans la mesure où nous proposons d'évaluer l'impact des changements dans les composants des collections de tests qui évoluent sur les performances des systèmes IR, sans viser des modification des systèmes.

D'autres travaux ont porté sur l'étude de la stabilité des évaluations des systèmes sur de multiples sous-collections. Sanderson et al. [7] a évalué les performances du système et exploré les similitudes entre les répartitions aléatoires des collections. Par rapport à notre approche, ce travail 1) n'a pris en compte aucun aspect d'évolution temporelle des collections, et 2) a été réalisé avant l'avènement des représentations de texte à l'aide de modèles de transformeurs. Les mêmes remarques s'appliquent à Ferro et al. [8], qui utilisent des fragments (shards) de collection définis de manière aléatoire comme méthodologie pour étudier comment différents facteurs affectent les performances de recherche: ce travail a conclu que l'interaction entre les fragments de corpus et les requêtes a un impact significatif sur les performances du système recherche d'information. Cependant, ce travail ne se préoccupait pas de l'évolution des données et il a été réalisé avant l'émergence des plongements. Nous allons nous inspirer des ses shards,

mais nous voulons utiliser les plongements et nous centrer sur des collections évolutives.

En restant sur l'étude de la stabilité des résultats des systèmes de recherche d'information, Gonzalez-Saez et al. [9] s'est concentré sur les sous-ensembles aléatoires des collections, en ajoutant des contraintes *d'intersection* entre les sous-ensembles. Ce travail a exploré dans quelle mesure les modifications du corpus de document, des requêtes et de l'évaluation conduisent à des classements similaires des systèmes par les mesures dévaluations. Même si aucun aspect temporel, ni utilisation de représentations continues n'ont été explorés dans [9], nous réutiliserons l'idée de d'intersection contrôlée, mais pour étudier les points qui nous intéressent.

3. Construction de collections intermédiaires entres des collections évolutives

Notre idée est de proposer de créer des collections intermédiaires entre des collections de tests qui sont déjà évolutives, et de caractériser les différences entre ces collections intermédiaires et les collections sources, afin d'affiner l'étude des éléments qui sont corrélés aux performances des système de recherche d'information. Dans la suite, nous considérerons que les états successifs d'une collection de test qui évolue sont en fait chacun une collection de test différente.

Notre proposition repose sur les éléments suivants:

1. deux collections de tests C_1 et C_2, composées de documents (resp. $C_1 doc$ et $C_2 doc$), de requêtes (resp. $C_1 req$ et $C_2 req$) et jugements de pertinences (resp. $C_1 per$ et $C_2 per$), avec comme hypothèse que C_2 est une évolution de C_1. Ce dernier point implique que C_1 et C_2 ont beaucoup d'éléments en commun, en particulier au niveau des documents et des requêtes ;
2. l'ensemble des requêtes communes (c'est-à-dire ayant la même chaîne de caractères) à $C_1 req$ et $C_2 req$, noté $Q_{inter_{1,2}}$;
3. un intervalle $[\tau_b; \tau_t]$ qui caractérise le niveau d'intersection choisi entre $C_1 doc$ et $C_2 doc$.

A partir de ces éléments, nous allons générer, à partir de C_1 et C_2, une collection de test notée $C_{1,2,inter_{[\tau_b;\tau_t]}}$ intermédiaire entre C_1 et C_2. Cette collection intermédiaire contiendra uniquement des documents provenants de $C_1 doc$ et $C_2 doc$, les requêtes $Q_{inter_{1,2}}$, et les jugements de pertinence cohérents avec le corpus de $C_{1,2,inter_{[\tau_b;\tau_t]}} doc$ et les requêtes $Q_{inter_{1,2}}$.

La génération de $C_{1,2,inter_{[\tau_b;\tau_t]}}$ repose sur les étapes suivantes :

1. La détermination des documents du corpus $C_2 doc$ qui ne sont pas dans $C_1 doc$, appelons cet ensemble $C_{2_{new}} doc$;
2. La sélection, parmi $C_{2_{new}} doc$, de ceux qui valident un critère fixé par rapport à $C_1 doc$. Cette étape peut prendre plusieurs formes, mais nous proposons de nous baser sur des représentations de chaque document sous forme de plongements, et d'utiliser les similarités cosinus. Avec ces plongements, nous définissons le critère de sélection par un intervalle $[\tau_b; \tau_t]$. Cet intervalle décrit des pourcentages par rapport à la taille de $C_{2_{new}} doc$, et est utilisé comme suit : on trie les documents de $C_{2_{new}} doc$ par ordre croissant de similarité avec $C_1 doc$ (cette similarité est définie comme le cosinus minimum entre le document de $C_{2_{new}} doc$ considéré et tous les documents de $C_1 doc$), et on garde les documents qui sont entre τ_b% et τ_t% de cette liste ;

3. La création de $C_{1,2,inter_{[\tau_b;\tau_t]}}doc$ par l'union de C_1doc et des documents de l'étape 2 ci-dessus ;
4. La création de $C_{1,2,inter_{[\tau_b;\tau_t]}}req$ comme étant égale à $Q_{inter_{1,2}}$;
5. La création de $C_{1,2,inter_{[\tau_b;\tau_t]}}per$ en ajoutant à C_1per les évaluations de C_2per limitées aux requêtes de $Q_{inter_{1,2}}$ et aux documents de l'étape 2.

En faisant varier τ_b et τ_t, il est donc possible de construire plusieurs collections intermédiaires entre C_1 et C_2. Si les bornes de l'intervalle $[\tau_b; \tau_t]$ sont proches de zéro, alors la collection intermédiaire est proche de C_1, et les bornes de l'intervalle $[\tau_b; \tau_t]$ sont proches de un, alors la collection intermédiaire plus éloigné de C_1 car les documents de C_2 ajoutés sont les plus dissimilaires de C_1. Nous allons expérimenter plusieurs collections intermédiaires dans la partie 6.

4. Caractéristiques des collections étudiées

Dans cette partie, nous nous intéressons à caractériser une collection C. Les descriptions que nous proposons sont applicables à toute collection de test composée de documents, requêtes et jugements de pertinence, mais notre objectif est de s'en servir comme source de calculs de dissimilarités entre collections de tests qui sont proches.

Nous rappelons que notre objectif est de déterminer si des variations de caractéristiques de collections sont corrélées à des variations de performances de SRI. Pour cela, nous allons définir des caractéristiques pour les corpus ainsi que pour les requêtes, afin de pouvoir les comparer pour plusieurs collections. Nous choisissons de caractériser une collection de test C au travers de ces éléments centraux, que sont le corpus de document $Cdoc$ et l'ensemble des requêtes $Qreq$.

Pour le corpus $Cdoc$, nous définissons une représentation basée sur un regroupement (clustering) de plongements, afin d'avoir une représentation à la fois précise et compacte d'un corpus. Les étapes sont les suivantes :

- Pour chaque d de $Cdoc$, nous calculons son plongement, v_d. Les approches les plus performantes sur des tâches de traitement de langue naturelle sont des approches à base de *Tansformers* comme XLNet [10], Longformer [11], MPNet [12] RoBERTa [13], ou DistilBERT [14]. Dans notre cas, nous devons considérer le fait que les plongements doivent être compatibles avec des documents longs, et dans cette optique des modèles MPNet [12] et Longformer [11] sont préférables.
- Nous appliquons ensuite un regroupement (par exemple par Kmeans [15]) afin de déterminer n clusters, chacun identifié par son centroïde $cent_n$.
- $Cdoc$ est donc représenté par l'ensemble des centroïdes $cent_n$.

Cette étape repose sur les paramètres suivants : le calcul de plongement utilisé, l'algorithme de regroupement, le nombre de clusters voulus. Il est alors possible de comparer les corpus de deux collections, en se basant sur des calculs de distances dans l'espace de plongements, entre les documents d'une collections et les centroïdes de l'autre.

Pour les requêtes q de $Creq$, nous nous reposons également sur des plongements. Cependant, nous faisons la choix de ne pas utiliser les plongements du texte des requêtes elles-mêmes, car le texte d'une requête est très court et qu'il n'est pas forcément syntaxiquement correct. Notre idée est alors de faire reposer la représentation de q sur les plongements des documents qui sont pertinents d'après les valeurs de pertinences $Cper$. Cette représentation est moins focalisée que le texte de la requête, mais les plongements des documents sont supposés être plus précis. Cette idée est librement inspirée des travaux de Rocchio [16] sur le bouclage de pertinence, et est décrite par :

- Pour chaque q de $Creq$, nous calculons un plongement, v_q. Pour cela, nous sélections tous les documents pertinents pour q, nous calculons leur plongement (similairement à ce qui est décrit pour les documents). v_q est ensuite égal à la moyenne des plongements de ces documents.
- $Creq$ est donc représenté par l'ensemble des v_q.

Il est alors possible de comparer les requêtes communes de deux collections, en se basant sur des calculs de distances dans l'espace de plongements entre les représentations des mêmes requêtes.

5. Dissimilarité entre les collections

En se basant sur les caractéristiques décrites ci-dessus, nous proposons les mesures suivantes de similarité entre une collection A et B, basées sur les espaces de plongements :

- Pour les corpus, nous agrégeons les dissimilarités cosinus (notée $discos$[1]) entre chaque centroïde des clusters du corpus de A et de la représentation du corpus de B : nous avons choisi dans nos expérimentations (cf partie 6) d'explorer les résultats en utilisant deux agrégations de ces dissimilarités élémentaires : le minimum (qui donne des dissimilarités notées $dis_{doc_{MIN}}$) et le maximum (qui donne des dissimilarités notées $dis_{doc_{MAX}}$);
- Pour les requêtes, nous nous calculons la moyenne de la similarité de la représentation de chaque requête de A et de B. Dans le cas des requêtes, nous calculons le cosinus entre les représentations de la même requête dans les deux collections. Cette dissimilarité est notée dis_{req}

Ces propositions ont comme hypothèse que les espaces de représentations des corpus et des requêtes sont les mêmes.

6. Expérimentations

6.1. Cadre expérimental

Nous définissons ici les expérimentations que nous avons menées pour valider notre approche.

Nous avons choisi de nous baser sur les deux derniers tours *rounds*, notés respectivement C_4 et C_5, de la collection TREC-COVID [3]. Ces collections rentrent bien, par construction,

[1] $discos(a,b) = \frac{1-cos(a,b)}{2}$

dans le cadre de collections qui évoluent au cours du temps car les tours représentent des *photographies* à certains moments des informations obtenues au fil de l'eau. Ces deux collections contiennent respectivement 157 000 et 191 000 documents, ainsi que 46 000 et 69 000 évaluations de pertinence. Nous utilisons les 45 requêtes communes entre des collections pour définir Q_{inter}. A partir de ces deux collections, nous avons créé deux collections de test intermédiaires avec les paramètres suivants, en nous basant sur les descriptions de la partie 3 :

- $\mathrm{C}_{4,5,inter_{[0;0,4]}}$ qui est intermédiaire entre TC_4 et TC_5, et avec un intervalle de sélection $[0\%; 40\%]$. Cette collection est donc une extension de C_4 avec 40% des nouveau documents de C_5 les plus proches de C_4 ;
- $\mathrm{C}_{4,5,inter_{[0,6;1]}}$ qui est intermédiaire entre TC_4 et TC_5, et avec un intervalle de sélection $[60\%; 100\%]$. Cette collection est donc une extension de C4 avec 40% des nouveau documents de C_5 les plus éloignés de C_4.

D'après ce que nous avons décrit précédemment, notre proposition repose sur l'utilisation de plongements de documents. Après une série d'expérimentations non décrites ici, nous avons opté pour l'architecture MPNet [12] pour ces calculs, qui est capable de gérer des documents longs. D'autre part, nous avons choisi d'appliquer un clustering des Kmoyennes avec 500 clusters, à la suite d'expérimentations préliminaires non-décrites ici.

Nos expérimentations utilisent les évaluation de systèmes de recherche d'information. Nous avons choisi d'utiliser dans ce travail des systèmes non-neuronaux, en employant des variations du modèle BM25 proposées par pyterrier [17], avec les paramètres par défaut et les traitements suivants : sans antidictionnaire et sans troncature (sAsT), avec l'antidictionnaire par défaut de pyterrier et sans troncature (aAsT), sans antidictionnaire et avec troncature (sAaT), et enfin avec antidictionnaire et avec troncature (aAaT). Ces variations nous permettrons d'étudier la sensibilité des systèmes à ces paramètres.

Les similarités entre ces collections décrites plus hauts seront comparées à la *différence relative* de MAP (Mean Average Precision), notée $\Delta(MAP)$ dans la suite.

6.2. Résultats préliminaires

Avec le cadre défini plus haut, nous sommes en mesure de calculer les performances des 4 systèmes de RI sur les 4 collections considérées (C4, C5, $\mathrm{C}_{4,5,inter_{[0;0,4]}}$, $\mathrm{C}_{4,5,inter_{[0,6;1]}}$. Comme nos calculs ne sont pas symétriques (aussi bien pour les différences relatives de mAP que les différences entre collections) nous calculons les 12 différences relatives de mAP et les différences des collections. Sur ces couples de valeurs, il ne reste ensuite qu'à calculer les corrélations entre ces ensembles de 12 valeurs.

Le tableau 1 présente la corrélation entre les résultats obtenus par les 4 versions de BM25 et les différentes différences entre les 4 collections. Ces valeurs de corrélations sont calculées par le coefficient de Pearson. Nous utilisons de plus la *p-value* des ces coefficients afin de mesurer la significativité statistique des valeurs obtenues[2].

[2]La fonction *pearsonr* de la librairie python *scipy.stats* a été utilisée pour calculer les corrélations et leur p-value.

Corrélation	sAsT	aAsT	sAaT	aAaT
$dis_{doc_{MIN}} \times \Delta(MAP)$	0.4924	0.4807	0.5123*	0.5015*
$dis_{doc_{MAX}} \times \Delta(MAP)$	-0.1885	-0.1724	-0.2160	-0.1992
$dis_{req} \times \Delta(MAP)$	0.8405**	0.8381**	0.8401**	0.8377**

Table 1
Corrélation (coefficient de Pearson) entre la différence relative de MAP entre les corpus pour les 4 systèmes testés et les différences calculées entre les corpus. ** dénote une p-value < 0.1. ** dénote une p-value < 0.05.

6.3. Dicussion

De l'analyse du tableau 1, nous pouvons tirer un certain nombre d'enseignements :

- Les corrélations les plus élevées de la table 1 apparaissent avec les similarités entre les collections basées sur les requêtes, avec des valeurs de corrélations positives et supérieures à 83%. Il est à noter que ces valeurs sont associées à des p-values inférieures à 5%, ceci veut dire que vous avons une grande confiance dans ces résultats. Ces valeurs de coefficients sont donc très grandes, et ceci pour toutes les variations des SRI testées. On en conclue que, les différences entre collections calculées sont donc de bons représentants des différences de MAP.
- Dans le cas de la représentation des collections basées uniquement sur les documents, les résultats sont moins clairs : les similarités reposant sur le maximum de similarités entre centroïdes ont toutes des corrélations négatives et proche de -20% avec les différences relatives de mAP, ce qui n'est pas concluant. Par contre, on remarque qu'en considérant le minimum de similarités entre les centroïdes les corrélations sont positives et proches de 50%, avec des valeurs de significativité statistiques inférieures à 0,1 pour les deux systèmes qui utilisent de la troncature, sAaT et aAaT.

Il résulte de cette discussion que l'utilisation d'un collection de test par les plongements des documents pertinents des requêtes semble être un bonne manière d'estimer des différences relatives de MAP.

7. Conclusion

Dans le travail présenté ici, nous avons proposé de fournir un cadre pour comparer les similarités entre des collections de test et leur corrélation avec les performances de plusieurs variations de systèmes de recherche d'information. Nous utilisons des plongements pour représenter les collections, et nous proposons des manières de les comparer. A partir des expérimentations préliminaires que nous avons effectuées, nous trouvons que l'utilisation de plongements des documents pertinents pour caractériser les requêtes d'une collection de test est corrélée aux variations de performance en terme de MAP.

Nos expérimentations sont préliminaires, car elles ne considèrent que quelques collections, en se basant sur deux collections intermédiaires. Il est donc nécessaire de continuer cette étude sur des plus amples ensembles de collections de test. Pour cela, nous générerons de nouvelles

collections intermédiaires par le processus décrit en partie 3. Ce travail est limité par certains paramètres, en particulier le choix des plongements, le nombre de clusters considérés lors de la représentation des corpus. Ces éléments devront être étudiés davantage à l'avenir. Une étude des résultats de systèmes de recherche neuronaux devra également être conduite pour valider nos résultats. Pour les représentations des collections en utilisant leurs requêtes, nous avons fixé que nous nous basions sur des requêtes communes : ce choix est limitant, et devra être dépassé pour permettre de mieux intégrer l'évolution des requêtes.

Remerciements

Ce travail a été partiellement financé par le projet Kodicare, ANR-19-CE23-0029, de l'Agence Nationale de la Recherche.

References

[1] M. Sanderson, Test collection based evaluation of information retrieval systems, Foundations and Trends® in Information Retrieval 4 (2010) 247–375. URL: http://dx.doi.org/10.1561/1500000009. doi:10.1561/1500000009.

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: 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. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[3] E. M. Voorhees, I. Soboroff, T. Alam, W. Hersh, K. Roberts, D. Demner-Fushman, K. Lo, L. Wang, S. Bedrick, Trec-covid: Constructing a pandemic information retrieval test collection, ACM SIGIR Forum (2021). doi:https://doi.org/10.1145/3451964.3451965.

[4] C. Hauff, Predicting the effectiveness of queries and retrieval systems, SIGIR Forum 44 (2010) 88. URL: https://doi.org/10.1145/1842890.1842906. doi:10.1145/1842890.1842906.

[5] S. Datta, D. Ganguly, D. Greene, M. Mitra, Deep-qpp: A pairwise interaction-based deep learning model for supervised query performance prediction, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, WSDM '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 201–209. URL: https://doi.org/10.1145/3488560.3498491. doi:10.1145/3488560.3498491.

[6] N. Nagpal, Query expansion for information retrieval using word embeddings: A comparative study, 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS) (2022) 289–293.

[7] M. Sanderson, A. Turpin, Y. Zhang, F. Scholer, Differences in effectiveness across subcollections, 2012, pp. 1965–1969. doi:10.1145/2396761.2398553.

[8] N. Ferro, Y. Kim, M. Sanderson, Using collection shards to study retrieval performance effect sizes, ACM Trans. Inf. Syst. 37 (2019). URL: https://doi.org/10.1145/3310364. doi:10.1145/3310364.

[9] G. Gonzalez-Saez, P. Mulhem, L. Goeuriot, P. Galuščáková, Multi-element protocol on IR experiments stability: Application to the TREC-COVID test collection ⋆, in: CIRCLE (Joint Conference of the Information Retrieval Communities in Europe), Samatan, France, 2022. URL: https://hal.science/hal-03719613.

[10] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

[11] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, CoRR abs/2004.05150 (2020). URL: https://arxiv.org/abs/2004.05150. arXiv:2004.05150.

[12] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: http://arxiv.org/abs/1907.11692, cite arxiv:1907.11692.

[14] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).

[15] M. Yuan, J. Zobel, P. Lin, Measurement of clustering effectiveness for document collections, Information Retrieval Journal 25 (2022) 239–268. URL: https://doi.org/10.1007/s10791-021-09401-8. doi:10.1007/s10791-021-09401-8.

[16] J. J. Rocchio, relevance feedback in information retrieval, https://sigir.org/files/museum/pub-08/XXIII-1.pdf, 1965. Accédé le 25/01/2024.

[17] C. Macdonald, N. Tonellotto, Declarative experimentation in information retrieval using PyTerrier, in: Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval, ICTIR 2020, ACM, 2020. URL: https://doi.org/10.1145%2F3409256.3409829. doi:10.1145/3409256.3409829.