TournaRank : Quand la recherche d'information devient un tournoi entre documents

Gilles Hubert, Yoann Pitarch, Karen Pinel-Sauvagnat, Ronan Tournier, Léa Laporte

▶ **To cite this version:**

Gilles Hubert, Yoann Pitarch, Karen Pinel-Sauvagnat, Ronan Tournier, Léa Laporte. TournaRank : Quand la recherche d'information devient un tournoi entre documents. 15e Conference francophone en Recherche d'Information et Applications (CORIA 2018), May 2018, Rennes, France. pp.1-16. hal-02305352

HAL Id: hal-02305352
https://hal.science/hal-02305352

Submitted on 4 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:
http://oatao.univ-toulouse.fr/22427

Official URL

DOI : https://doi.org/10.24348/coria.2018

To cite this version: Hubert, Gilles and Pitarch, Yoann and Pinel-Sauvagnat, Karen and Tournier, Ronan and Laporte, Léa *TournaRank : Quand la recherche d'information devient un tournoi entre documents.* (2018) In: 15e Conference francophone en Recherche d'Information et Applications (CORIA 2018), 14 May 2018 - 18 May 2018 (Rennes, France).

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

TournaRank : Quand la recherche d'information devient un tournoi entre documents

Gilles Hubert[*], Yoann Pitarch[*], Karen Pinel-Sauvagnat[*], Ronan Tournier[*], Léa Laporte[**]

[*] *IRIT UMR 5505, Université de Toulouse, France*
[**] *LIRIS UMR 5205, INSA Lyon, France*

RÉSUMÉ. *De nombreuses approches supervisées utilisant les caractéristiques des documents ont été proposées pour l'ordonnancement de documents. Un inconvénient est qu'elles requièrent une phase d'apprentissage. Dans cet article, nous proposons TournaRank, une approche non supervisée d'ordonnancement de documents inspirée des compétitions sportives. Les documents sont représentés par un ensemble de caractéristiques et s'affrontent lors de tournois. Un tournoi est vu comme une séquence de matchs au cours desquels deux documents s'affrontent sur la base des valeurs de leurs caractéristiques. À l'issue du tournoi, les documents sont ordonnés dans l'ordre décroissant des scores obtenus durant le tournoi. Notre approche est entièrement paramétrable (type de tournoi, règle des matchs, etc). Nous présentons les résultats de notre approche sur trois collections (Robust2004, Web2014, LETOR3.0) et la comparons à différentes approches de l'état de l'art en Learning to Rank et fusion des résultats. Cet article est une version résumée de (Hubert* et al.*, 2018).*

ABSTRACT. *Numerous supervised ranking models based on document features have been proposed in Information Retrieval. Such models require a learning phase. We propose TournaRank, an unsupervised approach inspired by sport competition principles. Documents compete against each other in tournaments using features as evidences of relevance. Tournaments consist of sequences of matches, which involve pairs of documents playing in turn their features. Once a tournament is ended, documents are ranked according to their scores obtained during the tournament. This principle is generic and flexible since it can be applied to any collection type and different variants can be considered (the tournament type, the match rules, etc). TournaRank was evaluated on three collections (TREC Robust2004, Web2014, and LETOR3.0) and compare to various state-of-the-art Learning to Rank and fusion approaches.*

MOTS-CLÉS : *Modèle de RI, Représentation centrée caractéristiques, Tournoi.*

KEYWORDS: *IR model, Feature-based representation, Tournament.*

1. Introduction

En Recherche d'Information (RI), proposer à un utilisateur une liste ordonnée de documents implique d'être capable d'en évaluer la pertinence vis-à-vis du besoin de l'utilisateur exprimé sous forme de requête. De nombreux modèles tels que BM25, Tf.Idf ou la mesure Cosinus ont été proposés pour cette tâche. Des travaux récents considèrent une représentation des documents sous forme de vecteurs de caractéristiques et proposent des approches d'apprentissage supervisées ou non supervisées pour prédire la pertinence d'un document vis-à-vis d'une requête. Les approches supervisées d'apprentissage d'ordonnancement (*Learning to rank*) (Liu, 2011) produisent en général d'excellents résultats, à condition que des données d'apprentissage soient disponibles en quantité suffisante. À notre connaissance, les approches non supervisées, qui ne nécessitent pas de phase d'apprentissage, n'ont que peu été étudiées dans ce domaine.

Dans cet article, notre objectif est double : tirer partie de la représentation centrée sur les caractéristiques des documents pour combiner différentes facettes de pertinence au sein d'une approche non-supervisée et dépasser le cadre du simple appariement requête-document en introduisant un nouveau modèle de comparaison de paires de documents. Nous proposons TournaRank, une approche non supervisée inspirée des principes des compétitions sportives. Les documents, représentées comme des vecteurs de caractéristiques (variables) s'affrontent lors de tournois constitués de matchs. Ces caractéristiques, telles que des fréquences de termes ou des scores issus de modèles de RI, sont utilisées comme attaques ou défenses durant les matchs et les documents sont ordonnés selon le nombre décroissant de points remportés lors des matchs. Notre approche est flexible et paramétrable au niveau des caractéristiques utilisées pour représenter les documents, du type de tournoi et de la façon dont les documents s'affrontent. À notre connaissance, il s'agit de la première méthode proposée utilisant ces principes. Une analyse de l'influence de chaque paramètre et de la qualité de l'approche est menée sur trois collections de référence : TREC Robust2004, Web2014 et LETOR3.0.

Cet article est une version synthétisée et traduite en français d'un article international accepté pour publication (Hubert *et al.*, 2018). Il est structuré de la façon suivante. La section 2 introduit brièvement l'état de l'art. Nous présentons notre approche dans les sections 3 et 4. Le protocole expérimental est décrit en section 5 et nous détaillons les résultats en section 6. La section 7 conclut l'article.

2. État de l'art

Notre approche présente des similitudes avec les techniques de fusion de résultats et d'apprentissage d'ordonnancement des documents, et peut être utilisée pour leur ré-ordonnancement.

Les approches de fusion de résultats permettent de combiner des résultats de recherche obtenus à partir de différents modèles ou différentes reformulations de re-

quêtes, pour fournir à l'utilisateur une liste unifiée de résultats, obtenue par agrégation sur les rangs ou les scores attribués aux documents dans les différents résultats. Les principales approches d'agrégation sur les rangs utilisées en RI sont des algorithmes heuristiques, comme Borda (de Borda, 1781) ou RRF (Cormack *et al.*, 2009). Les approches d'agrégation sur les scores utilisent une fonction de calcul qui combine les scores de chaque document dans les différents résultats. Les approches les plus utilisées en RI sont CombSum et CombMNZ (Fox et Shaw, 1994).

Les approches d'apprentissage d'ordonnancement optimisent automatiquement une fonction d'ordonnancement à l'aide de données d'apprentissage. Elles utilisent une représentation des paires requête-documents sous forme de vecteurs de caractéristiques, qui sont généralement des scores issus de modèle de RI, associés à des jugements de pertinence. De nombreuses approches ont été proposées dans ce cadre (Burges *et al.*, 2005 ; Chapelle et Keerthi, 2010 ; Xu et Li, 2007).

Enfin, les approches de ré-ordonnancement ont pour objectif de ré-ordonner une liste de documents restitués par un moteur de recherche, pour en améliorer la qualité globale selon une certaine métrique d'évaluation. Les approches de ré-ordonnancement sont regroupées en deux catégories (Ye *et al.*, 2011). La première rassemble des approches basées ressources utilisant des informations additionnelles, comme des thésaurus (Qu *et al.*, 2001), des préférences utilisateurs (Fathy *et al.*, 2014) ou des comportements utilisateurs (Cai *et al.*, 2014) pour ré-ordonner les documents. D'autres méthodes considèrent également des graphes de documents et exploitent les relations inter-documents, telles que les hyperliens (Page *et al.*, 1999 ; Kleinberg *et al.*, 1999) ou les relations bibliographiques (Picard et Savoy, 2000). La deuxième catégorie, à laquelle notre approche appartient, utilise *les documents comme unique source d'information*. Les approches de cette catégorie considèrent différents types de caractéristiques, comme la date des documents (Zhang *et al.*, 2009), la taille du document ou la proximité moyenne des termes de la requête (Ye *et al.*, 2011).

3. TournaRank

3.1. *Concepts et notations*

Considérons une collection de documents $\mathcal{C}$, un ensemble de caractéristiques $\mathcal{F} = \{f_i\}$ et une requête q. Notons $\mathcal{C}_\mathcal{T}$ l'ensemble des documents qualifiés, selon la valeur d'une certaine caractéristique, pour le tournoi $\mathcal{T}$. Un tournoi $\mathcal{T}$ est une séquence de matchs entre paires de documents. Plusieurs types de tournois peuvent être considérés (cf. section 3.2). Un match $M(d_i, d_j)$ entre deux documents d_i and d_j est défini suivant un ensemble de règles (cf. section 3.3.2). Chaque document est représenté sous la forme d'un vecteur de caractéristiques $f^d \subset \mathcal{F}$ et possède une jauge de vie notée Λ, dont la valeur décroit durant le match selon les coups reçus. Au cours d'un match, chaque document suit sa propre stratégie, notée σ_d, qui consiste à ordonner ses caractéristiques. Les différentes stratégies possibles sont décrites en section 3.3.3. Le résultat du match $M(d_i, d_j)$ est représenté par la paire (p_i, p_j) de points gagnés par d_i et d_j. Le nombre de points octroyés en cas de victoire (resp. défaite)

est noté pv (resp. pd). Un match est ainsi représenté formellement par une fonction $M : \mathcal{C}_\mathcal{T} \times \mathcal{C}_\mathcal{T} \rightarrow \{pw, pd, 0\} \times \{pw, pd, 0\}$. Le nombre total de points remportés par un document d à la fin du tournoi est noté w_d.

3.2. *Définition des types de tournoi*

3.2.1. *Système de tournoi Round Robin*

Le tournoi *Round Robin*, noté $\mathcal{T}^{RR}$, est inspiré d'un type de tournoi sportif dans lequel chaque document d_i joue contre tous les documents qualifiés $d_j \in \mathcal{C}_\mathcal{T}$. À la fin du tournoi, le nombre de points w_{d_i} remportés par le document d_i est donné par $w_{d_i} = \sum_{d_j \in \mathcal{C}_\mathcal{T} - \{d_i\}} p_i$, avec $(p_i, p_j) = M(d_i, d_j)$, où p_i est le nombre de points remportés par d_i pendant son match contre d_j.

3.2.2. *Système de tournoi suisse*

Le système de tournoi suisse est inspiré des tournois d'échecs. Il vise à limiter le nombre de matchs à effectuer, tout en étant plus juste et fiable que des systèmes par élimination directe. Dans ce type de tournoi, des groupes de documents sont créés et chaque document affronte les autres documents de son groupe, et non tous ceux du tournoi. Les matchs sont programmés tour par tour entre concurrents ayant le même nombre cumulé de points après chaque tour. L'algorithme est le suivant. Le nombre de tours r est prédéfini. Une première étape consiste à créer une partition $\mathcal{G}$ des documents, sur la base du nombre de points w remportés par chacun d'eux. Une fois les groupes créés, une fonction d'appariement est utilisée pour créer les paires de documents d'un groupe qui s'affronteront au cours d'un match. Ainsi, pour chaque groupe $G_w \subseteq \mathcal{G}$, un graphe de tous les documents de G_w est créé et un algorithme de couplage est utilisé pour créer les paires (Edmonds, 1965). Tout document non apparié est ajouté au groupe qui a le nombre de points le plus proche. Le processus est répété jusqu'à qu'il ne reste qu'un seul ou aucun document non apparié. Tout document non apparié sera joué au tour suivant. Chaque document joue au plus un match par tour. Une fois l'appariement effectué, les matchs programmés sont joués et les points gagnés sont ajoutés au score actuel w_{d_i} (resp. w_{d_j}) du document d_i (resp. d_j). Le processus est répété jusqu'à ce que les r tours aient été joués. Nous notons les tournois suisses à r tours $\mathcal{T}^{SS}(r)$.

3.2.3. *Tournoi en deux phases*

Les tournois en deux phases intègrent une étape de sélection visant à identifier les documents qui seront les mieux classés tout en limitant le nombre de matchs. Dans une première étape, l'ensemble de documents est divisé en trois sous-ensembles selon la valeur d'une caractéristique qualifiante considérée (1/3 haut, 1/3 milieu et 1/3 bas de liste). Les documents de chaque sous-ensemble sont répartis aléatoirement dans ρ groupes construits de façon homogène, puis un tournoi est réalisé au sein de chaque groupe. Dans une seconde étape, les N % de documents ayant eu le plus grand score dans chaque groupe sont sélectionnés pour participer à un dernier tournoi. Ils sont

ensuite classés par valeur décroissante de score final. Les tournois suisses et *Round Robin* en deux phases sont respectivement notés $\mathcal{T}^{PSS}(r, \rho, N)$ et $\mathcal{T}^{PRR}(\rho, N)$.

3.3. *Déroulement des matchs*

3.3.1. *Comparaison entre caractéristiques*

Durant chaque match, les documents s'affrontent en comparant deux à deux les valeurs de leurs caractéristiques. Pour effectuer la comparaison, nous introduisons deux concepts clés associés à chaque caractéristique : une fonction d'ordre, notée $\preceq_i$, permettant de déterminer quel document l'emporte selon une caractéristique donnée, et une fonction de distance, notée φ_i, permettant de déterminer l'impact de chaque « coup » échangé lors du match. Nous supposons que, pour chaque caractéristique f_i, il existe une seule valeur optimale de pertinence (resp. minimale), notée $\triangle_i$ (resp. $\triangledown_i$). Les fonctions d'ordre et de distance dépendent de la position de $\triangle_i$ dans la distribution des valeurs de la caractéristique f_i. En cas de valeurs manquantes, nous supposons que f_i^d vaut $\triangledown_i$, c.-à-d., $\forall d'\ f_i^d \preceq f_i^{d'}$. Trois cas sont considérés.

Dans le premier cas, $\triangle_i$ est la plus grande valeur dans la distribution de f_i. Un exemple typique est la mesure cosinus, dont la plus grande valeur possible est 1, représentant une correspondance parfaite entre la requête et le document, soit $\triangle_i = 1$. Pour deux documents d and d', nous avons $d \preceq_i d'$ si $f_i^d \leq f_i^{d'}$ et $\varphi_i(f_i^d, f_i^{d'}) = \frac{|f_i^d - f_i^{d'}|}{std(f_i)}$. L'écart type de f_i au dénominateur permet de lisser la distribution.

Dans le deuxième cas, $\triangle_i$ est la plus petite valeur de la distribution de f_i. Les distances à la requête, comme la distance euclienne, sont des exemples typiques. La plus petite valeur possible est alors 0, représentant une correspondance parfaite entre la requête et le document, soit $\triangle_i = 0$. Étant donné deux documents d and d', nous avons alors $d \preceq_i d'$ si $f_i^d \geq f_i^{d'}$ et $\varphi_i(f_i^d, f_i^{d'}) = \frac{|f_i^d - f_i^{d'}|}{std(f_i)}$.

Dans le troisième cas, $\triangle_i$ n'est ni la plus grande ni la plus petite valeur de la distribution de f_i. Un exemple typique est le temps passé sur une page (Liu et Belkin, 2015). Pour deux documents d et d', nous avons $d \preceq_i d'$ si $|f_i^d - \triangle_i| \geq |f_i^{d'} - \triangle_i|$ et $\varphi_i(f_i^d, f_i^{d'}) = ||f_i^d - \triangle_i| - |f_i^{d'} - \triangle_i||$.

3.3.2. *Règles des matchs*

Au cours d'un match, deux documents s'affrontent en jouant à leur tour leurs caractéristiques selon la procédure présentée dans l'algorithme 1. Les deux documents débutent leur match avec des jauges de vie Λ de même taille (lignes 3–4). Chaque document ordonne les caractéristiques qu'il souhaite jouer selon une certaine stratégie $\sigma^{(\cdot)}$ (lignes 1–2). La stratégie adoptée par le document d est notée σ^d.

Le document jouant (ou *attaquant*) en premier est choisi aléatoirement (ligne 5). À chaque tour, l'attaquant sélectionne sa meilleure caractéristique, qui est retirée de sa stratégie : elle ne pourra plus être jouée (ligne 8 ou 12). Le document adversaire retire

la caractéristique correspondante de sa stratégie (ligne 2) et en récupère la valeur. Les deux valeurs sont alors comparées (ligne 2) et la jauge de vie du document avec la plus petite valeur selon la relation $\preceq_i$ est diminuée de la distance φ_i entre les deux valeurs. Ce processus est itéré jusqu'à ce que toutes les caractéristiques aient été jouées ou que la jauge de vie d'un document atteigne 0. Le document ayant le plus de vie gagne le match. Chaque victoire rapporte pw points au document gagnant et 0 au perdant. En cas d'égalité, chaque document remporte pd points.

Illustrons ce fonctionnement par un exemple (cf. figure 1). Chaque caractéristique prend ses valeurs dans l'intervalle $[0, 1]$, où 1 est la valeur optimale (cas 1 section 3.3.1). Les documents d_1 and d_2 commencent leur match avec la jauge de vie $\Lambda = 3$ et leurs caractéristiques ordonnées selon la figure 1. Nous posons $\forall f_i,\ std(f_i) = 0.5$. Le document d_1 commence le match. Il attaque d_2 en utilisant la caractéristique f_1 (sa « meilleure »). Comme $f_1^{d_1}(= 0.9)$ est supérieure $f_1^{d_2}(= 0.4)$, nous obtenons à la fin du tour $\lambda_{d_1} = 3$ et $\lambda_{d_2} = 3 - (\frac{0.9-0.4}{0.5}) = 2$. f_1 est retirée des stratégies des deux documents. d_2 joue à son tour la caractéristique f_5. Les jauges de vie sont mises à jour de la façon suivante : $\lambda_{d_1} = 3 - (\frac{0,3-0,2}{0,5}) = 2.8$ et $\lambda_{d_2} = 2$. d_1 joue alors f_6 et nous avons $\lambda_{d_1} = 2.8$ et $\lambda_{d_2} = 2.5 - (\frac{0.8-0}{0.5}) = 0.9$. d_2 utilise ensuite f_3, d'où $\lambda_{d_1} = 2.8$ and $\lambda_{d_2} = 0.9 - (\frac{0.7-0.2}{0.5}) = -0.1$. Comme $\lambda_{d_2} < 0$, d_2 est *mort*, le document d_1 remporte le match et obtient pw points (par exemple 3 points comme au football).

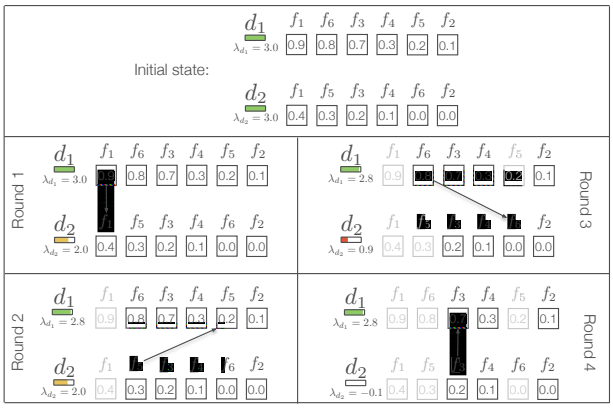


Figure 1 – Exemple d'un match avec $\Lambda = 3.0$

3.3.3. *Stratégies des documents*

Durant un tournoi, tous les documents adoptent le même type de stratégie $\sigma^{(\cdot)}$. Il existe $|\mathcal{C}_\mathcal{T}|!$ stratégies potentielles. La stratégie adoptée par un document est un élément clé pour sa victoire. Une première stratégie naïve, notée σ^v, est l'ordonnancement par valeur. Elle consiste à jouer les caractéristiques par ordre décroissant valeur (cf. figure 1). Formellement, considérons σ_d^v la stratégie d'ordonnancement par

	f_1	f_2	f_3	f_4	f_5	f_6
d_1	2	7	4	2	7	3
d_2	8	10	7	8	5	9

Tableau 1 – Classement des documents selon les caractéristiques f_1 à f_6 pour d_1 et d_2

valeur jouée par le document d, avec f_i et f_j deux caractéristiques dans σ_d^v. f_i est jouée avant f_j si $f_i^d > f_j^d$. En cas d'égalité des valeurs ($f_i^d = f_j^d$), f_i est jouée en premier si $i < j$. Une autre stratégie est l'ordonnancement par rang, notée σ^r. Elle consiste à ordonner les caractéristiques selon les rangs des documents dans les listes induites par les caractéristiques. En cas d'égalité des rangs, le choix de la caractéristiques est effectué aléatoirement. Considérons les documents d_1 et d_2 (cf. figure 1), $|\mathcal{C}_\mathcal{T}| = 10$, et le tableau 1 indiquant, pour chaque caractéristique f_i le rang des documents d_1 et d_2 selon leur relation d'ordre associée $\preceq_i$. La stratégie d'ordonnancement par rang pour le document d_1 est $f_1, f_4, f_6, f_3, f_2, f_5$ tandis que celle du document d_2 est $f_5, f_3, f_1, f_4, f_6, f_2$. L'adaptation des stratégies en cours de match ou de tournoi et l'exploration de stratégies supervisées font partie de nos perspectives de recherche futures.

4. Ré-ordonner avec TournaRank

TournaRank est une méthode de ré-ordonnancement : étant donnée une requête, un tournoi peut être lancé sur un ensemble de documents déjà ordonné. La liste de documents à ré-ordonner est celle induite par la caractéristique utilisée pour la qualification des documents. $\mathcal{C}_\mathcal{T}$ est considéré comme un ensemble ordonné, tel que $d_i > d_j$ si d_i est mieux classé que d_j selon la caractéristique qualifiante. L'ordre initial devrait de plus être pris en compte pendant le match et un avantage devrait être donné aux documents gagnant contre d'autres initialement mieux classés. Nous proposons de prendre en compte cet aspect en introduisant un facteur d'amplification $\alpha > 1$, selon deux scénarios d'amplification : *Upper* et *Seed* . Dans le scénario *Upper*, un document gagnant voit ses points amplifiés s'il était initiallement moins bien classé que son opposant. Dans le scénario *Seed*, le document gagnant voit ses points amplifiés si son opposant faisait partie des X documents les mieux classés initialement (cf. algorithme 1). Nous notons un tournoi sans amplification $\mathcal{T}^{(\cdot)}$ (avec $(\cdot) \in \{$*RR, SS, PRR, PSS*$\}$) et un tournoi avec amplification *Upper* (resp. *seed*) $\mathcal{T}^{(\cdot)_u}(\alpha)$ (resp. $\mathcal{T}^{(\cdot)_s}(\alpha, X)$).

5. Protocole expérimental

5.1. *Méthodologie*

À travers nos expérimentations, nous souhaitons répondre aux quatre questions suivantes :

Q1. Comment l'ensemble de caractéristiques influence-t-il le classement final ?

Q2. Quel est l'impact des différents paramètres sur les performances ?

Q3. Quelle combinaison de paramètres produit la meilleure liste de documents ?

Q4. Notre approche produit-elle de meilleures résultats que l'état de l'art ?

5.2. *Collections*

Dans nos expérimentations, nous considérons les collections TREC Robust2004, Web2014 et LETOR3.0. TREC Robust2004, constituée de 528,155 documents homogènes (disques 4 et 5 sans les documents Congressional Record—CR—) et 249 requêtes (Voorhees, 2004), est la collection fournie pour la tâche « Robust Retrieval » de la campagne TREC 2004. Web2014 est la collection fournie pour la tâche Web de TREC 2014. Considérée comme une collection de documents hétérogènes volumineuse, elle est composée de la collection ClueWeb12-full[1], comprenant 733 millions de pages (27.3 TB) et d'un total de 50 requêtes (Collins-Thompson *et al.*, 2014). Enfin, LETOR3.0 est une collection de référence en apprentissage d'ordonnancement[2]. Elle est composée de 7 jeux de données, un basé sur la collection Ohsumed, les autres (HP2003, HP2004, NP2003, NP2004, TD2003, TD2004) sur la collection « .Gov », et de 50 à 150 requêtes par jeu de données (Liu, 2011).

5.3. *Description des caractéristiques*

Nous avons utilisé des ensembles de caractéristiques différents par collection. Pour LETOR3.0, nous avons considéré directement les caractéristiques fournies dans les jeux de données correspondants (Liu, 2011). Pour Robust2004, nous avons défini 13 caractéristiques. Les caractéristiques f_1 à f_{10} sont utilisées en apprentissage d'ordonnancement (Liu, 2011). Les trois dernières correspondent à des modèles de l'état de l'art implémentés dans Indri[3], c.-à-d., f_{11} correspond au Tf.Idf, f_{12} au modèle Okapi, et f_{13} au modèle Indri (Metzler, 2011). Ces 13 caractéristiques ont également été utilisées pour la collection Web2014, complétées par deux caractéristiques spécifiques au Web, les scores PageRank et Spam[4]. Nous avons regroupé ces caractéristiques en trois classes : indépendantes de la requête ($\mathcal{F}_{QI}$), dépendantes de la requêtes($\mathcal{F}_{QD}$) et de type modèle ($\mathcal{F}_M$). Toutes ces caractéristiques sont décrites dans le tableau 2. Elles correspondent au scénario 1 décrit dans la section 3.3.1. Toutes ont été calculées sur le document entier et normalisées entre 0 et 1 (normalisation de type max).

1. `http://www.lemurproject.org/clueweb12.php/`
2. `http://research.microsoft.com/en-us/um/beijing/projects/letor/ /letor3dataset.aspx`
3. `http://www.lemurproject.org/doxygen/lemur/html/IndriRunQuery.html`
4. `http://lemurproject.org/clueweb12/related-data.php`

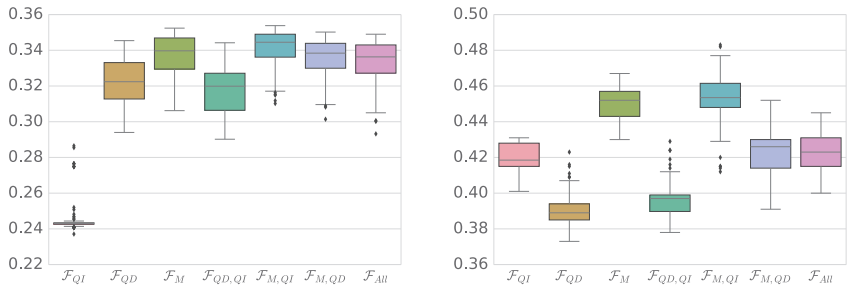
5.4. *Protocole*

Le tableau 3 présente les valeurs des paramètres utilisés pour l'évaluation de notre approche. Deux protocoles de gestion de la jauge de vie sont évalués : (1) diminuer la jauge de vie de 1 (noté 1 dans le tableau 3) ou (2) diminuer la jauge de vie à l'aide de la fonction φ_i, distance normalisée entre les valeurs de deux caractéristiques (cas 1 dans la section 3.3.1). Nous considérons différentes valeurs initiales pour la jauge de vie Λ selon le cas. Dans le cas (1), la jauge de vie peut être fixée à 20% ou 50% du nombre de caractéristiques (c.-à-d., un document peut *mourir* avant que toutes les caractéristiques n'aient été jouées), ou à ∞, c.-à-d., tous les matchs doivent avoir joué toutes les caractéristiques pour se terminer. Dans le cas (2), nous fixons la jauge de vie à 100% ou 200% du nombre de caractéristiques, ou à ∞.[5] Le pourcentage appliqué est supérieur à 100 car l'impact de chaque variable dans ce cas est supérieur à 1. Dans toutes les expériences, nous avons fixé pw (resp. pd) à 3 (resp. 1), selon les règles du football visant à promouvoir les victoires par rapport aux égalités.

Nous avons évalué la performance de notre approche à l'aide des mesures usuelles MAP, P@20 et MRR. Des tests statistiques exacts de significativité ont été appliqués selon les recommandations de (Smucker *et al.*, 2007). Pour chaque collection, nous avons choisi comme caractéristique qualifiante celle conduisant à la liste de documents restitués de meilleure qualité. Nous considérons ici 50 documents qualifiés.

Pour répondre à la question **Q1**, nous avons fait varier l'ensemble de caractéristiques sur les collections Web2014 et Robust2004 (cf. tableau 3, ligne 1). Pour chaque ensemble, nous avons mené 100 expériences en fixant aléatoirement les autres paramètres parmi les valeurs présentées dans le tableau 3 (lignes 2–5). Nous avons adopté un protocole similaire pour répondre à la question **Q2**. Pour chaque paramètre à évaluer, c.-à-d., le type de tournoi, la technique d'amplification, la gestion de la jauge de vie et le type de stratégie adoptée par les documents, nous avons mené 100 expériences en fixant aléatoirement les autres paramètres. Pour répondre aux questions **Q3** et **Q4**, nous avons comparé trois variantes de TournaRank à des approches de l'état de l'art. Deux variantes ont été choisies conformément aux résultats obtenus aux questions **Q1** et **Q2**, tandis que la troisième a été obtenue par une étape de validation croisée de 5 échantillons optimisant la P@20 (cf. section 6.4). TournaRank étant une approche non supervisée, l'étape de validation croisée sert à sélectionner la meilleure combinaison de paramètres pour le jeu de test. Nous considérons deux types d'approches de références auxquels nous comparons TournaRank : les approches de fusion et les techniques d'apprentissage d'ordonnancement. Nous considérons les approches de fusion de l'état de l'art Borda (de Borda, 1781), RRF avec k=60 (Cormack *et al.*, 2009), CombMNZ et CombSum (Fox et Shaw, 1994)) et 8 algorithmes d'apprentissage d'ordonnancement (MART, RankNet, RankBoost, AdaRank, Coordinate

5. Une étude préliminaire nous a permis de montrer que considérer une jauge de vie $\geq 200\%$ est similaire à ∞.



(a) Robust2004 (b) Web2014

Figure 2 – Variation des résultats selon l'ensemble des caractéristiques (P@20).

Ascent, LambdaMART, ListNet, et Random Forests)[6]. Ces approches ont été entraînées pour optimiser la P@20.

6. Résultats

6.1. *Influence de l'ensemble de caractéristiques considéré (Q1)*

La Figure 2 présente la variation des résultats obtenus suivant l'ensemble de caractéristiques considéré pour la P@20 (similaire aux autres métriques). Les comportements des ensembles de caractéritiques sont cohérents sur les deux collections, à l'exception des ensembles constitués uniquement de caractéristiques indépendantes de la requête. En effet, les résultats pour ces ensembles sont significativement moins bons sur Robust2004, mais équivalents à ceux des autres ensembles sur Web2014. La seule caractéristique indépendante de la requête sur Robust2004 est la taille du document, peu efficace pour ordonner les documents contrairement aux scores PageRank et Spam disponibles sur Web2014, expliquant ce comportement. Quelle que soit la collection, les meilleurs résultats sont obtenus pour l'ensemble de caractériques $\mathcal{F}_{M,QI}$, constitué de caractéristiques indépendantes de la requête et de caractéristiques de type « Modèle ». Nous observons que les combinaisons de caractéristiques hétérogènes (de type différents) produisent de meilleurs résultats que l'utilisation d'ensembles homogènes. Les performances de notre approche sont robustes sur les différents jeux de données, quels que soient les ensembles de caractéristiques considérés.

6.2. *Influence des paramètres (Q2)*

Les expérimentations montrent que les performances de TournaRank selon les métriques d'évaluation sont similaires quel que soit le type de tournoi considéré, bien

6. Nous avons utilisé la librairie RANKLIB (https ://sourceforge.net/p/lemur/wiki/RankLib/).

que les résultats soient plus variables sur Robust2004. Par manque de place, les graphiques ne sont pas reproduits ici.[7] De plus, l'utilisation du facteur d'amplification n'a pas d'influence significative sur les résultats. Nous recommandons donc de ne pas utiliser de facteur d'amplification, afin de limiter le nombre de paramètres à fixer. L'utilisation de stratégies différentes par les documents pour l'ordonnancement des caractéristiques conduit à des résultats équivalents en matière de performance. Cela peut être dû au fait que les listes partagent les mêmes documents en plus haute position. Les expérimentations montrent par ailleurs que la gestion de la barre de vie a une influence sur la qualité des résultats. Sur les deux collections considérées, utiliser la fonction φ_i conduit à de meilleurs résultats que de réduire la jauge de vie de 1. Enfin, l'analyse des temps de calcul a fait apparaître un résultat surprenant. En effet, les tournois suisses sont globalement plus gourmands en tant de calcul que les tournois Round Robin, alors que leur complexité théorique est plus faible. Cela est dû à la phase d'appariement qui contrebalance le gain en nombre de matchs. Conformément à nos attentes, les tournois en deux phases sont plus efficients. Sur la base cette étude, nous recommandons l'utilisation de tournois *Round Robin* en deux phases, représentant un compromis entre temps de calcul et qualité d'ordonnancement.

6.3. *Analyse des configurations de TournaRank (Q3)*

Nous comparons trois configurations différentes de TournaRank. Dans la première, *TournaRank (median)*, les paramètres de TournaRank sont choisis par rapport aux meilleurs résultats médians de l'analyse précédente, soient : $\mathcal{F}_{M,QI}$ (pour Web2014 et Robust2004)[8], $\mathcal{T}^{PRR}(2, 20)$, $\mathcal{T}^{(\cdot)_s}(\alpha = 3, X = 10)$, distance ($\Lambda = \infty$), σ^r. Dans la deuxième configuration, *TournaRank (max)*, les paramètres correspondent aux meilleurs résultats maximaux de l'analyse des paramètres, à savoir : $\mathcal{F}_{M,QI}$ (pour Web2014 et Robust2004)[10], $\mathcal{T}^{RR}$, $\mathcal{T}^{(\cdot)_s}(\alpha = 3, X = 20)$, distance ($\Lambda = 200\%$), σ^r. Pour la dernière configuration, *TournaRank (cross)* les paramètres sont déterminés par validation croisée en 5 échantillons sur l'union de 50 % des meilleurs valeurs médianes et de 50 % des meilleures valeurs maximales. La configuration apprise sur le jeu d'apprentissage (cf. paramètres en gras du tableau 3) est utilisée sur le jeu de test.

Les résultats de cette analyse sont présentés dans le tableau 4. Les différentes configurations obtiennent des résultats globalement similaires quelle que soit la métrique d'évaluation considérée. Les performances étant légèrement meilleures pour la configuration *max*, nous recommandons son utilisation pour des expériences préliminaires sur de futures collections. Par ailleurs, cette configuration conduit à de meilleurs résultats que *TournaRank (cross)* pour toutes les métriques pour 4 jeux de données sur 9, et pour deux métriques sur la majorité des jeux de données (8 sur 9). Il s'agit d'une configuration robuste pour tous les jeux de données considérés.

7. Disponibles dans (Hubert *et al.*, 2018).

8. Sur LETOR3.0, nous avons utilisé $\mathcal{F}_{All}$. Pour les approches de fusion, l'ensemble de caractéristiques $\mathcal{F}_{M,QI}$ a été considéré.

6.4. *Comparaison avec les approches de l'état de l'art (Q4)*

Nous comparons les configurations précédentes à différentes approches de fusion de résultats et d'apprentissage d'ordonnancement de l'état de l'art. Par souci de brièveté, nous ne considérons que la meilleure et la moins bonne approche d'apprentissage d'ordonnancement ainsi que la meilleure approche de fusion pour chaque collection. Les résultats sont présentés dans le tableau 4.

Nous remarquons que TournaRank est une approche de ré-ordonnancement performante : elle produit de meilleurs résultats que le classement initial pour au moins une métrique sur la majorité des jeux de données (6/9), et pour toutes les métriques sur 4 jeux données. De plus, TournaRank obtient de meilleurs résultats que la meilleure approche de fusion sur la majorité des jeux de données (5/9) pour toutes les métriques. Elle est significativement moins performante que la meilleure approche de fusion sur un seul jeu de données (Web2014). Par ailleurs, TournaRank est systématiquement meilleure que la moins bonne des approches d'apprentissage d'ordonnancement selon les métriques MRR et P@20. Lorque toutes les métriques sont considérées, TournaRank est plus performante que la moins bonne approche d'apprentissage d'ordonnancement pour 4 des 9 jeux de données. Considérant la meilleure approche d'apprentissage d'ordonnancement, les résultats sont variables selon la métrique. La meilleure approche d'apprentissage d'ordonnancement est plus performante selon la MAP, tandis que TournaRank est meilleure en matière de MRR et équivalente selon la P@20.

7. Conclusion

Dans cet article, nous avons proposé et évalué TournaRank, une approche non-supervisée originale et générique pour l'ordonnancement et le ré-ordonnancement de documents, indépendante de la collection et du domaine. Les documents sont représentés sous la forme de vecteurs de caractéristiques utilisés par les documents pour s'affronter lors de tournois composés de matchs. Nos résultats sur trois collections de l'état de l'art sont encourageants. Dans le futur, nous prévoyons d'étudier l'influence d'ensembles de caractéristiques plus hétérogènes ainsi que d'autres types de tournois pour lesquels les documents ne jouent pas au tour par tour.

8. Bibliographie

Burges C., Shaked T., Renshaw E., Lazier A., Deeds M., Hamilton N., Hullender G., « Learning to Rank Using Gradient Descent », *Proc. of ICML'05*, p. 89-96, 2005.

Cai F., Liang S., de Rijke M., « Personalized Document Re-ranking Based on Bayesian Probabilistic Matrix Factorization », *Proc. of SIGIR'14*, p. 835-838, 2014.

Chapelle O., Keerthi S. S., « Efficient Algorithms for Ranking with SVMs », *Inf. Retr.*, vol. 13, n° 3, p. 201-215, 2010.

Collins-Thompson K., Macdonald C., Bennett P. N., Diaz F., Voorhees E. M., « TREC 2014 Web Track Overview », *Proc. of The 23rd Text REtrieval Conf.*, TREC'14, 2014.

Cormack G. V., Clarke C. L. A., Buettcher S., « Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods », *Proc. of SIGIR'09*, p. 758-759, 2009.

de Borda J.-C., « Mémoire sur les élections au scrutin », *Histoire de l'Académie Royale des Sciences*, 1781.

Edmonds J., « Paths, trees, and flowers », *Can. J. Math.*, vol. 17, n° 3, p. 449-467, 1965.

Fathy N., Gharib T. F., Badr N., Mashat A. S., Abraham A., « A Personalized Approach for Re-ranking Search Results Using User Preferences », *J. Univers. Comput. Sci*, vol. 20, n° 9, p. 1232-1258, 2014.

Fox E. A., Shaw J. A., « Combination of multiple searches », *The 2nd Text REtrieval Conf.*, TREC-2, p. 243-252, 1994.

Hubert G., Pitarch Y., Pinel-Sauvagnat K., Tournier R., Laporte L., « TournaRank : When retrieval becomes document competition », *Information Processing & Management*, vol. 54, n° 2, p. 252 - 272, 2018.

Kleinberg J. M., Kumar R., Raghavan P., Rajagopalan S., Tomkins A. S., « The Web As a Graph : Measurements, Models, and Methods », *Proc. of COCOON'99*, Springer-Verlag, p. 1-17, 1999.

Liu J., Belkin N. J., « Personalizing information retrieval for multi-session tasks : Examining the roles of task stage, task type, and topic knowledge on the interpretation of dwell time as an indicator of document usefulness », *JASIST*, vol. 66, n° 1, p. 58-81, 2015.

Liu T.-Y., *Learning to Rank for Information Retrieval*, Springer, 2011.

Metzler D., *A Feature-Centric View of Information Retrieval*, Springer, 2011.

Page L., Brin S., Motwani R., Winograd T., The PageRank Citation Ranking : Bringing Order to the Web., Technical Report n° 1999-66, November, 1999.

Picard J., Savoy J., « A Logical Information Retrieval Model Based on a Combination of Propositional Logic and Probability Theory », *in* F. Crestani, G. Pasi (eds), *Soft Computing in Information Retrieval : Techniques and Applications*, Physica-Verlag HD, p. 225-258, 2000.

Qu Y., Xu G., Wang J., « Rerank Method Based on Individual Thesaurus », *Proc. of NTCIR-2*, 2001.

Smucker M. D., Allan J., Carterette B., « A Comparison of Statistical Significance Tests for Information Retrieval Evaluation », *Proc. of CIKM'07*, p. 623-632, 2007.

Voorhees E. M., « Overview of the TREC 2004 Robust Track », *Proc. of the 13th Text REtrieval Conf.*, TREC'04, 2004.

Xu J., Li H., « AdaRank : A Boosting Algorithm for Information Retrieval », *Proc. of SIGIR'07*, p. 391-398, 2007.

Ye Z., Huang J. X., Lin H., « Incorporating Rich Features to Boost Information Retrieval Performance : A SVM-regression based re-ranking approach », *Expert Syst. App.*, vol. 38, n° 6, p. 7569-7574, 2011.

Zhang R., Chang Y., Zheng Z., Metzler D., Nie J.-y., « Search Result Re-ranking by Feedback Control Adjustment for Time-sensitive Query », *Proc.of NAACL-Short'09*, p. 165-168, 2009.

Algorithm 1: $M(\Lambda, d_i, d_j, b, \alpha)$

Input: $\Lambda, d_i, d_j, b, \alpha$
Output: (p_i, p_j)

$\sigma_{d_i} \leftarrow$ récupérerStratégie(d_i,$\sigma^{(\cdot)}$);
$\sigma_{d_j} \leftarrow$ récupérerStratégie(d_j,$\sigma^{(\cdot)}$);
$\lambda_{d_i} \leftarrow \Lambda$;
$\lambda_{d_j} \leftarrow \Lambda$;
joueurCourant $\leftarrow$ aléatoire(d_i,d_j);
while ($\lambda_{d_i} > 0$ *et* $\lambda_{d_j} > 0$) *et* ($|\sigma_{d_i}| > 0$ *ou* $|\sigma_{d_j}| > 0$) **do**
    if *joueurCourant* = d_i **then**
        $f_{jouee} \leftarrow extraireDe(\sigma_{d_i})$;
        $\sigma_{d_j} \leftarrow \sigma_{d_j} - f_{jouee}$
        if $f^{d_j}_{jouee} \preceq_{jouee} f^{d_i}_{jouee}$ **then**
            $\lambda_{d_j} \leftarrow \lambda_{d_j} - \varphi_{jouee}(f^{d_i}_{jouee}, f^{d_j}_{jouee})$;
        else
            $\lambda_{d_i} \leftarrow \lambda_{d_i} - \varphi_{jouee}(f^{d_i}_{jouee}, f^{d_j}_{jouee})$;
        if $|\sigma_{d_j}| > 0$ **then** $joueurCourant \leftarrow d_j$;
    else
        $f_{jouee} \leftarrow extraireDe(\sigma_{d_j})$;
        $\sigma_{d_i} \leftarrow \sigma_{d_i} - f_{jouee}$
        if $f^{d_i}_{jouee} \preceq_{jouee} f^{d_i}_{jouee}$ **then**
            $\lambda_{d_i} \leftarrow \lambda_{d_i} - \varphi_{jouee}(f^{d_j}_{jouee}, f^{d_i}_{jouee})$;
        else
            $\lambda_{d_j} \leftarrow \lambda_{d_j} - \varphi_{jouee}(f^{d_j}_{jouee}, f^{d_i}_{jouee})$;
        if $|\sigma_{d_i}| > 0$ **then** $joueurCourant \leftarrow d_i$;
if $\lambda_{d_i} > \lambda_{d_j}$ **then**
    if *(*$b = upper$ *et* $d_j > d_i$*) OU (*$b = seed$ *et* d_i *dans le top* $X\%$*)* **then**
        return $(\alpha * pw, 0)$;
    else
        return $(pw, 0)$;
else if $\lambda_{d_j} > \lambda_{d_i}$ **then**
    if *(*$b = upper$ *et* $d_i > d_j$*) OU (*$b = seed$ *et* d_j *dans le top* $X\%$*)* **then**
        return $(0, \alpha * pw)$;
    else
        return $(0, pw)$;
else
    return (pd, pd);

ID	Caractéristique	Robust2004	Web2014	Type
f_1	$\sum_{t_i \in q \cap d} TF(t_i, d)$	$\surd$	$\surd$	$\mathcal{F}_{QD}$
f_2	$\sum_{t_i \in q \cap d} IDF(t_i)$	$\surd$	$\surd$	$\mathcal{F}_{QD}$
f_3	$\sum_{t_i \in q \cap d} TF(t_i, d) \cdot IDF(t_i)$	$\surd$	$\surd$	$\mathcal{F}_{QD}$
f_4	$\sum_{t_i \in q \cap d} log(TF(t_i, d))$	$\surd$	$\surd$	$\mathcal{F}_{QD}$
f_5	$LEN(d)$	$\surd$	$\surd$	$\mathcal{F}_{QI}$
f_6	$\sum_{t_i \in q \cap d} \frac{TF(t_i,d)}{LEN(d)}$	$\surd$	$\surd$	$\mathcal{F}_{QD}$
f_7	$\sum_{t_i \in q \cap d} \frac{log((TF(t_i,d))}{LEN(d)}$	$\surd$	$\surd$	$\mathcal{F}_{QD}$
f_8	$\sum_{t_i \in q \cap d} log\left(\frac{\lvert C \rvert}{TF(t_i,C)} + 1\right)$	$\surd$	$\surd$	$\mathcal{F}_{QD}$
f_9	$\sum_{t_i \in q \cap d} TF(t_i, d) \cdot (log(\lvert C \rvert . IDF(t_i))$	$\surd$	$\surd$	$\mathcal{F}_{QD}$
f_{10}	$\sum_{t_i \in q \cap d} log\left(\frac{TF(t_i,d)}{LEN(d)} \cdot \frac{\lvert C \rvert}{TF(t_i,C)} + 1\right)$	$\surd$	$\surd$	$\mathcal{F}_{QD}$
f_{11}	Tf.Idf (Implémentation Indri)	$\surd$	$\surd$	$\mathcal{F}_M$
f_{12}	Okapi-BM25 avec $k1 = 1.2$, $b = 0.75$, and $k3 = 7$ (Implémentation Indri)	$\surd$	$\surd$	$\mathcal{F}_M$
f_{13}	Modèle Indri	$\surd$	$\surd$	$\mathcal{F}_M$
f_{14}	Score PageRank		$\surd$	$\mathcal{F}_{QI}$
f_{15}	Score Spam		$\surd$	$\mathcal{F}_{QI}$

Tableau 2 – Description des caractéristiques sur Web2014 et Robust2004. $\mathcal{F}_{QD}$, $\mathcal{F}_{QI}$ et $\mathcal{F}_M$ signifient *Dépendant de la requête*, *Indépendant de la requête* et *Modèle*.

Paramètre	Valeur
$\mathcal{F}$	$\boldsymbol{\mathcal{F}_{\mathbf{M}}}$; $\boldsymbol{\mathcal{F}_{\mathbf{QD}}}$; $\mathcal{F}_{QI}$; $\boldsymbol{\mathcal{F}_{\mathbf{M,QD}}}$; $\boldsymbol{\mathcal{F}_{\mathbf{M,QI}}}$; $\mathcal{F}_{QD,QI}$; $\boldsymbol{\mathcal{F}_{\mathbf{All}}} = \mathcal{F}_{\mathbf{M,QD,QI}}$
$\mathcal{T}$	$\boldsymbol{\mathcal{T}^{\mathbf{RR}}}$; $\boldsymbol{\mathcal{T}^{\mathbf{SS}}}(\mathbf{r = 10})$; $\boldsymbol{\mathcal{T}^{\mathbf{SS}}}(\mathbf{r = 20})$; $\boldsymbol{\mathcal{T}^{\mathbf{SS}}}(\mathbf{r = 50})$; $\boldsymbol{\mathcal{T}^{\mathbf{PRR}}}(\rho = \mathbf{2}, \mathbf{N = 10})$; $\boldsymbol{\mathcal{T}^{\mathbf{PRR}}}(\rho = \mathbf{2}, \mathbf{N = 20})$; $\mathcal{T}^{PRR}(\rho = 5, 10)$; $\boldsymbol{\mathcal{T}^{\mathbf{PRR}}}(\rho = \mathbf{5}, \mathbf{N = 20})$; $\mathcal{T}^{PSS}(r = 10, \rho = 2, N = 10)$; $\mathcal{T}^{PSS}(r = 10, \rho = 2, N = 20)$; $\mathcal{T}^{PSS}(r = 10, \rho = 5, N = 10)$; $\mathcal{T}^{PSS}(r = 10, \rho = 5, N = 20)$; $\boldsymbol{\mathcal{T}^{\mathbf{PSS}}}(\mathbf{r = 20}, \rho = \mathbf{2}, \mathbf{N = 10})$; $\boldsymbol{\mathcal{T}^{\mathbf{PSS}}}(\mathbf{r = 20}, \rho = \mathbf{2}, \mathbf{N = 20})$; $\mathcal{T}^{PSS}(r = 20, \rho = 5, N = 10)$; $\mathcal{T}^{PSS}(r = 20, \rho = 5, N = 20)$; $\boldsymbol{\mathcal{T}^{\mathbf{PSS}}}(\mathbf{r = 50}, \rho = \mathbf{2}, \mathbf{N = 10})$; $\mathcal{T}^{PSS}(r = 50, \rho = 2, N = 20)$; $\mathcal{T}^{PSS}(r = 50, \rho = 5, N = 10)$; $\boldsymbol{\mathcal{T}^{\mathbf{PSS}}}(\mathbf{r = 50}, \rho = \mathbf{5}, \mathbf{N = 20})$
Amplification	$\boldsymbol{\mathcal{T}}^{(\cdot)}$; $\boldsymbol{\mathcal{T}^{(\cdot)_{\mathbf{u}}}}(\boldsymbol{\alpha} = \mathbf{3})$; $\boldsymbol{\mathcal{T}^{(\cdot)_{\mathbf{u}}}}(\boldsymbol{\alpha} = \mathbf{5})$; $\boldsymbol{\mathcal{T}^{(\cdot)_{\mathbf{s}}}}(\boldsymbol{\alpha} = \mathbf{3}, \mathbf{X = 10})$; $\boldsymbol{\mathcal{T}^{(\cdot)_{\mathbf{s}}}}(\boldsymbol{\alpha} = \mathbf{3}, \mathbf{X = 20})$; $\mathcal{T}^{(\cdot)_s}(\alpha = 5, X = 10)$; $\boldsymbol{\mathcal{T}^{(\cdot)_{\mathbf{s}}}}(\boldsymbol{\alpha} = \mathbf{5}, \mathbf{X = 20})$
Impact	$1\ (\Lambda = 20\%)$; $1\ (\Lambda = 50\%)$; $1\ (\Lambda = \infty)$; **distance** $(\boldsymbol{\Lambda = 100\%})$; **distance** $(\boldsymbol{\Lambda = 200\%})$; **distance** $(\boldsymbol{\Lambda = \infty})$
Type de stratégie	$\boldsymbol{\sigma^{\mathbf{v}}}$;$\boldsymbol{\sigma^{\mathbf{r}}}$

Tableau 3 – Valeurs de paramètres utilisées dans nos expérimentations. Les paramètres en gras sont utilisés pendant la phase de validation croisée de 5 échantillons.

	Run	MAP	P@20	MRR
Web2014	Median	0.1192††	**0.4670**††	0.4786*
	Max	0.1190††*	0.4580†	0.4921
	Cross	0.1176†*	0.4420	0.5585†
	RankBoost	**0.5091**	0.4650	0.5400
	ListNet	0.4753	0.4110	0.4200
	Borda	0.1223††	0.4660††	**0.5746**††
	Indri	0.1132	0.4250	0.4299
Robust2004	Median	0.1858†	0.3494	0.6906
	Max	0.1864††*	0.3534	0.6893
	Cross	0.1798	0.3473	0.6769
	CA	**0.4835**	**0.3639**	0.5480
	ListNet	0.4605	0.3167	0.4800
	RRF	0.1832	0.3536	**0.7000**
	Indri	0.1817	0.3490	0.6773
HP2003	Median	0.6449†	0.0530	0.6859†
	Max	0.6430†	0.0533	0.6856†
	Cross	0.6348††*	0.0530	0.6784†
	RankBoost	**0.7538**	**0.0550**	0.6800
	AdaRank	0.2965	0.0417	0.1933
	CombMNZ	0.6634	0.0530	0.7029
	Initial ranking	0.7024	0.0537	**0.7375**

	Run	MAP	P@20	MRR
HP2004	Median	0.5990	0.0507	0.6120
	Max	0.5988	0.0507	0.6121
	Cross	0.5514	0.0487	0.5636
	RankBoost	**0.6783**	**0.0513**	0.5733
	ListNet	0.1194	0.0227	0.0933
	CombMNZ	0.5547	0.0500	0.5673
	Initial ranking	0.6169	**0.0513**	**0.6239**
NP2003	Median	0.6522†*	0.0457	0.6496†*
	Max	0.6522†*	0.0457	**0.6522**†*
	Cross	0.6477†*	0.0453	0.6471†*
	RankBoost	**0.6671**	**0.0473**	0.5533
	AdaRank	0.2319	0.0313	0.1400
	RRF	0.5959	0.0457	0.5951
	Initial ranking	0.5777	0.0433	0.5792
NP2004	Median	0.5901	0.0473	0.5954
	Max	0.5912	0.0473	0.5962
	Cross	0.5185	**0.0480**†	0.5245
	RF	0.5974	**0.0480**	0.4800
	ListNet	0.1289	0.0313	0.0267
	CombMNZ	**0.6204**	0.0460†	**0.6193**†
	Initial ranking	0.5197	0.0427	0.5186

	Run	MAP	P@20	MRR
TD2003	Median	0.1762	0.0980	0.4481
	Max	0.1760	0.1020	0.4505
	Cross	0.1634	0.0970	0.4376
	RankBoost	**0.3929**	**0.1220**	0.3400
	ListNet	0.2669	0.0770	0.1200
	CombMNZ	0.1757	0.1010	0.4444
	Initial ranking	0.1816	0.0970	**0.4535**
TD2004	Median	0.1474**	0.1613	0.4607
	Max	0.1479**	0.1613	0.4650*
	Cross	0.1366	0.1587	**0.4671***
	LambdaMART	**0.4193**	**0.1827**	0.4267
	ListNet	0.2345	0.0913	0.1600
	CombMNZ	0.1309†	0.1600	0.4010
	Initial ranking	0.1513	0.1620	0.4646
OHSUMED	Median	0.2523††	0.4349	0.6719†
	Max	0.2525††	0.4363	0.6740†*
	Cross	0.2574†*	0.4363	0.6738†
	RankBoost	**0.5363**	0.4429	0.6226
	MART	0.4978	0.4165	0.5566
	RRF	0.2480††	0.4325	0.6370††
	Initial ranking	0.2675	**0.4453**	**0.7342**

Tableau 4 – Comparaison des trois variantes TournaRank (Median, Max et Cross) avec des algorithmes d'apprentissage d'ordonnancement et des méthodes de fusion. Pour chaque collection, la 4ème (resp. 5ème) ligne présente la meilleure (resp. la moins bonne) approche d'apprentissage d'ordonnancement, tandis que la 6ème ligne présente la meilleure méthode de fusion sur le jeu de données. † et $\star$ traduisent la significativité ($p < 0.05$) relativement au classement initial et à la meilleur méthode de fusion respectivement. Des symboles doublés indique une significativité au seuil $p < 0.01$. Les tests statistiques non pas été appliqués sur les approches d'apprentissage d'ordonnancement. CA et RF signifient "Coordinate Ascent" et "Random Forests".