À la recherche des paramètres des modèles de RI

Parantapa Goswami, Massih-Reza Amini, Eric Gaussier
LIG - Université de Grenoble/CNRS
firstname.lastname@imag.fr

RÉSUMÉ. *Nous abordons ici le problème de l'estimation des paramètres des modèles standard de la recherche d'information sur de nouvelles collections pour lesquelles aucun jugement de pertinence n'est disponible. Pour cela, nous nous reposons sur des collections passées pour lesquelles des jugements de pertinence sont disponibles et introduisons une nouvelle représentation des requêtes indépendante de la collection considérée. À partir de cette représentation et des collections passées, nous apprenons une fonction de régression capable de fournir, pour une nouvelle requête, une valeur à chaque paramètre des modèles standard de la recherche d'information. Les expériences menées sur des collections standard montrent le bien fondé de l'approche suivie, qui fournit des résultats significativement meilleurs que ceux obtenus en utilisant les valeurs par défaut des paramètres ou en utilisant des jugements de pertinence sur la nouvelle collection.*

ABSTRACT. *We address in this study the problem of estimating standard IR model parameters on new collections for which no relevance judgements are available. To do so, we rely on past collections with relevance judgements and introduce a new, collection independent query representation. From this representation as well as from past collections, we learn a regression function linking queries to parameter values. This regression function is then applied to each new query so as to determine the value of the parameter of the IR model considered. The experiments, conducted on standard IR collections, reveal that this approach significantly outperforms the approach based on default parameter values as well as the one making use of relevance judgements on the new collection to find global values for the IR model parameters.*

MOTS-CLÉS : *Modèles de RI, estimation des paramètres.*

KEYWORDS: *IR models, parameter estimation.*

1. Introduction

Construire des jugements de pertinence lorsque l'on veut déployer des modèles de recherche d'information (RI) sur de nouvelles collections (correspondant à de nouveaux domaines ou de nouvelles langues) est une opération coûteuse ; pour éviter cette étape coûteuse, on utilise en pratique très souvent les valeurs par défaut des paramètres des modèles de RI (c'est typiquement ce qui se passe dans les campagnes TREC et CLEF dès lors qu'une nouvelle collection est introduite).

Nous nous intéressons dans cette étude à l'estimation des paramètres des modèles de RI sur de nouvelles collections pour lesquelles aucun judgment de pertinence n'est disponible. Nous utiliserons les judgments de pertinence disponibles sur certaines collections afin de réaliser cette estimation, et nous concentrerons sur les modèles `BM25` (Robertson *et al.,* 2009), les modèles de langue (`LM`) (Ponte *et al.,* 1998) et les modèles *DFR* d'information (`LGD`) (Clinchant *et al.,* 2010). Ce choix est motivé par le fait que ces modèles sont parmi les modèles les plus utilisés et les plus performants à l'heure actuelle. Ils constituent de plus des attributs des systèmes d'apprentissage d'ordonnancement, qui peuvent être déployés dès qu'un certain nombre de retours utilisateurs (données de *clicks* par exemple) a été collecté. La question qui nous intéresse est donc : *comment inférer les paramètres des modèles de RI pour de nouvelles collections sans jugement de pertinence à partir des jugements de pertinence connus sur d'autres collections ?*

La méthode que nous proposons dans cette étude pour répondre à cette question permet :

– d'obtenir des paramètres des modèles propres à chaque requête ; les modèles de RI sont donc adaptés à chaque requête ;

– d'obtenir des représentations indépendantes des langues ; les modèles peuvent donc être utilisés sur de nouvelles collections rédigées dans de nouvelles langues.

Dans ce qui suit, nous présentons tout d'abord les travaux reliés dans la section 2. Notre approche est décrite dans la section 3, puis illustrée sur différentes collections dans la section 4. Enfin, la section 5 conclut cet article.

2. Etat de l'art

Les modèles standard de la RI reposent sur un nombre limité de paramètres, comme les paramètres b et k de `BM25` (Robertson *et al.,* 1994), le paramètre μ de lissage des modèles de langue (Jelinek *et al.,* 1980) ou le paramètre c du modèle `LGD` (Clinchant *et al.,* 2010). Ces paramètres dépendent de la collection de documents et des requêtes considérées. L'étude présentée dans (Zhai *et al.,* 2001) montre l'intérêt qu'il peut y avoir à adapter ces paramètres aux collections et requêtes considérées. Dans le cas où l'on dispose de requêtes avec jugements de pertinence, les valeurs de ces paramètres sont en général obtenues par une recherche en ligne, la valeur maximisant la mesure retenue (la `MAP` par exemple) sur l'ensemble des requêtes avec

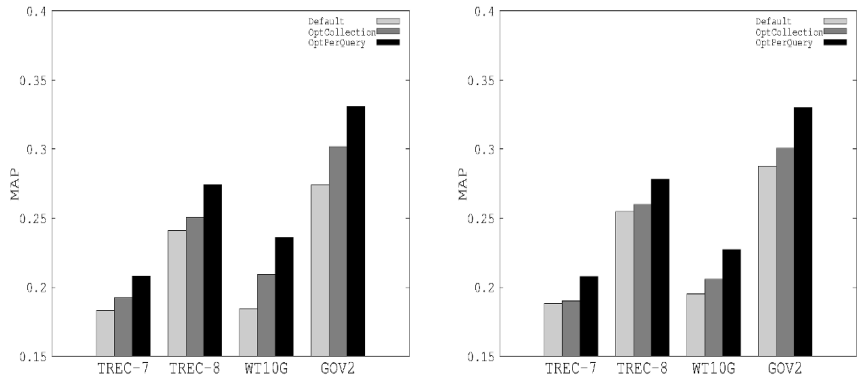
jugements de pertinence, étant sélectionnée pour la collection considérée. Ce genre d'approches fournit en général de bons résultats mais nécessite des jugements de pertinence.

Créer des jugements de pertinence, même en nombre limité, est une tâche pénible et coûteuse (Carterette, 2007). Ceci a conduit au développement de méthodes visant à estimer les paramètres de certains modèles de RI en mode non supervisé, comme par exemple (Zhai *et al.,* 2001, Tao *et al.,* 2006) pour le modèle de langue ou (Goswami *et al.,* 2013b) pour les modèles d'information. Toutefois, ces approches ne sont pas génériques dans la mesure où les méthodes proposées ne s'appliquent qu'à une classe restreinte de modèles. D'autres approches ont plutôt tenté d'utiliser des annotations sur des collections passées, appelées collections *source*, pour lesquelles des jugements de pertinence sont disponibles, afin d'estimer les paramètres des modèles de RI sur de nouvelles collections, appelées collections *cible*, pour lesquelles aucun jugement de pertinence n'est disponible. Ces approches rentrent dans le cadre de l'apprentissage par transfert, principalement développé pour la classification et la régression. Le transfert est tantôt réalisé à partir d'exemples (*instance-based transductive transfer learning*, (Sugiyama *et al.,* 2008)) ou à partir d'attributs (*feature-based transductive transfer learning*, (Blitzer *et al.,* 2008)).

Plus récemment, plusieurs études ont porté sur l'utilisation de l'apprentissage par transfert pour l'apprentissage de fonctions d'ordonnancement en RI (Chen *et al.,* 2008, Chen *et al.,* 2010, Gao *et al.,* 2010). Alors que la première de ces études fait l'hypothèse qu'un certain nombre de jugements de pertinence est disponible sur la la collection cible, les deux dernières se dispensent de cette hypothèse. Le transfert est réalisé, dans ces dernières études, en "tirant" les modèles appris sur la collection source vers la collection cible. Ceci se fait par une pondération des requêtes et/ou des documents de la collection source en fonction de leur distance aux requêtes et/ou documents de la collection cible. Les éléments source les plus proches des éléments cible auront plus de poids dans l'apprentissage et "tireront" donc le modèle appris vers la collection cible. Cette approche n'est toutefois valide que si les collections source et cible considérées sont proches l'une de l'autre. L'étude présentée dans (Goswami *et al.,* 2013a) revient en partie sur cette limitation en sélectionnant la meilleure collection source à prendre en compte, et en apprenant directement une fonction d'ordonnancement sur la collection cible.

Notre but dans cette étude est différent car nous nous intéressons aux modèles de RI standard et non à des fonctions d'ordonnancement générales (dont les modèles de RI standard sont des ingrédients). Les approches mentionnées ci-dessus ne peuvent être directement utilisées pour estimer les valeurs des paramètres de ces modèles sur des collections sans jugement de pertinence.

Enfin, notre approche comporte des similarités avec celle développée dans (Lv *et al.,* 2009) et qui vise à apprendre, requête par requête, le paramètre de modèles de rétro-pertinence (*feedback*). Le cadre retenu diffère toutefois du nôtre, qui est celui de la recherche d'information *ad hoc*. Cette différence de cadre a des implications sur les représentations proposées.



(a) BM25 (b) LGD

Figure 1. *Performance des modèles de RI avec les valeurs par défaut des paramètres (■), les valeurs optimisées sur toutes les requêtes (■) et les valeurs optimisées par requête (■). Les résultats sont en terme de MAP sur les collections TREC-7,8, WT10G et GOV2 pour (a) BM25 et (b) LGD*

3. Estimation des paramètres des modèles de RI sur des collections sans jugement de pertinence

La situation qui nous intéresse consiste en une collection source $\mathcal{S}$, composée d'un ensemble de documents $\mathcal{D}^s$, d'un ensemble de n requêtes $\mathcal{Q}^s = \{q_1^s, \ldots, q_n^s\}$ et d'un ensemble de judgments de pertinence pour chaque requête $\mathcal{Q}^s$. Nous considérons de plus une collection cible $\mathcal{T}$, composée d'un ensemble de documents $\mathcal{D}^t$ et d'un ensemble de u requêtes $\mathcal{Q}^t = \{q_1^t, \ldots, q_u^t\}$. Chaque requête q, quelle que soit la collection, est constituée d'un ensemble de termes $q = \{w_1^q, \ldots, w_{|q|}^q\}$.

Notre but est d'estimer les paramètres des modèles de RI sur la collection $\mathcal{T}$ en se servant des éléments de la collection $\mathcal{S}$. La première question que l'on peut se poser est de savoir si cela est vraiment intéressant. En d'autres termes, peut-on espérer avoir des résultats significativement meilleurs que ceux obtenus avec les valeurs par défaut des paramètres ? Pour répondre à cette question, nous avons calculé la MAP (Mean Average Precision) de trois modèles standard de RI (BM25, le modèle de langue avec lissage de Dirichlet LM et le modèle log-logistique LGD) sur quatre collections (TREC-7, TREC-8, WT10G and GOV2)[1], suivant trois méthodes : (a) en utilisant les paramètres par défaut des modèles, (b) en optimisant les paramètres des modèles à l'aide des jugements de pertinence, globalement sur l'ensemble des requêtes, et (c) en optimisant les paramètres des modèles à l'aide des jugements de pertinence requête par requête. Ce dernier cadre fournit la borne supérieure des performances des modèles de RI considérés ici.

1. Les collections sont décrites dans la section 4.

Les résultats obtenus sont présentés dans la figure 1. Comme on peut le remarquer, la différence entre les résultats obtenus à partir des valeurs par défaut et ceux obtenus à partir des valeurs optimisées varie suivant les modèles et les collections. Cette différence est relativement petite lorsque les paramètres sont optimisés globalement, et plus importante lorsque les paramètres sont optimisés requête par requête (elle est de l'ordre de 2-3% sur les collections TREC-7,8, et de 5-6% sur les collections GOV2 et WT10G, pour tous les modèles). Ceci indique tout d'abord que les valeurs par défaut peuvent produire, sur certaines collections, de très bons résultats ; il n'est pas garanti, sur ces collections, d'améliorer significativement les résultats. Ceci indique aussi que, si l'on veut vraiment dépasser les résultats obtenus avec les valeurs par défaut, il est nécessaire de se reposer sur des méthodes qui estiment les paramètres requête par requête.

Au vu de ces éléments, nous nous reposons sur l'approche suivante : nous calculons tout d'abord sur la collection source, pour tous les modèles et suivant la méthode (c) ci-dessus, un vecteur de paramètres optimisés pour chaque requête source $\mathcal{Q}^s$, dénoté $\mathbf{c}^s_{opt} = (c_{q^s_1}, \ldots, c_{q^s_n})^\top$, où $^\top$ représente la transposée ; nous apprenons ensuite une fonction de regression entre les requêtes source et $\mathbf{c}^s_{opt}$, et utilisons cette fonction pour prédire les valeurs des paramètres pour les requêtes cible. Bien sûr, cela nécessite de se reposer sur une représentation commune pour les requêtes source et cible.

Une représentation commune pour requêtes source et cible doit garantir que des requêtes qui mettent en jeu des types de mots similaires soient proches l'une de l'autre. Par "type", nous entendons ici les caractéristiques générales des mots (mots fréquents dans certains documents et pas d'autres, avec une fréquence documentaire plus ou moins élevée, ...). Pour construire une telle représentation commune, nous représentons tout d'abord chaque mot comme un vecteur à 4 dimensions :

$$\mathbf{w}^q = (idf(w^q), \mu(w^q), \sigma(w^q), sk(w^q)) \in \mathbb{R}^4 \quad [1]$$

où $idf(w)$, $\mu(q)$, $\sigma(w)$ et $sk(w)$ représentent respectivement l'inverse de la fréquence documentaire de w, et les estimés empiriques du premier, deuxième et troisième moment de la distribution des fréquences de w^q sur les documents de la collection (les deux premiers moments correspondent à la moyenne et à la variance). Une requête est alors représentée par l'ensemble des vecteurs des mots qu'elle contient :

$$q = \{\mathbf{w}^q_1, \ldots, \mathbf{w}^q_{|q|}\} \quad [2]$$

Par exemple, à partir de la requête (extraite de la collection WT10G décrite dans la section 4) *q : lava lamps*, on obtient les vecteurs représentatifs pour *lava* et *lamps* :

lava : $(6.7762, 0.0012, 0.0050, 137.0581)^\top$

lamps : $(5.6177, 0.0059, 0.0355, 81.8577)^\top$

où $^\top$ dénote la transposée. La représentation de la requête est alors :

$$\begin{aligned} q \quad = \quad & \{(6.7762, 0.0012, 0.0050, 137.0581)^\top, \\ & (5.6177, 0.0059, 0.0355, 81.8577)^\top\} \end{aligned}$$

La représentation ci-dessus présente l'avantage d'être indépendante de la collection. De plus, la prise en compte des trois premiers moments et de l'IDF fournit un résumé de la distribution des mots dans les collections de documents. Ainsi, deux requêtes sont proches dans cette représentation si elles contiennent des mots qui se comportent de façon similaire au vu de leurs distributions sur les documents de leurs collections. Dans la mesure où les paramètres des modèles de RI sont fortement dépendants de ces distributions, les paramètres des modèles pour deux requêtes différentes seront d'autant plus proches que ces requêtes mettent en jeu des mots qui ont des distributions similaires, ce que permet de capturer la représentation ci-dessus.

Toutefois, cette représentation, fondée sur un ensemble de vecteurs, ne permet pas d'utiliser directement l'ensemble des fonctions de régression existantes (qui requièrent pour la plupart une représentation vectorielle). Nous proposons ici deux représentations vectorielles dérivées de la représentation ci-dessus :

1) Le vecteur final représentant une requête (source ou cible) est obtenu en prenant la moyenne des vecteurs des mots qu'elle contient :

$$\mathbf{q} = \sum_{i=1}^{|q|} \frac{\mathbf{w}_i^q}{|q|} \qquad [3]$$

Chaque requête est donc finalement représentée par un vecteur d'un espace vectoriel à 4 dimensions indépendant du vocabulaire des collections considérées.

2) À partir du vecteur précédent, nous calculons, pour chaque requête q, un nouveau vecteur, noté $\phi(\mathbf{q})$, fondé sur la similarité (produit scalaire) entre $\mathbf{q}$ et chaque requête source :

$$\phi(\mathbf{q}) = (\langle \mathbf{q}, \mathbf{q}_1^s \rangle, \cdots, \langle \mathbf{q}, \mathbf{q}_n^s \rangle)^\top \qquad [4]$$

Nous obtenons dans ce cas un vecteur à n dimensions dont les coordonnées sont d'autant plus importantes que la requête est proche de la requête source considérée.

On peut alors apprendre, pour chaque paramètre des modèles de RI, à partir de la collection source, une fonction de régression reliant $\mathbf{q}$ ou $\phi(\mathbf{q})$ à une valeur du paramètre. L'ensemble du processus est résumé dans les étapes suivantes.

Pour chaque paramètre du modèle de RI considéré :

1) Calculer $\mathbf{c}_{opt}^s = (c_{q_1^s}, \ldots, c_{q_n^s})^\top$ et apprendre une fonction de régression f reliant $(\mathbf{q}_1^s, \cdots, \mathbf{q}_n^s)$ ou $(\phi(\mathbf{q}_1^s), \cdots, \phi(\mathbf{q}_n^s))$ à $(c_{q_1^s}, \ldots, c_{q_n^s})$;

2) Pour chaque requête q^t, calculer $c_{q^t} = f(\mathbf{q}^t)$ ou $c_{q^t} = f(\phi(\mathbf{q}^t))$

L'apprentissage de la fonction de régression f est réalisé dans cette étude à l'aide des SVR (Support Vector Regression) (Vapnik, 2000).

Collection	$\mathcal{N}$	l_{avg}	Index size	#queries
GOV2	25,177,217	646	19.6 GB	100
WT10G	1,692,096	398	1.3 GB	100
TREC-3	741,856	261	427.7 MB	50
TREC-4	567,529	323	379.0 MB	50
TREC-5	524,929	339	378.0 MB	50
TREC-6,7,8	528,155	296	373.0 MB	50
CLEF-3	169,477	301	126.2 MB	60

Tableau 1. *Caractéristiques des différentes collections utilisées dans nos expériences, triées par taille*

4. Expériences

Nous présentons dans cette section les expériences réalisées afin d'évaluer les résultats par l'approche décrite ci-dessus.

4.1. *Collections*

Toutes nos expériences sont réalisées à partir de la plate-forme Terrier v3.5 (terrier.org) et portent sur sept collections, dont une est tirée de la campagne CLEF[2] et six de la campagne TREC[3]. Le tableau 1 résume les caractéristiques de ces collections. WT10G est constituée des collections de la piste Terabyte de TREC-9 et TREC-10 ; GOV2 est constituée des collections de la piste Terabyte de TREC-2004 and TREC-2005. Sur toutes collections, les mots vides ont été supprimés et les mots pleins racinisés. Ces deux opérations ont été réalisées à l'aide des outils Terrier (liste de mots vides et raciniseur de Porter).

Sauf indication contraire, nous utilisons CLEF-3, TREC-3,4,5,6 comme collection source, et TREC-7,8 ainsi que WT10G et GOV2 comme collections cibles. Les résultats sont évalués en terme de MAP (*Mean Average Precision*) et de précision à 10 documents P@10. Les jugements de pertinence sur les collections cibles ne sont considérés que pour évaluer les résultats finals. Nous utilisons de plus un test de Wilcoxon, avec une p-valeur fixée à $p = 0.05$, pour juger du caractère significatif ou non de la différence entre deux résultats.

Comme mentionné auparavant, nous avons choisi trois modèles de RI parmi les plus utilisés : BM25, le modèle de langue avec lissage de Dirichlet (LM) et le modèle log-logistique LGD. Nous avons comparé les résultats obtenus avec les valeurs des paramètres de ces modèles appris suivant l'approche décrite dans la section précédente, avec les valeurs optimisées et les valeurs optimales de ces paramètres, et enfin avec

2. www.clef-campaign.org
3. trec.nist.gov

b (BM25)	$0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0$
μ (LM)	$10, 25, 50, 75, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1500, 2000, 2500,$ $3000, 4000, 5000, 10000$
c (LGD)	$0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20.0$

Tableau 2. *Ensemble des valeurs des paramètres considérées pour chaque modèle*

les valeurs par défaut de ces paramètres. Ces valeurs par défaut sont : $b = 0.75$ pour BM25, $\mu = 2500.0$ pour LM et $c = 1.0$ pour LGD. Pour BM25 nous avons considéré, suivant en cela une pratique courante de ce modèle, que les deux autres paramètres k_1 et k_2 sont fixes ($k_1 = 1.2$ et $k_2 = 8.0$) ; seul le paramètre b est appris. Les vecteurs $\mathbf{c}_{opt}^s$ des valeurs optimales des paramètres sont obtenus, pour chaque modèle de RI et requête de la collection source, en sélectionnant, par une recherche en ligne, la valeur du paramètre du modèle qui fournit la meilleure MAP pour cette requête. Les valeurs considérées dans cette recherche en ligne sont standard et sont données dans le tableau 2.

Pour l'apprentissage de la fonction de régression, nous avons utilisé l'implantation ϵ-SVR disponible dans LIBSVM[4] (ϵ a été fixé à 0.1 et l'hyper-paramètre C a été obtenu par validation croisée sur l'ensemble d'apprentissage).

4.2. *Résultats expérimentaux*

Nous commençons notre étude expérimentale par une comparaison des deux représentations choisies : $\mathbf{q}$ et $\phi(\mathbf{q})$. Les résultats obtenus avec ces deux représentations sont donnés dans le tableau 3. Comme nous pouvons le constater, la représentation simple, $\mathbf{q}$ fournit des résultats qui sont quasi-systématiquement meilleurs que ceux fournis par la représentation plus complexe $\phi(\mathbf{q})$, et ce pour tous les modèles et toutes les collections. Si cette dernière représentation utilise un espace vectoriel plus important, les caractéristiques des requêtes (par rapport aux distributions des mots les constituant) sont en partie noyées dans la similarité avec les requêtes source, ce qui explique à notre avis que la représentation plus simple soit finalement plus efficace. Nous nous reposons dans la suite sur cette représentation, que nous noterons simplement SVR.

Nous comparons maintenant les résultats obtenus avec les paramètres appris avec notre approche et ceux obtenus avec la valeur par défaut de ces paramètres.

4.2.1. *Valeurs apprises vs valeurs par défaut*

Les tableaux 4 et 5 présentent les résultats obtenus avec les valeurs par défaut des paramètres, notés def, et les valeurs apprises, notées SVR, pour la MAP (tableau 4) et pour la précision à 10 documents (P@10, tableau 5). Comme on peut le constater sur le

4. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

		TREC-7		TREC-8		GOV2	
		MAP	P@10	MAP	P@10	MAP	P@10
BM25	SVR-$\mathbf{q}$	**0.1903**	0.4300	**0.2495**	0.4720	**0.3031***	**0.5879***
	SVR-$\phi(\mathbf{q})$	0.1895	0.4280	0.2473	0.4700	0.2921	0.5616
LM	SVR-$\mathbf{q}$	**0.1914**	**0.4120***	**0.2473**	**0.4480***	**0.2944***	**0.5576***
	SVR-$\phi(\mathbf{q})$	0.1884	0.3940	0.2447	0.4380	0.2868	0.5494
LGD	SVR-$\mathbf{q}$	0.1900	0.4420	0.2564	**0.4660***	0.2962	0.5646
	SVR-$\phi(\mathbf{q})$	0.1887	0.4460	0.2551	0.4540	0.2990	0.5727

Tableau 3. *Comparaison des deux représentations vectorielles proposées pour chaque requête en termes de MAP et P@10, pour les différents modèles de RI sur les collections TREC 7-8 et GOV2. Un résultat en gras est significativement meilleur que l'autre pour une p-valeur fixée à 0.05 ; un résultat en gras avec un astérisque est significativement meilleur pour une p-valeur fixée à 0.025 level*

tableau 4, la méthode que nous avons décrite précédemment fournit des résultats, en termes de MAP, significativement meilleurs que ceux obtenus avec les valeurs par défaut, et ce pour quasiment tous les modèles et collections (la seule exception concerne le modèle LGD pour les collections TREC-8 et WT10G pour lesquelles, s'il y a bien amélioration, celle-ci n'est pas significative). La même conclusion s'impose pour la P@10 (les seules exceptions sont cette fois toujours le modèle LGD pour les collections TREC-8 et WT10G, et le modèle BM25 pour la collection TREC-8). Il faut toutefois noter que nous avons appris la fonction de régression sur la MAP et non sur la P@10. Cette fonction est donc optimisée pour la MAP. Il est bien sûr possible d'optimiser la fonction de régression sur la P@10 dans le cadre que nous avons défini. Nous avons choisi ici la MAP car c'est la mesure la plus utilisée en pratique. Les résultats que nous présentons pour la P@10 servent ici à illustrer le bon comportement de la méthode sur une autre mesure, même si celle-ci n'est pas la mesure ciblée.

Enfin, et comme conjecturé dans la section 3, la différence entre les résultats obtenus avec les valeurs par défaut et ceux obtenus avec l'approche proposée varie d'une collection et d'un modèle à l'autre. Comme mentionné ci-dessus, cette différence n'est pas significative pour LGD sur TREC-8, cas qui correspond à la plus petite différence entre valeurs par défaut et bornes supérieures (voir la figure 1). Il y a dans ce cas peu de marge d'amélioration par rapport aux valeurs par défaut.

4.2.2. *Valeurs apprises vs valeurs optimisées et valeurs optimales*

Nous comparons maintenant les résultats obtenus avec notre approche et ceux correspondant à deux situations "idéales". La première situation "idéale" correspond au cas où des jugements de pertinence sont connus pour un certain nombre de requêtes cible (50% des requêtes cible dans nos expériences). On peut alors utiliser ces requêtes pour sélectionner, par une recherche en ligne sur la base des valeurs des paramètres données dans le tableau 2, la meilleure valeur, c'est-à-dire celle qui fournit la meilleure MAP, pour l'ensemble des requêtes cible avec jugement de pertinence. Nous avons réa-

		TREC-7	TREC-8	WT10G	GOV2
BM25	def	0.1828	0.2409	0.1842	0.2739
	SVR	**0.1903**†	**0.2495**†	**0.2052**†	**0.3031**†
LM	def	0.1863	0.2401	0.2040	0.2798
	SVR	**0.1914**†	**0.2473**†	**0.2102**†	**0.2944**†
LGD	def	0.1882	0.2547	0.1949	0.2876
	SVR	**0.1900**†	**0.2564**	**0.2002**	**0.2962**†

Tableau 4. *Comparaison, sur la MAP, entre les modèles de RI utilisant les valeurs par défaut des paramètres (def) et les valeurs apprises par régression (SVR). Les meilleurs résultats, pour chaque modèle sur chaque collection, sont en gras ; les résultats en gras avec † sont significativement meilleurs que les autres (test de Wilcoxon avec une p-valeur de 0.05)*

		TREC-7	TREC-8	WT10G	GOV2
BM25	def	0.4180	**0.4740**	0.2910	0.5384
	SVR	**0.4300**†	0.4720	**0.3090**†	**0.5879**†
LM	def	0.3920	0.4320	0.2930	0.5545
	SVR	**0.4120**†	**0.4480**†	**0.3010**	**0.5576**†
LGD	def	0.4280	**0.4740**	0.2870	0.5414
	SVR	**0.4420**†	0.4660	**0.2910**	**0.5646**†

Tableau 5. *Comparaison, sur la P@10, entre les modèles de RI utilisant les valeurs par défaut des paramètres (def) et les valeurs apprises par régression (SVR). Les meilleurs résultats, pour chaque modèle sur chaque collection, sont en gras ; les résultats en gras avec † sont significativement meilleurs que les autres (test de Wilcoxon avec une p-valeur de 0.05)*

lisé 10 découpages aléatoires (50-50) des requêtes cible pour chaque collection cible, avons utilisé 50% de ces requêtes pour la recherche en ligne et 50% pour le test. Cette méthode sera dénotée 10split dans la suite. Le tableau 6 fournit les résultats obtenus en comparant cette méthode à celle proposée auparavant qui n'utilise aucun jugement de pertinence sur la collection cible. Comme on peut le constater, la variance est toujours très faible, ce qui indique une certaine stabilité des résultats par rapport aux requêtes choisies pour la recherche en ligne (cette stabilité s'explique aussi par la stabilité des modèles de RI par rapport aux valeurs des paramètres qu'ils utilisent). Un autre fait important à noter est que notre approche, sans jugement de pertinence, est soit meilleure soit similaire à l'approche idéale avec jugements de pertinence (25 requêtes avec jugements de pertinence sont utilisées pour les collections TREC-7,8 et 50 pour les collections WT10G et GOV2). Ceci valide l'approche que nous proposons, capable de fournir de meilleurs résultats sans jugement de pertinence.

		TREC-7	TREC-8	WT10G	GOV2
BM25	10split$_{mean}$	0.1820	0.2453	0.1961	**0.3054**
	10split$_{var}$	0.00029	0.00121	0.00046	0.00022
	SVR	**0.1903**†	**0.2495**	**0.2052**†	0.3031
LM	10split$_{mean}$	0.1821	**0.2545**	0.1967	**0.3019**
	10split$_{var}$	0.00035	0.00048	0.00052	0.00019
	SVR	**0.1914**†	0.2473	**0.2102**†	0.2944
LGD	10split$_{mean}$	0.1803	**0.2608**	0.1924	**0.3026**
	10split$_{var}$	0.00039	0.00048	0.00053	0.00023
	SVR	**0.1900**†	0.2564	**0.2002**	0.2962

Tableau 6. *Comparaison des modèles de RI avec les valeurs optimisées des paramètres estimées sur 10 découpages aléatoires (10split ; 50% des jugements de pertinence cible sont utilisés pour optimiser les valeurs des paramètres) et les valeurs apprises (SVR ; aucun jugement de pertinence cible n'est utilisé) en terme de MAP. Les meilleurs résultats, pour chaque modèle sur chaque collection, sont en gras ; les résultats en gras avec † sont significativement meilleurs que les autres (test de Wilcoxon avec une p-valeur de 0.05)*

		TREC-7	TREC-8	WT10G	GOV2
BM25	10split$_{mean}$	0.4192	0.4552	**0.3094**	**0.6018**†
	10split$_{var}$	0.00053	0.00149	0.00048	0.00041
	SVR	**0.4300**†	**0.4720**†	0.3090	0.5879
LM	10split$_{mean}$	**0.4224**	**0.4536**	**0.3076**	**0.5679**
	10split$_{var}$	0.00115	0.00059	0.00062	0.00096
	SVR	0.4120	0.4480	0.3010	0.5576
LGD	10split$_{mean}$	0.4304	0.4584	0.2908	**0.5901**†
	10split$_{var}$	0.00167	0.00079	0.00037	0.00088
	SVR	**0.4420**†	**0.4660**	**0.2910**	0.5646

Tableau 7. *Comparaison des modèles de RI avec les valeurs optimisées des paramètres estimées sur 10 découpages aléatoires (10split ; 50% des jugements de pertinence cible sont utilisés pour optimiser les valeurs des paramètres) et les valeurs apprises (SVR ; aucun jugement de pertinence cible n'est utilisé) en terme de P@10. Les meilleurs résultats, pour chaque modèle sur chaque collection, sont en gras ; les résultats en gras avec † sont significativement meilleurs que les autres (test de Wilcoxon avec une p-valeur de 0.05)*

La seconde situation "idéale" que nous considérons est celle où tous les jugements de pertinence des requêtes cible sont utilisés pour obtenir, requête par requête, la meilleure valeur possible des paramètres. Encore une fois, la recherche de cette meilleure valeur se fait par une recherche en ligne, sur la base des valeurs données dans le tableau 2. La différence avec la situation précédente réside dans le fait que tous les jugements de pertinence cible sont utilisés, et que la recherche en ligne est effectuée requête par requête (alors que les paramètres étaient optimisés sur un ensemble de requêtes dans le cas précédent). Si notre première situation idéale correspond à une réalité pratique (celle dans laquelle l'on dispose d'un certain nombre d'annotations sur la collection cible), la seconde situation idéale que nous considérons ici, et que nous avons déjà considéré en section 3[5], n'a aucune réalité pratique et n'est là que pour fournir une borne supérieure à l'approche que nous avons développée.

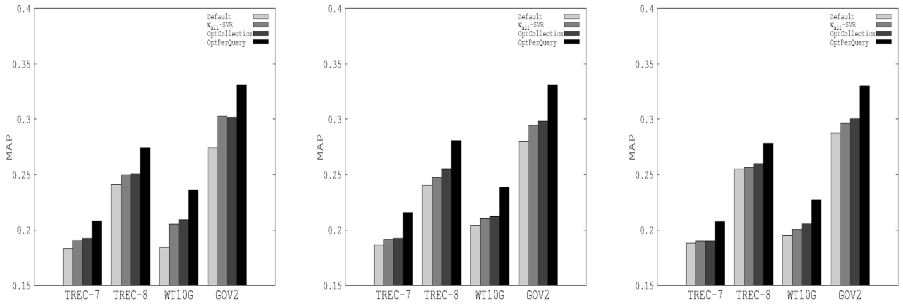
La comparaison de notre approche et de cette solution idéale est fournie dans la figure 2 (cette figure contient aussi, à des fins de complétude, les résultats par défaut et ceux correspondant à une optimisation globale, utilisant tous les jugements de pertinence sur les requêtes cible). Comme on peut le constater, les résultats obtenus par notre approche sont très proches de ceux obtenus en considérant tous les jugements de pertinence cible et en opérant une optimisation globale, sur l'ensemble des requêtes, des valeurs des paramètres (deuxième et troisième barres de chaque série d'histogrammes). L'utilisation d'un test de Wilcoxon n'a en fait montré aucune différence significative entre notre approche et cette approche optimale, et ce quel que soit le modèle, quelle que soit la collection. Ceci indique que notre approche est au moins aussi bonne qu'une approche optimisant les paramètres des modèles de façon globale (les stratégies par défaut, `def`, et d'optimisation standard sur la base d'un ensemble de jugements de pertinence, `10split`, sont de telles approches, qui, comme on l'a vu, fournissent de moins bons résultats que notre approche).

Le second point que nous pouvons remarquer est que notre approche est encore en dessous (entre 2 et 4%) de la borne supérieure qu'il est possible d'atteindre sur ces collections. Il y a donc encore de la marge pour améliorer la solution que nous proposons ici.

4.2.3. *Impact sur le temps de traitement*

Dans la mesure où notre approche nécessite une étape supplémentaire pour chaque requête, à savoir le calcul du paramètre du modèle de RI considéré, il est important de connaître l'impact sur le temps de traitement d'une requête de cette étape additionnelle. Les vecteurs de mots définis par l'équation 1 peuvent être calculés au moment de l'indexation de la collection et stockés de façon similaire à l'IDF. La représentation de la requête donnée dans l'équation 3 peut alors être obtenue sans coût additionnel, en utilisant les fichiers d'index standard. L'étape qui requiert un temps supplémentaire est bien celle qui correspond à l'application de la fonction de régression sur la requête cible. Toutefois, cette application requiert un nombre limité d'opérations puisque l'es-

5. Voir la figure 1 par exemple.



(a) BM25 (b) LM (b) LGD

Figure 2. *Comparaison des modèles de RI avec les valeurs optimales par requête des paramètres (■), les valeurs optimales sur l'ensemble des requêtes (■), les valeurs apprises par régression SVR (■) et les valeurs par défaut (■). Les résultats sont en terme de MAP sur les collections TREC-7,8, WT10G et GOV2 pour (a) BM25 (b) LM et (c) LGD*

pace vectoriel considéré est de dimension 4. On peut donc conjecturer que l'impact sur le temps de traitement d'une requête sera limité.

Le tableau 8 fournit la différence (en millisecondes) entre le temps de traitement moyen par requête pour notre approche et pour l'approche utilisant les valeurs par défaut des paramètres. Comme on peut le constater, cette différence est de l'ordre de $15 - 30$ millisecondes, ce qui rend l'approche proposée ici utilisable en pratique.

	Temps de traitement additionnel (millisec.)			
	TREC-7	TREC-8	WT10G	GOV2
BM25	31.22	31.03	16.81	16.99
LM	30.89	31.80	17.23	16.94
LGD	31.51	31.41	16.98	17.46

Tableau 8. *Temps supplémentaire (en millisecondes) par requête pour la méthode SVR par rapport à l'utilisation des valeurs par défaut*

5. Conclusion

Nous avons présenté dans cette étude une nouvelle méthode pour prédire, requête par requête, les paramètres des modèles standard de RI sur des collections pour lesquelles aucun jugement de pertinence n'est disponible. Pour cela, nous avons utilisé des collections source avec jugements de pertinence et introduit des représentations des requêtes indépendantes des collections et fondées sur les distributions des mots dans les collections considérées. Nous avons appris une fonction simple de régression à partir de ces représentations, reliant requêtes et valeurs des paramètres.

Les expériences menées sur des collections standard de la RI ont montré :

1) Que la méthode proposée est significativement meilleure, en termes de MAP, que celle utilisant les valeurs par défaut des paramètres ; elle est également au moins aussi bonne, et dans certains cas meilleure, que la stratégie consistant à se reposer sur un ensemble de jugements de pertinence cible et à apprendre des valeurs globales (c'est-à-dire pour l'ensemble des requêtes) des paramètres ;

2) Qu'il n'y a pas de différence significative, sur les collections considérées, entre la méthode proposée et celle "idéale" utilisant tous les jugements de pertinence et optimisant globalement les paramètres (cette méthode "idéale" n'a bien sûr pas de réalité pratique et fournit juste une borne supérieure aux méthodes globales sur l'ensemble des requêtes) ;

3) Que l'impact sur le temps de traitement pour une requête est de l'ordre de 15-30 millisecondes. Le temps additionnel requis par la méthode proposée correspond à l'application de la fonction de régression à chaque requête.

Dans la suite, nous comptons explorer de nouveaux espaces de représentation des requêtes, avec l'espoir de capturer au mieux les caractéristiques importantes de chaque requête, tout en restant indépendant des collections. Nous comptons aussi étudier l'utilisation de régressions multiples de façon à estimer simultanément plusieurs paramètres (ceci pourrait être utile pour le modèle BM25 mais aussi pour certains modèles d'apprentissage d'ordonnancement).

6. Bibliographie

Blitzer J., Crammer K., Kulesza A., Pereira F., Wortman J., « Learning Bounds for Domain Adaptation », *Advances in Neural Information Processing Systems (NIPS 20)*, 2008.

Carterette B., « Robust test collections for retrieval evaluation », *Proceedings of the* 30^{th} *annual international ACM SIGIR conference*, 2007.

Chen D., Xiong Y., Yan J., Xue G.-R., Wang G., Chen Z., « Knowledge transfer for cross domain learning to rank », *Information Retrieval*, 2010.

Chen D., Yan J., Wang G., Xiong Y., Fan W., Chen Z., « TransRank : A Novel Algorithm for Transfer of Rank Learning », *ICDM Workshops*, IEEE Computer Society, p. 106-115, 2008.

Clinchant S., Gaussier E., « Information-based models for ad hoc IR », *Proceedings of the* 33^{rd} *annual international ACM SIGIR conference*, 2010.

Gao W., Cai P., Wong K.-F., Zhou A., « Learning to rank only using training data from related domain », *Proceedings of the* 33^{rd} *annual international ACM SIGIR conference*, 2010.

Goswami P., Amini M. R., Gaussier E., « Transferring knowledge with source selection to learn IR functions on unlabeled collections », *Proceedings of the* 22^{nd} *ACM International Conference on Information and Knowledge Management (CIKM)*, 2013a.

Goswami P., Gaussier E., « Estimation of the Collection Parameter of Information Models for IR », *Proceedings of the* 35^{th} *European Conference on Advances in Information Retrieval (ECIR)*, 2013b.

Jelinek F., Mercer R. L., « Interpolated estimation of Markov source parameters from sparse data », *Proceedings of the Workshop on Pattern Recognition in Practice*, 1980.

Lv Y., Zhai C., « A comparative study of methods for estimating query language models with pseudo feedback », *Proceedings of the* 18^{th} *ACM Conference on Information and Knowledge Management (CIKM)*, 2009.

Ponte J. M., Croft W. B., « A Language Modeling Approach to Information Retrieval », *Proceedings of the* 21^{st} *annual international ACM SIGIR conference*, 1998.

Robertson S. E., Walker S., « Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval », *Proceedings of the* 17^{th} *annual international ACM SIGIR conference*, 1994.

Robertson S. E., Zaragoza H., « The Probabilistic Relevance Framework : BM25 and Beyond », *Foundations and Trends in Information Retrieval*, 2009.

Sugiyama M., Nakajima S., Kashima H., Buenau P. V., Kawanabe M., « Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation », *Advances in Neural Information Processing Systems (NIPS 20)*, 2008.

Tao T., Zhai C., « Regularized estimation of mixture models for robust pseudo-relevance feedback », *Proceedings of the* 29^{th} *annual international ACM SIGIR conference*, 2006.

Vapnik V. N., *The nature of statistical learning theory*, Springer, 2000.

Zhai C., Lafferty J. D., « A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval », *Proceedings of the* 24^{th} *annual international ACM SIGIR conference*, 2001.