Contrainte de correspondance Document-Document pour la RI. Application à la Divergence de Kullback-Leibler

Philippe Mulhem, Jean-Pierre Chevalet

Contrainte de correspondance Document-Document pour la RI. Application à la Divergence de Kullback-Leibler.

Philippe Mulhem, Jean-Pierre Chevalet

Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France
CNRS, LIG, F-38000 Grenoble, France
{Philippe.Mulhem, Jean-Pierre.Chevallet}@imag.fr

RÉSUMÉ. *Cet article décrit une contrainte d'un modèle de recherche d'information décrivant les comportement attendu d'un système si un document du corpus est posé en requête, la contrainte DDMC (Document-Document Matching Constraint). Cette contrainte n'étant pas vérifiée par un modèle classique de recherche d'information (modèle de langue basé sur un calcul de négative de Divergence de Kullback-Leibler avec lissage de Jelinek-Mercer), nous présentons une modification de ce dernier modèle qui permet de vérifier DDMC. Une dernière partie présente des expérimentations menées afin de vérifier que notre modification n'impacte pas la qualité des réponses d'un système, tout en garantissant la vérification de DDMC.*

ABSTRACT. *This paper defines a new axiomatic constraint, namely DDMC (Document-Document Matching Constraint), for information retrieval that depicts the behavior of a matching if a corpus document is used as a query. The DDMC constraint is not verified by a classical IR model like the Language Model based on Jelinek-Mercer smoothing and Kulback-Leibler Divergence. We introduce a modification of this model to validate DDMC. An experiment conducted on two corpus show that the modification of the reference model does not degrade significantly the results, and validates the DDMC.*

MOTS-CLÉS : *Fichier inverse, fonction de correspondance, contrainte de la fonction de correspondance.*

KEYWORDS: *Inverted file, matching function, matching constraint.*

1. Introduction

Depuis le début des travaux dans le domaine de la Recherche d'Information (RI), les comparaisons entre les modèles (ou les systèmes) ont été l'un des centres des préoccupations. L'avènement du paradigme de Cranfield pour les évaluations expérimentales des systèmes a permis au domaine de progresser à grands pas. Cependant, ces évaluations atteignent leur limite dans des contextes interactifs et personnalisés. Parallèlement, depuis les années 2000, est apparu une approche, initiée par Fang, Tao et Zai en 2004 (Fang *et al.*, 2004), définissant des heuristiques à valider pour les modèles de RI.

Ces travaux à base d'heuristiques ont été révolutionnaires, mais il est difficile de déterminer leur complétude. Nous nous préoccupons ici du cas de requêtes "par l'exemple" qui, selon nous, soulève un nouveau problème lié au fait que si un document D du corpus est utilisé comme requête, alors les contraintes actuelles ne garantissent pas que ce document D du corpus a la valeur de correspondance la plus élevée de tous les documents du corpus. Le travail décrit dans cet article définit donc une contrainte additionnelle aux heuristiques définies dans (Fang *et al.*, 2004), (Fang et Zhai, 2005), (Fang *et al.*, 2011), (Clinchant et Gaussier, 2011) et (Cummins et O'Riordan, 2007). Cette contrainte, appelée DDMC, pour *Document-Document Matching Constraint*, est décrite par :

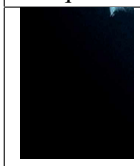
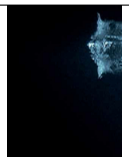
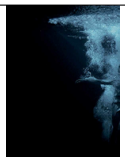
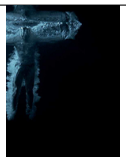
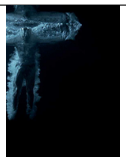
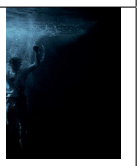
La valeur de correspondance entre un document D du corpus utilisé comme requête et ce même document dans le corpus est supérieure ou égale à celle entre D utilisé comme requête et chacun des autres documents du corpus.

Une telle contrainte n'est pas liée spécifiquement au tf, à l'idf ou à la taille des documents, mais est plus globale car elle définit une contrainte par rapport à tous les autres documents. En notant C le corpus, et en utilisant la notation $f(Q, D)$ pour dénoter la formule de correspondance entre une requête Q et un document D, DDMC est formalisée par : $\forall D \in C, \nexists D' \in C, D' \neq D, f(D, D') > f(D, D)$

Dans le cas de requêtes sur des documents textuels, la vérification de cette contrainte n'a pas lieu d'être car les requêtes sont courtes (Robertson, 2000) et que ce cas n'apparait donc pas dans la pratique. Dans le cas de requêtes par l'exemple, comme des recherches d'images par similarité visuelle, nous estimons que cette contrainte doit être vérifiée théoriquement par le modèle utilisé. Ici, nous nous intéressons à un modèle qui a fait ses preuves aussi bien pour la recherche de textes que celle de la recherche d'images : le modèle de langue basé sur un lissage de Jelinek-Mercer, utilisant une fonction de correspondance basée sur la négative de la Divergence de Kullback-Leibler (notée nKLD). Le tableau 1 présente un exemple réel avec ce modèle de langue sur une approche classique à base de sacs de *mots visuels* (Csurka *et al.*, 2004) sur une base d'images tirée d'un projet industriel. Ce tableau montre le résultat d'une requête (une image du corpus) et les cinq premiers résultats obtenus sur un corpus de quelques centaines d'images. On remarque que : i) l'image requête est quasiment unie, elle ne contient quasiment qu'un terme visuel du vocabulaire (noir uni), et ii) que les images qui répondent le mieux suivant le modèle de langue contiennent ce terme majoritaire-

ment, mais également d'autres éléments visuels. Ce résultat déroutant est obtenu car le modèle de l'image est lissé dans le corpus et qu'il n'est pas lissé quand l'image est utilisée en temps que requête.

Tableau 1. *Exemple de résultat pour une requête provenant du corpus.*

requête	résultat 1	résultat 2	résultat 3	résultat 4	résultat 5
					

Le travail décrit dans cet article a pour objectif de proposer un moyen d'éviter ce comportement aberrant pour un utilisateur. Notre proposition repose sur l'utilisation d'un modèle capable de réaliser la correspondance sur les requêtes lissées. Pour que ce modèle soit réellement utilisable dans le cadre de grands corpus, nous le rendons compatible avec les fichiers inverses (cf. (Mulhem et Chevallet, 2014)).

Le plan de cet article est le suivant : nous décrivons les travaux relatifs à notre approche en section 2. La section 3 décrit le comportement de trois exemples de modèles de référence en recherche d'information par rapport à la contrainte DDMC. Notre proposition de modèle de langue vérifiant DDMC est détaillée en section 3, avec une évaluation partielle de notre proposition pour les axiomes existants. La section 4 est dédiée aux résultats expérimentaux obtenus sur une implantation de nos propositions, et nous concluons en section 5.

2. Etat de l'art

L'évaluation des propositions en Recherche d'Information a toujours été l'un des points clés de ce domaine de recherche.

En l'absence de comparaisons théoriques, le paradigme de Cranfield a été celui utilisé majoritairement depuis les années 70. Avec ce paradigme, de type "boite noire", on analyse les résultats fournis par un système, pour chaque requête d'un ensemble, sur un corpus prédéfini. Cette analyse consiste à comparer les résultats obtenus à une "vérité" établie par des humains. La comparaison porte le plus souvent sur des estimations numériques de la capacité des systèmes à renvoyer tous les documents pertinents (rappel) et uniquement des documents pertinents (précision). Ce paradigme à été fondateur pour la Recherche d'Information, mais en plus de ses limitations couramment admises (pertinence utilisateur binaire, ...), il faut noter que l'on évalue des systèmes et pas leurs modèles sous-jacents. Or, il n'est pas rare de constater que différentes implantations d'un même modèle (avec les même paramètres) ne donnent pas exactement les mêmes résultats (cf. (Beigbeder, 2014)).

Depuis l'année 2004, une approche radicalement différente a été proposée dans (Fang *et al.*, 2004) : l'idée est de définir des contraintes heuristiques sur les modèles eux-mêmes, et d'estimer si la vérification de ces contraintes est effectivement liées à la qualité des systèmes implantant ces modèles. Les contraintes sont basées sur les éléments fondamentaux reconnus de la Recherche d'Information : les tf , les idf, la taille des documents, et les interactions entre ces paramètres. Pour résumer ces heuristiques, elles sont liées à des dérivées partielles de la formule de correspondance théorique étudiée suivant les paramètres tf, idf, et la taille des documents. Des travaux ultérieurs à (Fang *et al.*, 2004) (par exemple (Fang et Zhai, 2005) (Fang *et al.*, 2011) (Clinchant et Gaussier, 2011) (Cummins et O'Riordan, 2007)) ont raffiné et corrigé certaines expressions, mais le principe est resté inchangé.

Les contraintes heuristiques existantes sont très importantes, mais dans le cas de requêtes par l'exemple (images par exemple), le comportement attendu d'un modèle de RI devrait à notre avis renvoyer le score de correspondance maximal si l'on pose en requête un document de la collection. Ceci n'est pas du tout avéré a priori pour les modèles de langues utilisant une divergence de Kullback-Leibler par exemple. Dans le cas de requêtes "courantes" en RI, courtes, ce point n'a pas nécessairement à être soulevé (Robertson, 2000), mais dans le cas des images il est à notre avis primordial.

3. DDMC pour des modèles de RI de référence

Cette partie décrit le comportement de fonctions de correspondances de modèles populaires de Recherche d'Information pour la contrainte DDMC. Nous passons en revue le modèle vectoriel avec pondération $tf.idf$, le modèle BM25 et le modèle de langue par négative de divergence de Kullback-Leibler et lissage de Jelinek-Mercer. Pour montrer que BM25 et le modèle de langue considéré ne vérifient pas inconditionnellement DDMC, nous nous contentons de fournir un contre-exemple. Notre proposition dans la section suivante modifiera la formule "classique" du modèle de langue pour garantir DDMC.

3.1. *DDMC pour le modèle vectoriel*

Si nous utilisons une approche de modèle vectoriel basée sur des tf.idf sans normalisation pivotée, il est relativement simple de constater que, si la pondération est la même pour les documents et les requêtes, la correspondance par cosinus sera égale à 1, valeur maximale. Plus précisément, si un document D du corpus est utilisé comme requête, alors il sera représenté par la même vecteur que le document dans le corpus : i) les valeurs de tf seront les mêmes pour tous les termes pour les deux représentations, et les valeurs d'idf, basées sur le corpus, seront également les mêmes pour les deux représentations du document. Un tel modèle vectoriel vérifie donc bien DDMC.

3.2. *DDMC pour le modèle BM25*

Nous utilisons la formule suivante du modèle réputé BM25 (Singhal, 2001) :

$$f_{BM25}(Q,D) = \sum_{t\in D\cap Q} ln\frac{N-df_t+0.5}{df_t+0.5} \cdot \frac{(k_1+1).\#t,D}{k_1.((1-b)+b.\frac{|D|}{avdl})+\#t,D} \cdot \frac{(k_3+1).\#t,Q}{k_3+\#t,Q} \quad [1]$$

avec $\#t,Q$ et $\#t,D$ le tf du terme t dans Q et D, $|D|$ la taille du document en nombre de termes, N la taille du corpus, df_t le nombre de documents du corpus indéxés par t, et $avdl$ la taille moyenne des documents du corpus.

Prenons un vocabulaire à deux termes, t_1 et t_2, un corpus de $N = 1000$ documents, avec $df_{t_1} = 501$ et $df_{t_2} = 100$. Considérons d'autre part deux documents D_1 et D_2 tels que $|D_1| = |D_2| = 10$, avec $\#t_1, D_1 = \#t_2, D_1 = 5$, $\#t_1, D_1 = 0$ et $\#t_2, D_2 = 10$. En posant $k_1 = k_2 = b = 1$ et $avdl = 10$, nous obtenons :

$$\begin{aligned} f_{BM25}(D_1,D_1) &= 6,029 \\ f_{BM25}(D_1,D_2) &= 6,645 \end{aligned}$$

D_2 répond alors mieux que D_1 à la requete D_1. Le contrainte DDMC n'est donc pas validée sur cet exemple.

3.3. *DDMC pour la divergence de Kullback-Leibler (KLD) avec lissage de Jelinek-Mercer (JM)*

Nous nous intéressons à la divergence de Kullback-Leibler dans les modèles de langues (qui généralise l'approche de probabilité de génération de requêtes, *Query Likelihood*). Ce choix provient du fait que cette divergence est celle qui a montré son intérêt théorique (compatibilité avec le bouclage de pertinence), et qui expérimentalement est compatible avec des fichiers inverses, condition *sine qua non* pour être utilisée dans des systèmes de recherche d'information.

La divergence négative de Kullback-Leibler est exprimée par :

$$nKL(Q||D) = -\sum_{t\in T} P(t|Q).\log\frac{P(t|Q)}{P(t|D)} \quad [2]$$

L'approche courante avec ce modèle, est de supposer que les probabilités pour les documents sont lissées, et que la probabilité pour la requête est non-lissée P_{ML} (estimation de la probabilité par maximum de vraisemblance), ce qui mène habituellement à la formule (en éliminant les termes non-présents dans la requête) suivante :

$$nKL(Q||D) = -\sum_{t\in Q} P(t|Q).\log\frac{P(t|Q)}{P(t|D)} \quad [3]$$

Malheureusement, avec l'expression [3], la contrainte DDMC n'est pas garantie, car si un document D est utilisé comme requête, sa représentation n'est pas lissée, alors que sa représentation dans le corpus est, elle, lissée. Il est dès lors possible qu'un autre document D' du corpus corresponde mieux au document D non-lissé que le document D lissé du corpus.

Supposons un vocabulaire de 3 termes, deux documents ayant les occurrences des termes fournies en tableau 2. Supposons d'autre part que les probabilités P(t|C) sur le

Tableau 2. *Occurrences des termes du vocabulaire dans D1 et D2, et probabilités P_{ML} par maximum de vraisemblance.*

t	t1	t2	t3
#t,D1	0	45	0
#t,D2	3	45	3
P_{ML}(t|D1)	0	1	0
P_{ML}	0,059	0,882	0,059

corpus C sont celles du tableau 3.

Tableau 3. *Probabilités des termes dans le corpus estimée par maximum de vraisemblance.*

	t1	t2	t3
P_{ML}(t|C)	1/3	1/3	1/3

Les probabilités lissées (cf. formule [4] plus loin) des termes pour les documents, en fixant $\lambda = 0,2$, sont alors celles du tableau 4.

Tableau 4. *Probabilités lissées de D1 et D2.*

document	t1	t2	t3
$P_\lambda(t\|D1)$	0,067	0,867	0,067
$P_\lambda(t\|D2)$	0,0474	0,7063	0,047

En utilisant la négative de la divergence de Kullback-Leibler, nous allons simuler une requête posée par le document D_2 : nous comparons donc d'un côté D_2 (estimé par maximum de vraisemblance) et D_1 lissé noté $D_{1\lambda}$, et de l'autre D_2 (estimé par maximum de vraisemblance) et D_2 lissé noté $D_{2\lambda}$. Nous obtenons les valeurs suivantes :

– nKL(D_2 || $D_{1\lambda}$) = -0,001

– nKL(D_2 || $D_{2\lambda}$) = -0,223

Un système de recherche d'information basé sur de telle probabilités retournera D_1 avant D_2 comme réponse, ce qui ne vérifie pas la contrainte DDMC décrite dans l'introduction.

4. Un modèle de divergence de KL avec lissage JM vérifiant DDMC

Dans cette partie, nous décrivons notre proposition pour garantir DDMC, puis nous montrons dans un deuxième temps que cette proposition peut être implantée avec une structure de fichiers inverses.

4.1. *Proposition utilisant un lissage de la requête*

Ce n'est pas la divergence elle-même qui est la cause de la non-vérification de DDMC, mais le fait que documents et requêtes ne sont pas traités de la même manière. Si nous voulons les traiter de la même manière, la solution que nous choisissons est de lisser la requête avec des probabilités provenant du corpus. Comme nous sommes dans le cas de requêtes par l'exemple, on peut considérer qu'elles sont longues. Dans ce cas, Zhai et Lafferty ont montré que le lissage de Jelinek-Mercer donne des résultats supérieurs à un lissage de Dirichlet (Zhai et Lafferty, 2001). Pour un document D, un lissage de Jelinek-Mercer (basé sur le corpus C) utilisé en Recherche d'Information, définit une probabilité P_λ telle que :

$$P_\lambda(t|D) = (1-\lambda).P_{ML}(t|D) + \lambda.P_{ML}(t|C) \quad [4]$$

Nous reprenons-donc la formule [2], en utilisant non pas une maximisation de vraisemblance pour la requête, mais en la lissant par le corpus, suivant le même lissage de Jelinek-Mercer (avec le même paramètre λ que pour les documents du corpus). Notre proposition est à rapprocher de (Louis et Nenkova, 2009), qui utilise dans un contexte d'évaluation de résumés automatiques une telle idée. Même si nous de redéfinissons pas une nouvelle divergence, nous choisissons d'appeler cette fonction de correspondance $nKLS$ (pour *negative Smoothed Kullback-Leibler*) :

$$nSKL(Q||D) = -\sum_{t\in T} P_\lambda(t|Q).\log\frac{P_\lambda(t|Q)}{P_\lambda(t|D)} \quad [5]$$

Il est simple de vérifier qu'une telle formulation vérifie bien la contrainte DDMC : si la requête est lissée avec un lissage de Jelinek-Mercer basé sur les probabilités provenant du corpus, alors si nous utilisons une requête qui est le document D, nous obtenons :

$$nSKL(D||D) = -\sum_{t\in T} P_\lambda(t|D).\log\frac{P_\lambda(t|D)}{P_\lambda(t|D)} \quad [6]$$

Comme $log(1) = 0$, la divergence obtenue par la requête D est alors égale à 0. Cette divergence est la valeur maximale théorique, encore plus contraignante que DDMC. **La formule [2] vérifie donc bien DDMC**.

4.2. *Compatibilité du modèle avec les fichiers inverses*

Comme notre proposition fixe que les requêtes sont lissées, ont pourrait supposer que l'utilisation de fichiers inverses est impossible à réaliser. Nous prouvons ici que, même avec des requêtes lissées par les probabilités du corpus, il est possible d'utiliser une expression, garantissant le même ordre des réponses que [2], compatible de classe 2 avec des fichiers inverses suivant la dénomination de (Mulhem et Chevallet, 2014), qui stipule que le formule de correspondance peut être exprimée comme une fonction dépendant de l'intersection des termes de la requête et du document, et éventuellement d'une constante dépendant uniquement du document.

La première étape vers une expression compatible avec des fichiers inverses consiste à séparer dans [5] les termes présents/absents dans la requête Q et présents/absents du document D. Cette séparation nous permet d'introduire les notations $P_{s\lambda}$ pour la probabilité lissée P_λ pour un terme présent (*seen*) et $P_{u\lambda}$ pour la probabilité lissée pour un terme non-présent (*unseen*) :

$$\begin{aligned} nSKL(Q||D) &= -\sum_{t\in Q, t\in D} P_{s\lambda}(t|Q).\log\frac{P_{s\lambda}(t|Q)}{P_{s\lambda}(t|D)} \\ &\quad -\sum_{t\in Q, t\notin D} P_{s\lambda}(t|Q).\log\frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)} \\ &\quad -\sum_{t\notin Q, t\in D} P_{u\lambda}(t|Q).\log\frac{P_{u\lambda}(t|Q)}{P_{s\lambda}(t|D)} \\ &\quad -\sum_{t\notin Q, t\notin D} P_{u\lambda}(t|Q).\log\frac{P_{u\lambda}(t|Q)}{P_{u\lambda}(t|D)} \end{aligned} \quad [7]$$

Le premier élément de la formule [7] dépend des termes en commun dans D et Q, on n'a donc pas besoin de le modifier car il est déjà utilisable sur des fichiers inverses. Le dernier élément de la formule [7] est égal à zéro dans le cas d'utilisation d'un lissage de Jelinek-Mercer, car $P_{u\lambda}(t|Q) = P_{u\lambda}(t|D) = \lambda.P_{ML}(t|C)$.

Nous nous préoccupons donc des deux parties restantes de la formule [7]. Si nous retranchons et ajoutons la constante $\sum_{t \in Q} P_{s\lambda}(t|Q).\log \frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)}$ à la deuxième partie de la formule [7], nous avons :

$$\begin{aligned}\sum_{t \in Q, t \notin D} P_{s\lambda}(t|Q).\log \frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)} &= \sum_{t \in Q, t \notin D} P_{s\lambda}(t|Q).\log \frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)} \\ &\quad - \sum_{t \in Q} P_{s\lambda}(t|Q).\log \frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)} \\ &\quad + \sum_{t \in Q} P_{s\lambda}(t|Q).\log \frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)} \qquad [8]\end{aligned}$$

Ce qui donne après simplification :

$$\begin{aligned}\sum_{t \in Q, t \notin D} P_{s\lambda}(t|Q).\log \frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)} &= - \sum_{t \in Q, t \in D} P_{s\lambda}(t|Q).\log \frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)} \\ &\quad + \sum_{t \in Q} P_{s\lambda}(t|Q).\log \frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)} \qquad [9]\end{aligned}$$

La dernière partie de la formule [9] est une constante pour une requête car, comme nous l'avons noté précédemment $P_{u\lambda}(t|D) = \lambda.P_{ML}(t|C)$. Ce qui donne (en utilisant le le symbole $\propto_Q$ pour dénoter "est égal, à une constante dépendante uniquement de Q près") :

$$\sum_{t \in Q, t \notin D} P_{s\lambda}(t|Q).\log \frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)} \propto_Q - \sum_{t \in Q, t \in D} P_{s\lambda}(t|Q).\log \frac{P_{s\lambda}(t|Q)}{P_{u\lambda}(t|D)} \qquad [10]$$

Si nous retranchons et ajoutons la constante $\sum_{t \in D} P_{u\lambda}(t|Q).\log \frac{P_{u\lambda}(t|Q)}{P_{s\lambda}(t|D)}$ à l'avant-dernière partie de la formule [7], nous avons :

$$\begin{aligned}\sum_{t \notin Q, t \in D} P_{u\lambda}(t|Q).\log \frac{P_{u\lambda}(t|Q)}{P_{s\lambda}(t|D)} &= \sum_{t \notin Q, t \in D} P_{u\lambda}(t|Q).\log \frac{P_{u\lambda}(t|Q)}{P_{s\lambda}(t|D)} \\ &\quad - \sum_{t \in D} P_{u\lambda}(t|Q).\log \frac{P_{u\lambda}(t|Q)}{P_{s\lambda}(t|D)} \\ &\quad + \sum_{t \in D} P_{u\lambda}(t|Q).\log \frac{P_{u\lambda}(t|Q)}{P_{s\lambda}(t|D)} \qquad [11]\end{aligned}$$

Ce qui donne après simplification :

$$\sum_{t\notin Q, t\in D} P_{u\lambda}(t|Q).\log\frac{P_{u\lambda}(t|Q)}{P_{s\lambda}(t|D)} = -\sum_{t\in Q, t\in D} P_{u\lambda}(t|Q).\log\frac{P_{u\lambda}(t|Q)}{P_{s\lambda}(t|D)} - \sum_{t\in D} P_{u\lambda}(t|Q).\log\frac{P_{s\lambda}(t|D)}{P_{u\lambda}(t|Q)} \quad [12]$$

Le dernier élément de la formule [12] est une constante pour un document D, car comme nous l'avons décrit plus haut, $P_{u\lambda}(t|Q)$ avec un lissage de Jelinek-Mercer peut être exprimée indépendamment de la requête par $\lambda.P_{ML}(t|C)$. Par souci de simplicité, cette constante pour le document est notée K_D.

Nous réintégrons tous ces éléments dans la formule initiale [7], en utilisant explicitement $\lambda.P_{ML}(t|C)$ nous obtenons :

$$\begin{aligned} nSKL(Q||D) \quad \propto_Q \quad & \sum_{t\in Q, t\in D} -P_{s\lambda}(t|Q).\log\frac{P_{s\lambda}(t|Q)}{P_{s\lambda}(t|D)} \\ & +P_{s\lambda}(t|Q).\log\frac{P_{s\lambda}(t|Q)}{\lambda.P(t|C)} \\ & +\lambda.P_{ML}(t|C).\log\frac{\lambda.P_{ML}(t|C)}{P_{s\lambda}(t|D)} + K_D \end{aligned} \quad [13]$$

Pour l'implantation de la formule [13], on peut encore combiner les deux premiers termes, et développer la formule lissée de la requête, ce qui donne la fonction expérimentée dans la suite :

$$nSKL(Q||D) \quad \propto_Q \sum_{t\in Q, t\in D} \quad ((1-\lambda).P_{ML}(t|Q).\log\frac{P_{s\lambda}(t|D)}{\lambda.P_{ML}(t|C)}) + K_D \quad [14]$$

La formule [14], malgré le fait qu'elle intègre un lissage de la requête, est compatible de niveau 2 avec l'approche à base de fichier inverse, telle qu'exprimée dans (Mulhem et Chevallet, 2014). Cette compatibilité stipule que la fonction de correspondance peut s'exprimer sous la forme de la somme de composants liées aux termes présents à la fois dans le document et la requête, et d'une valeur dépendant uniquement du document (qui peut donc être calculée a priori). Ce point est fondamental pour que cette formule soit implantée dans un système de recherche d'information.

4.3. *Validation des axiomes de base par nSKL*

Cette partie a pour objectif de déterminer dans quelle mesure $nSKL$ vérifie les axiomes définis par Fang, Tao et Zhai en 2004 (Fang *et al.*, 2004).

La contrainte TFC1 est exprimée comme suit : "pour une requête Q composée d'un terme w, pour deux documents D_1 et D_2 de même taille $|D_1| = |D_2|$, tels que $c(w, D_1) > c(w, D_2)$, avec $c(t, D)$ le nombre d'occurrences du terme t dans le document D, alors $nSLK(Q||D_1) > nSLK(Q||D_2)$". Pour traiter cette contrainte axiomatique, nous faisons appel à une expression de nSKL réécrite en prenant en compte que la requête est composée d'un terme et que la probabilité de ce terme dans la requête non lissée est égal à 1, et en utilisant explicitement un autre terme w' n'apparaissant pas dans la requête. En développant les probabilités lissées, nous obtenons :

$$\begin{aligned} nSKL(Q||D_1) \quad \propto_Q \quad & (1-\lambda).\log(\frac{1-\lambda}{\lambda}.\frac{c(w,D_1)}{|D_1|.P_{ML}(w|C)}+1) \\ + \quad & \lambda.P_{ML}(w|C).\log(\frac{1-\lambda}{\lambda}.\frac{c(w,D_1)}{|D_1|.P_{ML}(w|C)}+1) \\ + \quad & \lambda.P_{ML}(w'|C).\log(\frac{1-\lambda}{\lambda}.\frac{c(w',D_1)}{|D_1|.P_{ML}(w'|C)}+1) \qquad [15] \\ + \quad & \sum_{t\in D_1\setminus\{w,w'\}} \lambda.P_{ML}(t|C).\log(\frac{1-\lambda}{\lambda}.\frac{c(t,D_1)}{|D_1|.P_{ML}(t|C)}+1) \end{aligned}$$

Dans la formule [15], seule la première ligne est multipliée par $1-\lambda$, les autres étant multipliées par $\lambda.P_{ML}(.|C)$. Si nous reprenons une hypothèse utilisée dans (Fang *et al.*, 2004) qui stipule que pour un terme t porteur de sens on peut estimer que $P(t|C)$ est de l'ordre de $\frac{1}{avdl}$, dans ce cas les valeurs de la deuxième et troisième lignes de la formule [15] sont négligeables car les documents (de campagnes courantes en RI) sont de taille de plusieurs centaines de termes en moyenne, alors que λ est plutôt compris entre 0,2 et 0,4. D'autre part, les auteurs de (Clinchant et Gaussier, 2010) ont indiqué que l'on peut évaluer TFC1 comme la dérivée de la fonction de correspondance par rapport aux occurrences du terme considéré. En notant c la valeur $c(w, D_1)$, et en négligeant les parties comme vu ci-dessus, nous avons :

$$\frac{\partial nSKL(Q||D_1)}{\partial c} \quad = \quad \frac{(1-\lambda)^2}{\lambda.|D_1|.P_{ML}(w|C)-\lambda.c+c} \qquad [16]$$

Cette dérivée est strictement positive si $\lambda \in [0, 1[$, donc, si l'on néglige les deuxième et troisième parties de la formule [15], et en se basant sur l'hypothèse qui indique qu'un terme porteur de sens apparaît en moyenne une fois dans chaque document, nSKL valide TFC1. On peut signaler que le fait de supposer, dans la formule [15], que seules des occurrences d'un seul terme, w', sont modifiées n'est pas limitatif, car dans le cas où plusieurs termes en dehors de w sont modifiés, alors ces autres éléments sont également considérés comme négligeables.

Pour la contrainte TFC2 de (Fang *et al.*, 2004), il faut vérifier le signe de la dérivée seconde de la formule [15] par rapport à c, ce qui donne :

$$\frac{\partial^2 nSKL(Q||D_1)}{\partial c^2} = \frac{(1-\lambda)^3}{(\lambda.|D_1|.P_{ML}(w|C) - \lambda.c + c)^2} \quad [17]$$

Cette dérivée est strictement négative si $\lambda \in [0, 1[$ avec les même hypothèses que pour TFC1. nSKL valide donc TFC2.

Nous nous intéressons maintenant aux contraintes qui portent sur la normalisation de la longueur des documents. La contrainte LNC1 stipule que "soit Q une requête et D_1 et D_2 deux documents ; si pour un mot $w' \notin q$, $c(w', D_2) = c(w', D_1) + 1$, mais pour tous les termes w de la requête, $c(w, D_1) = c(w, D_2)$, alors $nSKL(Q||D_1) \geq nSKL(Q||D_2)$". LNC1 est basée sur la dérivée de la formule [15] par rapport à la taille du document que nous notons d (cf. (Clinchant et Gaussier, 2010)). Avec la même hypothèse que précédemment pour ne garder que la première et la dernière partie de [15], en se limitant à une requête à un terme nous avons :

$$\begin{aligned}\frac{\partial nSKL(Q||D_1)}{\partial d} &= -\sum_{t\in Q, t\in D_1} \frac{c(t,D_1).(\lambda-1)^2}{d.(\lambda.d.P_{ML}(t|C) - \lambda.c(t,D_1) + c(t,D_1))} \\ &+ \sum_{t\in D_1\setminus\{w,w'\}} \frac{\lambda.(\lambda-1).c(t,D_1).P_{ML}(t|C)}{d.(\lambda.d.P_{ML}(w|C) - \lambda..c(t,D_1) + c(t,D_1))} \quad [18]\end{aligned}$$

On constate que cette expression est strictement négative car $\lambda < 1$, la contrainte LNC1 est donc vérifiée pour une requête à un terme[1]. La contrainte LNC2 correspond au cas où un document est concaténé k fois. Dans ce cas, il est aisé de voir que la formule [15] donne le même résultat pour le document initial et le document concaténé, la contrainte LNC2 est donc validée par nSKL pour une requête à un terme.

Nous nous intéressons à la contrainte spe_TDC de (Clinchant et Gaussier, 2010). Cette expression est en fait exprimée par la dérivée par rapport au nombre d'occurence du terme dans le corpus, proportionnelle à la probabilité $P_{ML}(t|C)$, que nous notons p. En calculant la dérivée de la formule [15] par rapport à p, nous obtenons :

$$\frac{\partial nSKL(Q||D_1)}{\partial p} = -\frac{c(t,D_1).(1-\lambda)^2}{|D_1|.\lambda.(\frac{c(t,D_1).(1-\lambda)}{|D_1|.\lambda.p} + 1).p^2} \quad [19]$$

Cette dérivée est négative, donc la contrainte spe_TDC est validée par nSKL.

La dernière contrainte que nous devons vérifier est appelée TF-LNC. Elle décrit le comportement du paramètre d'occurrence des termes par rapport à celui de la taille des documents. L'idée est que le paramètre de la fréquence des termes doit primer

1. Le cas de requêtes à plusieurs termes n'est pas considéré ici.

sur l'évolution de la taille des documents dans certaines conditions. En l'absence de validation théorique simple de cette contrainte pour nSKL, nous présentons ici une première étape, numérique, de cette validation. Nous avons créé toutes les variations d'occurrences entre 1 et 20 (par pas de 5) des termes de documents sur un vocabulaire de 20 termes, et nous avons vérifié qu'une variation d'une occurence du terme de la requête dans le document, pour une valeur fixe des probabilités des termes dans le corpus (0.00001), vérifie la contrainte TF-LNC. Cette vérification n'est pas une preuve, mais elle valide très partiellement la contrainte TF-LNC pour nSKL. Des études ultérieures seront menées pour renforcer cette validation.

Nous avons donc réalisé dans cette partie une étude préliminaire sur la validation par nSKL des contraintes axiomatiques de base de la recherche d'information tirées de (Fang *et al.*, 2004 ; Clinchant et Gaussier, 2010). Cette étude montre que la formule proposée valide, suivant les hypothèses posées, les contraintes courantes de recherche d'information.

5. Expérimentations

Nous avons testé notre proposition en implantant la formule [14] décrite ci-dessus en partie 4. Nous réalisons un premier jeu de test sur un exemple-type de recherche par l'exemple : la recherche d'images. Nous utilisons la collection de bâtiments de Zürich appelée *Zubud* (Shao *et al.*, 2003). Cette collection est composée d'un corpus de 1005 images de 201 bâtiments, et d'un ensemble de test de 115 images requêtes. Des résultats reportés dans (Obdrzalek et Matas, 2003) atteignent 100%, la problématique liée à cette collection pourrait donc être considérée comme résolue, cependant notre objectif ici est vérifier que notre proposition obtient des résultats similaires à l'approche de référence, mais en gardant à l'esprit que notre proposition possède l'avantage de garantir que si un document est utilisé comme requête il aura la valeur de correspondance maximale. Les caractéristiques visuelles extraites sont des SIFT couleurs sur un échantillonnage spatial dense, en utilisant le logiciel développé par Koen van De Sande (van de Sande *et al.*, 2010). Un regroupement (*clustering*) des caractéristiques est effectué par un algorithme de K-moyennes, définissant un vocabulaire de taille 2000 dimensions. Cette approche est habituellement appelée sac de mots visuels (cf. (Csurka *et al.*, 2004)). Pour le modèle de langues considéré, nous avons utilisé le lissage de Jelinek-Mercer avec 9 valeurs de λ de 0,1 à 0,9 par pas de 0,1.

Les résultats obtenus sont présentés dans le tableau 5. Nous constatons dans ce tableau que les résultats obtenus avec notre proposition sont très proches de ceux de référence obtenu par Kullback-Leibler et un lissage de Jelinek-Mercer : les différences relatives par rapport à nKL sont au maximum de 1,1%. Dans tous les cas ces valeurs de sont pas significatives selon un t-test pairé bilatéral avec un seuil de significativité de 5 %. Dans les deux cas, comme nous pouvions nous y attendre, les meilleurs résultats sont obtenus avec des valeurs de λ faibles, et plus faibles pour nSKL par rapport à nKL. Ceci peut s'expliquer que le lissage étant aussi utilisé dans la requête, il doit être faible pour garantir des bons résultats. Nous concluons donc ici que pour cet exemple,

le fait de rajouter le lissage de la requête n'impacte pas significativement la qualité des réponses d'un système de recherche d'information.

Tableau 5. *MAP, Rang Réciproque Moyen, précision à 10 documents et taux de reconnaissance à 1 document pour la collection Zubud*

divergence (paramètre)	MAP	MRR	P@10	Taux de Reco.
nKL ($\lambda = 0,2$)	0.6988	0.9072	0.3826	0.8782
nSKLD ($\lambda = 0,1$)	0.6949 (-0,6 %)	0.9029 (-0,5 %)	0.3783 (-1,1 %)	0.8696 (-1,0 %)

Un second jeu de tests est effectué sur des documents textes, utilisant sur la collection TREC-6 ad-hoc (Voorhees et Harman, 2000), en utilisant une référence avec un lissage de Jelinek-Mercer. Cette collection contient environ 550 000 documents, et l'évaluation utilise 50 requêtes. Nous avons posé les requêtes en utilisant les champs $topic$, $description$ et $narrative$, ce qui correspond à des requêtes "longues" selon la terminologie de (Voorhees et Harman, 2000). Nous voulons étudier le comportement de notre proposition dans un cas relativement défavorable, où même les requête "longues" sont plus courtes que les documents. Les résultats obtenus sont présentés dans le tableau 6. Nous voyons que les résultats de notre proposition sont toujours inférieurs à notre référence. Cependant, des t-tests sur ces différences pairés bilatéraux, avec un seuil de significativité de 0.05, ne permettent pas de conclure sur une différence significative.

Tableau 6. *MAP, précision à 10, précision à 30 documents pour TREC-6 ad-hoc*

divergence (paramètre)	MAP	P@10	P@30
nKL ($\lambda = 0,7$)	0.2646	0.4400	0.3307
nSKLD ($\lambda = 0,4$)	0.2477 (-6,4 %)	0.4360 (-0,9 %)	0.3247 (-1,8 %)

Ces deux résultats montrent que sur les expérimentations menées les résultats obtenus par notre proposition, qui garantit DDMC, sont très proches de ceux obtenus par l'approche de référence.

6. Conclusion

Dans cet article, nous nous sommes intéressés à une caractéristique théorique des modèles de langue de recherche d'information, qui ne permet pas de garantir que si un document du corpus est posé en requête, alors il est renvoyé en tête par le système de recherche d'information. Cette caractéristique peut paraître anecdotique, mais elle peut se révéler très déstabilisante pour un utilisateur. Nous avons choisi de définir une heuristique des systèmes de recherche d'information, notée DDMC, pour *Document-Document Matching Constraint*, qui indique le comportement attendu dans un tel cas. Une fois cette heuristique définie, le travail décrit ici a porté sur la définition d'une

formule d'un modèle de langue de recherche d'information capable de garantir cette heuristique. Nous avons donc travaillé sur un modèle très classique et réputé pour les requêtes longues : le calcul de la négative de la divergence de Kullback-Leibler en lissant les documents par un lissage de Jelinek-Mercer. Notre proposition est de lisser également les requête de la même manière. Un résultat de ce travail fournit une expression d'une telle correspondance compatible avec une implantation avec les fichier inverses. Nous avons par ailleurs établi une étude préliminaire, avec certaine hypothèses et limitations qui devront être raffinées, de la compatibilité de notre proposition avec les contrainte axiomatiques couramment admise pour les modèles de recherche d'information. Des expérimentations ont montré que notre proposition est capable d'obtenir des résultats comparables à ceux d'une approche de référence, tout en garantissant explicitement la vérification de la contrainte DDMC.

Ce travail va se poursuivre dans plusieurs directions. Une première consiste à travailler en profondeur la contrainte DDMC pour déterminer dans quelle mesure cette formulation de la contrainte est la meilleure et dans quelle mesure elle est vérifiée par d'autres modèles de l'état de l'art. Notre proposition sera davantage mise en perspective par rapport aux contraintes axiomatiques de la recherche d'information. Nous étudierons ensuite la transposition de ce travail à d'autres modèles de RI, comme BM25 par exemple, ou bien pour des modèles de langue avec lissage de Dirichlet.

Remerciements

Ce travail a été réalisé avec le support partiel de la région Rhône-Alpes pour le projet GUIMUTEIC.

7. Bibliographie

Beigbeder M., « Les outils pour la RI », , http ://www.asso-aria.org/documents/earia/earia2014/slides/6-BEIGBEDER.pdf, 2014. [Online ; Ecole d'Automne en Recherche d'Information et Applications EARIA 2014].

Clinchant S., Gaussier E., « Information-based models for ad hoc IR », *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, ACM, p. 234-241, 2010.

Clinchant S., Gaussier É., « Do IR models satisfy the TDC retrieval constraint », *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, p. 1155-1156, 2011.

Csurka G., Dance C. R., Fan L., Willamowski J., Bray C., « Visual categorization with bags of keypoints », *In Workshop on Statistical Learning in Computer Vision, ECCV*, p. 1-22, 2004.

Cummins R., O'Riordan C., « An axiomatic comparison of learned term-weighting schemes in information retrieval : clarifications and extensions », *Artif. Intell. Rev.*, vol. 28, n° 1, p. 51-68, 2007.

Fang H., Tao T., Zhai C., « A Formal Study of Information Retrieval Heuristics », *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, ACM, New York, NY, USA, p. 49-56, 2004.

Fang H., Tao T., Zhai C., « Diagnostic Evaluation of Information Retrieval Models », *ACM Trans. Inf. Syst.*, vol. 29, n° 2, p. 1-49, 2011.

Fang H., Zhai C., « An Axiomatic Approach to IR–UIUC TREC 2005 Robust Track Experiments », *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, Maryland, November 15-18, 2005*, 2005.

Louis A., Nenkova A., « Automatically Evaluating Content Selection in Summarization Without Human Models », *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 1 - Volume 1*, Association for Computational Linguistics, p. 306-314, 2009.

Mulhem P., Chevallet J., « Correspondances compatibles avec les fichiers inverses pour la recherche d'information », *CORIA 2014 - Conférence en Recherche d'Infomations et Applications- 11th French Information Retrieval Conference. CIFED 2014 Colloque International Francophone sur l'Ecrit et le Document, Nancy, France, March 19-23, 2014.*, p. 353-368, 2014.

Obdrzalek S., Matas J., « Image Retrieval Using Local Compact DCT-Based Representation. », *in* B. Michaelis, G. Krell (eds), *DAGM-Symposium*, vol. 2781 of *Lecture Notes in Computer Science*, Springer, p. 490-497, 2003.

Robertson S. E., « Salton Award Lecture : On theoretical argument in information retrieval », *SIGIR Forum*, vol. 34, n° 1, p. 1-10, 2000.

Shao H., Svoboda T., Gool L. V., ZuBuD — Zürich buildings database for image based recognition, Technical Report n° 260, Computer Vision Laboratory, Swiss Federal Institute of Technology, March, 2003.

Singhal A., « Modern information retrieval : a brief overview », *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 24, p. 2001, 2001.

van de Sande K. E. A., Gevers T., Snoek C. G. M., « Evaluating Color Descriptors for Object and Scene Recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, n° 9, p. 1582-1596, 2010.

Voorhees E. M., Harman D., « Overview of the sixth text REtrieval conference (TREC-6) », *Inf. Process. Manage.*, vol. 36, n° 1, p. 3-35, January, 2000.

Zhai C., Lafferty J., « A study of smoothing methods for language models applied to Ad Hoc information retrieval », *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, ACM, New York, NY, USA, p. 334-342, 2001.