Publié dans les
Actes 11e Journées internationales d'analyse statistique des données textuelles JADT 2012, 753-766, 2012
qui doivent être utilisés pour toute référence à ce travail

Etude comparative de l'efficacité du dépistage de l'information dans des manuscrits médiévaux

Nada Naji, Jacques Savoy

Institut d'informatique

Université de Neuchâtel - rue Emile Argand 11 - 2000 Neuchâtel - Suisse

Abstract

This paper presents, evaluates and compares the effectiveness of information retrieval (IR) for medieval manuscripts when facing with noisy texts. The corpus used in our experiments is based on a well-known medieval epic poem written in Middle High German dating to the thirteenth century (Parzival). An error-free transcription of the poem was created manually and made available by experts. This error-free transcription represents our baseline that we used to assess the performance levels. In practice, the document noise could be caused by different sources (e.g., spelling variations due to non-normalized medieval text, recognition mistake). To overcome these difficulties, we suggest several query expansion strategies, hence allowing some form of spelling variation between the requests and the searchable items. To analyze these performances under several conditions, we have evaluated five IR models, three forms of stemming and three text representations. We show that incorporating the maximum spelling variation possibilities in the query expansion process does not produce the best results, while a wiser and more conservative approach of involving expansion terms yields better performance levels.

Keywords: IR effectiveness with noisy text, medieval manuscripts, spelling variations, digital libraries, OCR.

Résumé

Cette communication présente et évalue l'efficacité de la recherche d'information (RI) dans un corpus de manuscrits médiévaux en présence d'erreurs. Dans nos évaluations, nous avons retenu le corpus *Parzival*, poème épique allemand du 13ième siècle. Une version sans erreur a été créée manuellement à l'aide d'experts afin de servir de base de comparaison. La présence de bruit s'explique par des variations d'écriture provenant d'une orthographe médiévale non standardisée ou d'erreurs lors de la reconnaissance. Afin de permettre un appariement entre des formes différentes se référant au même objet, nous avons conçu et évalué plusieurs stratégies d'expansion massive de la requête. Pour vérifier leur efficacité sous diverses conditions, nous avons retenus cinq modèles de recherche, trois formes de suppression des séquences terminales (*stemming*) et trois représentations des textes. Si le recours à une expansion maximale des requêtes ne permet pas d'obtenir la meilleure qualité de recherche, cette dernière s'obtient par un choix plus judicieux des termes à inclure dans la requête étendue.

Mots-clés : Recherche d'information dans des textes scannés, manuscrits médiévaux, OCR, variation orthographique, bibliothèque numérique.

1. Introduction

Les deux dernières décennies ont vu l'apparition de plusieurs projets de bibliothèques numériques ayant une dimension nationale (par exemple, Gallica en France) ou internationale (The European Library ou Europeana). Si la préservation de notre héritage culturel constitue l'intérêt majeur de tels projets, l'accès aisé et distribué à leur contenu représente un deuxième objectif important. Notre projet se situe dans cette perspective, mais en se limitant à des sources écrites et, en particulier, à des manuscrits médiévaux. D'un point de vue technique, leur gestion présente une tâche complexe. Ainsi, durant les phases de traitement d'images et de reconnaissance des caractères, nous devons tenir compte des enluminures entourant le texte, des commentaires insérés entre les lignes ou dans les marges, ainsi que de la présence

de trous et taches sur le support. Cependant, l'objectif de cet article se limite à la mise au point et à l'analyse de l'efficacité d'outils de recherche documentaire pour de tels corpus. Le traitement d'images (numérisation, génération des métadonnées) a été réalisé par le groupe DIVA (Université de Fribourg) tandis que la reconnaissance des caractères a été accomplie par le groupe Computer Vision (Universität Bern).

Lorsque l'on travaille avec des manuscrits médiévaux, il s'avère illusoire de disposer d'une transcription sans erreur et une tolérance aux erreurs de reconnaissance doit être admise. Le taux d'erreur va dépendre de plusieurs facteurs, dont la qualité du système de reconnaissance, le niveau de contraste entre la couleur du support et celle de l'encre, la régularité de l'écriture manuelle, etc. On doit admettre qu'un taux d'erreur élevé comme, par exemple, de l'ordre de 25 % va rendre difficile une recherche d'information de bonne qualité.

De plus, on doit souligner que l'orthographe des manuscrits médiévaux n'est pas normée, rendant ainsi plus difficile un appariement adéquat entre formes différentes, mais se référant au même objet, comme par exemple dans l'ancien français, *jor*, *jur* et *jour*, *cheveus*, *chevex* ou *chevels* (Andrieux-Reix *et al*., 2000). Ce phénomène constitue un second obstacle à une recherche documentaire efficiente. On peut également ajouter que la syntaxe est non seulement différente de la grammaire contemporaine, mais elle présente également des variations d'une région à l'autre, voire d'un auteur à l'autre.

La suite de cette communication est organisée de la manière suivante. La prochaine section décrit l'état actuel des connaissances, tandis que la troisième section décrit le corpus utilisé et notre méthodologie d'évaluation. La quatrième section présente les modèles d'indexation et de recherche servant de base à nos expériences. La cinquième section évalue et analyse les diverses stratégies de recherche sur l'ensemble de nos manuscrits médiévaux.

2. Etat des connaissances

La mise au point d'un moteur de recherche efficient pour des corpus de documents historiques constitue un problème non résolu de manière satisfaisante, même en face d'une demande croissante de la part des archives, bibliothèques et du public en général (Toni *et al*., 2004; Nicholas *et al*., 2005). Un des principaux défis est de permettre une recherche documentaire face à des documents présentant de nombreuses erreurs (Callan *et al*., 2002).

Afin d'étudier cette question, la campagne d'évaluation TREC-5 (*confusion track*) a organisé une série d'expériences (Voorhees & Garofolo, 2005). Dans ce cadre, trois versions différentes d'un même corpus, écrit en anglais, ont été créées. La première correspond à une version sans erreur et sert de base de référence. Les deuxième et troisième versions sont obtenues par reconnaissance optique des caractères sur la base d'une version imprimée. Le taux d'erreur par caractère s'élève respectivement à 5 % et à 20 %. Afin de mesurer la qualité de réponse fournie par le système de dépistage de l'information, on a recours à la moyenne de l'inverse du rang ou MRR (*mean reciprocal rank*). Pour une requête donnée, cette mesure se calcule comme l'inverse du rang de la première réponse juste dépistée (Buckley & Voorhees, 2005). Cette mesure reflète l'attente d'un utilisateur intéressé par une ou un petit nombre de bonnes réponses à son interrogation.

Lors de la campagne TREC-5, le meilleur système a obtenu une valeur MRR de 0,735 sur la version sans erreur et un MRR de 0,574 en présence d'un taux d'erreur de reconnaissance de 5 % (soit une dégradation relative de l'ordre de 22 %). Lorsque le corpus présente un taux d'erreur de 20 %, la performance s'élève à 0,498, pour une différence relative de -32 %. Des niveaux similaires de dégradation ont été annoncés par d'autres systèmes (Voorhees &

Garofolo, 2005). Ces taux d'erreurs élevés ne sont pas une fatalité car Tagva *et al*. (1994) indiquent qu'en présence d'images scannées en haute définition et avec une reconnaissance optique performante, le taux d'erreur peut être limité aux environs de 2 %.

Toutefois ces expériences ont été conduites sur la langue anglaise et avec des textes imprimés. Lorsque l'original est un manuscrit médiéval, sur du papier légèrement coloré, présentant des taches, trous, ou point de suture, le taux d'erreur s'élève au-dessus des 2 %. Ainsi, des études menées sur des manuscrits de Georges Washington, datant de la fin du XVIIIe siècle, (Toni *et al*., 2004; Nicholas *et al*., 2005) indiquent des taux d'erreur supérieurs (environ 50%).

Lorsque l'on considère des langues médiévales, nous devons faire face à des variations tant dans l'orthographe que dans la grammaire. Ainsi, pour traiter les variantes orthographiques de l'anglais de l'époque de Shakespeare dans les pièces de théâtre ou les poèmes, quelques approches ont été proposées (Craig & Whipp, 2010; Pilz *et al*., 2006). A cette époque, ces variations peuvent s'illustrer par le nom même de Shakespeare et peuvent apparaître sous cinq formes différentes (Shakper, Shakspe, Shaksper, Shakspere ou Shakspeare) mais aucune ne correspond à l'écriture moderne. Avec la langue allemande, nous devons également faire face à la présence de mots composés. Ainsi, le mot frigidaire *Kühlschrank* correspond à la concaténation des mots *kühl* (froid) et *Schrank* (étagère). Selon les évaluations comparatives de la campagne CLEF (Peters *et al*., 2004), la décomposition automatique des mots allemands permet d'accroître la performance d'un système de recherche. En effet, dans cette langue, le même concept peut s'écrire comme un mot composé ou comme une séquence de mots simples, comme par exemple, *Bankpräsident* ou *Präsident der Bank*. On notera toutefois que la fréquence des mots composés est plus élevée dans l'allemand contemporain (soit 25 %) que dans l'allemand du Moyen Age (environ 7 %) (Gardt *et al*., 1999).

3. Corpus et méthodologie d'évaluation

Le corpus utilisé dans nos expériences est extrait du poème épique Parzival (ou Perceval) attribué à Wolfram von Eschenbach. Une première version de ce récit, datée du début du XIIIe siècle, est écrite en moyen haut-allemand. Actuellement, on dénombre plusieurs versions avec leurs variations propres. Notre corpus provient du manuscrit *codex 857* de la bibliothèque de St-Gall (Fischer *et al*., 2007). Afin de vérifier la qualité de nos traitements, une version sans erreur a été générée manuellement avec l'aide d'experts du domaine.

3.1. Reconnaissance de l'écriture manuelle

Dans le projet HisDoc, les pages du manuscrit ont été numérisées, puis les zones de texte ont été identifiées, ainsi que le nombre de lignes et de colonnes par page. Au besoin, l'orientation des lignes a été ajustée et chaque ligne a été clairement identifiée. Sur cette base, la reconnaissance des caractères a été réalisée au moyen d'un modèle de Markov caché (Fischer *et al*., 2007). Ensuite, à l'aide d'un modèle de langue, on détermine la séquence des mots la plus probable. Après cette opération, le système de reconnaissance conserve également les sept formes les plus probables, triées selon leur valeur de vraisemblance. Ainsi pour chaque mot, nous sommes en mesure de connaître les substitutions possibles. Par exemple, lorsque l'on inspecte le vocable *man* dans le verset "*dem man dirre aventivre giht*" on retrouve les paires : [*man* #36007, *min* #35657, *mat* #35453, *nam* #35425, *arm* #35296, *nimt* #35278, *gan* #35266]. Le vocable *man* apparaît en première position et correspond à un choix correct. Après chaque possibilité, nous disposons du logarithme de la vraisemblance qui s'élève, par exemple, à 36 007 pour le vocable *man* ou 35 266 pour le mot *gan*. Nous utiliserons cet exemple, nommé *man*, plusieurs fois dans la suite de cette communication.

Dans ce processus de reconnaissance, nous avons admis l'hypothèse du vocabulaire clos qui stipule que chaque mot apparaissant dans la partie test a été vu, au moins une fois, lors de l'apprentissage. Diverses versions du système ont été conçues et implémentées pour aboutir à un taux de reconnaissance des mots s'élevant à 94 %. Le texte disponible pour la recherche d'information contient donc un taux d'erreur par mot de l'ordre de 6 %. Ces documents correspondent à la version bruitée de Parzival utilisée dans nos expériences.

3.2. Génération des divers corpus d'évaluation

En suivant une tradition des médiévistes, chaque verset formera un document et constituera une réponse possible du système de recherche. Dans la version actuelle d'évaluation, nous disposons de 1 328 documents représentant un sous-ensemble du corpus Parzival. Un autre ensemble, disjoint du premier, comprend 4 477 versets qui ont été utilisés lors de la phase d'entraînement. Ces lignes seront ignorées dans nos évaluations.

Ayant accès non seulement aux mots retenus par le système de reconnaissance, mais également aux six autres paires (mot, valeur de vraisemblance), nous pouvons analyser la qualité de différentes solutions. Comme premier corpus d'évaluation, nous pouvons nous limiter à la meilleure solution déterminée par le système de reconnaissance (soit tenir compte uniquement du mot *man* dans notre exemple précédent). Ce corpus possèdera le nom de MS1 (l'unique Meilleure Solution).

Sachant que chaque document est bref, la présence d'une erreur de reconnaissance ne permettra habituellement pas un appariement entre la requête et le document visé. Un appariement plus flexible entre les mots présents dans le document et la demande de l'usager doit être mis en place. Afin d'offrir une telle possibilité, nous avons généré trois corpus additionnels nommés MS3, MS7 et MSδ. Les versions MS3 et MS7 s'obtiennent de manière similaire au corpus MS1, mais en prenant les trois, respectivement les sept premiers choix pour chaque mot. En reprenant l'exemple précédent, le deuxième mot de notre verset (*man*) comprendra les trois mots *man*, *min* et *mat* dans la version MS3 et toutes les possibilités dans la version MS7. Signalons que l'usager consultant le corpus Parzival ne voit pas les alternatives possibles à chaque mot. Ces dernières sont prises en compte uniquement par les modules d'indexation et de recherche.

Pour générer le corpus MSδ, nous avons tenu compte de la vraisemblance afin d'inclure un nombre variable d'alternatives par mot. Ainsi, si les valeurs de vraisemblance des autres possibilités sont proches de la meilleure, ces alternatives seront incluses lors de l'indexation et de la recherche. Comme limite, nous reprendrons toutes les formes possédant un logarithme de la valeur de vraisemblance ayant une différence inférieure à δ % du maximum. Ce paramètre a été fixé à 1,5 % dans nos expériences. Si l'on reprend l'exemple précédent, le système sélectionnera, en plus du vocable *man*, le mot *min*. En effet, ce dernier possède une valeur de vraisemblance de 35 657, soit 1 % inférieure au maximum (36 007).

Avec le recours au corpus étendu MS3, MS7 ou MSδ, nous tenons compte d'alternatives au meilleur mot choisi lors de la reconnaissance, permettant ainsi d'inclure des variations de graphie. Par exemple, le nom propre *Parzival* apparaît dans les manuscrits sous les formes *Parcifal*, *Parcival* et *Parzifal*. Toutes ces alternatives sont admissibles et doivent être considérées comme se référant à la même personne. Comme autres exemples de variations, on peut signaler l'emploi d'un *f* à la place d'un *v* et vice-versa. Ainsi, le mot oiseau peut s'écrire *vogel* ou *fogel* tandis que le mot poisson peut apparaître comme *fisch* ou *visch*.

En plus de ces variations de graphie, nous devons tenir compte des variations dues à la morphologie. Ainsi, des flexions peuvent modifier la fin des noms, adjectifs ou verbes pour signaler le genre, le nombre, le temps ou la personne. Comme pour les langues latines ou salves, l'allemand dispose également de cas grammaticaux influençant les suffixes. Ainsi, le vocable *Parzival* peut apparaître comme *Parcivale*, *Parcivals* ou *Parcivalen*.

3.3. Génération automatique des requêtes

La création d'un jeu de requêtes avec leurs jugements de pertinence constitue une tâche manuelle non négligeable, coûteuse en temps et ressources. Comme alternative, plusieurs études ont proposé une génération automatique, tout en s'approchant de la qualité d'une création manuelle (Azzopardi & de Rijke, 2006; Callan & Connell, 2001; Jordan *et al*., 2006). En se limitant à la recherche d'un objet connu, de telles simulations présentent un intérêt indéniable. En effet, dans un tel scénario, le nombre de réponses correctes se limite souvent à l'unité. Dans notre projet, nous avons choisi ce principe d'évaluation et avons généré les requêtes selon l'approche proposée par Azzopardi & de Rijke (2006) (voir tableau 1).

Initialisation de la requête Q = {}
Sélection d'un document d_k comme bonne réponse avec la probabilité Prob[d_k]
Sélection de la taille de la requête s avec une probabilité Prob[s]
Répétition s fois {
Sélection d'un terme t_i de d_k selon un modèle de langue avec une probabilité Prob[$t_i \mid \theta_d$]
Ajout de t_i a la requête Q }
Retourner d_k et Q définissant la paire (objet connu / requête).

Tableau 1. *Algorithme de génération de requêtes et réponses selon Azzopardi & de Rijke (2006)*

Dans notre adaptation de cet algorithme, les mots brefs composés de trois lettres ou moins sont exclus de la requête. De plus, les formes entrant dans les 150 mots les plus fréquents sont aussi éliminées. Finalement, nous avons généré deux jeux de requêtes comprenant chacun 60 interrogations. Le premier jeu contient uniquement des requêtes courtes exprimées par un seul mot, (jeu noté Q1) tandis que le deuxième ensemble inclus des interrogations composées de trois termes (Q3). Dans cette communication, nos expériences seront essentiellement basées sur le jeu Q1, avec mention de résultats sur le jeu étendu Q3. Pour définir la probabilité de sélectionner un document (Prob[d_k]), nous avons choisi une distribution uniforme. Lors du choix des termes de la requête, la probabilité Prob[$t_i \mid \theta_d$] donne une chance proportionnelle à la longueur de chaque mot d'un document ; plus le mot sera long, plus grande sera sa chance d'être sélectionné.

3.4. Génération des requêtes bruitées

Ces premiers jeux de requêtes contiennent des termes extraits directement des documents pertinents. Ils ne comprennent pas de faute d'orthographe. De plus, les variations graphiques ou liées à la flexion morphologique ne sont pas incluses. Ainsi, si le terme *Parzivale* est présent dans le document pertinent, l'usager ne devrait pas se limiter à cette seule graphie, mais devrait pouvoir écrire *Parcifal* (différente graphie) ou *Parzivals* (changement de flexion).

Pour tenir compte de ces variations, nous avons repris notre jeu de requêtes courtes (Q1) afin de toujours modifier le terme recherché. Dans ce nouveau jeu, nommé QM1, la forme présente dans la requête sera toujours absente du document pertinent. Afin de générer ces alternatives possibles, par rapport à un terme donné, nous avons repris notre corpus MS7.

Dans ce cas, nous disposons de sept alternatives pour chaque mot. En reprenant notre exemple avec le mot *man*, nous disposons de six autres mots reliés à cette entrée. En parcourant tout le corpus, nous avons regroupé toutes les alternatives proposées lorsque le mot *man* était rencontré. Ensuite, toutes ces formes sont triées par ordre de fréquence d'apparition pour former l'entrée du mot *man* dans notre dictionnaire des alternatives. Pour générer le jeu QM1, nous avons substitué le mot *man* par son alternative la plus fréquente.

3.5. Mesures d'évaluation

Pour mesurer la qualité de la réponse fournie par un moteur de recherche, la campagne d'évaluation TREC-5 avait retenue la moyenne de l'inverse du rang ou MRR (*mean reciprocal rank*) (Voorhees & Garofolo, 2005; Buckley & Voorhees, 2005). Cette mesure convient bien à notre modèle d'usager, soit une personne recherchant un article connu dont il existe souvent une seule bonne réponse. Cependant, si le système de dépistage retourne le verset précédent ou le suivant, la mesure MRR le considère comme faux. Nous ne pensons pas qu'une décision aussi stricte convienne bien à notre contexte. Retrouver la ligne immédiate adjacente devrait plutôt être vue comme une réponse partiellement exacte et une certaine souplesse devrait donc être admise. Toutefois cette possibilité est ignorée par la mesure MRR que l'on peut donc analyser comme une mesure pessimiste de la vraie efficience du système de recherche perçue par l'usager.

4. Stratégies d'indexation et de recherche

En recherche d'information nous disposons de plusieurs modèles et stratégies d'indexations (Manning *et al*., 2008). Face à un corpus donné, il s'avère impossible de déterminer *a priori* quel modèle, quels paramètres et quelle stratégie d'indexation offriront les meilleures performances. Dans ce but, nous pouvons indexer les documents (et les requêtes) en considérant les mots présents, sans aucune forme de normalisation. Toutefois, certaines formes très fréquentes et peu ou pas porteuses de sens peuvent être éliminées. Dans notre corpus, nous pouvons retenir quatre formes correspond à ce critère, soit *der, daz, ir* et *er*. Comme alternative, nous pouvons également éliminer, par exemple, les 150 formes les plus fréquentes.

Afin d'améliorer l'appariement requête-document, les formes fléchies peuvent être réduites à leur racine. En effet, les variations de genre, de nombre ou les cas grammaticaux ne doivent pas empêcher un appariement sémantique. Dans ce but, nous avons conçu et implémenté un enracineur léger pour le moyen haut-allemand. Cette normalisation des formes suit les principes du S-stemmer de la langue anglaise (Harman, 1991) supprimant uniquement la marque du pluriel. Notre solution cherche à éliminer les suffixes '-e', '-en', et '-er', sous la contrainte que la partie résultante se compose d'au moins trois lettres. Comme autre solution, nous avons développé un enracineur plus agressif éliminant la majorité des suffixes flexionnels et quelques suffixes dérivationnels. Une telle approche avait été choisie par le meilleur système lors de la campagne d'évaluation TREC-5 (Ballerini *et al*., 1997).

Au lieu de considérer les mots, l'indexation peut représenter les documents et requêtes sur la base de séquences de k caractères (McNamee & Mayfield, 2004). Par exemple, en fixant $k = 4$, le vocable *ordinateur* génèrera la séquence "ordi", "rdin", "dina", … et "teur". Le recours à une telle représentation ne requiert pas la présence d'un enracineur. Les séquences finales correspondant à des suffixes étant très fréquentes, elles possèderont un poids très faible voire nulle. Comme autre possibilité, nous pouvons limiter chaque mot à ses k

premiers caractères (troncature à k). En fixant $k = 4$, le mot *ordinateur* sera représenté par la séquence "ordi".

Les termes (mots, racines, séquence de k lettres, etc.) doivent ensuite être pondérés. Comme première approche, on peut alors recourir au modèle vectoriel *tf idf* (Manning *et al*., 2008) tenant compte de la fréquence d'occurrence (tf_{ij}) du terme t_j dans le document d_i et de la fréquence documentaire (df_j). Cette dernière se retrouve dans la composante *idf* définie comme $idf_j = \log_2(n/df_j)$ avec n indiquant le nombre de documents dans le corpus.

Pour tenir compte de la longueur variable des documents, Buckley *et al*. (1996) proposent une pondération plus complexe connue sous l'acronyme *Lnu-ltu*. Dans cette notation, la pondération appliquée aux termes des documents suit la formulation *Lnu* tandis que les termes de la requête seront pondérés selon l'équation *ltu* (voir équation 1). Ce modèle présentait la meilleure performance lors de la campagne TREC-5 (Ballerini *et al*., 1997).

Dans l'équation 1, w_{ij} indique le poids attribué au terme t_j apparaissant dans le document d_i (ou la requête pour le poids w_{qj}), nt_i correspond au nombre de termes dans le document d_i, et *pivot* et *slope* sont deux constantes.

$$w_{ij} = \frac{\left(\left(\ln(tf_{ij})+1\right) \Big/ \left(\ln(\text{mean } tf)+1\right)\right)}{(1-slope)\cdot pivot + slope \cdot nt_i} \qquad w_{qj} = \frac{\left(\ln(tf_{qj})+1\right)\cdot idf_j}{(1-slope)\cdot pivot + slope \cdot nt_i} \tag{1}$$

En plus de ces solutions, basées sur le modèle vectoriel, nous avons considéré le modèle probabiliste Okapi (Robertson *et al*., 2000) basé sur la formulation suivante :

$$w_{ij} = [(k_1+1)\cdot tf_{ij}] \,/\, (\mathrm{K} + tf_{ij}) \quad \text{dans laquelle} \quad \mathrm{K} = k_1 \cdot [(1\text{-}b) + ((b \cdot l_i) \,/\, mean\ dl)] \tag{2}$$

dans laquelle l_i représente la longueur du document d_i, b (= 0,55) et k_1 (= 1,2) sont deux constantes tandis que *mean dl* indique la longueur moyenne des documents.

Comme deuxième modèle probabiliste, nous avons retenu le modèle I(n_e)B2, un des membres de la famille *Divergence from Ramdomness* (Amati & van Rijsbergen, 2002). Dans ce dernier cas, la pondération w_{ij} du terme d'indexation t_j dans le document d_i, combine deux mesures d'information, à savoir :

$$w_{ij} = \mathrm{Inf}^1_{ij} \cdot \mathrm{Inf}^2_{ij} = \mathrm{Inf}^1_{ij} \cdot (1 - \mathrm{Prob}^2_{ij}) \quad \text{avec } \mathrm{Prob}^2_{ij} = 1 - [(tc_j+1) \,/\, (df_j \cdot (tfn_{ij}+1))]$$
$$\text{et } \mathrm{Inf}^1_{ij} = tfn_{ij} \cdot \log_2[(n+1) \,/\, (n_e+0{,}5)] \quad \text{avec } n_e = n \cdot [1 - [(n\text{-}1)/n]^{tc_j}] \tag{3}$$

avec tc_i indiquant le nombre d'occurrence du terme t_j dans toute la collection.

Enfin, nous avons repris un modèle de langue (LM) dans lequel les probabilités sont estimées en se basant sur les fréquences d'occurrences dans le document d_i et dans le corpus C. Dans cet article, nous avons repris le modèle de Hiemstra (2000), décrit dans l'équation 4 combinant une estimation basée sur le document (Prob[t_j | d_i]) et sur le corpus (Prob[t_j | C]).

$$\mathrm{Prob}[d_i \mid q] = \mathrm{Prob}[d_i] \,.\, \textstyle\prod_{t_j \in Q} (\lambda_j \,.\, \mathrm{Prob}[t_j \mid d_i] + (1\text{-}\lambda_j) \,.\, \mathrm{Prob}[t_j \mid C]) \tag{4}$$
$$\mathrm{Prob}[t_j \mid d_i] = tf_{ij} \,/\, nt_i \quad \text{et} \quad \mathrm{Prob}[t_j \mid C] = df_j \,/\, lc \quad \text{avec } lc = \textstyle\sum_k df_k$$

dans laquelle λ_j est un facteur de lissage (une constante pour tous les termes t_j, et qui est fixée à 0,35) et lc indique la taille du corpus C.

5. Evaluation

5.1. Evaluation avec des requêtes sans erreur

En s'appuyant sur les connaissances actuelles en recherche d'information, nous pouvons penser qu'une représentation par mots devrait fournir une bonne performance. Comme enracineur, une approche éliminant les suffixes flexionnels des noms et adjectifs ainsi que les principaux suffixes dérivationnels devrait fournir la meilleure performance. De plus, nous avons appliqué un processus automatique de décomposition des mots composés permettant d'augmenter légèrement la qualité de la réponse. Ces conditions ont été choisies pour notre première série d'expériences décrite dans le tableau 2. Au niveau des modèles de recherche, nous avons repris deux modèles vectoriels (*tf idf* et *Lnu-ltu*), deux modèles probabilistes (Okapi et I(n_e)B2) ainsi qu'un modèle de langue (LM).

Modèles	parfait	MSδ	MS1	MS3	MS7
Okapi	0,612	0,603	0,584	0,400*	0,318*
I(n_e)B2	0,612	0,595	0,584	0,400*	0,318*
LM	0,612	0,595	0,584	0,408*	0,309*
Lnu-ltu	0,612	0,621	0,584	**0,410***	0,322*
tf idf	**0,627**	**0,637**	**0,589**	0,396*	**0,327***
Différence %		-0,8 %	-4,9 %	-34,5 %	-48,2%

Tableau 2. *Mesure de MRR avec différents corpus de même que la version sans erreur, indexation par mots et enracineur agressif (requête Q1, 60 requêtes)*

Dans le tableau 2, nous avons repris les différents corpus comprenant un nombre variable de mots sélectionnés pour chaque mot manuscrit. Nous avons également repris notre version sans erreur (performances indiquées sous la colonne "parfait"). En comparaison avec cette version sans erreur, la dernière ligne du tableau 2 indique que le corpus MSδ ou MS1 présente une différence moyenne relativement faible (de 0,8 % et 4,9 %). Par contre, l'inclusion d'un plus grand nombre de termes additionnels tend à détériorer la performance moyenne. Afin de vérifier si les différences de performances entre corpus étaient significatives, nous avons sélectionné la colonne "parfait" comme base de référence. Un astérisque indique une différence significative (test *t* bilatéral, $\alpha = 5\%$).

Le choix du corpus MSδ semble donc représenter les meilleures conditions. Afin d'analyser d'autres formes de représentation ou l'emploi d'un enracineur moins agressif, nous avons conduit une deuxième série d'expériences décrite dans le tableau 3. Comme première alternative, nous proposons d'ignorer toute forme d'enracineur (colonne "aucun"). Comme second choix, nous avons appliqué un enracineur léger (colonne "léger") éliminant uniquement les suffixes flexionnels des noms et adjectifs. Enfin, les performances obtenues par un enracineur agressif sont reprises sous la colonne "agressif". Comme alternative, nous pouvons recourir aux séquences de 4-grammes ou limiter chaque mot à ses *k* premiers caractères (troncature à 4).

Selon les performances indiquées dans le tableau 3, on constate que le modèle *tf idf* présente presque toujours la meilleure évaluation. Dans la dernière ligne de ce tableau, nous avons indiqué le pourcentage moyen de différence avec la première série d'expériences (aucune normalisation). Ces différences relatives restent faibles avec une indexation par *n*-grammes, variant de -0.73 %. Pour la troncature à 4, la différence demeure importante, -23.05% en moyenne, rendant cette approche sans intérêt dans notre contexte. Lors de l'indexation par

mots, l'emploi d'enracineur tend à réduire la performance moyenne de -8,97 % pour un enracineur léger et à -15.78 % pour une stratégie agressive.

	Indexation par mots			4-grammes	tronc-4
Modèles	aucun	léger	agressif		
Okapi	0.7166	0.6568	0.6030	0.7251	0.5591
I(n_e)B2	0.7166	0.6518	0.5950	0.7238	0.5554
LM	0.7275	0.6647	0.6210	0.7011	**0.5885**
Lnu-ltu	0.7166	0.6449	0.5951	0.7114	0.5362*
tf idf	**0.7447**	**0.6790**	**0.6365**	**0.7343**	0.5480
Différence %		-8.97%	-15.78%	-0.73%	-23.05%

Tableau 3. *Mesure de MRR avec le corpus MSδ, cinq modèles de recherche, et différentes formes d'indexation possibles avec les requêtes Q1 (60 requêtes)*

Afin de déterminer si les différences de performance peuvent être analysées comme significatives, nous avons appliqué le test bilatéral de Student (seuil de signification $\alpha = 5$ %). La meilleure performance dans chaque colonne sert de base de comparaison. Si un modèle présente une performance moyenne significativement différente, elle sera signalée par un '*'. Comme le montre le tableau 3, les différences entre le modèle *tf idf* et les autres sont souvent non-significatives avec une indexation par mots. Lors de l'indexation par *n*-grammes, le modèle *tf idf* propose la meilleure solution, bien que la différence avec d'autres modèles probabilistes ne soit pas statistiquement significative.

5.2. Evaluation avec des requêtes bruitées

En utilisant directement le jeu de requêtes bruitées QM1, la performance moyenne est quasi nulle, quelque soit l'indexation ou le modèle de recherche choisi. En effet, le terme recherché étant absent du document pertinent, l'appariement s'avère impossible. Nous devons donc prévoir d'assouplir l'appariement entre le terme présent dans la requête et ceux apparaissant dans le document visé.

Dans ce but, nous avons décidé d'étendre automatiquement la requête en lui ajoutant des termes reliés à celui donné par l'usager. Cette adjonction vise à fournir au système des variations morphologiques (*man* → *mane*, *manen*, …), graphiques (*man* → *maen)* ou à améliorer la reconnaissance incorrecte des mots (*min* → *man*). Toutefois, une expansion trop massive risque de réduire la qualité de la réponse par la présence trop abondante de termes sans intérêt par rapport au document souhaité.

Comme première approche, nous avons repris notre corpus MS7. Avec cet outil, pour chaque terme nous possédons une liste triée par probabilité lors de la reconnaissance automatique. En reprenant l'exemple du mot *man*, celui-ci apparaît dans la liste avec, entre autres, *min*, *mat*, *nam*, *arm*, *nimt*, et *gan.* Dans ce cas, nous proposons d'ajouter la première (*min*), les trois premières (*min*, *mat*, *nam*) ou toutes les alternatives possibles. Les requêtes ainsi étendues n'apportent qu'une amélioration faible de la qualité de réponse. Les documents souhaités ne sont que rarement dépistés.

Comme deuxième approche, nous élargissons la requête selon la liste des alternatives présente notre dictionnaire (voir section 3.4). Dans notre précédent exemple, nous pourrions retrouver dans notre dictionnaire des alternatives l'entrée suivante : *man* → *min*, *mat*, *nam*, *arm*, *nimt*, et *gan*. Si le terme *man* apparaît dans la requête, nous ajoutons la meilleure alternative, soit le mot *min* dans notre exemple. La qualité de la recherche s'en trouve améliorée mais sans atteindre un niveau satisfaisant.

Comme troisième approche, nous lisons notre dictionnaire des alternatives de manière inverse. Ainsi, si la requête comprend le terme *nam*, nous consultons notre dictionnaire et relevons toutes les entrées ayant ce terme parmi les trois premières alternatives. Dans notre exemple, le terme *nam* apparaît comme troisième alternative du mot *man*. Ce dernier sera inclus dans la requête élargie. Cette approche permet d'améliorer l'efficience de la recherche et les documents souhaités sont très souvent dépistés, mais sans forcément apparaître en première position.

Comme stratégie d'expansion automatique de la requête, nous avons décidé de combiner ces deux dernières approches. Les résultats obtenus avec le corpus sans erreur sont repris dans le tableau 4. Dans ce cas, en partant du jeu de requête QM1 les performances moyennes se situent au environ de 0. Après l'élargissement automatique, nous constatons que la performance moyenne s'élève à environ 0,6 avec l'indexation par mots et sans suppression automatique des suffixes. Ainsi, en moyenne, le document pertinent est retrouvé en position 1 / 0,6 = 1,7. L'emploi d'un enracineur léger ou agressif tend à réduire cette performance moyenne. L'indexation par 4-grammes n'apporte pas un niveau de qualité intéressant.

	Indexation par mots			4-grammes	tronc-4
Modèles	aucun	léger	agressif		
Okapi	0.616	0.526	0.486	0.488	0.420
I(n_e)B2	0.590	0.520	0.466	**0.526**	0.419
LM	**0.622**	**0.528**	**0.491**	0.497	**0.433**
Lnu-ltu	0.583	0.508	0.459	0.471	0.431
tf idf	0.589	0.502	0.469	0.502	0.422
Différence %		-13.92%	-20.98%	-17.20%	-29.18%

Tableau 4. *Mesure MRR avec le corpus parfait, cinq modèles de recherche, différentes indexations avec les requêtes étendues (60 requêtes).*

Finalement, dans ces conditions, l'emploi d'une approche probabiliste basée sur un modèle de langue (LM) fournit la meilleure performance, si l'on considère l'indexation par mots avec divers enracineurs. Pour l'indexation par 4-grammes, la meilleure approche serait le modèle I(n_e)B2 qui ne possède pas une performance significativement supérieure au modèle LM.

5.3. Analyse de quelques requêtes

Les requêtes comprenant un seul mot peuvent comprendre un *hapax legomenon* (*df* = 1) apparaissant uniquement dans la bonne réponse. Par exemple, la requête courte n° 4 contient le mot *machete* tiré du verset "*er machete ê daz er gein ir sp(ra)ch*" qui est aussi la réponse attendue. Dans le corpus MS1, le mot reconnu pour *machete* est *machen*, ce qui est une erreur. Dès lors, le système de dépistage (modèle Okapi) s'avère incapable de faire le lien entre la forme *machete* et le terme *machen*. Un problème similaire survient avec les requêtes courtes n° 11, 25 et 43.

En utilisant le corpus MSδ, le système de dépistage ne possède pas uniquement le terme *machen*, mais retrouve également le mot *machete* (deuxième choix lors de la reconnaissance). Comme ce dernier est inclus dans la représentation du vers, le système de dépistage (modèle Okapi) retourne en première position le passage souhaité. Avec le corpus MS3 ou MS7, le verset est aussi retourné mais au rang 3 (avec le corpus MS3) ou 18 (corpus MS7). Cette diminution de la qualité de réponse provient du fait que le mot *machete* est inclus, de manière erronée, dans d'autres passages.

6. Conclusion

Dans cette communication, nous avons étudié l'efficience d'un système de dépistage de l'information œuvrant sur un corpus de manuscrits médiévaux. Cet ensemble de textes a été numérisé, puis un système de reconnaissance des caractères a déterminé les sept mots possibles pour chaque mot écrit. Disposant également d'une version sans erreur, nous avons calculé le taux d'erreur en reconnaissance des mots qui s'élève à environ 6 %.

Les manuscrits servant de base à nos expériences ont été écrits au début du XIIIe siècle en moyen haut-allemand. Comme pour d'autres langues de cette époque, l'orthographe n'est pas normée, permettant à des graphies différentes de désigner le même objet. Afin d'autoriser des appariements entre formes distinctes, mais désignant le même objet, nous avons généré différents corpus présentant un nombre variable d'alternatives par mot présent dans le manuscrit original. Lors de l'indexation, nous avons comparé l'efficience d'un enracineur léger ou plus agressif, de même que des représentations basées sur des séquences de *k* caractères.

Comme modèle d'usager, nous avons repris la recherche d'un document connu dans laquelle le système est jugé sur sa capacité à dépister cet objet unique. En s'appuyant sur la moyenne de l'inverse du rang (MRR ou *mean reciprocal rank*), nos expériences indiquent que la différence moyenne de performance entre une version sans erreur et une version avec 6 % d'erreurs de reconnaissance peut rester faible (entre 1 et 5 %, voir tableau 2).

En présence d'un corpus de textes médiévaux, nous devons également tenir compte des erreurs de reconnaissance, des variations de graphie ou d'une morphologie flexionnelle plus complexe. Pour simuler ces variations orthographiques, nous avons généré un jeu de requêtes courtes écrites sous une forme différente que celles présentes dans le document souhaité. En combinant deux stratégies d'expansion automatique des requêtes, la mesure MRR moyenne passe de 0 à 0,6 ; le document souhaité se retrouve, en moyenne, dépisté au rang 1,7.

Remerciements

Cette recherche a été financée en partie par le Fonds national suisse pour la recherche scientifique (subside n^0 CRSI22_125220).

Références

Amati G. & van Rijsbergen C.J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems*, 20(4), pages 357-389.

Andrieux-Reix N., Croizy-Naquet C., Guyot F. & Opperman E. (2000). *Petit traité de langue française medievale*. Presses Univeristaires de France, Paris.

Azzopardi L. & de Rijke M. (2006). Automatic construction of known-item finding test beds. *Proceedings ACM SIGIR*, pages 603–604.

Ballerini J.P., Büchel M., Domering R., Knaus D., Mateev B., Mittendorf E., Schäuble P., Sheridan P. & Wechsler M. (1997). SPIDER retrieval system at TREC-5. *Proceedings of TREC-5,* NIST Publication #500-238, pages 217-228.

Buckley C., Singhal A., Mitra M. & Salton G. (1996). New retrieval approaches using SMART. *Proceedings of TREC-4*, NIST Publication #500-236, pages 25-48.

Buckley C. & Voorhees E. (2005). Retrieval system evaluation. In: Voorhees, E.M., Harman, D.K. (Eds) *TREC. Experiment and Evaluation in Information Retrieval*, The MIT Press, Cambridge (MA), pages 53-75.

Callan J. & Connell M. (2001). Query-based sampling of text databases. *Information Systems*, 19(2), pages 97–130.

Callan J., Kantor P. & Grossman D. (2002). Information retrieval and OCR: From converting content to grasping meaning. *SIGIR Forum*, 36(2), pages 58-61.

Craig H. & Whipp R. (1020). Old spellings, new methods: Automated procedures for indeterminate linguistic data. *Literary & Linguistic Computing*, 25(1), pages 37-52.

Eguchi K., Oyama K., Ishida E., Kando N. & Kuriyama K. (2003). Overview of the web retrieval task. *Proceedings Third NTCIR Workshop*, NII Publication.

Fischer A., Wüthrich M., Liwicki M., Frinken V., Bunke H., Viehhauser G. & Stolz M. (2007). Automatic transcription of handwritten medieval documents. *Proceedings 15th International Conference on Virtual Systems and Multimedia*.

Gardt A., Hauss-Zumkehr U. & Roelcke T. (1999). *Sprachgeschichte als Kulturgeschichte.* Walter de Gruyter, Berlin.

Harman D. (1991). How effective is suffixing?. *Journal of the American Society for Information Science,* 42(1), pages 7-15.

Hiemstra D. (2000). *Using Language Models for Information Retrieval*. CTIT Ph.D. Thesis.

Jordan C., Watters C. & Gao Q. (2006). Using controlled query generation to evaluate blind relevance feedback algorithms. *Proceedings of the sixth ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 286–295.

Manning C.D., Raghavan P. & Schütze H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge (UK).

McNamee P. & Mayfield J. (2004). Character n-gram tokenization for European language text retrieval. *IR Journal*, 7(1-2), pages 73-97.

Nicholas R., Toni M. & Manmatha R. (2005). Boosted decision trees for word recognition in handwritten document retrieval. *Proceedings of the ACM-SIGIR*, pages 377-383.

Peters C., Gonzalo J., Braschler M. & Kluck M. (2004). *Comparative Evaluation of Multilingual Information Access Systems*. LNCS #3237, Springer-Verlag, Berlin.

Pilz T., Luther W., Fuhr N. & Ammon U. (2006). Rule-based search in text databases with nonstandard orthography. *Literacy & Linguistic Computing*, 21(2), pages 179-186.

Rey A. Duval F. & Siouffi G. (2007). *Mille ans de langue française. Histoire d'une passion*. Perrin, Paris.

Robertson S.E., Walker S. & Beaulieu M. (2000). Experimentation as a way of life: Okapi at TREC. *Information Processing & Management*, 36(1), pages 95-108.

Sakai T. (2006). Bootstrap-based comparison of IR metrics for finding one retrieval document. *Proceedings AIRS*, LNCS #4182, Springer-Verlag, Berlin, pages 374-389.

Tagva K., Borsack J. & Condit A. (1994). Results of applying probabilistic IR to OCR text. *Proceedings of the ACM-SIGIR*, pages 202-211.

Toni M., Manmatha R. & Lavrenko V. (2004). A search engine for historical manuscript images. *Proceedings of the ACM-SIGIR*, pages 369-376.

Voorhees E.M. & Garofolo J.S. (2005). Retrieving noisy text. In: Voorhees, E.M., Harman, D.K. (Eds) *TREC. Experiment and Evaluation in Information Retrieval*, The MIT Press, Cambridge (MA), pages 183-197.