Chapitre 2

Recherche d'information et systèmes de questions-réponses

2.1. Introduction

Les moteurs de questions-réponses actuels correspondent généralement à la concaténation de plusieurs systèmes indépendants. Parmi les plus importants, on trouve les analyseurs de questions (catégorisation, extraction de focus…), les analyseurs de textes (étiqueteurs d'entités nommées, étiqueteurs morpho-syntaxiques, analyseurs syntaxiques profonds…), les moteurs de recherche documentaire (extraction d'une liste de documents ou de passages depuis le corpus de recherche) et enfin les modules d'extraction de réponses candidates. Ceci correspond à une vision séquentielle de la recherche d'informations de type questions-réponses (QR) pour laquelle, à partir de l'analyse d'une question, on va réduire le champ de recherche depuis le corpus jusqu'à la réponse en passant par un ensemble de documents, un ensemble de passages de ces documents, un ensemble de phrases extraites des passages… Ce découpage a une double origine : la préexistence de chacun des modules (évalués par l'intermédiaire de campagnes d'évaluation indépendantes : TREC, MUC…) et l'incapacité des techniques d'analyse « profonde » à opérer directement sur de grands corpus.

La vision minimaliste de la tâche questions-réponses correspond ainsi à une séquence d'opérations mises en œuvre par des logiciels éprouvés sur des textes de même nature mais dans des contextes différents. L'enjeu est alors double. Il consiste d'une part à vérifier, pour chacun des modules, que les approches les plus performantes dans un contexte autre que questions-réponses, sont les mêmes pour cette tâche et, d'autre part, à faire en sorte de diminuer l'indépendance de ces

---

Chapitre rédigé par Patrice BELLOT et Mohand BOUGHANEM.

modules afin de prendre en compte le problème dans sa globalité. Parmi tous les modules en œuvre dans questions-réponses et que l'on pourrait étudier sous cet angle, c'est celui de la recherche de documents et de passages qui est traité dans ce chapitre. Une différence essentielle entre la recherche d'informations (RI) et questions-réponses réside dans la nature de la requête : dans un cas, RI, il s'agit d'une description de l'information recherchée (qui contient bien souvent les mots des documents qui intéressent l'utilisateur, *i.e.* une forme de « réponse ») et dans l'autre cas d'une question qui ne contient justement pas les mots recherchés (la réponse). Le problème sous-jacent est alors : « *À quel point a-t-on besoin d'approches spécifiques de recherche d'informations pour questions-réponses ?* ».

Après avoir présenté les critères d'évaluation des modules de RI et certains résultats obtenus lors de campagnes de test, nous décrirons quelques-uns des principaux modèles de RI et plusieurs pré et post traitements utilisés en QR dont l'enrichissement et la réécriture de requêtes ainsi que la recherche de passages.

2.1.1. *Évaluation de la recherche documentaire dans les moteurs QR*

L'objectif de cette section est à la fois de rappeler les critères d'évaluation utilisés en recherche documentaire mais aussi de faire le lien avec questions-réponses en présentant des mesures adaptées. Il s'agit de différencier l'évaluation intrinsèque de la recherche d'informations *dans* questions-réponses (les documents trouvés contiennent-ils ou non une réponse correcte dans le contexte de la question) de l'évaluation globale (de deux moteurs de recherche documentaire, lequel permet au final de mieux répondre aux questions). Posons en premier lieu le problème du référentiel d'évaluation.

2.1.2. *Évaluation stricte* vs. *évaluation tolérante*

Si la question du référentiel sur lequel évaluer les systèmes était déjà cruciale en recherche documentaire, elle l'est d'autant plus en questions-réponses où c'est un couple qui doit être trouvé : réponse correcte et documents supports dans lesquels une expression *équivalente* de la réponse est présente et apparaît dans un contexte identique à celui de la question (exemple de réponse : *2005* ; exemple de phrase où une information *équivalente* est trouvée : « *5 ans après l'an 2000* »). Le document support permet de vérifier la capacité du système à puiser les réponses dans les documents (par opposition à une base de données) et à minimiser la chance d'obtenir par hasard la réponse. Construire un référentiel fiable en vue d'une évaluation automatique d'un système correspondrait non seulement à énumérer toutes les manières d'exprimer la réponse correcte mais également à identifier les documents de la collection qui en contiennent l'expression en contexte.

Dans la littérature, on appelle « évaluation stricte » une évaluation où le référentiel est constitué de couples (réponse correcte – généralement une expression régulière –, document support) où les deux composantes sont prises en compte. On appelle « évaluation tolérante » (*lenient*) une évaluation où les réponses du système sont uniquement comparées aux expressions régulières du référentiel sans tenir compte du document support.

Nota Bene. La plupart des articles discutant de questions-réponses en dehors du cadre strict des campagnes d'évaluation utilisent ces deux modes d'évaluation comme un pis aller à défaut d'une évaluation manuelle très coûteuse. Malheureusement, plusieurs études ont montré le peu de fiabilité que l'on pouvait accorder aussi bien aux évaluations strictes que tolérantes [LIN 05, VOO 03]…

2.1.2.1. *Mesures d'évaluation classiques en recherche documentaire*

Le souci d'évaluer les performances des systèmes de recherche d'information (SRI) date des années 50 [BER 55] et les techniques d'évaluation sont elles-mêmes évaluées entre autres dans [SAR 95]. L'évaluation a été abordée selon deux angles principaux : l'efficience et l'efficacité. L'efficience mesure la capacité d'un SRI à répondre rapidement à une requête de l'utilisateur. L'efficacité, quant à elle, mesure la capacité d'un SRI à sélectionner uniquement des documents pertinents. Nous nous intéressons à ce dernier facteur dans cette section.

L'évaluation de la qualité d'un SRI en termes de pertinence des résultats est une tâche subjective puisqu'elle dépend de référentiels – listes des bonnes réponses – construits par des individus qui s'accordent dans seulement 70 % ou 80 % des cas sur la pertinence ou non d'un document [HAR 95]. D'autre part, la notion de pertinence d'un document dépend des objectifs (compréhension d'un problème ou simple description, compléments d'information ou recherche d'une présentation générale) et du point du vue de l'utilisateur. Malgré la complexité de cette tâche, plusieurs métriques reconnues et largement utilisées en RI ont été proposées, les plus importantes sont [FLU 04] :

Précision. La précision mesure la proportion de documents pertinents dans la liste de documents retournés par le SRI.

Rappel. Le rappel rend compte de la quantité de bonnes réponses par rapport au nombre de documents pertinents dans le corpus. Autrement dit, le rappel est le taux de documents pertinents *trouvés* par rapport au nombre de documents pertinents *à trouver*. Le nombre total de documents pertinents dans une collection peut être obtenu si les documents du corpus sont connus et jugés par des individus. Pour questions-réponses, il correspond au nombre total de documents qui contiennent une réponse exacte et supportée.

Courbe rappel / précision. La précision peut être calculée à différents niveaux de rappel. Supposons par exemple qu'à une requête donnée correspondent deux documents pertinents, que le SRI propose 100 documents et que les documents pertinents sont en position 2 et 8. Pour atteindre un rappel de 50%, il faut atteindre la position 2 de la liste. À ce niveau, la précision est de 50% (1 document sur les 2 premiers est pertinent). Pour atteindre 100% de rappel, il faut aller jusqu'à la position 8 : la précision est de 25% (2 documents sur 8 sont pertinents).

La précision moyenne. Elle tient compte à la fois du rappel et de la précision. Elle correspond à la moyenne des précisions non interpolées associées aux positions – dans la liste des documents trouvés – de chaque document pertinent à trouver. Lorsqu'un document pertinent du référentiel n'est pas présent dans la liste des documents rapportés, la valeur qui lui associée est nulle. La précision moyenne est la moyenne arithmétique de toutes ces valeurs.

2.1.2.2. *Mesure de recherche documentaire spécifique à QR*

Au-delà des mesures classiquement utilisées en questions-réponses (*cf.* chapitre 1), certaines sont définies pour mesurer uniquement la phase de recherche documentaire : le *Mean Reciprocal Document Rank* (MRDR) donne une indication sur le rang moyen du premier document contenant une bonne réponse [PRA 99]. Le *Reciprocal Document Rank* (RDR) vaut 1 si le premier document contient la bonne réponse, ½ si le premier ne contient pas de bonne réponse mais le second oui *etc*. Si aucun document trouvé ne contient de bonne réponse le RDR vaut 0. Le MRDR est la moyenne de ces valeurs.

2.2. Quelques résultats obtenus lors des campagnes d'évaluation

L'une des premières questions que l'on doit poser concerne la qualité du filtrage effectué par le module de recherche documentaire : les documents retenus contiennent-ils la bonne réponse à la question posée ?

Pour la campagne d'évaluation TREC-9 [VOO 00], NIST et AT&T fournissaient pour chaque question la liste des 50 premiers documents trouvés par le moteur de recherche vectoriel *SMART* [BUC 85] dans une collection de 979 000 documents. Cela permettait aux équipes participantes de ne pas avoir à travailler sur un module de recherche documentaire pour se concentrer sur l'extraction d'information. Parmi les 693 questions posées, 193 étaient des variantes de 54 des 500 premières questions. Prager *et al*. [PRA 01] disposaient de leur propre moteur de recherche documentaire au sein de *GuruQA*[1], lui-même basé sur *Guru* [BRO 97]. Ils ont ainsi

[1] Les questions étaient posées au moteur GuruQA sous une forme lemmatisée et enrichies par le type de réponse cherchée (avec, en parallèle, un étiquetage en entités nommées des

pu comparer les résultats à partir des deux moteurs, *SMART* et *Guru*. Pour les deux systèmes, le MRDR valait 0,49 *i.e.* qu'en moyenne le premier document contenant la bonne réponse figurait en position 2. Avec *GuruQA*, une bonne réponse se trouvait parmi les 50 premiers documents pour 542 questions et parmi les 200 premiers documents pour 576 questions. Les résultats globaux sont sensiblement identiques avec *SMART* même si les documents trouvés ne sont pas toujours les mêmes : parmi les 682 documents contenant une bonne réponse et trouvés par l'un ou l'autre des systèmes (50 documents par question furent retenus), seuls 483 l'ont été par les deux systèmes simultanément. Ces différences se traduisent par des écarts de performance suivant les types de question. Il est cependant probable que ce ne soit pas tant les moteurs qui expliquent ces écarts que l'enrichissement des questions avec des informations sémantiques (type de la réponse cherchée, type des entités nommées rencontrées) qui a pu être réalisée sur les questions posées à *GuruQA* mais pas à AT&T *SMART*.

Pour la campagne d'évaluation TREC-2001 [VOO 01], NIST fournissait pour chaque question la liste des 1 000 premiers documents trouvés par le moteur de recherche PRISE sans garantie qu'ils contiennent les bonnes réponses (à comparer avec seulement 50 documents pour TREC-9 avec le moteur AT&T *SMART*). D'après l'analyse conduite par K. Litkowski [LIT 02], une bonne réponse se trouvait dans les 10 premiers documents trouvés pour 311 des 500 questions de TREC-10. Pour 26 autres questions il fallait chercher entre le 11è et le 20è document et pour 32 autres questions entre le 21è et le 50è document. Parmi les 122 questions n'ayant pas de bonne réponse dans les 50 premiers documents, 49 n'avaient pas de réponse du tout parmi les 979 000 documents de la collection. Autrement dit, *PRISE* a échoué à classer des bons documents pour 73 questions sur 500 et il n'était guère utile d'aller chercher des réponses au-delà du 10è document.

Toujours sur les données de TREC-2001, Brill *et al.* ont relevé que seulement 37 questions ont leur réponse dans au moins 25 documents différents de la collection et 138 questions dans au moins 10 documents différents [BRI 01]. Cela implique que les stratégies de résolution des questions mettant en œuvre la redondance de l'information doivent être différentes sur des collections fermées comme celle de TREC et des corpus ouverts tels que le Web.

En exploitant le moteur de recherche *Lucene* [CUT] durant la campagne QA@CLEF 2006 [GIL 06b], nous avons obtenu pour 38 questions sur 156 questions

---

documents du corpus) ainsi que par des synonymes. Le moteur retrouvait alors des passages de une, deux ou trois phrases en leur associant un score *ad-hoc* basé sur les occurrences communes avec la question. Seul le meilleur passage étant retenu pour chaque document, cela induisait directement une liste de documents pouvant être comparée avec celle trouvée par AT&T avec SMART.

factuelles, un document en rang 1 qui contenait la bonne réponse[2]. Pour ces 38 questions, le score moyen du premier document retourné par *Lucene* est de 0,94 (écart-type 0,082) alors que le score moyen sur les 156 questions est de 0,92 (écart-type 0,118). La différence entre les scores obtenus ne permet pas d'en faire un critère décisif pour la sélection de la réponse.

De manière générale, la connaissance des performances d'un système de recherche documentaire aide à déterminer le seuil au-delà duquel il est inutile — ou trop risqué — d'aller chercher des réponses dans les documents.

2.3. Les principaux modèles de recherche d'informations

Les modèles de recherche d'informations peuvent intervenir durant au moins deux des étapes d'un processus de questions-réponses : la recherche de documents pouvant contenir une réponse et la recherche de passages au sein de ces documents [CUI 05, KAS 97, TEL 03]. Un processus de recherche documentaire en texte intégral se décompose quant à lui en au moins deux parties, précédées par une série de pré-traitements (section 2.3.1) :

1– l'indexation des documents de la collection complète : l'indexation est un processus permettant d'extraire à partir d'un document les unités d'indexation (termes) ciblées (mots, couples de mots, syntagmes *etc.*). Cette étape produit pour chaque document une liste de termes caractérisant son contenu. Ces termes sont alors regroupés dans une structure appelée fichier inverse, dans laquelle chaque terme est relié à la liste des documents dans lesquels il apparaît, associé à sa poids dans chacun d'eux (fréquence d'apparition, TF.IDF, …) ;

2– la recherche proprement dite à partir d'une requête booléenne ou bien écrite en langage naturel : il s'agit de mesurer, grâce aux informations enregistrées dans l'index, la ressemblance – le score – entre chaque document et la requête. Les documents trouvés sont ordonnés en fonction de leur score et sont proposés à l'utilisateur ou, dans le cas d'un moteur de questions-réponses, au module de traitement suivant : segmenteur thématique ou, directement, extracteur de réponse.

Un modèle de RI a pour but de fournir une formalisation du processus de recherche d'informations. Il doit accomplir plusieurs rôles dont le plus important est de fournir un cadre théorique pour la modélisation de la mesure de pertinence. Plusieurs modèles ont été décrits dans de nombreux ouvrages [BAE 99]. Après une brève évocation des pré-traitements possibles du corpus, nous présentons dans cette section trois modèles courants se distinguant par leur cadre théorique.

---

[2] Ce résultat est obtenu à partir des écritures des bonnes réponses trouvées par les participants à CLEF 2006 : il est probable qu'il soit en réalité supérieur à 38 si l'on envisage des écritures non trouvées mais pourtant exactes et supportées.

2.3.1. Pré-traitements des corpus

Des prétraitements peuvent être effectués sur les documents de la collection afin de normaliser les entrées de l'index. Ils concernent à la fois le jeu de caractères (conservation ou non des accents autres marques diacritiques, de la distinction majuscules) et les mots eux-mêmes (reconnaissance de mots composés, normalisation des dates et des nombres…).

Étiquetage morpho-syntaxique et *stemming*. Une des manières d'améliorer la recherche documentaire en résolvant certaines ambiguïtés et en diminuant la taille de l'index est de reconnaître les différentes formes fléchies d'un mot. Le *stemming* est une technique qui permet de généraliser les mots suivant leurs variantes morphologiques. La plupart des *stemmers* , utilisent des règles prédéfinies de réduction des mots (pour l'anglais, élimination des suffixes en 'al' : *revival* → *reviv* ; remplacement des suffixes en 'izer' par 'ize' : *digitizer* → *digitize etc*.). L'utilisation d'un *stemmer* s'accompagne quelquefois d'une baisse de la précision à cause de réductions abusives.

Lemmatisation. La lemmatisation est une autre possibilité de réduction des mots vers une racine commune en utilisant conjointement un étiqueteur morpho-syntaxique et un dictionnaire de formes fléchies. Elle permet de réduire les formes fléchies d'un mot en leur lemme : forme au singulier pour les mots employés au pluriel et forme à l'infinitif pour les verbes conjugués *etc*. En outre, il est possible de résoudre quelques ambiguïtés sur les mots (*portes* en tant que verbe par opposition à (des) *portes*, substantif) et d'éliminer automatiquement un certain nombre de mots outils tels les articles, pronoms, déterminants *etc*. sans être obligé d'employer une liste prédéfinie. S'il a été montré que la lemmatisation permettait d'améliorer la recherche documentaire, il semblerait qu'un simple *stemmer* soit préférable pour questions-réponses car ce que l'on souhaite, lors de l'étape de recherche documentaire, est de maximiser le rappel (sans trop sacrifier la précision) alors que c'est plutôt le contraire sinon [MON 03].

2.3.2. *Le modèle booléen*

Le modèle booléen est le plus simple des modèles de RI. C'est aussi le premier qui s'est imposé dans le monde de la recherche d'informations. Il est basé sur la théorie des ensembles et l'algèbre de Boole. Le modèle booléen considère que les termes de l'index sont présents ou absents dans un document. L'impact de la formulation de la requête est ici particulièrement fort. Soit la requête correspond à une conjonction de mots-clés et la forte précision induite peut se traduire par un silence élevé, soit il s'agit d'une disjonction aboutissant à un meilleur rappel mais à une précision affaiblie qui nécessitera des filtrages ultérieurs.

Dans sa forme classique, le modèle booléen ne permet pas d'ordonner les documents trouvés en fonction de la question. Cette limitation est généralement contournée par le module suivant de la chaîne de traitement qui peut être soit un module de recherche de passages soit l'extracteur de la réponse lui-même qui va attribuer un score à chaque réponse potentielle. Une autre manière d'attribuer des scores dans le booléen consiste « à faire du booléen » à partir du « vectoriel » en appliquant les opérateurs logiques sur les listes de documents correspondant à chacun des mots-clés.

2.3.3. *Le modèle vectoriel*

Le modèle vectoriel représente les documents du corpus et les requêtes par des vecteurs de mots clés. Ces mots clés sont eux-mêmes extraits des textes lors de la phase d'indexation. Pour chaque document, un poids est attribué à chacun des mots clés qu'il contient. Dans le modèle vectoriel, le vecteur requête est représenté dans le même espace que les vecteurs documents. Le vecteur requête peut alors être comparé à chacun des vecteurs documents. Cette comparaison correspond au calcul d'une similarité (ou distance) entre les vecteurs documents et le vecteur requête. Les différentes valeurs de similarité permettent d'ordonner les documents trouvés.

Si deux documents partagent le même nombre de mots communs avec la requête, seule une pondération non binaire des mots permet de différencier (et donc d'ordonner) ces documents. D'une manière générale, la pondération permet d'établir, aussi bien dans les documents que dans la requête, un ordre d'importance entre les mots sur lequel se base le calcul de similarité. L'étude des pondérations est donc particulièrement importante et constitue une partie majeure des travaux en recherche documentaire.

Un document D (resp. une requête R) est représentée par un vecteur de la forme :

$$\vec{D} = \begin{pmatrix} m_1, w_{1,D} \\ m_2, w_{2,D} \\ \cdots \\ m_n, w_{n,D} \end{pmatrix} \qquad [2.1]$$

avec m_i le i-ème mot de la liste de tous les mots retenus et $w_{i,D}$ le poids de m_i dans le document D (resp. R). Dans le cas où les m_i n'appartient pas au document D (resp. à la requête R) : $w_{i,D} = 0$.

Les mots sont généralement supposés comme étant mutuellement indépendants. Ceci est une forte simplification puisqu'il est clair que la présence ou l'absence d'un mot dans un texte dépend des autres mots qui constituent ce texte.

2.3.3.1. *Mesures de similarité*

Les similarités sont calculées en tenant compte des poids des mots dans les documents et dans la requête. Lorsque aucune technique de regroupement de mots ou d'enrichissement n'est utilisée, la valeur de similarité tient uniquement compte des mots communs à la requête R et au document D :

$$s\left(\vec{R}, \vec{D}\right) = \sum_{m_i \in R \cap D} w_{i,R} \cdot w_{i,D} \qquad [2.3]$$

Ainsi, un document qui n'a pas un seul mot commun avec la requête ne peut pas être trouvé par le système de recherche alors qu'il peut malgré tout contenir la réponse recherchée.

2.3.3.2. *Pondération des mots : les critères tf et idf*

Lorsque les poids w valent systématiquement 1 pour les mots contenus dans D ou R, la mesure de similarité (formule 2.3) revient à dénombrer le nombre de mots communs à la requête et au document. Une pondération plus fine permet de distinguer, dans un document et dans une requête, les mots importants de ceux qui le sont moins. Dans une optique de recherche documentaire, un mot important est un mot qui permet de fortement caractériser un document vis à vis des requêtes pouvant être posées par l'utilisateur et par rapport à l'ensemble des documents du corpus cible. Autrement dit, un mot est important si sa sélection prioritaire par rapport aux autres mots candidats permet d'améliorer les résultats de la recherche.

Globalement, favoriser les mots qui apparaissent souvent dans les documents revient à augmenter le nombre de documents trouvés et donc à favoriser le rappel. Inversement, favoriser les mots qui apparaissent rarement, revient à sélectionner des documents spécifiques et a donc tendance à améliorer la précision.

Si les mots les plus importants ne sont pas ceux qui apparaissent dans un grand nombre de documents, ce ne sont pas non plus seulement ceux qui apparaissent dans un faible nombre de documents (adopter cette solution pour seul critère reviendrait à privilégier fortement les fautes de frappe ou les orthographes rares – transcriptions de noms propres non courantes par exemple –). Il est raisonnable de penser que plus un mot apparaît dans un document, plus le sens de ce mot influe sur la thématique du document. Ceci est essentiellement valable pour certaines catégories de mots (noms propres, substantifs…). On peut alors entrevoir la possibilité de pondérer

différemment les mots en fonction de leur type : en fait, cette distinction est implicite pour la plupart des pondérations du fait de fréquences d'apparition différentes des catégories morpho-syntaxiques.

Finalement, pour un document donné, un mot important est un mot qui est fréquent dans ce document mais qui ne se retrouve que dans un faible nombre de documents dans le corpus. Cette définition de l'importance d'un mot correspond à introduire les facteurs TF (*term frequency*) et IDF (*inverse document frequency*) exprimant respectivement le nombre d'occurrences d'un mot donné dans un document et le nombre de documents qui contiennent ce mot dans le corpus. Ces critères sont généralement combinés [SAL 75] selon une expression du type :

$$w_i = \mathrm{tf}(i).\mathrm{idf}(i) \qquad [2.4]$$

Normalisation des poids. Si la pondération w tient compte du critère TF, une mesure de similarité telle que le produit scalaire (formule 2.4) a tendance à favoriser les documents longs par rapport aux documents courts puisque les mots ont plus de chances d'y apparaître souvent. La normalisation des poids en fonction de la taille des vecteurs permet de minimiser l'avantage des documents longs. Parmi les deux normalisations les plus employées, citons:

$$w_{i,D} = \frac{w_{i,D}}{\sum_j w_{j,D}} \text{ et } w_{i,D} = \frac{w_{i,D}}{\sqrt{\sum_j w_{j,D}^2}} \qquad [2.5]$$

Lorsque la norme euclidienne est choisie pour normaliser les poids, le calcul de la similarité (2.4) se ramène à celui du cosinus :

$$\begin{aligned} \mathrm{s}\left(\vec{R},\vec{D}\right) &= \sum_{m_i \in R \cap D} \frac{w_{i,R}}{\sqrt{\sum_j w_{j,R}^2}} \cdot \frac{w_{i,D}}{\sqrt{\sum_j w_{j,D}^2}} \\ &= \frac{\sum_{m_i \in R \cap D} w_{i,R} \cdot w_{i,D}}{\sqrt{\sum_j w_{j,R}^2 \cdot \sum_j w_{j,D}^2}} = \frac{\vec{R} \cdot \vec{D}}{\left\|\vec{R}\right\|_2 \cdot \left\|\vec{D}\right\|_2} = \cos\left(\vec{R},\vec{D}\right) \end{aligned} \qquad [2.6]$$

Notons toutefois que comme cela a été montré dans [SIN 96], le cosinus a tendance, dans la pratique, à privilégier les documents courts par rapport aux documents longs. Afin de mieux prendre en compte le critère « taille des

documents », plusieurs autres mesures de pondération ont vu le jour, grâce notamment aux campagnes TREC. Les deux plus importantes sont celle proposée dans le système *Okapi* (voir formule 2.20 décrite dans la section 2.3.3) et celle proposée dans [SIN 96], donnée comme suit :

$$W_{ij} = \frac{\left(1 + \log\left(tf_{ij}\right) \middle/ 1 + Pivot\right)}{\left(1 - Slope\right) * Pivot + Slope * dl} \qquad [2.7]$$

avec Slope = 0,2 et Pivot = 150 des constantes fixées.

Pondérations différentes pour les requêtes et les documents. Parmi les très nombreuses pondérations proposées, l'une des plus classiques et des plus performantes en moyenne est la suivante :

$$w_{m_i d} = \frac{tf\left(m_i, d\right) \cdot \log \frac{N}{n(m_i)}}{\sqrt{\sum_{m_j \in d} \left(tf\left(m_j, d\right) \cdot \log \frac{N}{n(m_j)} \right)^2}} \qquad [2.8]$$

$$\text{et } w_{m_i q} = \left(0,5 + 0,5 \frac{tf\left(m_i, q\right)}{\max\limits_{m_j \in q} tf\left(m_j, q\right)} \right) \cdot \log \frac{N}{n(m_i)} \qquad [2.9]$$

Prise en compte de la négation. Par opposition aux approches booléennes, les approches numériques au sens large ne considèrent les mots de la requête que d'un point de vue « positif ». La plupart des systèmes numériques ne prennent pas correctement en compte les requêtes qui incluent une description de ce qui n'est *pas* recherché. Il serait particulièrement intéressant d'introduire dans la détermination des poids un domaine de valeurs exprimant un point de vue « négatif » nuancé. Ainsi, certains mots de la requête permettraient de rejeter plus ou moins fortement certains documents ou segments de documents mais cette possibilité a été à notre connaissance fort peu approfondie. Dans les systèmes QR, la négation est généralement prise en charge par le dernier module : l'extracteur de réponse.

2.3.4. *Le modèle probabiliste*

Le modèle probabiliste [ROB 76] permet de représenter le processus de recherche comme un processus de décision : le coût, pour l'utilisateur, associé à la récupération d'un document doit être minimisé. Autrement dit, un document n'est proposé à l'utilisateur que si le coût associé à cette proposition est inférieur à celui de ne pas le retrouver. La règle de décision équivaut à proposer un document à l'utilisateur seulement si le rapport entre la probabilité qu'il soit pertinent et celle qu'il ne le soit pas est supérieur à un seuil donné. Une autre manière de voir le modèle probabiliste est de considérer que celui-ci cherche à modéliser l'ensemble des documents pertinents, autrement dit à estimer la probabilité qu'un mot donné apparaisse ou n'apparaisse pas dans de tels documents. La solution à ce problème pourrait passer par un processus itératif durant lequel l'utilisateur sélectionne des documents qu'il juge pertinents parmi ceux qui lui sont proposés.

Soit q une requête et d_j un document. Le modèle probabiliste tente d'estimer la probabilité que l'utilisateur trouve intéressant le document d_j sachant la requête q. On suppose qu'il existe alors l'ensemble R des documents pertinents. Soit $\overline{R}$ le complément de R. Le modèle attribue à chaque document d_j un score comme étant le rapport entre la probabilité de pertinence du document et sa probabilité de non pertinence. Cette quantité ne pouvant être calculée qu'à la condition de savoir définir *a priori* la pertinence d'un document en fonction de q (ce que l'on ne sait faire), il est nécessaire de la déterminer à partir d'exemples de documents pertinents. En appliquant la règle de Bayes :

$$P\left(R\middle|d_j\right) = \frac{P(R) \times P\left(d_j|R\right)}{P\left(d_j\right)} \qquad [2.10]$$

le score vaut :

$$s\left(d_j, q\right) = \frac{P\left(d_j|R\right) \times P(R)}{P\left(d_j\middle|\overline{R}\right) \times P\left(\overline{R}\right)} \approx \frac{P\left(d_j|R\right)}{P\left(d_j\middle|\overline{R}\right)} \qquad [2.11]$$

$P(d_j|R)$ correspond à la probabilité de sélectionner aléatoirement d_j dans l'ensemble des documents pertinents et $P(R)$ la probabilité qu'un document choisi aléatoirement dans la collection est pertinent. $P(R)$ et $P(\overline{R})$ étant indépendants de q, leur calcul n'est pas nécessaire pour ordonner les scores. En faisant l'hypothèse que les mots apparaissent indépendamment les uns des autres dans les textes (hypothèse naturellement fausse... mais réaliste à l'usage), les probabilités se réduisent à celles des sacs de mots :

$$P\left(d_j|R\right) = \prod_{i=1}^{n} P\left(d_{j,i}|R\right) \qquad [2.12]$$

avec $d_{j,i}$ une variable aléatoire désignant la présence ou l'absence du mot m_i dans le document d_j. En supposant que ces variables sont indépendantes, alors la probabilité de sélectionner aléatoirement d_j dans l'ensemble des documents pertinents est égal au produit des probabilités d'appartenance des mots de d_j dans un document de R (choisi aléatoirement) et des probabilités de non appartenance à un document de R des mots non présents dans d_j:

$$s\left(d_j,q\right) \approx \frac{\left(\prod_{m_i \in d_j} P\left(m_i,R\right)\right) \times \left(\prod_{m_i \notin d_j} P\left(\overline{m}_i,R\right)\right)}{\left(\prod_{m_i \in d_j} P\left(m_i,\overline{R}\right)\right) \times \left(\prod_{m_i \notin d_j} P\left(\overline{m}_i,\overline{R}\right)\right)} \qquad [2.13]$$

avec $P(m_i,R)$ la probabilité que le mot m_i soit présent dans un document sélectionné aléatoirement dans R et $P(\overline{m}_i,R)$ la probabilité que le mot m_i ne soit pas présent dans un document sélectionné aléatoirement dans R. Cette équation peut être coupée en deux parties suivant que le mot appartient ou non au document d_j. Soit $p_i = P(m_i \in d_j \mid R)$ et $q_i = P(m_i \in d_j \mid \overline{R})$.

Ceci implique que $1 - p_i = P(m_i \notin d_j \mid R)$ et que $1 - q_i = P(m_i \notin d_j \mid \overline{R})$. Il est enfin généralement supposé que pour les mots n'apparaissant pas dans la requête : $p_i = q_i$. Ainsi :

$$\begin{aligned} s\left(d_j,q\right) &\approx \prod_{m_i \in d_j} \frac{p_i}{q_i} \times \prod_{m_i \notin d_j} \frac{1-p_i}{1-q_i} \\ &\approx \prod_{m_i \in d_j \cap q} \frac{p_i}{q_i} \times \prod_{m_i \notin d_j, m_i \in q} \frac{1-p_i}{1-q_i} \\ &= \prod_{m_i \in d_j \cap q} \frac{p_i(1-q_i)}{q_i(1-p_i)} \times \prod_{m_i \in q} \frac{1-p_i}{1-q_i} \end{aligned} \qquad [2.14]$$

Le deuxième terme de ce produit étant indépendant de d_j, il peut être ignoré pour ordonner les scores de chaque document. En passant au logarithme, on obtient :

$$s\left(d_j, q\right) \approx \sum_{m_i \in d_j \cap q} \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \qquad [2.15]$$

Estimation des paramètres. Lors de la première itération, aucun document pertinent n'a encore été trouvé, il est nécessaire de poser les valeurs de $P(m_i, R)$ et de $P(m_i, \overline{R})$. On suppose ainsi qu'il y a une chance sur deux qu'un mot quelconque de l'index soit présent dans un document pertinent et que la probabilité qu'un mot soit présent dans un document non pertinent est proportionnelle à sa distribution dans la collection (étant donné que le nombre de documents non pertinents est généralement bien plus grand que celui des pertinents) :

$$P\left(m_i \middle| R\right) = 0,5 \text{ et } P\left(m_i \middle| \overline{R}\right) = n_i / N \qquad [2.16]$$

avec n_i le nombre de documents qui contiennent m_i dans la collection et N le nombre total de documents de la collection. Ces valeurs doivent être estimées lors de chaque itération en fonction des documents qu'elles permettent de trouver (et, éventuellement de la sélection de ceux qui sont pertinents par l'utilisateur). De nombreuses méthodes d'apprentissage ont été proposées, à partir d'approches bayésiennes, de modèle 2-Poisson ou bien de mixture de n distributions de Poisson. De manière simplifiée, disons que ces travaux ont permis d'intégrer les poids des mots dans les documents et dans la requête dans le calcul du score :

$$s\left(d_j, q\right) \approx \sum_{m_i \in d_j \cap q} w_{m_i, d_j} \cdot w_{m_i, q} \cdot \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \qquad [2.17]$$

qui, lorsqu'on n'a pas de données sur l'ensemble des documents pertinents, se réduit en un classique produit scalaire :

$$s\left(d_j, q\right) \approx \sum_{m_i \in d_j \cap q} w_{m_i, d_j} \cdot w_{m_i, q} \qquad [2.18]$$

Les pondérations *Okapi* (le SRI supportant le modèle probabiliste). Lors des différentes campagnes d'évaluation TREC, de nombreuses pondérations ont été testées au sein des systèmes *Okapi* [ROB 94] et *Inquery* [ALL 96] dont BM25. C'est également ce type de pondération qui est utilisé par de nombreux systèmes de questions-réponses [BRI 01]. Soit N le nombre de documents dans la collection, $n(m_i)$ le nombre de documents contenant le mot m_i, R le nombre de documents connus comme étant pertinents pour la requête q, $r(m_i)$ le nombre de documents de R contenant le mot m_i, $tf(m_i, d_j)$ le nombre d'occurrences de m_i dans d_j, $tf(m_i, q)$ le nombre d'occurrences de m_i dans q, $l(d_j)$ la taille en nombre de mots de d_j, $\overline{l}$ la taille

moyenne des documents du corpus et k_i et b des paramètres dépendants de la requête et du corpus. Le poids d'un mot est défini par :

$$w(m_i) = \log \frac{(r(m_i) + 0,5)/(R - r(m_i) + 0,5)}{(n(m_i) - r(m_i) + 0,5)/(N - n(m_i) - R + r(m_i) + 0,5)} \quad [2.19]$$

qui se réduit à $w(m_i) = \log\left(\frac{N - n(m_i) + 0,5}{n(m_i) + 0,5}\right)$ lorsque l'on ne se trouve pas dans un processus itératif d'apprentissage en interaction avec l'utilisateur et lorsque l'on n'a pas connaissance de l'ensemble des documents pertinents.

$$s(d_j, q) \approx \sum_{m_i \in d_j \cap q} \left(w_{m_i} \times \frac{(k_1 + 1) \cdot tf(m_i, d_j)}{K + tf(m_i, d_j)} \times \frac{(k_3 + 1) \cdot tf(m_i, q)}{k_3 + tf(m_i, q)} \right) \quad [2.20]$$

$$\text{avec } K = k_1 \cdot \left((1 - b) + b \cdot \frac{l(d_j)}{\bar{l}} \right) \quad [2.21]$$

et k_1 , k_3 et b des constantes (valeurs utilisées dans les différentes campagnes TREC : $k_1 = 2$, $k_3 = 8$, $b = 0,75$), $l(d_j)$ la longueur du document d_j, calculée en nombre de mots ou en taille de fichier en octets et $\bar{l}$ la longueur moyenne des documents.

De nombreuses variantes. Certains auteurs ont suggéré de pondérer les mots en fonction de l'écart de leur distribution observée en corpus par rapport à des distributions aléatoires dont le principal avantage est de ne pas nécessiter de paramètres dont les valeurs doivent être apprises sur les données [AMA 02]. Cela a conduit à la génération de plusieurs modèles dont certains ont obtenu des résultats meilleurs que BM25 sur les collections TREC *ad-hoc*. Il a également été proposé de tenir compte d'un plus grand nombre de caractéristiques que celles énoncées précédemment, par exemple la variance des occurrences dans la collection [GRE 02]. L'impact de ces propositions sur QR reste à mesurer.

2.3.5. *Quelques autres modèles*

La complexité de la tâche de modélisation de la pertinence a conduit la communauté RI à s'attaquer à cette problématique selon différents angles. Chaque angle tente de trouver le cadre théorique le plus adéquat pour de mieux modéliser cette notion ou encore pour étendre les modèles existants.

Une des extensions que l'on peut citer est le modèle booléen étendu [SAL 83], combinaison du modèle booléen et du modèle vectoriel. Son objectif est de prendre en compte les poids des mots (autres que 0 et 1) dans le calcul de la pertinence d'un document vis à vis d'une requête écrite sous forme d'une expression booléenne. Une autre extension du modèle vectoriel est le modèle LSI (*Latent semantic Indexing*) [DEE 90]. Le but de LSI est de corriger les défauts du modèle vectoriel liés à la non prise en compte des variations linguistiques (principalement synonymie) des mots. Ce modèle utilise les techniques de l'analyse en composante principale sur l'espace des mots afin de le ramener à un espace de concepts. Pour ce faire, LSI exploite les corrélations (co-occurrences) entre les mots afin de regrouper ceux qui sont susceptibles de représenter un même concept dans une même classe. On obtient ainsi une représentation conceptuelle des documents, ce qui limite l'impact de la variation dans l'utilisation des mots dans les documents. LSI s'est révélé performant dans de nombreuses applications dont le résumé orienté requête [FAV 06].

Dans la lignée de nouveaux modèles de RI, l'un des modèles clé qui a vu le jour dans la fin des années 90 est le modèle de langue introduit par Ponte [PON 98]. Ce modèle probabiliste est basé sur les modèles de langages répandus en linguistique. Ces modèles tentent de capter les régularités d'une langue (succession possible/probable de mots *etc.*) en observant des phrases (mots) dans un corpus d'entraînement. Un modèle de langage estime en fait la probabilité d'avoir une séquence donnée de mots dans un langage donné. Son utilisation en RI consiste à considérer que tout document est représenté par son modèle de langage. La pertinence d'un document vis-à-vis d'une requête est traduite par la probabilité que la requête puisse être générée à partir du modèle de langage du document [BOU 04]. Intuitivement ceci revient à mesurer si la requête et le document proviennent de la même source modélisée par un modèle de langage.

2.4. L'enrichissement et la réécriture de requêtes

Rocchio [ROC 71] a publié le premier travail majeur sur l'enrichissement automatique de requêtes afin de palier les phénomènes de synonymie et de polysémie. L'idée centrale était d'aller puiser dans les documents que l'on sait être pertinents les variantes des mots de la requête et dans les documents non pertinents les mots qu'il ne faut au contraire pas trouver. Ce processus « requête, enrichissement, nouvelle requête » peut être itéré.

Dans le modèle vectoriel, ce processus correspond à une modification du vecteur requête. Soient R_0 la requête initiale, d le nombre de documents pertinents, n le nombre de documents non pertinents, D_i un document pertinent, N_i un document non pertinent. La requête enrichie R_1 se calcule de la manière suivante (les facteurs β et γ sont généralement choisis proportionnellement à $1/d$ et à $1/n$ respectivement) :

$$\overrightarrow{R_1} = \alpha\overrightarrow{R_0} + \beta\sum_{i=1}^{d}\overrightarrow{D_i} - \gamma\sum_{j=1}^{n}\overrightarrow{N_j} \qquad [2.22]$$

La formulation de Rocchio s'adapte également au modèle probabiliste (section 2.3.4). Selon la définition initiale du modèle probabiliste (formule 2.15), il est possible d'obtenir un premier jeu de documents qui seront examinés par l'utilisateur selon les paramètres initiaux de la formule 2.16.

Après sélection d'un ensemble de documents pertinents et d'un autre de documents non pertinents par l'utilisateur, ces probabilités deviennent :

$$P(k_i|D) = \frac{|D_i|}{|D|} \;;\; P(k_i|\overline{D}) = \frac{n_i - |D_i|}{N - |D|} \qquad [2.23]$$

avec $|D_i|$ le nombre de documents pertinents contenant k_i et $|D|$ le nombre de documents pertinents.

Pour éviter certains problèmes avec des valeurs faibles de $|D_i|$ et de $|D|$, Yu *et al.* [YU 83] ont proposé les définitions suivantes :

$$P(k_i|D) = \frac{|D_i| + \frac{n_i}{N}}{|D| + 1} \;;\; P(k_i|\overline{D}) = \frac{n_i - |D_i| + \frac{n_i}{N}}{N - |D| + 1} \qquad [2.24]$$

Dans le cas de plusieurs itérations, et contrairement au modèle vectoriel, il n'est pas tenu compte des probabilités des itérations précédentes. En outre, les fréquences d'apparition des mots ne sont pas considérées et il n'est pas possible d'ajouter de nouveaux mots à la requête. Pour toutes ces raisons l'enrichissement fonctionne généralement moins bien dans le modèle probabiliste que dans le modèle vectoriel [BAE 1999]. Croft [CRO 83] a toutefois proposé une extension du modèle probabiliste qui tient compte des fréquences des mots dans les documents. Selon ce modèle, la similarité entre un document et une requête est :

$$\mathrm{sim}(d_j, q) = \sum_{i=1}^{t} w_{i,q}.w_{i,d_j}\left(\log\frac{P(k_i|D)}{1-P(k_i|D)} + \log\frac{1-P(k_i|\overline{D})}{P(k_i|\overline{D})} + C\right)\left(K + (1-K)\frac{f_{i,j}}{\max(f_{i,j})}\right) \qquad [2.25]$$

avec C et K deux constantes et $f_{i,j}$ le nombre d'occurrences de k_i dans d_j.

Enrichissement automatique. Les moteurs de recherche ont tendance à ramener plus de documents pertinents en tête de liste qu'en queue. En fonction de ce constat, il est possible d'effectuer un enrichissement automatique en « pariant » que les x premiers documents sont des documents pertinents et en procédant ensuite

exactement comme pour l'enrichissement interactif (la rétroaction négative en moins). En ce qui concerne l'extraction automatique de mots proches de ceux de la requête, on peut l'étendre à l'ensemble des documents trouvés (voire de la collection entière) et non plus uniquement aux documents sélectionnés par l'utilisateur. C. Monz a montré [MON 03] qu'un tel processus dégradait les performances de son système de questions-réponses (avec recherche documentaire vectorielle) alors qu'il était efficace sur la seule tâche recherche documentaire *ad-hoc* de TREC-8. La solution proposée (mais non testée) est d'employer une analyse locale pour l'enrichissement tel que décrit dans [XU 96].

Enrichissement à partir de ressources externes. Il peut s'agir tout d'abord d'ajouter dans la requête les différentes écritures possibles des noms propres et des acronymes, de corriger (ajouter ?) certaines erreurs d'orthographe ou tout simplement d'ajouter/remplacer les caractères non standards (accents, majuscules). Ces traitements, parfois effectués sur les corpus eux-mêmes durant l'indexation, ne sont pas sans danger : ajout d'ambiguïtés, modification du sens de la requête, perte de précision de la recherche (éventuellement au profit d'un meilleur rappel). De nombreux auteurs ont proposé d'enrichir les requêtes à l'aide des informations sémantiques de WordNet [MIL 90] parmi lesquelles les synonymes. Il n'a été malheureusement été constaté d'amélioration des résultats que dans des cas restreints [BAZ 05]. Il est également montré qu'une amélioration par hyponymie améliore plus nettement les résultats que par synonymie. Une autre piste consiste à enrichir la requête avec des informations que l'on s'attend à trouver dans la réponse : par exemple les unités pour des questions appelant des réponses numériques (ajout de *km* par exemple pour une question portant sur une distance).

Réécriture de requêtes. Dans le cadre de l'exploitation des moteurs de recherche sur le Web, Radev *et al*. [RAD 01] ont cherché à déterminer automatiquement quelle est la meilleure écriture d'une question pour chaque moteur du Web visé. Leur approche s'appuie sur les modèles de traduction qui, dans ce cadre, « traduisent » une question en une requête à l'aide d'opérateurs simples (insertion, suppression, traduction, ajout de guillemets pour les expressions figées, d'opérateurs rendant le mot nécessaire ou bien au contraire indésirable) et d'une phase d'apprentissage automatique à partir de couples questions / réponses *ad-hoc*.

Pour la tâche questions-réponses, C. Mons [MON 03] a proposé plusieurs stratégies pour transformer la question d'origine en « requête » pondérée qui peut être vue comme une réécriture : l'idée est de profiter de variantes des questions et de l'analyse des performances de chacune d'elles pour estimer le poids des mots à utiliser. La méthode d'apprentissage est basée sur les arbres de décision de type M5' qui sont présents dans l'environnement WEKA[3]. Si les résultats obtenus sont

[3] http://www.cs.waikato.ac.nz/ml/weka/

intéressants, le problème de la généralisation à des questions ouvertes reste posé (données d'apprentissage en quantité insuffisante).

Une autre stratégie de recherche consiste à transformer la question en une expression de ce que pourrait être la réponse cherchée (patrons de réécriture). Cette stratégie a été suivie par Brill *et al.* [BRI 01] lors de la campagne TREC-2001 et dans un certain sens par Echihabi et Marcu [ECH 03] qui établissent un modèle de transformation des phrases trouvées vers les questions et, ainsi, estiment une mesure de ressemblance (ils utilisent à cet effet le modèle canal-bruité déjà utilisé en traduction automatique ou en reconnaissance de la parole).

2.5. La recherche de questions similaires

La recherche de questions similaires et, plus largement, la gestion de l'historique d'utilisation, constitue une piste de recherche intéressante. Durant la campagne TREC-9, le jeu de questions de test contenait 193 variantes de 54 questions. Il peut s'agir de variantes plus ou moins éloignées comme par exemple « *What is the moral status of human cloning ?* » et « *What are the ethical issues for human cloning ?* » mais aussi de reformulations telles que « *What attracts tourists in Reims ?* » et « *What are tourist attractions in Reims ?* ». Malheureusement, peu de travaux spécifiques ont pris cette caractéristique en compte.

Tomuro & Litinen [TOM 04] ont exploité conjointement plusieurs critères pour mesurer l'appariement d'une question avec une autre : nombre de mots communs et cosinus, distances entre les mots par rapport aux *synsets* de WordNet et identification du type de réponse cherchée [TOM 04]. Dans une même optique, ils ont défini des patrons de réécriture permettant de « normaliser » les questions par extraction de leur forme canonique grâce à l'usage d'un analyseur syntaxique : passage de la voie passive à la voie active, réécritures de formes interrogatives similaires par l'usage d'un pronom interrogatif unique… [LYT 02, TOM 03]. Durant la campagne TREC-9, Harabagiu *et al.* [HAR 01] ont exploité efficacement une mesure de similarité entre questions à partir des couples (mot, étiquette morpho-syntaxique du mot) mais, pour être réellement opérantes, de telles approches nécessitent probablement une analyse en profondeur des questions.

2.6. La recherche de passages

Après avoir évoqué des traitements en amont de la recherche de documents, nous nous penchons sur la recherche de passage qui est une étape importante puisqu'elle permet non seulement de réduire le champ d'extraction de la réponse à certaines

parties des documents trouvés mais également de justifier aux yeux de l'utilisateur la réponse par un extrait de document [LIN 03].

Dans sa forme la plus simple, la recherche de passages correspond à l'extraction d'un certain nombre de phrases contiguës formant un bloc ayant une similarité particulièrement élevée avec la question. La similarité entre une question et un passage peut se calculer de la même manière que pour un document mais la plus petite taille du texte rend les mesures classiques moins performantes. Cela dit, il est intéressant de relever, à la vue des résultats de la campagne TREC-8, que la recherche de passages suffit à elle seule pour une tâche de questions réponses dans laquelle l'utilisateur ne demande pas une réponse précise mais se contente d'un extrait de texte de 250 caractères [SIN 00, VOO 05].

2.6.1. *Comparaison de différentes approches*

Tellex *et al*. [TEL 03] ont comparé huit méthodes de segmentation appliquées à la tâche questions-réponses sur les données de la campagne TREC-10 :

– méthode 1 : classe les passages candidats en fonction du nombre de mots ou bien de *stems* qu'ils ont en commun avec la question ;

– méthode 2 : calcule une similarité de type *Okapi*/BM25 entre la question et les passages (fenêtre glissante sur les documents) ;

– méthode 3 : emploie une variante de celle employée dans le système MultiText [CLA 00]. Elle calcule des scores de densité en favorisant les passages courts contenant des mots rares dans les documents (*IDF* élevées). Les passages retenus doivent commencer et se terminer par des mots de la question ;

– méthode 4 : emploie une méthode proposée dans [ITT 01] qui associe à chaque passage potentiel une combinaison linéaire de scores calculés en fonction des poids (*IDF*) des mots communs avec la question ou dont un synonyme est trouvé dans WordNet mais aussi des poids des mots qui sont dans la question mais pas dans le passage, du nombre de mots communs et du nombre de mots communs adjacents à la fois dans le passage et dans la question ;

– méthode 5 : inspirée par le système *SiteQ* [LEE 01], elle combine le score de plusieurs phrases adjacentes afin de trouver le meilleur passage (la longueur optimale trouvée est de trois phrases) ;

– méthode 6 : emploie la méthode du système de l'Université d'Alicante [VIC 01] qui calcule le cosinus entre la question et les passages candidats (les chercheurs d'Alicante trouvent une longueur de passage optimale de 20 phrases alors que Tellex *et al.* aboutissent à 6 phrases et nous à 3 sur CLEF 2006) ;

– méthode 7 : emploie la méthode de l'Université de Sud Californie [HOV 01] qui ordonne les phrases en différenciant les noms propres des autres mots dans le

calcul des scores mais aussi les mots trouvés en commun avec la requête (graphie identique) des mots dont seuls les *stems* sont en commun ;

– méthode 8 : correspond à la fusion des résultats obtenus par les méthodes 4, 5 et 7 en fonction des rangs de chaque passage retenu et du nombre de passages extraits de chaque document par chacune des trois méthodes initiales.

Les expériences effectuées à partir de ces huit méthodes ont été réalisées d'une part avec le moteur *Prise* (les requêtes étaient les questions telles quelles), d'autre part avec le moteur *Lucene* [CUT] en mode booléen (conjonction des mots des questions après élimination des mots outils) et enfin à partir d'un oracle ne retenant que les documents connus comme contenant une réponse correcte aux questions. Chaque méthode pouvait retourner jusqu'à vingt passages par question à partir des 200 premiers documents trouvés (seul le meilleur passage d'un document a été retenu). Pour une évaluation stricte (section 2.1.1.1.), avec *Prise*, le MRR varie entre 0,189 (méthode 1) ou 0,242 (méthode 1 avec *stems*) et 0,358 (méthode 5) et le pourcentage de questions auxquelles le système n'a pas su répondre correctement entre 52 % (méthode 1) ou 58,6 % (méthode 1 avec *stems*) et 39,6 % (méthode 4). Avec *Lucene*, le MRR varie entre 0,271 (méthode 1) ou 0,25 (méthode 1 avec *stems*) et 0,354 (méthode 3) et le pourcentage de questions auxquelles le système n'a pas su répondre correctement entre 49,4 % (méthode 1) ou 52,6 % (méthode 1 avec *stems*) et 48 % (méthode 5). Les différences entre les résultats obtenus par chacune des méthodes en employant *Prise* se sont montrées statistiquement significatives contrairement à celles observées en employant *Lucene*. Les conclusions de Tellex *et al.* sont que l'emploi de méthodes de recherche de passages (plutôt que de recherche de documents seul) avec *Lucene* permet de mieux identifier les bonnes réponses (amélioration de la précision, hausse du MRR) tandis qu'il permet de répondre à plus de questions avec *Prise* (amélioration du rappel). Ils remarquent également que la méthode 4 n'est pas sensible au moteur de recherche documentaire employé en amont de la recherche de passages contrairement à la méthode 1.

Au final, les scores obtenus à partir de *Lucene* soulignent qu'une approche booléenne obtient d'aussi bonnes performances qu'une approche numérique avec *Prise* et que la prise en compte de la densité d'occurrences des mots de la question dans les passages (méthode 4) et de la différenciation des mots suivant leur nature syntaxique (méthode 7) conduisent à des améliorations significatives.

Ordonnancement des phrases. Au-delà de la définition de scores permettant de sélectionner des passages de documents, quelques auteurs ont proposé des mesures pour travailler sur les phrases elles-mêmes. Parmi eux, Radev *et al.* [RAD 02] calculent, pour chaque phrase des documents trouvés, une combinaison linéaire des poids de chaque unigramme, bigramme et trigramme de la question dans la phrase. Ce modèle a donné de meilleurs résultats qu'une variante de la mesure *Okapi*

(formule 2.20) appliquée aux documents sur TREC-8. Pour 132 questions sur 200, la réponse correcte se trouve parmi les 20 premières phrases trouvées.

2.6.2. *Densité prenant en compte le type de la réponse attendue*

Nous avons proposé [GIL 06a] de définir une notion de densité inspirée par les fonctions de score citées dans [TEL 03]. Comme elles, elle permet de choisir un passage en fonction des mots qu'il a en commun avec une question mais elle tient compte en outre des phrases adjacentes dans le document et de la présence ou non du type de réponse recherchée. Nous avons défini un ensemble d'« objets caractéristiques » o_i, extraits de la question Q afin d'aboutir à une requête enrichie. Cet ensemble est constitué des lemmes des mots (après élimination des mots outils), des types d'entités nommées (EN) présentes – noms propres, dates, lieux… –, et du/des type(s) de réponse attendue lorsque celle-ci est une entité nommée. L'idée est de favoriser les passages qui ont non seulement des traits en commun avec la question mais qui contiennent en outre des mots susceptibles d'être la réponse à la question. Ceci illustre l'intégration possible dans le processus de RI de caractéristiques de la tâche questions-réponses (ici le type de réponse attendue).

Pour chacune des occurrences o_w des « objets caractéristiques » w rencontrés, à l'intérieur de chaque document, une distance moyenne $\mu(o_w)$, évaluée en « nombre d'objets », est calculée entre l'occurrence courante o_w et celles des autres objets caractéristiques de la requête, ou de leur plus proche occurrence en cas de présences multiples, au sein du document. Un score de densité est attribué à chaque occurrence de chaque objet caractéristique du document selon la définition suivante :

$$\text{densité}(o_w, d) = \frac{\log\left(\mu(o_w) + \left(|w| - |w, d|\right) \times p\right)}{|w|} \qquad [2.26]$$

avec d un document, o_w une occurrence d'un objet caractéristique w (lemme, type d'entité nommée, type de réponse attendue), p une pénalité, $|w|$ le nombre des objets caractéristiques différents dans la requête q et $|w, d|$ le nombre des objets caractéristiques différents de la requête q présents dans d.

La densité d'une occurrence est ainsi estimée en fonction de la distance qui le sépare des autres objets caractéristiques, du nombre de ces objets dans la question, et enfin, du nombre d'objets communs entre la question et le document. La pénalité, fixée empiriquement, a pour rôle de plus ou moins favoriser une forte proximité de quelques objets communs avec la requête par rapport à une proximité plus faible d'un plus grand nombre d'objets communs. Au contraire, lorsque tous les objets de la requête sont trouvés, la pénalité ne doit pas intervenir.

Un score s est ensuite attribué à chaque phrase S comme étant le meilleur score obtenu par une occurrence d'un objet caractéristique qu'elle contient :

$$s(S,d) = \max_{o_w \in S} s(o_w, d) \qquad [2.27]$$

Chaque phrase est ensuite étendue à un « passage » constitué de la phrase qui la précède et de la phrase qui la suit (lorsqu'elles existent). Cela permet de compenser quelque peu une éventuelle perte de contexte. Le score d'un passage est celui de sa phrase centrale.

2.6.3. *Perspectives dans la recherche de passages*

À notre connaissance, et contrairement à l'un de ses usages en recherche documentaire [BEL 01, CAL 94], la recherche de passages n'est pas utilisée en questions-réponses pour *améliorer la recherche de documents* où la prise en compte de similarités locales permet à certaines informations de ne pas être noyées dans la globalité du document. Une telle rétroaction (recherche de documents puis recherche de passages aboutissant au ré-ordonnancement des documents trouvés puis de nouveau recherche de passages) mériterait certainement d'être expérimentée. Notons toutefois que Rasofolo et Savoy [RAS 03] ont proposé de combiner une mesure de densité — qui tient compte des paires de mots sans nécessiter qu'ils soient adjacents — avec une distance de type *Okapi* (*cf.* formule 2.20) sur le document entier afin d'obtenir le score d'un passage.

Une autre perspective concerne l'une des principales critiques que l'on peut formuler à l'encontre des approches précédentes est qu'elles ne tiennent pas compte de l'ordre des mots dans les phrases et encore moins de leurs rôles syntaxiques et sémantiques. Plusieurs auteurs ont par exemple proposé d'utiliser des modèles de langage probabilistes afin de tenir compte des relations pouvant exister entre les mots d'une question et favoriser ainsi les documents et les passages dans lesquels des relations identiques apparaissent [GAO 04]. Par exemple, dans la question extraite de la campagne EQueR [AYA 05] : « *Quand a été construite la maison d'arrêt de Fleury-Mérogis ?* », l'interrogation concerne la maison d'arrêt située à Fleury-Mérogis et non pas celle d'une autre ville. Cette propriété devrait être retrouvée dans les passages retenus. À défaut d'écarter ceux qui ne semblent pas vérifier cette contrainte de localisation (processus qui sera confié ultérieurement au module d'extraction de réponses), il faut privilégier ceux pour lesquels cela semble fort probable (proximité des mots *Fleury-Mérogis*, *maison d'arrêt* et d'une date dans le passage). Ce constat a été formulé dans [CUI 05], où il est montré l'apport significatif de cette approche par rapport aux méthodes uniquement lexicales.

2.7. Conclusion

Si les modèles de recherche d'informations (RI) qui ont été présentés dans ce chapitre peuvent paraître suffisamment performants pour la tâche questions-réponses, des progrès sont souhaitables. En moyenne, le deuxième document trouvé contient en effet la bonne réponse mais celle-ci peut demeurer inaccessible au module d'extraction de réponse : il pourrait être avantageux d'aller la chercher *plus loin*, au sein de documents *plus faciles*. C'est bien là l'une des difficultés majeures de questions-réponses : le besoin d'utiliser des techniques d'extraction « en profondeur », techniques qui, lorsqu'elles sont disponibles et adaptées à l'expression des réponses candidates dans les documents de la collection, ne sont pas toujours applicables en des temps ou occupation mémoire raisonnables. Dans la pratique, limiter la recherche de la réponse aux tout premiers documents trouvés empêche d'utiliser la redondance comme critère d'extraction. À l'inverse, ne pas se limiter aux premiers documents peut entraîner la prise en considération d'un trop grand nombre de réponses potentielles et donc diminuer la précision du système.

À l'heure actuelle, l'incapacité des systèmes à répondre correctement à certaines questions à partir de documents contenant pourtant les bonnes réponses confirme que la seule présence de ces dernières parmi les premiers documents n'est pas suffisante. Comme en recherche documentaire, il est souhaitable que le SRI soit capable de fournir l'ensemble des documents du corpus qui contiennent la réponse dans le contexte de la question, et seulement ceux-ci. Dans ce contexte, la marge de progression demeure réelle (les référentiels restent à construire) : amélioration de la précision et du rappel afin d'offrir plus de documents pertinents à explorer aux autres modules de la chaîne et variabilité de la granularité (du document aux passages). L'intégration des spécificités de questions-réponses dans les modèles de RI reste à réaliser avec comme double conséquence celle d'unifier, au moins en partie, les problématiques et, par ricochet, d'améliorer les performances des deux applications que sont la recherche documentaire et la recherche de réponses précises.

2.8. Bibliographie

[ALL 96] ALLAN J., CALLAN J., CROFT W.B., BALLESTEROS L., BYRD D., SWAN R., XU J., INQUERY Does Battle With TREC-6 The Sixth Text REtrieval Conference (TREC 6) , NIST Special Publication 500-240, p. 169-206, 1996.

[AMA 02] AMATI, G., VAN RIJSBERGEN, C.J., Probabilistic models of information retrieval based on measuring the divergence from randomness, ACM Trans. Inf. Syst., vol. 20, p. 357-389, 2002.

[AYA 05] AYACHE, C., GRAU, B., VILNAT, A., Campagne d'évaluation EQueR-EVALDA: Évaluation en question-réponse, TALN 2005, p. 6-10, 2005.

[BAE 99] BAEZA-YATES, R.A., *Modern Information Retrieval*, ACM Press, Addison-Wesley, 1999.

[BAZ 05] BAZIZ, M., BOUGHANEM, M., AUSSENAC-GILLES, N., A Conceptual Indexing Approach based on Document Content Representation, COLIS 2005 Context : nature, impact and role, Glasgow, Grande-Bretagne, p. 171-186, 2005.

[BEL 01] BELLOT, P., EL-BÈZE, M., Classification et segmentation de textes par arbres de décision, Technique et Science Informatiques (TSI), Hermes, 20, p. 397-424, 2001.

[BER 55] BERRY, Operational criteria for designing information retrieval systems, American Documentation, vol. 6, 1955, p. 93 à 101.

[BOU 04] BOUGHANEM, M., KRAAIJ, W., NIE, J.Y., Modèles de langue pour la recherche d'information, In: Ihadjadene, M. (ed.), *Les systèmes de recherche d'informations*, Hermes-Lavoisier, p. 163-182, 2004.

[BRI 01] BRILL, E., LIN, J., BANKO, M., DUMAIS, S., NG, A., Data-intensive question answering. Proceedings of the Tenth Text REtrieval Conference (TREC 2001). NIST Special Publication 500-250, p. 393–400, 2001.

[BRO 97] BROWN, E.W., CHONG, H.A.: The Guru System in TREC-6. Proceedings of TREC6. NIST Special Publication 500-240, p. 535-540, 1997.

[BUC 85] BUCKLEY, C.: Implementation of the SMART Information Retrieval System. Department of Computer Science, Cornell University, Ithaca, USA, 1985.

[CAL 94] CALLAN, J.P., Passage-level evidence in document retrieval, Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, Dublin, Ireland, p. 302-310, 1994.

[CLA 00] CLARKE, C.L.A., CORMACK, G.V., KISMAN, D.I.E., LYNAM, T.R., Question answering by passage selection (MultiText experiments for TREC-9), Proceedings of the Ninth Text REtrieval Conference (TREC-9), p. 673-684, 2000.

[CRO 83] CROFT W.B., « Experiments with representation in a document retrieval system », *Information Technology: Research Development*, vol. 2, n° 1, p. 1 à 21, 1983.

[CUI 05] CUI, H., SUN, R., LI, K., KAN, M.-Y., CHUA, T.-S., Question answering passage retrieval using dependency relations, Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, Brazil, p. 400-407, 2005.

[CUT] CUTTING, D.: The Lucene Search Engine. http://www. lucene. com.

[DEE 90] DEERWESTER, S.C., DUMAIS, S., LANDAUER, T.K., FURNAS, G.W., HARSHMAN, R.A., Indexing by Latent Semantic Analysis, Journal of the American Society of Information Science, 41, p. 391-407, 1990.

[ECH 03] ECHIHABI, A., MARCU, D., A Noisy-Channel Approach to Question Answering, Proceedings of the 41st Annual Meeting of the ACL, 2003.

[FAV 06] FAVRE, B., BÉCHET, F., BELLOT, P., BOUDIN, F., EL-BÈZE, M., GILLARD, L., LAPALME, G., TORRES-MORENO, J.-M., The LIA-Thales summarization system at DUC-2006, Document Understanding Conference (DUC-2006), New York (USA), 2006.

[FLU 04] FLUHR, C., L'évaluation des systèmes de recherche d'informations textuelles, In: Chaudiron, S. (ed.), *Evaluation des systèmes de traitement de l'information*, Hermes Lavoisier, p. 27-46, 2004.

[GAO 04] GAO, J., NIE, J.Y., WU, G., CAO, G., Dependence language model for information retrieval, Proceedings of the 27th annual international ACM SIGIR conference on Research and developement in information retrieval, p. 170-177, 2004.

[GIL 06a] GILLARD, L., BELLOT, P., EL-BÈZE, M., Influence de mesures de densité pour la recherche de passages et l'extraction de réponses dans un système de questions-réponses, 3è Conférence en Recherche d'Informations et Applications (CORIA), Lyon, France, p. 193-204, 2006.

[GIL 06b] GILLARD, L., SITBON, L., BLAUDEZ, E., BELLOT, P., EL-BÈZE, M., The LIA at QA@CLEF-2006, actes du Cross Language Evaluation Forum (CLEF 2006), Alicante, Espagne, 2006.

[GRE 02] GREIIF, W.R., MORGAN, W.T., PONTE, J.M., The role of variance in term weighting for probabilistic information retrieval, Proceedings of the eleventh international conference on Information and knowledge management, McLean, Virginia, USA, p. 252-260, 2002.

[HAR 01] HARABAGIU, S., MOLDOVAN, D., PASCA, M., MIHALCEA, R., SURDEANU, M., BUNESCU, R., GIRJU, R., RUS, V., MORARESCU, P., FALCON: Boosting Knowledge for Answer Engines, Proceedings of the 9thText REtrieval Conference (TREC 9), p. 479–488, 2001.

[HAR 95] HARMAN D.K., Overview of the Third Text REtrieval Conference (TREC-3), actes de Text REtrieval Conference TREC-3, Gaithersburg, USA (1994), NIST special publication 500-225, p. 1-19, 1995.

[HOV 02] HOVY, E., HERMJAKOB, U., LIN, C.Y., The Use of External Knowledge in Factoid QA, Proceedings of the Tenth Text REtrieval Conference (TREC 2001), p. 644-652, 2002.

[ITT 02] ITTYCHERIAH, A., FRANZ, M., ROUKOS, S., IBM's Statistical Question Answering System (TREC-10), Proceedings of the Tenth Text REtrieval Conference, p. 258-264, 2002.

[KAS 97] KASZKIEL, M., ZOBEL, J., Passage retrieval revisited, Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, Philadelphia, Pennsylvania, United States, p. 178-185, 1997.

[LEE 02] LEE, G.G., SEO, J., LEE, S., JUNG, H., CHO, B.H., LEE, C., KWAK, B.K., CHA, J., KIM, D., AN, J., SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow NLP, Proceedings of the Tenth Text REtrieval Conference (TREC 2001), p. 442-451, 2002.

[LIN 03] LIN, J., QUAN, D., SINHA, V., BAKSHI, K., HUYNH, D., KATZ, B., KARGER, D.R.: What Makes a Good Answer? The Role of Context in Question Answering, Human-Computer Interaction (INTERACT 2003), Zurich, Switzerland, 2003.

[LIN 05] LIN, J., Evaluation of resources for question answering evaluation, Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, Brazil, p. 392-399, 2005.

[LIT 02] LITKOWSKI, K.C., CL Research Experiments in TREC-10 Question Answering, The Tenth Text Retrieval Conference (TREC 2001) 500-250, 2002.

[LYT 02] LYTINEN, S., TOMURO, N., The use of question types to match questions in FAQFinder, AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases, p. 46–53, 2002.

[MIL 90] MILLER G.A., BECKWITH R., FELLBAUM C., GROSS D., MILLER K., « Introduction to WordNet : An Online Lexical Database », *International Journal of Lexicography*, vol. 3, n° 4, p. 235-244, (version révisée en 1993), 1990.

[MON 03] MONZ, C., *From Document Retrieval to Question Answering*, PhD dissertation, Institute for Logic, Language and Computation, Univ. Amsterdam, 2003.

[PON 98] PONTE, J.M., CROFT, W.B., A Language Modeling Approach to Information Retrieval, Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, Melbourne, Australia, p. 275-281, 1998.

[PRA 01] PRAGER, J., BROWN, E., RADEV, D.R., CZUBA, K.: One Search Engine or Two for Question-Answering. Proceedings of the TREC-9 Conference. NIST Special Publication 500-249, p. 235–240, 2001.

[RAD 02] RADEV, D., FAN, W., QI, H., WU, H., GREWAL, A., Probabilistic question answering on the Web, Proceedings International WWW Conference, Honolulu, Hawaii, USA, 2002.

[RAD 01] RADEV, D.R., QI, H., ZHENG, Z., BLAIR-GOLDENSOHN, S., ZHANG, Z., FAN, W., PRAGER, J., Mining the web for answers to natural language questions, Proceedings of the tenth international conference on Information and knowledge management, Atlanta, Georgia, USA, p. 143-150, 2001.

[RAS 03] RASOLOFO, Y., SAVOY, J., Term proximity scoring for keyword-based retrieval systems, Proceedings 25th European Conference on IR Research (ECIR 2003), p. 207–218, 2003.

[ROB 76] ROBERTSON S.E., SPARCK-JONES K., Relevance weighting of search terms, Journal of the American Society for Information Science, vol. 27, n° 3, p. 129-146, 1976.

[ROB 94] ROBERTSON S.E., WALKER S., HANCOCK-BEAULIEU M., GATFORD M., Okapi at Trec-3, actes de Text REtrieval Conference TREC-3, U.S. National Institute of Standards and Technology, Gaithersburg, USA, NIST special publication 500-225, p. 109-126, 1994.

[ROC 71] ROCCHIO J.J., « Relevance Feedback in Information Retrieval », in Salton G. (éd.), *The SMART Retrieval Storage and Retrieval System*, Englewood Cliffs, N.J., Pentice Hall Inc., p. 313-323, 1971.

[SAL 75] SALTON G., *Dynamic information and library processing*, Englewood Cliffs, USA, 1975.

[SAL 83] SALTON, G., FOX, E., WU, H., Extended boolean information retrieval, Communications of the ACM, 31, p. 1002-1036, 1983.

[SAR 95] SARACEVIC T., « Evaluation of evaluation in information retrieval », actes de la 18è ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle WA, USA, p. 138 à 145, 1995.

[SIN 96] SINGHAL, A., SALTON, G., MITRA, M., BUCKLEY, C., Document Length Normalization, Information Processing and Management, 32, p. 619-633, 1996.

[SIN 00] SINGHAL, A., ABNEY, S., BACCHIANI, M., COLLINS, M., HINDLE, D., PEREIRA, F., AT&T at TREC-8, Proceedings of the Eighth Text REtrieval Conference TREC, p. 317-330, 2000.

[TEL 03] TELLEX, S., KATZ, B., LIN, J., FERNANDES, A., MARTON, G., Quantitative evaluation of passage retrieval algorithms for question answering, Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Canada, p. 41-47, 2003.

[TOM 03] TOMURO, N., Interrogative Reformulation Patterns and Acquisition of Question Paraphrases, The Second International Workshop on Paraphrasing: Paraphrase Acquisition and Applications, p. 33-40, 2003.

[TOM 04] TOMURO, N., LYTINEN, S., Retrieval Models and Q and A Learning With FAQ Files, In: Maybury, M.T. (ed.), *New Directions in Question Answering*, The MIT Press, p. 183-194, 2004.

[VIC 01] VICEDO, J.L., FERRANDEZ, A., LLOPIS, F., University of Alicante at TREC-10, Proceedings of the Tenth Text REtrieval Conference (TREC 2001), p. 510-518, 2001.

[VOO 05] VOORHEES ELLEN, M., Question Answering in TREC, In: Voorhees Ellen, M., Harman, D.K. (eds.), *TREC - Experiment and Evaluation in Information Retrieval*, The MIT Press, p. 233-260, 2005.

[VOO 00] VOORHEES, E.M.: Overview of the TREC-9 question answering track. The Ninth Text Retrieval Conference (TREC-9). NIST Special Publication 500-249, p. 71–80, 2000.

[VOO 01] VOORHEES, E.M., Overview of the TREC 2001 Question Answering Track, The Tenth Text Retrieval Conference (TREC 2001), p. 42-50, 2002.

[VOO 03] VOORHEES, E.M., Evaluating the evaluation: a case study using the TREC 2002 question answering track, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, Edmonton, Canada, p. 181-188, 2003.

[XU 96] XU J., CROFT B., « Query expansion using local and global document analysis », *Proceedings ACM-SIGIR Conference on Research and Development in Information Retrieval* », Zurich, Suisse, p. 4-11, 1996.

[YU 83] YU C.T., BUCKLEY C., LAM K., SALTON G., « A generalized term dependence model in information retrieval », *Information Technology: Research Development*, vol. 2, n° 4, p. 129 à 154, 1983.

NB. Le lecteur pourra en outre se reporter aux actes de l'atelier de travail « *IR4QA: Information Retrieval for Question Answering* » de la conférence SIGIR 2004 (http://nlp.shef.ac.uk/ir4qa04/).