Questions-Réponses en domaine ouvert : sélection pertinente de documents en fonction du contexte de la question

Nicolas Foucault

UNIVERSITÉ PARIS SUD

Comprendre le monde, construire l'avenir®

UNIVERSITÉ PARIS-SUD

ÉCOLE DOCTORALE : ED 427
École doctorale d'Informatique

LABORATOIRE : LIMSI-CNRS
Laboratoire d'Informatique Mécanique et Sciences de l'Ingénieur (UPR 3251)

THÈSE DE DOCTORAT

DISCIPLINE : Informatique

Questions-Réponses en domaine ouvert : sélection pertinente de documents en fonction du contexte de la question

---

Soutenue le 16 décembre 2013
par

Nicolas Foucault

Directeur de thèse :	M^me^	Sophie ROSSET	**DR2–CNRS** (LIMSI–CNRS, Paris)
Co-directeur de thèse :	M.	Gilles ADDA	**IRHC–CNRS** (LIMSI–CNRS, Paris)

Composition du jury :

Président :	M^me^	Brigitte GRAU	**PU** (LIMSI–CNRS, ENSIIE, Paris)
Rapporteurs :	M^me^	Pascale SÉBILLOT	**PU** (IRISA–INSA, Rennes)
	M.	Patrice BELLOT	**PU** (LSIS, Université Aix-Marseille)
Examinateur :	M.	Thierry BACCINO	**PU** (LUTIN, Université Paris 8)

Thèse préparée au LIMSI-CNRS UPR 3251
Université Paris-Sud
BP133 – 91403 Orsay CEDEX

« Et Verbum caro factum est »

[Au commencement était le Verbe]

Verset 1 – **Prologue de l'évangile selon Jean** (traduction par Augustin Crampon 1864)

Résumé

Les problématiques abordées dans ma thèse sont de définir une adaptation unifiée entre la sélection des documents et les stratégies de recherche de la réponse à partir du type des documents et de celui des questions, intégrer la solution au système de Questions-Réponses (QR) RITEL du LIMSI et évaluer son apport.

Nous développons et étudions une méthode basée sur une approche de Recherche d'Information pour la sélection de documents en QR. Celle-ci s'appuie sur un modèle de langue et un modèle de classification binaire de texte en catégorie *pertinent* ou *non pertinent* d'un point de vue QR. Cette méthode permet de filtrer les documents sélectionnés pour l'extraction de réponses par un système QR.

Nous présentons la méthode et ses modèles, et la testons dans le cadre QR à l'aide de RITEL. L'évaluation est faite en français en contexte web sur un corpus de 500 000 pages web et de questions factuelles fournis par le programme Quaero. Celle-ci est menée soit sur des documents complets, soit sur des segments de documents. L'hypothèse suivie est que le contenu informationnel des segments est plus cohérent et facilite l'extraction de réponses. Dans le premier cas, les gains obtenus sont faibles comparés aux résultats de référence (sans filtrage). Dans le second cas, les gains sont plus élevés et confortent l'hypothèse, sans pour autant être significatifs. Une étude approfondie des liens existant entre les performances de RITEL et les paramètres de filtrage complète ces évaluations.

Le système de segmentation créé pour travailler sur des segments est détaillé et évalué. Son évaluation nous sert à mesurer l'impact de la variabilité naturelle des pages web (en taille et en contenu) sur la tâche QR, en lien avec l'hypothèse précédente.

En général, les résultats expérimentaux obtenus suggèrent que notre méthode aide un système QR dans sa tâche. Cependant, de nouvelles évaluations sont à mener pour rendre ces résultats significatifs, et notamment en utilisant des corpus de questions plus importants.

Mots-clefs : Traitement Automatique des Langues, Questions-Réponses, Recherche d'Information, Ritel, Quaero, sélection de documents, modèle de langue, classification de pages web, segmentation de pages web, apprentissage automatique.

Abstract

Title : Open domain question-answering : relevant document selection geared to the question

This thesis aims at defining a unified adaptation of the document selection and answer extraction strategies, based on the document and question types, in a Question-Answering (QA) context. The solution is integrated in RITEL (a LIMSI QA system) to assess the contribution.

We develop and investigate a method based on an Information Retrieval approach for the selection of relevant documents in QA. The method is based on a language model and a binary model of textual classification in *relevant* or *irrelevant* category. It is used to filter unusable documents for answer extraction by matching lists of a priori relevant documents to the question type automatically.

First, we present the method along with its underlying models and we evaluate it on the QA task with RITEL in French. The evaluation is done on a corpus of 500,000 unsegmented web pages with factoid questions provided by the Quaero program (i.e. evaluation at the document level or D-level). Then, we evaluate the method on segmented web pages (i.e. evaluation at the segment level or S-level). The idea is that information content is more consistent with segments, which facilitates answer extraction. D-filtering brings a small improvement over the baseline (no filtering). S-filtering outperforms both the baseline and D-filtering but not significantly. Finally, we study at the S-level the links between RITEL's performances and the key parameters of the method.

In order to apply the method on segments, we created a system of web page segmentation. We present and evaluate it on the QA task with the same corpora used to evaluate our document selection method. This evaluation follows the former hypothesis and measures the impact of natural web page variability (in terms of size and content) on RITEL in its task.

In general, the experimental results we obtained suggest that our IR-based method helps a QA system in its task, however further investigations should be conducted – especially with larger corpora of questions – to make them significant.

Keywords : Natural Language Processing, Question & Answering, Information Retrieval, Quaero, Ritel, document selection, web page classification, language modeling, web page segmentation, machine learning.

Remerciements

Durant cette thèse, j'ai eu la chance et le plaisir de connaître et d'interagir avec de nombreuses personnes, chercheurs ou non, qui ont, parfois sans le savoir, contribués au succès de ma thèse. J'ai bien conscience qu'il serait vain de vouloir remercier chacune d'entre elles. Ainsi, je tiens tout d'abord à dire un grand merci à toutes celles et ceux à qui je pense en rédigant ces remerciements, mais également à toutes celles et ceux que je pourrais avoir l'indélicatesse d'oublier dans cette pensée.

En particulier, je tiens à remercier Sophie Rosset et Gilles Adda qui m'ont guidé tout au long de ma thèse. La responsabilité d'un encadrement de thèse me semble être une savante alchimie entre *humanité* et *enseignement* pour gérer aussi bien l'homme que l'apprenti. Je peux dire qu'ils ont réussi dans les deux cas. À leur côté j'ai appris qu'*écouter* ne signifiait pas seulement *entendre*, mais aussi *comprendre*. J'ai notamment compris que le mieux est l'ennemi du bien : la rigueur scientifique ne peut et ne doit pas être synonyme de *perfection*, mais de *qualité*. L'*imperfection* est un bien nécessaire en recherche. On ne peut que l'accepter et, si possible, chercher à la maîtriser. Peut-être est-ce à l'aune des imperfections laissées volontairement vacantes qu'on est le plus à même d'évaluer la *maturité* d'une recherche et celle de son auteur ?

Je tiens également à remercier les membres de mon jury. Merci à Pascale Sébillot et à Patrice Bellot d'avoir été mes rapporteurs. Sans eux, la qualité de mon manuscrit ne serait pas ce qu'elle est aujourd'hui et je ne doute pas qu'avec le temps je ne leur en serait que plus reconnaissant. Merci à Brigitte Grau pour avoir présidé avec soin ma soutenance de thèse, pour ses remarques, ses questions, mais aussi pour son soutien, nos discussions et son inconditionnelle gaieté. Enfin, merci à Thierry Baccino d'avoir accepté, sans sourcillé, de s'être porté examinateur de mon travail, bien que réalisé dans un domaine relativement éloigné du sien. J'espère que nous aurons l'occasion de pousser plus avant ce désir de mettre en commun nos recherches respectives.

Je réalise aujourd'hui que la thèse est une aventure qui a commencé avant même que j'y songe, tout au long de mon parcours universitaire et, notamment, durant mes années à Lyon en sciences cognitives. Merci donc à Barbara Tillman et Nicolas Grimault de m'avoir transmis leur passion pour la recherche à travers mon stage de maîtrise au laboratoire de Neurosciences et Systèmes Sensoriels en psycho-acoustique et psycho-musicologie.

Je remercie aussi mes enseignants de Paris 7, Pascale Amsili et Benoît Crabbé qui m'ont donné le goût du TAL, Anne Abeillé et Clément Plancq du LLF avec qui j'ai fait mes armes dans ce domaine, et Danielle Godard pour le partage de son bureau et de son thé à cette époque-là.

Je remercie chaleureusement Silvia Quarteroni avec qui j'ai eu le loisir de travailler à l'*Università degli studi di Trento* autour de la thématique Questions-Réponses. Elle m'a acceuilli à bras ouverts alors que nous ne nous connaissions pas. C'est elle qui m'a donné envie de continuer à travailler dans ce domaine en thèse. De cette période en Italie et encore aujourd'hui, je dois aussi un grand merci à Pierre Andrews pour nos échanges, ces coups de pouces bibliographiques et méthodologiques.

La thèse se déroule dans un environnement de recherche. J'ai été ravi de passer ma thèse au LIMSI. J'ai eu l'opportunité de côtoyer différents laboratoires, je dois dire que l'atmosphère du LIMSI est particulièrement stimulante et revêt un esprit très familial. J'y ai été à l'aise, entouré par un personnel d'encadrement-recherche fort compétent qui m'a fait bénéficier de conditions de travail adéquates pour la mener.

Je tiens à saluer mes collègues et amis là-bas, Béatrice, Anne-Lyse, Alexander, Gabriel, Patrick, Mathieu et Mathieu, Sylvain, Andreea, Houda, Sami, Driss, Camille, Guillaume, Michael, Cyril, Virginie, Clément et Clément, Julieta, David, Olivier, Nadi et mes actuels collègues de bureau Thomas et Martin.

L'amitié étant un point particulièrement important au cours d'une thèse, je me dois de remercier ceux qui m'ont soutenu dans la durée, amis de longue date et amis plus récents, Ben, Ringo, Toutoune, Nh, Croco, Barb, Gab, Ju, Barnabe, Carmen, Rébécca, Emma, Misa, Do, Soria, Agathe, François, Maidi, Neysa et ma très chère colocataire, Virginia.

Merci du fond du coeur à ma famille, mes parents Anne et Thierry pour m'avoir toujours poussé vers ce que j'aime faire, à mon frère Antoine, bien parti pour de longues études, alors que tout comme moi il se dirigeait au contraire vers un cycle court ;p. Enfin merci à celle qui m'accompagne depuis peu d'être là, elle se reconnaîtra.

Nicolas Foucault,
Paris, 30 janvier 2014

Table des matières

Liste des tableaux

Table des figures

« Je ne sais qu'une chose, c'est que je ne sais rien. »

SOCRATE

Chapitre 1

Introduction

1.1 Problématique

POUVOIR, à partir d'un ensemble de documents, extraire une information, est un des plus anciens sujets traités en informatique. Ce sujet a débouché sur le domaine de recherche de la « Recherche d'Information » (RI), terme introduit en 1950 par Calvin Northrup Mooers. Mooers a conduit les travaux fondateurs dans ce domaine avec le développement du système de « Zato coding » [104].
Il est inutile de rappeler l'importance qu'a de nos jours le domaine de la recherche d'information, en particulier par l'utilisation massive des moteurs de recherche qui ont permis un accès facile et efficace aux quantités d'informations en croissance exponentielle disponibles sur Internet. Cet impact sociétal peut se mesurer en particulier par le fait qu'un moteur de recherche, GOOGLE, est devenu l'une des plus importantes et des plus profitables entreprises mondiales.

Mais les moteurs de recherche ont certaines limitations, en particulier lorsque la requête est faite dans l'intention d'obtenir une réponse précise : les systèmes de recherche renvoient un ensemble de documents pouvant contenir l'information demandée, à charge au demandeur de la trouver dans lesdits documents. Ceci a conduit au développement des recherches sur le principe des systèmes de réponses aux questions (ou bien questions-réponses) où on propose directement à l'utilisateur une réponse précise à sa question.

Dans le cadre des systèmes de questions-réponses, deux aspects sont particulièrement importants : d'une part la sélection de documents pertinents pour une question donnée et, d'autre part, les stratégies d'extraction employées pour trouver des réponses dans ces documents. Actuellement, ces deux aspects sont relativement indépendants. Ainsi, à partir d'une question, de tels systèmes génèrent dans un premier temps, une ou plusieurs requêtes. Celles-ci sont ensuite utilisées par un moteur d'indexation généraliste qui se charge de trouver les N documents les plus pertinents dans lesquels appliquer des stratégies de recherche et d'extraction de réponses.

Une solution envisageable pour lier ces deux aspects serait de catégoriser les documents en fonction de leur contenu et mettre en relation ce typage à celui des questions, dans le but de sélectionner les documents les plus appropriés à la recherche des réponses. La catégorisation de documents est un domaine de recherche très actif, en particulier pour les documents du web dont la quantité [46] et la diversité sont phénoménales : spams, pages de forum, blogues, journaux en ligne, etc. Ceci implique nécessairement de sélectionner l'information pertinente recherchée.

Pouvoir opérer une sélection pertinente des documents en fonction de leur type (catégorie) et de celui des questions (selon leur thème, leur forme, leur classe etc.) devrait permettre un gain d'efficacité dans le choix de stratégies d'extraction précises.

Les problématiques abordées dans ma thèse sont de définir une adaptation unifiée de la sélection des documents et des stratégies de recherche de la réponse à partir du type des documents et de celui des questions, d'intégrer la solution au système de questions-réponses du projet *Ritel* du LIMSI et d'en évaluer l'apport.

1.2 Cadre de la thèse

Nous précisons dans cette section les conditions théoriques et pratiques qui ont guidées cette thèse.

1.2.1 Domaine : Traitement Automatique des Langues

Mon travail de thèse s'inscrit dans le cadre plus large du traitement automatique des langues naturelles (ou TAL). Nous nous situons dans une approche statistique du TAL, au sens où nous nous reposons sur des modèles et méthodes issus de corpus, à l'aide d'apprentissage automatique.

Le TAL est un domaine issu de l'Intelligence Artificielle, et s'intéresse particulièrement à des problèmes de génération et de compréhension automatique des langues naturelles avec le support d'approches originaires des domaines de l'apprentissage automatique et de la linguistique computationnelle [17]. Ainsi, il n'est pas rare que les approches utilisées en TAL empruntent aux modèles probabilistes, à la théorie de l'information, et à l'algèbre linéaire [92].

Parmi les tâches importantes auxquelles s'est attaqué le TAL, on compte l'annotation, l'analyse syntaxique, la traduction, la classification de textes et les réponses à des questions, communément appelée tâche de Questions-Réponses. Ces deux dernières tâches sont souvent considérées dans la littérature comme des tâches de Recherche d'Information [136, 17].

1.2.2 Contexte : le projet Quaero

Quaero signifie « je cherche » en Latin. Quaero est un programme de recherche et d'innovation Européen franco-allemand [1] destiné au développement de nouvelles techniques et outils intégrés, pour l'accès et l'utilisation de contenus multimédias (c.-à-d. texte, son, image et vidéo), en analyse automatique, classification et indexation d'information.

Ce programme, qui a duré cinq années, a impliqué un grand nombre de laboratoires publics et d'acteurs industriels (PME, ETI et grands groupes). Le programme comportait plusieurs volets : un volet applicatif, un volet recherche et un volet corpus. L'ensemble du programme était fondé sur le principe de mise à disposition de technologies (développées dans le volet recherche) dans des applications industrielles (développées dans le volet applicatif). Les technologies, souvent innovantes, étaient évaluées de manière générique et en contexte applicatif, lors de campagnes d'évaluation objective et compétitive. Le volet corpus a fourni aux technologies les données nécessaires à l'apprentissage des modèles et à l'évaluation des systèmes. En particulier de nombreuses campagnes d'évaluation des systèmes de questions-réponses ont eu lieu dans le cadre du projet Quaero. Nous avons utilisé des corpus d'entraînement et d'évaluation élaborés dans ce cadre.

1. http://www.quaero.org/quaero-en-bref

1.2.3 Support : le système de questions-réponses de Ritel

Le projet *Ritel* [147] a vu le jour au LIMSI en 2004. L'objectif de ce projet est de réaliser un système de communication homme-machine (**RITEL** [36]) guidé par la parole, c'est-à-dire un système de dialogue dédié à l'interaction homme-machine en modalité orale. Ce système de dialogue est entendu comme un système de RI généraliste capable de trouver des réponses à des questions telles que *Qui est le Président du Sénat ?* ou *Quelle a été l'évolution du prix de l'essence ces dix dernières années ?* tout en permettant à l'utilisateur d'affiner ou préciser sa demande de manière interactive et dynamique.

Le projet Ritel se situe à la frontière de domaines distincts mais complémentaires comme la Recherche d'Information, le dialogue oral homme-machine et le TAL pour la compréhension et la génération d'énoncés en langue naturelle, et vise un type nouveau de dialogue entre l'homme et la machine au sens où il s'applique à des recherches collaboratives, basées sur une co-construction dynamique du sens et du domaine de l'interaction.

Ce projet a dû relever un certain nombre de défis audacieux dont les plus évidents sont : la reconnaissance de la parole qui doit être à grand vocabulaire et sur laquelle une contrainte temps réel s'applique, la gestion d'un dialogue en domaine ouvert (c.-à-d. dont le sujet de la conversation peut se rapporter à n'importe quelle thématique), la communication et l'échange de résultats entre un système de RI et un système de dialogue et enfin, la génération automatique de réponses dans le contexte des questions utilisateurs au sein du dialogue.

Dans notre travail de thèse, nous nous sommes intéressé uniquement au système de questions-réponses du projet Ritel.

1.3 Axes de recherche de la thèse

L'une des idées majeures sur laquelle s'est fondé l'ensemble des travaux expérimentaux présentés dans cette thèse est qu'afin d'obtenir les meilleurs résultats possibles, nous devions pouvoir mettre en commun autant que faire se peut, les technologies utilisées en RI et en QR. Cette conviction est en partie fondée sur les résultats de l'évaluation NTCIR-7, que nous détaillerons plus avant.

1.3.1 Questions-Réponses et Recherche d'Information

Les systèmes de questions-réponses (QR) peuvent se voir comme des systèmes de RI avancés. Dans un système de RI (p. ex. le moteur de recherche GOOGLE), l'utilisateur définit sa recherche par une requête contenant des mots-clés. Le résultat de la recherche est un ensemble de documents ou de liens vers des documents que l'utilisateur doit ensuite fouiller afin de trouver l'information qu'il recherche.

Par exemple, pour répondre à la question *Quel acteur joue le rôle principal dans le film Golden Eye ?* [2], sans rien savoir ni du film ni de l'acteur recherché, il est nécessaire à un internaute de suivre (voir figure 1.1) le 3ème résultat parmi les réponses proposées par Google à la requête *Golden Eye main actor* et ainsi vérifier que l'acteur Pierce Brosnan mentionné dans la section « Starring » de ce résultat est bien l'information qu'il recherche.

A contrario, les systèmes QR fournissent à l'utilisateur des réponses précises à leur question formulée en langue naturelle. Ainsi, pour la même recherche que celle faite précédemment à GOOGLE, on posera par exemple au système START [69, 70] la question *Who is the main actor in Golden Eye ?* qui pourra répondre en retour : *Pierce Brosnan* (voir figure 1.2).

2. Réponse : *Pierce Brosnan*

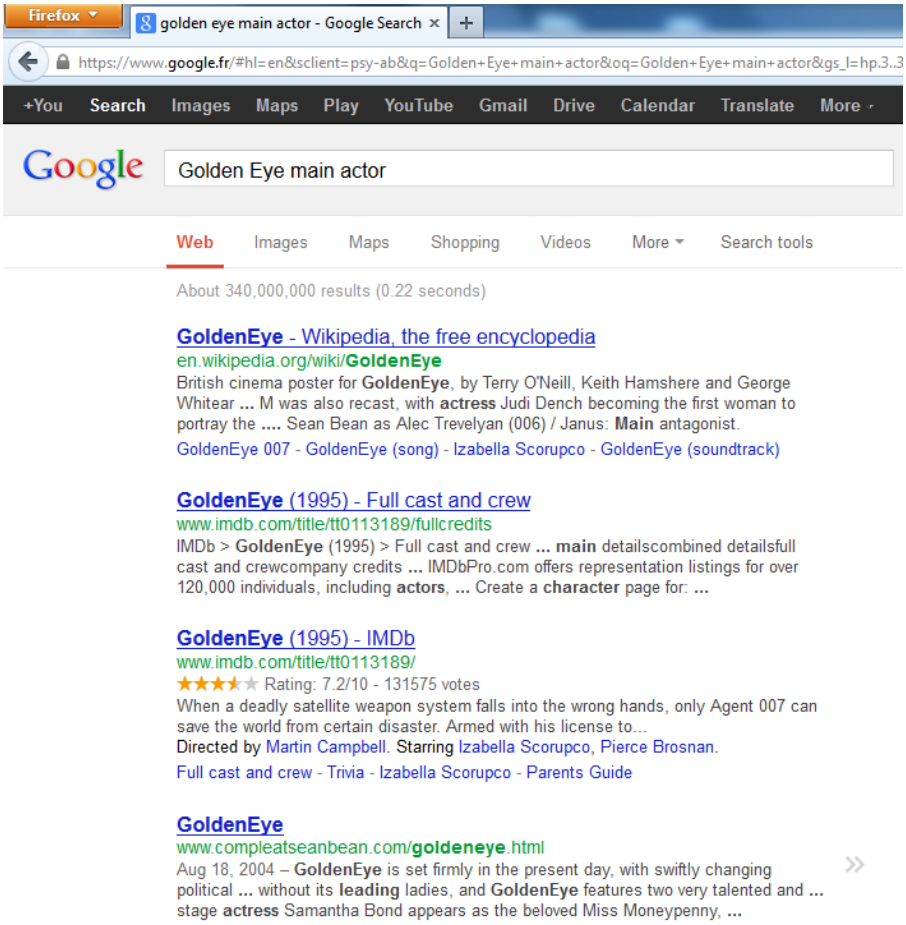


FIGURE 1.1 – Exemple de requête (*Golden Eye main actor*) et résultat fourni par un système de Recherche d'Information classique (le moteur de recherche de chez GOOGLE) pour trouver quel acteur tient le rôle principal dans le film *Golden Eye*.

Certains systèmes QR comme START peuvent poser des questions à l'utilisateur pour leur demander de préciser leur requête, par exemple, quand celle-ci est ambiguë. Dans le cas de la question *Who is the main actor in Golden Eye ?*, on voit à la figure 1.2b que START demande à l'utilisateur de préciser sa question afin de savoir si celle-ci porte sur le film *Golden Eye* de Martin Campell (1995) avec comme acteur principal Pierce Brosnan dans le rôle de James Bond ou bien *The Golden Eye* de William Beaudine (1948) avec l'acteur principal Roland Winters dans le rôle de Charlie Chan.

À la figure 1.2c, on voit que la réponse apportée par START concernant le film de Campbell est faite sous la forme d'un lien vers la page web IMDB relative à Pierce Brosnan (voir figure 1.3) dans le cas où un complément d'information serait nécessaire à l'utilisateur.

(a)

(b)

(c)

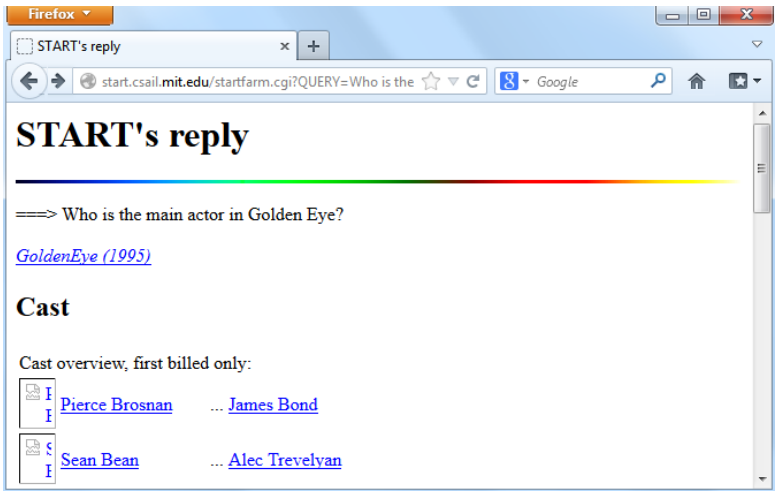


FIGURE 1.2 – Exemple de questions (a) et (b) et réponses (c) à l'aide du système de Questions-Réponses START.

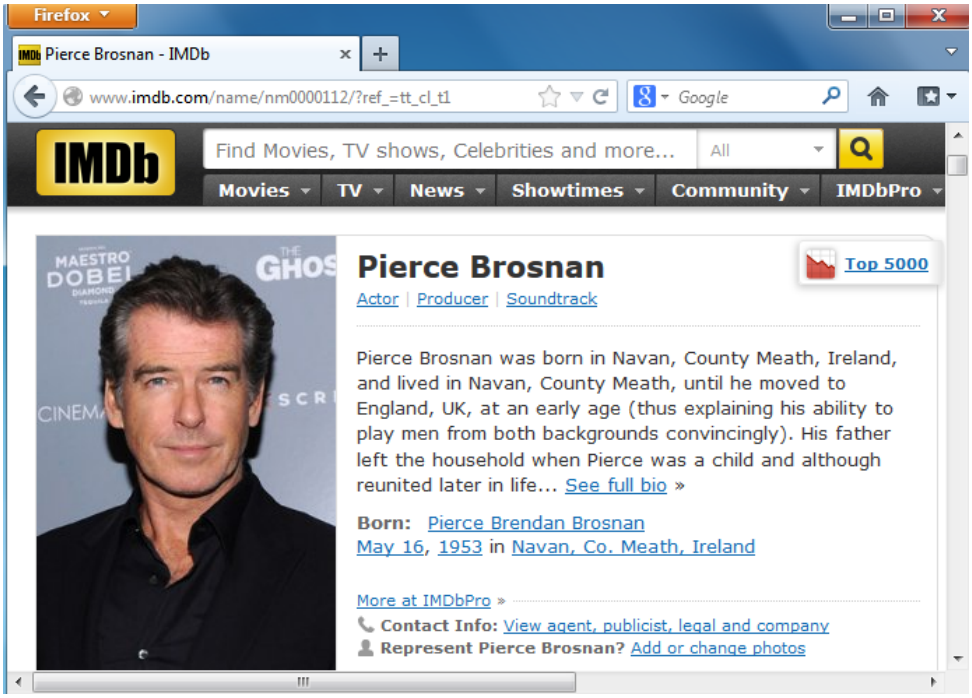


FIGURE 1.3 – Exemple de complément d'information fourni par le système de Questions-Réponses START à la réponse *Pierce Brosnan* pour la question : *Who is the main actor in Golden Eye ?*.

1.3.2 Sélection de documents, pertinence et Question-Réponses

Les systèmes de QR comme START et de RI comme GOOGLE sont souvent opposés dans la littérature TAL. Les premiers travaillent à l'aide d'indices linguistiques extraits de questions, analysent les documents à l'échelle d'extraits de documents et de phrases et en extraient des réponses précises destinées à l'utilisateur. Les seconds travaillent à l'aide de mots-clés trouvés dans des requêtes, n'effectuent pas ou peu d'analyses linguistiques profondes des textes, et mènent leur recherche à l'échelle globale des documents, qu'ils retournent à l'utilisateur.

Pourtant, ces deux types de systèmes se rejoignent sur certains points, et notamment sur leurs stratégies de recherche et de sélection de documents, et sur la manière de définir leur pertinence.

Ainsi, de nombreux systèmes QR (p. ex. RITEL) et RI (p. ex. GOOGLE) s'appuient sur des statistiques de co-occurrences des termes trouvés dans les documents afin de mesurer le degré de coïncidence (ou de pertinence) existant entre la demande d'un utilisateur et un document donné. Une mesure de pondération des termes d'un document comme TF-IDF [65] permet d'associer des poids à chaque terme des documents selon la tâche, afin d'affiner la recherche et la sélection des documents. Le degré de coïncidence entre la demande de l'utilisateur et un document peut être défini à l'aide de leur représentation vectorielle sur la base des termes qu'ils contiennent. Ce genre de méthodes et de mesures statistiques de pertinence pour les recherches a été utilisé à l'origine en RI pour la fouille textuelle. Elle a également été réutilisée avec succès par la suite en QR. Par exemple, GOOGLE comme START utilisent ce genre d'outils dans leur tâche.

D'autres systèmes s'appuient sur la notion de redondance. L'idée de la redondance est de donner à un document plus d'importance quand d'autres documents sélectionnés par un système pour une même demande (requête ou question) lui ressemblent fortement. La redondance est souvent utilisée par les moteurs de recherche dans leur tâche et l'est également dans certains systèmes de QR (p. ex. RITEL et YourQA [121]).

Un autre critère de pertinence, très fréquemment utilisé en RI par les moteurs d'indexation web, est le nombre de visites faites à un site ou une page web. Par exemple, GOOGLE utilise ce critère. Ainsi, dans un premier temps, à partir des mots-clés de recherche, le moteur de GOOGLE détermine quelles pages sélectionner dans ses index. À partir des pages trouvées et de leur fréquence de visites respectives, PageRank™ [15, 112] (l'algorithme de recherche et de sélection de documents de GOOGLE) détermine ensuite dans quelle mesure une page web P répond à une requête R, et où situer P parmi l'ensemble des résultats correspondant à R. Une telle sélection est motivée par l'idée que plus P a reçu de visites, plus elle mérite de figurer en tête du classement des pages établi par PageRank pour répondre à l'utilisateur. Ce genre de critère peut aussi être utilisé par les systèmes de QR en complément d'une analyse fine de documents, mais la pertinence d'un document sur la base du nombre de visites d'une page est relative et conduit à des erreurs. Notamment, elle a tendance à pénaliser la sélection et le classement des pages les moins visitées. Par exemple, GOOGLE ne fait pas référence au film *The Golden Eye* de Beaudine parmi ces 300 résultats de recherche à la requête : *main actor Golden Eye*.

Dans la section suivante, nous présentons en surface une campagne d'évaluation à la croisée des domaines du QR et de la RI, la campagne d'évaluation NTCIR-7 ACLIA [129]. Cette campagne montre bien le potentiel que représentent les stratégies de RI en QR et vice-versa, malgré un domaine d'application éloigné du nôtre dans cette thèse. C'est la seule campagne d'évaluation que nous ayons trouvée dans la littérature fondée sur l'idée que les domaines de la RI et du QR devraient unifier leurs efforts dans une même direction pour bénéficier de meilleures approches dans leur tâche respective. Certains travaux récents vont dans ce sens [89].

Nous présentons cette campagne, en mettant en avant les points essentiels qui nous ont motivé dans cette thèse à utiliser et étudier rigoureusement un modèle basé sur une approche de RI en QR pour la sélection pertinente de documents.

1.3.3 Exemple d'alliance entre RI et QR : la campagne NTCIR-7

Présentation générale de NTCIR-7 La campagne NTCIR entre dans le champ de l'Accès à l'Information. Ce domaine couvre des disciplines telles que la RI, le QR, le résumé automatique et l'Extraction d'Information.

Cette campagne a pour but d'évaluer l'efficacité des systèmes développés dans ces disciplines. Son cadre de travail porte sur l'étude des langues asiatiques et, en particulier, le chinois classique, le chinois simplifié et le japonais. L'objectif de cette campagne est double et concerne, d'une part, la création de corpus textuels multilingues à grande échelle et réutilisables pour l'expérimentation et, d'autre part, le développement d'infrastructures destinées à l'évaluation commune des systèmes orientés QR.

Spécificité de l'édition 2007 L'édition NTCIR de 2007 (appelée ACLIA) propose pour la première fois l'étude de questions complexes en QR multilingue (ou CCLQA) dans le contexte des langues asiatiques. Les questions sont réparties en 4 classes : *definition* ou DEF (p. ex. *What is the Human Genome Project ?*), *biography* ou BIO (p. ex. *Who is Kim Jong-Il ?*), *relationship* ou REL (p. ex. *Does Iraq possess uranium, and if so, where did it come from ?*) et *events* ou EVT (p. ex. *List major events in Saddam Hussein's life.*).

Évaluations Les évaluations sont faites selon 2 séries de défis. La première série présente 3 défis CCLQA, dédiés à l'étude des différentes parties de la chaîne QR : un défi pour l'analyse de la question, un autre pour la recherche d'information et un dernier pour la recherche et l'extraction de réponses. Ces deux derniers défis sont évalués d'après les réponses fournies par les systèmes aux questions qui leur ont été posées. La seconde série présente un unique défi, dédié à la RI multilingue (nommé CLIR) dont l'évaluation est menée à l'aide de jeux de requêtes construits à partir des questions posées aux systèmes QR des défis CCLQA.

Originalité des évaluations L'originalité ici consiste à proposer des évaluations centrées autour d'une architecture d'inter-échangeabilité des données d'entrée et de sortie des systèmes, à la fois entre systèmes et entre tâches. Ainsi, l'une des variantes du challenge CCLQA consiste à fournir à un système QR du challenge CCLQA les 50 documents les plus pertinents sélectionnés par un système RI lors du challenge IR4QA, et à mener la tâche d'extraction de réponses sur ces documents plutôt que sur ceux trouvés par ses propres moyens. De même, les systèmes RI peuvent décider d'utiliser les analyses fournies par les systèmes QR afin de mener leur recherche plutôt que d'utiliser les analyses de référence fournies par NTCIR. La seconde variante consiste à procéder non plus avec 50, mais avec 1000 documents.

Constats De manière générale, on constate qu'en collaborant, les systèmes obtiennent de meilleurs résultats que s'ils avaient fonctionné en autonomie ; de plus les systèmes collaboratifs sont ceux qui obtiennent les meilleurs résultats en général.

Cependant, on peut observer que les systèmes de QR ont soumis des réponses basées sur les sorties des systèmes de RI, mais l'inverse ne s'est pas fait. Cela relativise les résultats attendus de l'évaluation, dans la mesure où l'objectif principal du défi IR4QA était d'évaluer l'impact de la classification des questions selon leur catégorie (BIO, DEF, REL et EVT) sur la tâche de RI pour savoir si un système de RI, à l'image des systèmes QR, devraient procéder à une analyse de la requête en terme de classe afin d'en tirer avantage pour la recherche et la sélection de documents pertinents.

Par ailleurs, aucun des systèmes de RI du challenge IR4QA ne s'est servi des types de réponses fournis en référence par NTCIR pour caractériser les requêtes fournies aux systèmes. En effet le type d'une réponse ne conduit pas naturellement à typer un document ; il est donc nécessaire d'adapter spécifiquement un système de RI pour que ce genre d'information lui servent dans sa tâche et nécessite un coût que les participants n'ont pas jugé intéressant d'investir.

Le fait que les meilleurs résultats de CCLQA soient obtenus grâce aux sorties des systèmes d'IR4QA nous apprend donc qu'il n'est pas crucial d'appliquer pour les phases de recherche et de sélection de documents pertinents en QR des analyses TAL poussées lors de l'analyse de la question, aussi bien au niveau syntaxique (p. ex. définir des arbres syntaxiques sur la question qui peuvent permettre de préciser la recherche) que sémantique (p. ex. chercher le focus de la question ou le type de la réponse). Ceci est intéressant en particulier pour notre approche, car cela signifie que les techniques standards employées en RI peuvent être utiles en QR.

Au vu de ces résultats, il nous a donc semblé important de s'intéresser aux techniques de RI utilisées lors du challenge IR4QA de la campagne NTCIR-7 dans une perspective d'application en QR.

Le défi IR4QA de la campagne NTCIR-7

Parmi les techniques de RI les plus employées lors de la campagne NTCIR-7 par les systèmes en compétition, nous avons orienté notre analyse sur les systèmes qui utilisaient des moteurs et des stratégies de RI indépendantes d'un système en particulier, en mettant de côté les systèmes les plus faibles sur chacune des 3 langues évaluées. Les principales méthodes et systèmes que nous avons sélectionnés sont données au tableau 1.1 (pour le chinois simplifié uniquement).

P	**Moteur**	**Index**	N-grams (si c)	**Stratégies RI**	**#CS**
CYUT	Lucene	w		**ER** basée sur Wikipedia ; scoring/OkapiBM25	8
HIT	Indri	c	2	ranking de passages/ML (KL lissée)	4
KECIR	Lucene	w+c	1	**ER** optimisée/topic ; ranking/CL de scores (MV+ML)	5
MITEL	Lemur	w+c	1+2	3 index séparés ; ranking/CL de scores (KL lissée)	2
RALI	Indri	w		fenêtre glissante sur passages et ranking/ML (KL brute)	6
WHUCC	Lemur	c	1+2	**ER** ; scoring/OkapiBM25 ; reranking	7

TABLE 1.1 – Récapitulatif des stratégies RI (listées par ordre alphabétique) les plus utilisées au cours du challenge NTCIR7-IR4QA. **P** : participant. w : indexation au niveau des mots. c : indexation au niveau des caractères. Dans ce cas, on précise à la suite l'ordre du modèle N-grams utilisé : 1 (unigram) ou 2 (bi-grams). **ER** : expansion de requêtes. ML : modèle de langue. KL : scores basés sur la divergence de Kullback-Leibler. MV : modèle vectoriel. CL : combinaison linéaire de score. #CS : classement des systèmes sur le chinois simplifié.

On peut relever deux points dans ce tableau. Tout d'abord, on peut constater que les systèmes de RI se fondent, comme souvent en QR, sur des méthodes d'expansion de requêtes (ER). Celle de CYUT consiste à étendre les termes trouvés dans les requêtes par d'autres termes issus de liens hyper-textes en provenance de Wikipedia, via une pondération des clés de recherche basée sur OkapiBM25 [126] (une variante de TF-IDF). Celle de KECIR consiste à comparer trois ER différentes et à choisir la meilleure. L'une emprunte au concept de *relevance feedback* [4], l'autre se base sur une analyse en contexte local et la dernière s'appuie sur le moteur de recherche *Baidu*[3] ; de plus, KECIR optimise la taille des requêtes en fonction des catégories de thèmes possibles (BIO, DEF, REL et EVT). On peut relever cependant que parmi les meilleurs systèmes (MITEL, HIT et RALI), certains n'emploient pas d'expansion de requêtes.

Le second point est qu'il y a divergence sur les méthodes de calcul de scores de pertinence des documents : elles sont soit classiques (p. ex. OkapiBM25 chez WHUCC), soit complexes (p. ex. scoring par modèle de langue [63] et divergence de Kullback-Leibler chez HIT et RALI).

On voit dans ce tableau que les meilleures performances sont attribuées à MITEL. Une explication possible, à vérifier, est que cela est dû à l'interrogation d'index séparés. Les 2 meilleurs systèmes utilisent des modèles de langues et un scoring de pertinence des documents vis-à-vis de la requête établi selon des divergences de Kullback-Leibler [78] lissées (HIT et MITEL avec des lissages de type Jelinek–Mercer [160] et Dirichlet [11] respectivement) ou non (RALI). Les tests réalisés a posteriori par les participants ont montré que l'équipe du RALI avait eu un problème lors de la soumission, et que leurs résultats devaient en fait les positionner premier. Il est intéressant de voir qu'une méthode utilisée en extraction de passages pertinents en QR (une analyse des documents basée sur une fenêtre glissante) donne en RI de bons résultats pour la sélection de documents.

3. http://www.baidu.com. Baidu est le pendant chinois de GOOGLE.

Ces résultats nous ont incité à étudier en détail les liens existants entre RI et QR dans notre travail. Ils nous ont notamment motivé à utiliser un modèle de langue pour la sélection de documents pertinents en QR.

1.4 Approche de la thèse

Le travail de thèse présenté ici est fondamentalement expérimental. La démarche que nous avons choisie est de procéder à des analyses détaillées afin de pouvoir, lorsque cela est possible, en tirer des conclusions sur les effets individuels et joints des différents paramètres utilisés par nos modèles sur les résultats QR de RITEL.

Les résultats obtenus sont systématiquement atomisés et les conclusions apportées, interprétées sous différents angles et confrontées à des contrôles, des analyses et des études complémentaires.

Notre volonté a été, autant que possible, de rechercher l'objectivité et l'introspection des modèles évalués.

1.5 Apports de la thèse

- **Côté méthode** :
  > une méthode d'évaluation de la pertinence intrinsèque d'un document en QR ;
  > un système de filtrage pour la sélection pertinente de documents en QR ;
- **côté outil** :
  > un système de segmentation textuelle de pages web ;
- **côté étude** :
  > une étude approfondie de l'échelle (documents ou segments) de filtrage sur les performances QR ;
- **côté ressource** :
  > un corpus de segments pertinents en QR ;
  > une classification de référence du corpus de 500k pages web Quaero en français en catégorie *pertinent* ou *non pertinent* pour les recherches en QR.

Au moment de l'écriture de la thèse, ces apports ont donné lieu aux publications suivantes :

- N. Foucault, G. Adda, S. Rosset, *Language Modeling for Document Selection in Question Answering*. In Proceedings of Recent Advances in Natural Language Processing, pages 716-720. 12-14 September, 2011, Hissar, Bulgaria.
- N. Foucault, S. Rosset, G. Adda *Pré-segmentation de pages web et sélection de documents pertinents en Questions-Réponses*. TALN'13, 20ème conférence du traitement Automatique du Langage Naturel. Les Sables d'Olonne, France, Juin 2013.

1.6 Organisation du manuscrit

Ce manuscrit s'articule en 4 chapitres principaux : **le chapitre 2** traite de l'état de l'art des systèmes QR et en particulier des stratégies de RI (recherche et sélection de documents) employées dans ce domaine. Ce chapitre nous sert à définir le cadre de la thèse et son sujet. Nous introduisons les notions et les conventions du domaine en donnant au lecteur les clés pour lui permettre de répondre à la question : *qu'est-ce qu'un système de QR en domaine ouvert ?*

Le chapitre 3 présente une méthode statistique conçue dans le but d'évaluer la pertinence intrinsèque d'un texte pour la sélection de documents en QR. Cette méthode s'appuie sur un modèle de langue et un modèle de classification binaire de textes en catégorie *pertinent* ou *non pertinent* pour la sélection de documents en QR. La méthode est testée et évaluée dans le cadre QR à l'échelle des documents, sur un corpus de pages web français et un jeu de questions test fourni par le programme Quaero et le système RITEL du LIMSI.

Le chapitre 4 présente le système de segmentation textuelle de pages web que nous avons développé pour pouvoir appliquer la méthode d'évaluation de la pertinence intrinsèque d'un document, non plus à l'échelle des documents mais à l'échelle de segments (c.-à-d. sur des sous-parties de documents). Ce système est évalué à l'aide des mêmes corpus de test qu'au chapitre précédant et de RITEL. Nous présentons également dans ce chapitre une analyse détaillée de l'impact de la segmentation sur les performances de RITEL, pour chacune de ses étapes de traitement.

Le chapitre 5 présente l'application de la méthode d'évaluation de la pertinence intrinsèque d'un document à l'échelle des segments, grâce au système de segmentation textuelle présenté au chapitre précédant. Les évaluations sont menées sur les mêmes corpus de test et le même système QR que précédemment. Nous présentons également dans ce chapitre une étude approfondie des liens de dépendance apparents qui existent entre les performances QR de RITEL obtenues en évaluation à l'échelle des segments et les paramètres principaux des modèles sur lesquels la méthode s'appuie et, en particulier, sur les paramètres du modèle de langue que nous avons utilisés.

Nous clôturons ce manuscrit par la conclusion et les perspectives de recherches associés aux travaux présentés dans cette thèse, et sur l'ouverture du sujet traité dans un contexte plus large, celui des sciences cognitives.

Les versions complètes des différents extraits de pages web qui sont utilisés dans ce manuscrit pour supporter notre propos sont accessibles sur Internet à l'adresse suivante : `http://perso.limsi.fr/foucault/these/illustrations`. Nous avons préféré procéder ainsi, plutôt que de surcharger inutilement les sections où nous nous servons de ces extraits en en donnant les pages web au complet ou en produisant des annexes plus encombrantes que nécessaires.

« Information retrieval embraces the intellectual aspects of the description of information and its specification for search, and also whatever systems, technique, or machines that are employed to carry out the operation. » [105]

CALVIN NORTHRUP MOOERS

Chapitre 2

Les systèmes de Questions-Réponses

Sommaire

2.1 Introduction

Ce chapitre s'articule autour de trois sections principales. Tout d'abord, en adoptant un point de vue général, nous présentons dans la section 2.2 le domaine QR et son évolution depuis ses débuts dans les années 1950 jusqu'à nos jours. Ensuite, dans la section 2.3, nous abordons l'architecture et le fonctionnement des systèmes QR contemporains, de manière surfacique dans un premier temps, puis de manière approfondie autour des principales étapes de traitement d'une chaîne QR classique. Nous présentons également dans cette section les mesures les plus utilisées dans le domaine pour évaluer les performances des systèmes. Cette section permettra au lecteur de répondre à la question : *Qu'est-ce qu'un système QR en domaine ouvert ?*.

Finalement, la section 2.4 présente le système RITEL que nous avons utilisé durant ce travail de thèse. Son fonctionnement, en lien avec les notions et concepts qui auront été vus précédemment, est présenté. Nous prenons soin de détailler les spécificités et l'originalité de ce système.

2.2 Historique

Nous proposons un historique des systèmes QR et l'évolution de ce domaine dans le contexte du TAL. Nous mettons en perspective les théories ainsi que les campagnes d'évaluation qui nous ont semblé importantes pour comprendre l'essence même de la QR actuelle.

En **1950**, Turing est le premier à vouloir modéliser les rapports entre pensée et langage à l'aide d'une machine [148]. L'**intelligence artificielle** (IA) voit le jour quelques années plus tard en 1956 à l'occasion de l'école d'été organisée à l'université de Dartmouth. Les pères fondateurs de l'IA y sont tous réunis : Mc Carthy, Minsky, Newell et Simon. Les bases des premiers systèmes *intelligents* sont alors posées. L'intelligence de ces systèmes réside dans leur faculté à imiter le comportement humain, notamment pour ce qui touche au raisonnement, aux mathématiques, à la perception et la compréhension du langage. C'est dans ce contexte que les premiers travaux en **Traitement Automatique des Langues** (TAL) apparaissent et, avec eux, les premiers systèmes de questions-réponses (QR).

Les précurseurs des systèmes QR actuels remontent aux débuts des années **1960**. Simmons [140] en compte une quinzaine dès 1965, dont SAD-SAM [88] et BASEBALL [45]. Dans la même lignée, d'autres systèmes se succèdent jusqu'au début des années 1970, dont les plus connus sont : SIR [123], STUDENT [12] et LUNAR [157]. Tous ces systèmes sont présentés en détail dans [6]. Le système type tient en un programme, est dépourvu de modèles linguistiques et ne s'applique qu'en *domaine fermé †*. Les questions sont posées en langue naturelle et respectent de fortes contraintes. Par exemple, les questions posées à BASEBALL sont limitées à une clause sans connecteurs logiques ni comparatifs. Leur approche consiste à extraire des expressions et mots-clés trouvés dans les questions par **pattern-matching** lexico-syntaxique puis à les utiliser pour interroger une **base de données** en quête de réponses.

Si la philosophie des systèmes QR de l'époque est éloignée de celle des systèmes contemporains, il nous semble intéressant de souligner l'existence de SYNTHEX [139], absent de la plupart des études courantes dans le domaine, alors que ce dernier préfigure déjà de la philosophie QR employée aujourd'hui.

SYNTHEX a été créé par Simmons en 1961. Il répond à des questions factuelles en *domaine ouvert †* formulées librement en anglais. Il possède un index de phrases construit à partir d'un large corpus de textes : *The Golden Book Encyclopedia* [1]. Seuls les termes de contenu (c.-à-d. tout mot non fonctionnel) servent à l'indexation et aux recherches. L'approche de SYNTHEX suit trois étapes : interrogation de l'index à partir des termes de contenu trouvés dans la question, sélection des phrases pertinentes trouvées dans l'index (sur la base du nombre de termes communs avec la question) et génération de réponses pour chaque phrase sélectionnée. Ainsi, à la question *What do birds eat ?* et une phrase de l'index comme *Worms are eaten by birds*, SYNTHEX est capable de générer une réponse comme *Birds eat worms*.

Par la suite, des approches sémantiques sont peu à peu proposées et intégrées dans des systèmes de plus en plus modulaires mais qui continuent encore à ne s'appliquer qu'en domaine restreint. On voit désormais des modèles linguistiques intégrer les systèmes comme la théorie de Schank des dépendances conceptuelles [132], implémentée pour la première fois en 1973 dans MARGIE [135], puis réutilisée dans SAM [133] et PAM [155]. Les systèmes de l'époque utilisent des **structures de connaissances** dépendantes du domaine d'application et écrites à la main par des experts du domaine sous forme de règles et de bases de données.

1. http://en.wikipedia.org/wiki/Golden_Book_Encyclopedia

Cette évolution est liée au développement des systèmes de dialogue homme-machine (DHM) tels que SCHRDLU [156] et GUS [13], qui ouvrent la voie vers des études en TAL appliquées à des situations complexes en prise avec le monde réel. On peut citer comme exemple d'analyse complexe les tâches d'analyse thématique et d'analyse des intentions abordées plus tard dans des systèmes de dialogue avancés comme DIABOLO [110, 44]. SCHRDLU est un système que l'utilisateur dirige par la parole pour déplacer un bras virtuel et bouger des blocs de couleur sur une table. GUS est un système de dialogue qui permet de simuler des réservations de vols.

Ce dernier est souvent considéré comme le premier système orienté QR doué de « compréhension » du langage humain, au sens où GUS combine syntaxe, sémantique et raisonnement pour interpréter les demandes de l'utilisateur. Cette évolution est liée aussi aux théories émergentes en linguistique computationnelle, sous l'influence des sciences cognitives. En particulier, on peut citer les théories de Schank et Abelson [134] autour des *scripts*, *plans* et *buts* [2] intégrés dans SAM et PAM et celle des unités thématiques abstraites de Dyer [28] autour des *motivations* et *intentions* [3], chacune développée en psychologie cognitive pour la compréhension de texte.

Dans la continuité de SYNTHEX, Simmons crée en **1973** le premier algorithme générique de QR [141]. L'algorithme sélectionne dans une base de connaissances prédéfinies un ensemble de structures sémantiques pour une question donnée, afin de former une liste de réponses candidates. Les structures sélectionnées doivent partager les mêmes concepts lexicaux que ceux définis au niveau de la question. Ces structures représentent le sens des phrases. Ensuite, l'algorithme met en relation chaque réponse candidate avec la question, sélectionne les meilleures et produit des réponses.

Il faut attendre **1977** pour voir un système de QR capable de s'adapter en théorie à n'importe quel domaine d'application et voir l'usage de stratégies définies selon le type d'information recherchée, comme aujourd'hui en QR. C'est Lenhert qui la première le fera avec **QUALM** [83], un système de compréhension du langage supporté par SAM et PAM. QUALM lit les textes présentés par l'utilisateur, puis répond à ses questions à propos des histoires qui s'y trouvent racontées. Les questions posées ne portent que sur des faits (p. ex. le nombre de personnages décrits, le narrateur, le lieu du récit). Lenhert s'intéresse particulièrement au problème de la modélisation des questions. L'application de ses recherches en QR consiste à classer une question selon une hiérarchie de types génériques (p. ex. question de type *vérification*, *requête*, *procédure*) et, pour chaque classe de question, à définir des stratégies d'extraction spécifiques des réponses. Comme pour les autres systèmes de l'époque, QUALM sait où chercher les réponses, les textes lui étant fournis à l'avance. Dès lors, la **classification de questions** (QC) devient une composante majeure des systèmes de QR.

Les systèmes développés s'avèrent en pratique trop coûteux pour s'adapter à n'importe quel domaine d'application, car les structures de connaissances dont ils dépendent sont spécifiques à un domaine et nécessitent une adaptation importante pour être appliquées à un autre domaine.

Dans les années 80 et surtout 90, la notion de robustesse et d'analyse robuste est mise en avant. On passe de l'analyse phrastique complète de textes à des analyses de surface dont l'objectif est de caractériser les éléments rencontrés en fonction d'une typologie. Cette philosophie est celle appliquée dans les domaines de la **Recherche d'Information** (RI) et de l'**Extraction d'Information** (EI).

2. Les scripts décrivent des actions telles que : *aller au restaurant* ou *aller chez le docteur* en fonction des événements types qui les composent (p. ex. 1-aller au restaurant, 2-s'installer à une table, 3-lire le menu, 4-passer commande, etc.). Les plans sont eux aussi des structures de connaissances. Ils décrivent les moyens d'atteindre un but, étant donné un certain script.

3. Les unités de Dyer sont des structures de connaissances plus abstraites que les scripts.

En conséquence, les recherches en TAL migrent de l'étude de phénomènes linguistiques d'un point de vue cognitif à l'étude de phénomènes linguistiques du point de vue de la langue. Cette mutation engendre le développement de tâches de TAL dédiées à l'analyse de texte telles que le parsing de surface, l'annotation morpho-syntaxique. Le développement de ces tâches s'appuie très souvent sur des traitements statistiques.

Les systèmes de QR vont suivre cette tendance. En particulier, ils vont s'inspirer des avancées produites en EI dans le cadre des **campagnes d'évaluation MUC** organisées par la DARPA de 1987 (MUC-1 [144]) à 1998 (MUC-7 [93]). Dans ces campagnes d'évaluation, le but était d'extraire des informations trouvées dans des corpus de textes pour répondre à des requêtes, assimilables à des formulaires, à propos d'une thématique particulière (p. ex. le terrorisme dans MUC-3 [143] et les crash d'avions dans MUC-7). Pour ce faire, les systèmes d'EI s'appuient sur le repérage des **entités nommées** qui se trouvent dans les textes (c.-à-d. un mot ou bien un groupe de mots qui caractérisent par exemple un *lieu*, un *événement*, une *personne*). La détection des entités se fait principalement à l'aide de patrons d'extraction tels que : `<crash> s'est produit à <LIEU>`. En QR, ces entités sont utilisées pour caractériser le **type d'information à trouver afin d'extraire la partie de texte concernée, en fonction de la classe de la question**. Par exemple, à une question de type *qui*, on cherchera à détecter les entitées nommées de type *personne*.

De cette période, on peut aussi citer quelques réflexions intéressantes pour la problématique QR, comme les travaux de McCoy en QR conversationnel [94] sur la clarification et la correction de questions par rapport aux croyances de l'utilisateur, et les travaux de Kaplan en QR coopératif [68] sur la complétion de réponses pour la rectification des présupposés de l'utilisateur. Dans cette tendance, on trouve les travaux de Webber [154] sur la nature de la réponse à fournir à l'utilisateur et la distinction entre *réponse au sens propre* (c.-à-d. une réponse qui renseigne de façon stricte la demande de l'utilisateur) et *réponse étendue* (c.-à-d. *une réponse au sens propre*, si besoin accompagnée d'informations supplémentaires afin de clarifier, compléter, justifier et rectifier la demande de l'utilisateur). Enfin, on peut citer des travaux annonciateurs de futurs challenges QR tels que ceux proposés dans les campagnes d'évaluations TREC, comme les travaux de Wahlster [153] dédiés au traitement de questions *binaires* (c.-à-d. une question à laquelle la réponse est *oui* ou *non* ; *L'humain est-il un mammifère ?*) et de McKeown [95] sur la résolution de questions de *définition* (p. ex. *Qu'est-ce qu'un atome ?*).

Les années **1990** marquent le tournant en QR vers des **systèmes appliqués en domaine ouvert** (ODQA). Les systèmes adoptent la procédure de traitements standards suivante : analyse de la question, recherche d'information, extraction de réponses. Les contraintes de formulation des questions disparaissent et le nombre de types s'accroît (factuelle, définition, oui-non et d'autres encore sur lesquelles nous reviendrons). Lors des campagnes d'évaluation organisées dès la fin des années 90, les collections de documents sont figées, mais les systèmes peuvent rechercher des informations ailleurs que dans la collection pour valider une réponse trouvée. Le traitement des données issues du web constitue un nouveau défi à gérer pour les systèmes. En effet, ces derniers doivent par exemple régler les erreurs inhérentes à ce genre de données comme les erreurs d'encodage et éviter les baisses de performances qu'elles risquent d'amener.

Fréquemment, des ressources externes aux systèmes (p. ex. IMDB, Wikipedia[4] [71]) sont utilisées par ceux-ci pour aider les phases d'analyse et d'extraction de réponses. Certains travaux ont donné lieu à la constitution de bases encyclopédiques massives comme le *Cyc* de Lenhert [82] (3,3 millions de faits sur le monde et plus de 150K concepts) utilisé plutôt dans des applications industrielles (chez MySentient [22] et IBM avec PIQUANT [118]) et *WordNet*[5] [98] (environ 100K concepts) qui est très utilisé en TAL et, notamment, en QR.

4. http://www.wikipedia.org
5. http://wordnet.princeton.edu

WordNet est un réseau lexico-sémantique initialement défini pour l'anglais, à l'intérieur duquel les mots (p. ex. noms, verbes, adjectifs) sont regroupées dans des *synsets*. Tous les mots d'un synset sont vus comme des synonymes et correspondent à un concept d'ordre sémantique. Ils sont connectés les uns avec les autres par des relations conceptuelles (p. ex. hyperonymie) ou lexicales (p. ex. synonymie) permettant notamment de mesurer la distance qui sépare deux concepts.

L'un des premiers systèmes de type ODQA à utiliser WordNet est FAQFinder [16] de Tomuro. FAQFinder utilise WordNet pour affiner les recherches à l'intérieur d'une base de FAQ [16] . Si la question de l'utilisateur est suffisamment proche d'une question de la base de FAQ, alors la réponse qui lui correspond dans cette base est renvoyée à l'utilisateur. Ici, aucune recherche à l'intérieur des documents (blocs de réponses) n'est effectuée.

Sous l'impulsion des nouvelles approches développées en TAL et à l'intérêt renaissant accordé en général aux systèmes QR, un challenge en ODQA apparaît au sein des **campagnes d'évaluation TREC du NIST.** La première édition a lieu en 1999 [152]. Cet espace offre un cadre d'évaluation commun au développement des systèmes QR et marque l'essor de l'ODQA [57].

De nos jours, l'utilisation de **modèles statistiques** est de plus en plus fréquente en QR. Il suffit de regarder les travaux d'IBM pour s'en rendre compte et, en particulier, lors de leur participation aux campagnes QR de TREC avec le système PIQUANT [118, 19, 20, 18]. L'apprentissage de ces modèles rend nécessaire l'existence d'une très grande quantité de données correspondant à la tâche visée. Or il existe relativement peu de corpus spécifique de taille suffisante pour cela. Pour pallier cette carence, la recherche en QR s'est orientée vers des **systèmes hybrides** qui combinent théoriquement la force générique des modèles statistiques au savoir-faire des approches acquises en TAL ces dernières décennies. Un paradigme de système hybride efficace est le système QR hybride WATSON [33] d'IBM : grâce à ses modèles statistiques géants et une accumulation de connaissances générales et dédiées, IBM-Watson remporte en 2010 le jeu télévisé `Jeopardy!` [6] face à deux compétiteurs humain.

La figure 2.1 présente une chronologie récapitulative de l'évolution des systèmes QR, des années 50 à nos jours en mettant en avant les aspects TAL et les campagnes d'évaluation influentes en QA.

2.3 Architecture, fonctionnement et évaluation des systèmes QR

Nous présentons l'architecture et le fonctionnement des systèmes QR à la section 2.3.1, en mettant l'accent sur les principales étapes de traitement de la chaîne QR (sections 2.3.2, 2.3.3 et 2.3.4). Enfin, les mesures d'évaluation que nous avons utilisées durant nos travaux sont présentées à la section 2.3.5.

6. `http://www.jeopardy.com`, un jeu qui consiste à fournir des réponses aux participants qui trouvent la question associée.

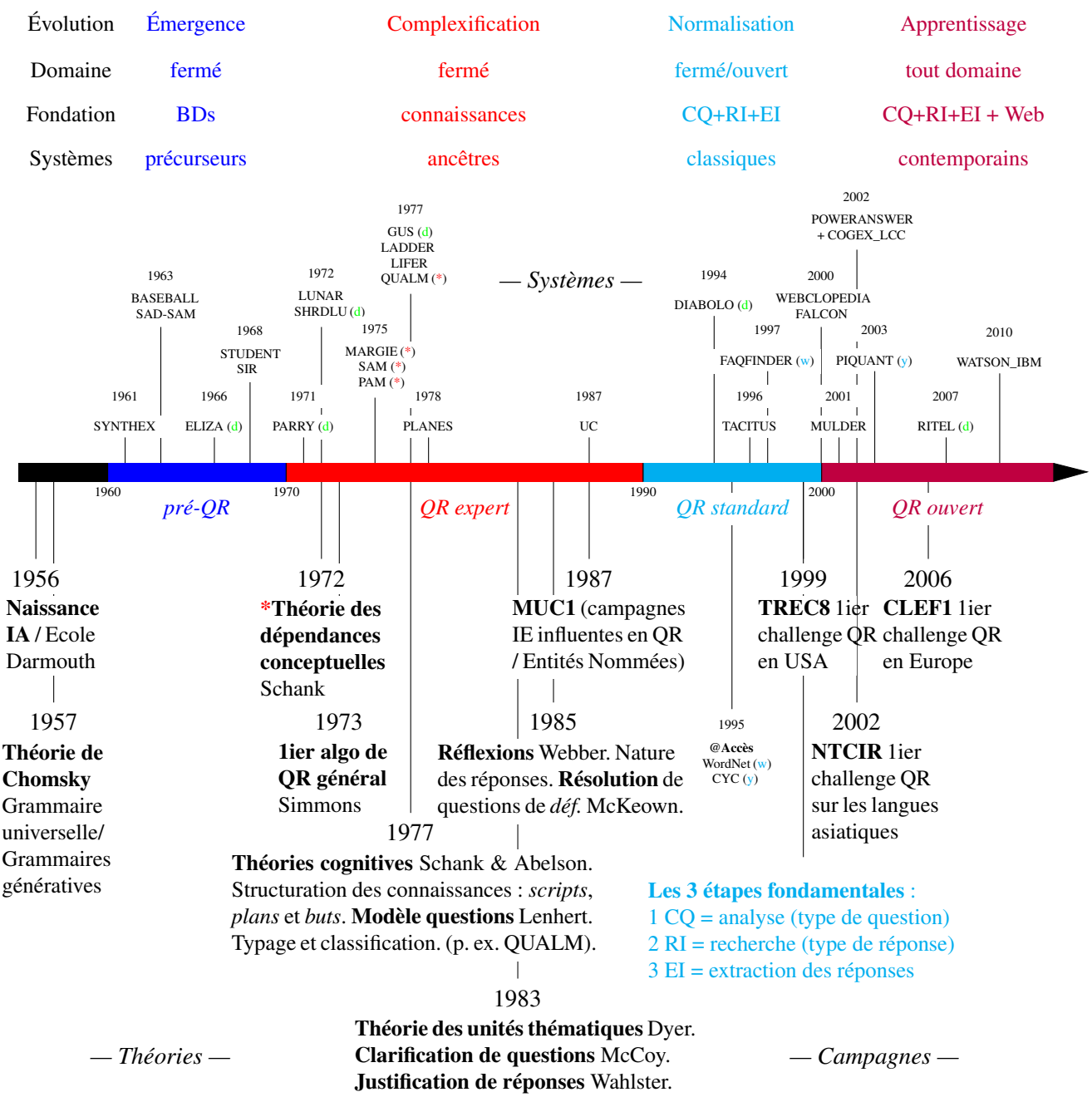


FIGURE 2.1 - Évolution du domaine Questions-Réponses. Partie haute de la chronologie : étapes majeures d'évolution, progression générale du domaine et apparition des systèmes. Partie base de la chronologie : évolution prise dans un contexte TAL large, vis-à-vis des théories et campagnes d'évaluations influentes en QR. (d) : système de dialogue.

2.3.1 Présentation générale des systèmes

L'architecture fonctionnelle classique des systèmes de QR, illustrée Figure 2.2, s'articule autour des trois grandes étapes de traitement suivantes [79, 43, 87] :

- l'**analyse de la question**. Cette étape consiste à examiner finement la question d'un utilisateur (le plus souvent d'un point de vue linguistique) en vue d'extraire les traits d'informations caractéristiques qui serviront aux recherches ;
- la **recherche d'information**. Cette étape consiste à chercher et sélectionner les documents (ou passages) les plus susceptibles de contenir des réponses à la question de l'utilisateur. Cette recherche s'effectue dans l'ensemble des documents dont dispose le système ;
- l'**extraction de réponses**. Cette étape consiste à détecter et à extraire dans les documents (ou passages) sélectionnés précédemment une ou plusieurs réponses destinées à l'utilisateur.

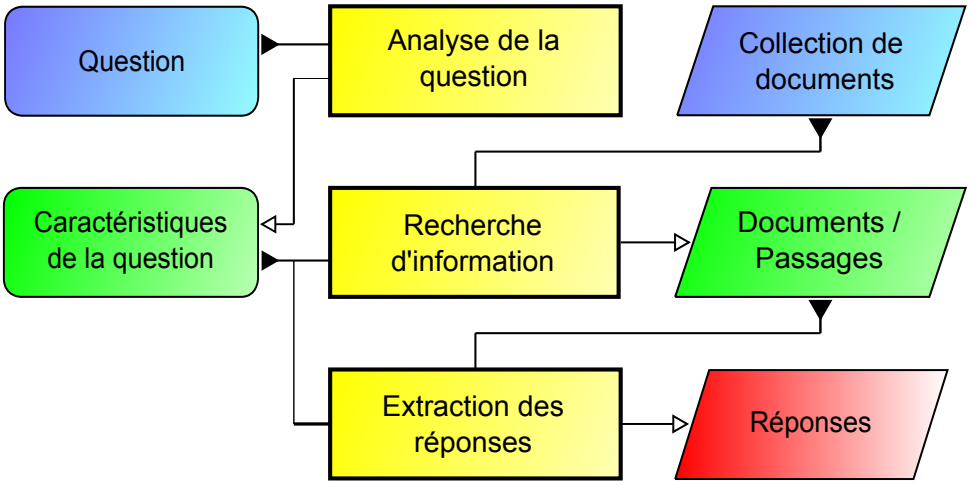


FIGURE 2.2 – Architecture fonctionnelle classique d'un système de QR. Bleu : les données fournies. Jaune : les modules de traitement. Vert / rouge : les résultats de traitements. Les résultats en arrondi ne dépendent que de la question, ceux en biseau dépendent de la question et des documents. Flèche noire : entrée de module. Flèche blanche : sortie de module.

L'**analyse de la question** est une procédure cruciale de la tâche QR puisqu'elle permet de spécifier *ce qu'il faut chercher* au cours de l'étape suivante de Recherche d'Information. L'analyse de la question permet d'extraire des traits caractéristiques qui sont utilisés lors de la recherche de documents et lors de la recherche et de l'extraction de candidats réponses. Ces traits sont utilisés pour créer des requêtes destinées au moteur de recherche. Puis, au cours de l'extraction des réponses, ces traits vont permettre de repérer dans les documents des éléments de réponses possibles. L'analyse de la question extrait, en particulier, des traits relatifs au type possible de réponses en fonction du genre de question. Par exemple le type de réponse attendu pour la question *Quand a été découverte l'Amérique ?* est DATE, car c'est le type de réponse auquel est habituellement associé une question de type *Quand*, alors que PERS (personne) est le type attendu de réponse pour la question *Qui a découvert l'Amérique ?*.

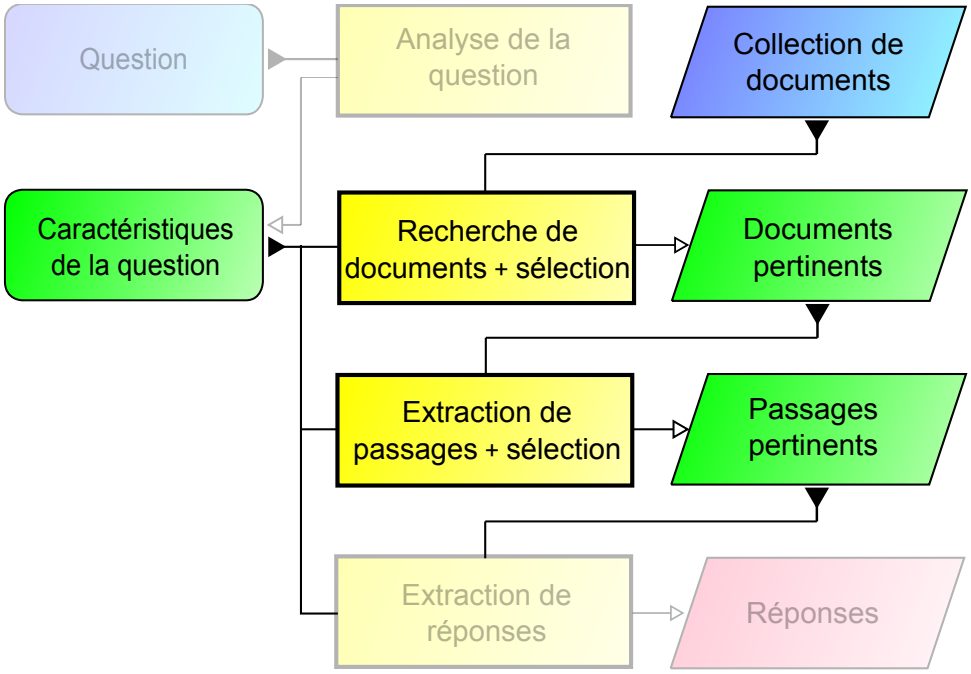


FIGURE 2.3 – Architecture fonctionnelle globale du module de Recherche d'Information (en clair), en connexion avec le reste du système QR (flouté). Bleu : les données fournies. Jaune : les modules de traitement. Vert / rouge : les résultats de traitement. Les résultats arrondis ne dépendent que de la question, ceux biseautés dépendent de la question et des documents. Flèche noire : entrée de module. Flèche blanche : sortie de module.

La **Recherche d'Information** a pour but de réduire l'espace de recherche lors de l'extraction des réponses. On peut décomposer cette étape en 4 phases comme illustré sur la figure 2.3 :

- La **première phase**, appelée *recherche de documents*, détermine l'ensemble des documents intéressants pour l'extraction de passages. Le moteur de recherche du système interroge une collection de documents grâce aux requêtes définies à partir de l'analyse de la question. Il récupère en retour une liste de documents.
- La **deuxième phase**, appelée *sélection de documents*, filtre les documents trouvés précédemment. Ce filtrage est fonction du degré d'information commun entre la question et les documents. Plus ce degré est grand, plus la pertinence d'un document est jugée élevée. Les documents les plus pertinents sont sélectionnés comme candidats à l'extraction des passages.
- La **troisième phase**, appelée *extraction de passages*, découpe les documents sélectionnés précédemment en passages d'un ou deux paragraphes tout au plus (parfois une phrase uniquement) [36, 121, 42]. La recherche de passages se fonde sur des heuristiques proches de celles utilisées pour la recherche des documents. Les passages collectés sont ensuite soumis à une sélection.
- La **dernière phase**, appelée *sélection de passages*, filtre l'ensemble des passages trouvés précédemment. Ce filtrage est fonction du degré d'information commun entre la question et les passages. Les passages les plus pertinents sont sélectionnés comme candidats à l'extraction de réponses.

L'**extraction de réponses** a pour but d'extraire des candidats réponses possibles des documents ou passages pertinents qui lui sont fournis au terme de la recherche d'information. Les stratégies employées au cours de cette étape sont fortement liées aux traits extraits au cours de l'analyse de la question, comme le type attendu des réponses. En effet, pour repérer dans les phrases analysées les éléments de la réponse, il est commun dans les systèmes de QR de détecter des *entités nommées* (EN). Comme mentionné à l'origine par Kripke en philosophie du langage dans sa théorie de la référence à propos des noms propres et du rapport entre signifiant et signifié pour le nommage [76], une EN s'apparente à un désignateur rigide qui réfère ou dénomme le même objet quelque soit l'univers considéré. Ainsi, les dates (p. ex. *12 octobre 1492*), les noms de personne (p. ex. *Christophe Colomb*), les noms de lieu (p. ex. *l'Amérique*) ou bien les noms d'organisation (p. ex. *République de Gênes*) sont des entités nommées. En QR, on a adopté la même référence pour désigner le type attendu de réponse et le type d'entités trouvées dans les documents. Par exemple, à chacune des EN citées plus haut, on associe respectivement les types : DATE, PERS, LOC (lieu) et ORG (organisation).

Si ces trois étapes régissent l'organisation traditionnelle des systèmes de QR, chaque système les implémente de façon différente et y intègre des composantes qui lui sont propres. C'est cette variabilité qui forme la richesse et la diversité architecturale des systèmes de QR, ainsi que nous pourrons le constater au fur et à mesure de notre discussion.

Certains systèmes possèdent un module d'**expansion de requête** [159, 8, 26], qui peut être utilisé par le module de recherche de documents, comme indiqué sur la figure 2.4. Il vise à enrichir les requêtes données au moteur chargé des recherches. Son objectif est d'élargir le champ des recherches du système (c.-à-d. améliorer son rappel). Cet enrichissement peut se faire par exemple en s'appuyant sur la lemmatisation des mots (par exemple, *explorent* → *explorer* ou encore *chevaux* → *cheval*) ou la recherche de synonymes. Ainsi, à partir de la requête *naviguer mer Christophe Colomb*, on a comme extensions possibles *naviguer traverser mer océan Christophe Colomb colonisateur coloniser colon*.

Certains systèmes (par exemple QRISTAL [81], FIDJI [106], RITEL) intègrent un module de **pré-traitement**, que nous présentons à la figure 2.5. Celui-ci est dédié aux traitements préalables à l'indexation des documents. Pour ce faire, les documents sont d'abords corrigés, transformés et réduits, puis enrichis. Ces deux phases de traitement des documents s'appellent la *normalisation* et l'*annotation*.

La **normalisation** consiste à préparer le contenu des documents avant l'annotation. C'est une tâche difficile, qui implique de nombreux traitements et qui conditionne le succès des systèmes dans leurs tâches. Selon la nature des données (textuelles versus multimédia), la normalisation se décline en deux types : *structurelle* et *textuelle*. Pour normaliser des documents issus du web, il est nécessaire d'appliquer les deux types de normalisation. La normalisation structurelle consiste à nettoyer le contenu source des pages web comme l'encodage, et les malformations dépendantes du format de données manipulées (p. ex. corriger les balises erronées dans les arbres de représentation Html et Xml). De plus, dans le cas de pages web, la publicité, le pied de page et les spams sont souvent éliminés. La normalisation textuelle, quant à elle, consiste à transformer un texte *brut* (p. ex. le contenu textuel extrait d'une page web) : l'orthographe et la ponctuation sont corrigées ; le texte est transformé dans une forme où les mots et les nombres sont correctement définis et délimités ; la ponctuation est séparée des mots et les phrases sont clairement formées. Des outils d'EI clé en main, spécifiques à l'extraction de contenu web sont dédiés à cette tâche (p. ex. Boilerpipe [74] et Justext [115]), mais aussi des *frameworks* de *crawling web* comme Scrapy ou des systèmes spécifiques de nettoyage de contenu web [29] (appelés « boilerplate removal ») comme ceux développés pour la campagne CLEANEVAL [5].

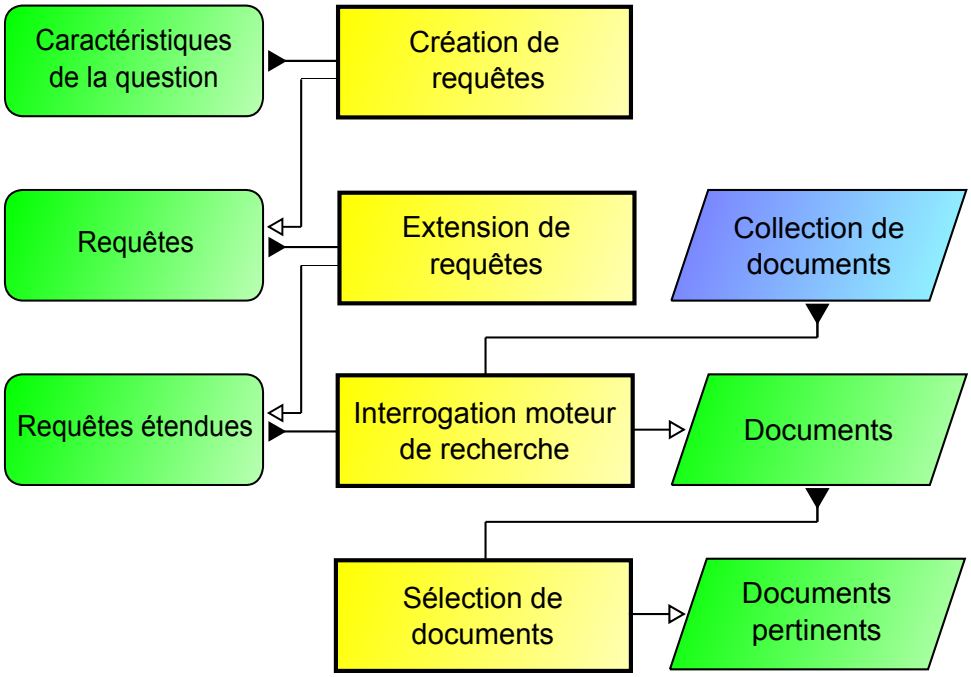


FIGURE 2.4 – Architecture fonctionnelle détaillée du module de recherche de documents avec expansion de requêtes, en lien avec le reste du module. Bleu : les données fournies. Jaune : les modules de traitement. Vert : les résultats de traitements. Les résultats arrondis ne dépendent que de la question, ceux biseautés dépendant de la question et des documents. Flèche noire : entrée de module. Flèche blanche : sortie de module.

L'**annotation** consiste à enrichir les documents pour aider les traitements d'extraction (p. ex. recherche de passages dans les documents pertinents, repérage de candidats réponses dans les meilleurs passages trouvés, et leur extraction). L'enrichissement des documents inclut entre autres le repérage des EN. L'annotation des EN se fait par la biais d'un jeu d'*étiquettes* (ou *labels*) correspondant aux types utilisés pour définir les différents type de la réponses. Par exemple, les étiquettes *loc* et *pers* de RITEL servent respectivement pour annoter des EN de type LOC et PERS, à l'analyse des documents comme à l'analyse des questions. Les labels d'annotation sont spécifiques à chaque système, bien qu'il puissent référer à des types d'EN communs à différents systèmes (voir section 2.3.2).

L'**indexation** consiste à stocker soit dans un (p. ex. FIDJI), soit dans plusieurs index (p. ex. QRISTAL) le contenu textuel des documents normalisés et annotés à l'aide de structures de représentation de texte compactes et simples d'accès, facilement exploitables et manipulables par un système en mémoire (contrairement à une stratégie qui manipulerait directement l'information d'origine des documents).

Certains systèmes QR appliquent des traitements suite à l'étape d'extraction des réponses, comme le *réordonnancement* et la *validation* de réponses.

Dans le cas où un système renvoie à l'utilisateur un ensemble de réponses, le **réordonnancement** a pour rôle de réordonner leur classement originel. Le but du réordonnancement est d'ajuster la position des réponses trouvées par un système de manière à placer dans le haut du classement les réponses jugées les plus pertinentes.

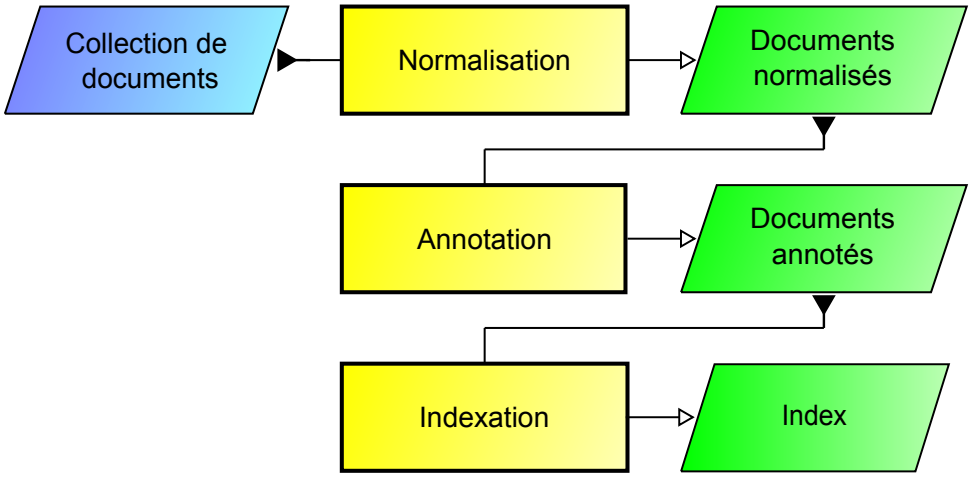


FIGURE 2.5 – Architecture fonctionnelle gobale du module de prétraitement. Jaune : les modules de traitements. Bleu : les données fournies. Vert : les résultats de traitements. Flèche noire : entrée de module. Flèche blanche : sortie de module.

Par exemple, le réordonnanceur de RITEL [9] réordonne chaque candidat réponse sur la base d'un calcul de similarité entre la question posée au système et le passage d'où il est extrait. Un algorithme de distance d'édition de type *Levenshtein* [7] [84] sert à mesurer la proximité entre une question et un passage grâce à un formalisme de structuration de phrases typées, organisées sous forme de constituants (ou *chunks*) syntaxiques hiérarchisés (sur lequel nous reviendrons plus loin).

La **validation de réponses** [2, 60] consiste à justifier la validité des réponses fournies par un système avant de procéder éventuellement à leur réordonnancement (p. ex. QAVAL [42]) ou dans l'optique d'évaluer automatiquement les réponses du système, comme dans le challenge AVE de la campagne d'évaluation CLEF [109, 113, 127]. Concrètement, la validation d'une réponse nécessite de connaître la question qui lui correspond et le fragment de texte duquel elle est extraite. L'approche générale consiste à considérer ce fragment comme un passage devant justifier la réponse [41]. Prenons par exemple le triplet de question, réponse et passage $< Q, R, P >$ suivant : Q : *Qui a le premier traversé l'Atlantique ?*, R : *Christophe Colomb*, P : *Christophe Colomb est la première personne de l'histoire moderne à traverser l'océan Atlantique*. Dans ce cas, la réponse est effectivement justifiée par le passage. Toutefois, on peut voir que si Christophe Colomb est bien énoncé dans le passage comme le premier à avoir traversé l'Atlantique, cela est exprimé d'une manière assez différente dans la question. Relier P et Q nécessite donc de nombreuses connaissances, par exemple savoir que l'Atlantique est un océan, que Christophe Colomb est une personne ou encore être capable de faire le lien de co-référence entre : (a) *le premier* (dans Q), et (b) *la première personne* et *Christophe Colomb* (dans R). C'est pourquoi la validation de réponses emploie des techniques relativement avancées du TAL, qui vont de la simple vérification entre le type d'EN identifiée et le type attendu de réponse (p. ex. YourQA, INAOE [66] et [55]) jusqu'à la résolution d'implications textuelles (p. ex. QAVAL). Pour valider une réponse par implication textuelle, une solution appliquée, par exemple dans le système de résolution d'inférences COGEX [145] du système POWERANSWER [103] développé au LCC, consiste à démontrer par réfutation que le passage justificatif implique à la fois la question et la réponse. Pour cela, l'idée est de définir dans un langage formel

7. Cette distance est une métrique appliquée à l'origine à des chaînes de caractères pour mesurer la différence entre deux séquences.

(p. ex. de type FOL), une hypothèse construite à partir de la question et une traduction du passage justificatif dans ce même formalisme. La mise en forme de l'hypothèse et du passage dans leur version formelle se fait par transformations automatiques à l'aide de règles de réécriture définies à la main [41], telle que des règles de substitution (p. ex. substitution du pronom interrogatif de la question par la référence correspondante trouvée dans la réponse). Enfin, un système de preuve, tel que le système de Gentzen [40, 39], est utilisé pour mener à bien la procédure de réfutation et valider/justifier ou non l'hypothèse.

On présente à la figure 2.6 l'architecture fonctionnelle avancée des systèmes de QR. Cette dernière correspond à celle de la figure 2.2, complétée par les modules de pré-processing et post-processing QR (validation/réordonnancement) décrits précédemment. C'est ce type d'architecture que suit le système RITEL utilisé dans cette thèse et présenté en fin de chapitre.

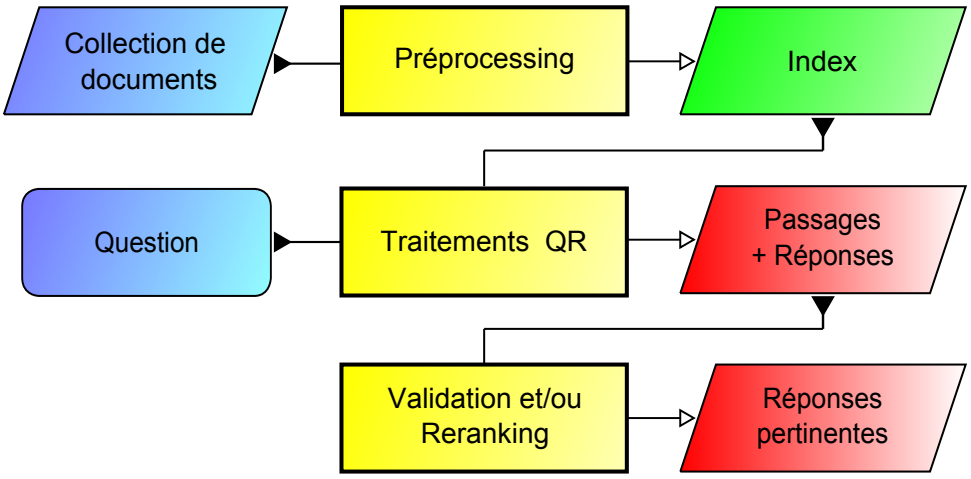


FIGURE 2.6 – Architecture fonctionnelle avancée des systèmes de QR. Jaune : les modules de traitements. Bleu : les données fournies. Vert / rouge : les résultats de traitements. Flèche noire : entrée de module. Flèche blanche : sortie de module.

Dans les sections suivantes, nous allons reprendre les traitements introduits précédemment, relatifs à l'analyse de la question, la Recherche d'Information et l'extraction de réponses, pour en donner un analyse détaillée respectivement aux sections 2.3.2, 2.3.3 et 2.3.4.

En l'absence de contexte dans lequel s'insèrerait la question (p. ex. au sein d'un dialogue dans le cadre de QR interactif ou en QR procédural où une série de questions et de réponses peut être échangée entre le système et l'utilisateur), les traits utilisés pour la recherche d'information et l'extraction de réponses ne dépendent que de la question. **C'est dans ce cadre que nous nous plaçons dans les sections qui suivent**.

2.3.2 Analyse de la question

L'analyse de la question est cruciale pour le succès d'un système QR car elle conditionne la réussite des étapes ultérieures de traitement. Elle permet de spécifier ce qu'il faut chercher (c.-à-d. les éléments de réponse).

L'analyse de la question se fait différemment d'un système à un autre. Néanmoins, il existe un certain nombre de traits (et de représentations) communs à la plupart des systèmes : des *traits lexico-syntaxiques* et des *traits sémantiques*. Les premiers dépendent de la représentation syntaxique de la question ainsi que d'informations morphologiques. Les seconds sont définis par des modèles sémantiques de la question, appuyés sur la syntaxe. Nous verrons que le pronom interrogatif joue un rôle prépondérant dans ces modèles.

Parmi les traits lexico-syntaxiques extraits à l'analyse d'une question, par exemple *Qui est Garry Kasparov ?*, on peut trouver des *mots-clés* (*Qui* ou bien *Kasparov*), les catégories morpho-syntaxiques (POS) des termes de la question (*Qui* : PROIms pour (pronom interrogatif), *Kasparov* : NPms (nom propre)) ; des fonctions grammaticales (*Kasparov* : sujet) ou encore des *relations de dépendance syntaxique* qu'entretiennent les mots-clés les uns avec les autres (c.-à-d. les relations de type prédicat-arguments ou des relations à longue distance, s'il y en a) et enfin, le *type de la question*.

Le type de la question correspond à la forme syntaxique de la question et sert dans le choix des patrons d'extraction de la réponse [43]. Ce trait est majoritairement caractérisé par le pronom interrogatif de la question. Par exemple, pour la question *Who is Garry Kasparov ?*, QALC [23] définira comme type de question *Who-Be-NP*.

Il est moins évident de lister avec précision les **traits sémantiques** extraits à l'analyse de la question, les études présentes dans la littérature étant parfois en désaccord sur la terminologie à employer pour les nommer, et les définitions à leur attribuer.

Cependant, il y a un certain accord à distinguer parmi les traits sémantiques extraits à l'analyse de la question : la *catégorie* [87, 121], la *classe (ou type attendu de réponse)* [83, 61, 23, 36, 121] et le *focus* [23, 83, 61] de la question. Nous allons détailler leur caractérisation.

La catégorie de la question correspond à la nature de la question et s'exprime sur la base du pronom interrogatif employé. La catégorie de la question conditionne le type de réponse attendu. Le type de réponse attendu est un trait essentiel à la réussite de la tâche en QR [101, 58, 85]. **L'analyse minimale** d'une question consiste donc à extraire la catégorie de la question et à en déduire le type attendu de la réponse.

Si la question porte sur un fait, sa catégorie est *factuelle* (p. ex. *What do eat birds ?*). Ces questions appellent une réponse courte en quelques mots [79] (*Birds eat worms*). Les entités nommées (EN) représentent très souvent le type de la réponse pour des questions factuelles. À la question *Où est Pondichéry ?*, le type attendu est donc LOC, car cette EN est caractéristique du pronom interrogatif *Où* pour une question factuelle. Certains interrogatifs marquent autant des questions factuelles que d'autres catégories de questions. Par exemple, *What* marque un fait dans *What do birds eat ?* et une définition dans *What is an atom ?* ; dans ce cas le type de la réponse correspond à la catégorie de la question (p. ex. DEF pour définition). De même, *pourquoi* et *comment* spécifient des questions *explicatives* (p. ex. *Pourquoi Galilée a-t-il été condamné ?*). L'aspect factuel d'une question sert comme critère discriminant aux systèmes pour guider la prédiction du type de réponse attendu en fonction de la catégorie de la question [23, 36, 121]. C'est le but du module de classification de question (QC) dont l'objectif cest d'attribuer à une question q une catégorie c parmi un ensemble de k catégories pré-définies de type de réponses. Ce module est implémenté dans la plupart des systèmes (p. ex. dans [85, 142, 121]).

Parfois, le type de réponse est spécifié à l'aide d'entités nommées plus précises que celles mentionnées plus tôt. Pour ce faire, les systèmes s'appuient sur des taxonomies de **classes de questions**, organisées en niveaux de granularité de plus en plus fin (p. ex. LOC couvrira des types plus fins tels que *pays* et *ville*. Ainsi une question comme *Où est Pondichéry ?* sera dans un premier temps classée *factuelle*, puis *LOC* et enfin, *pays*), par exemple les taxonomies de Li/Roth [85, 86] utilisée en QC (50 types), celle de Prager [117] (50 types) ou encore Hovy [59] (122 types). D'autres systèmes [23, 36] appliquent des taxonomies maison développées pour la détection et le repérage d'entités nommées en EI (la taxonomie de RITEL [36] avoisine les 300 entités).

Le focus de la question, quelle que soit l'approche considérée, détermine l'objet de la question et sert de point d'ancrage pour trouver la réponse. En fonction des approches, la nature du focus change, tantôt syntaxique (p. ex. QALC [23]) tantôt sémantique (p. ex. COGEX [145]) et sa forme varie (p. ex. lexicale ou conceptuelle). Dans QUALM [83], Lenhert définit le focus comme l'idée à laquelle se rapporte la question (le signifié [24]). Dans ce cas, le focus prend la forme du concept désigné par la classe de la question. Par exemple, dans *Quelle firme a créé DeepBlue, le système d'IA ayant battu le champion du monde d'échecs Garry Kasparov en 1997 ?* [8] le focus est *créateur* ou *concepteur* (l'entité nommée qui correspond à la réponse étant *ORG*, pour organisation). Dans QALC, Chalendar définit globalement le focus comme le sujet de la question. Le focus prend ici la forme des mots porteurs de cette fonction dans la question. Donc, pour la même question que précédemment, selon Chalendar, le focus est *Deep Blue* (l'élément de la question sur lequel porte la question).

Le tableau 2.1 contient des exemples illustrant ces différents aspects. On reprend ici la taxonomie de question du système YourQA comme illustrée par Quarteroni [120], suivant les mêmes exemples de questions, auxquels on a ajouté l'explicitation du focus en tenant compte de la différence de définition entre Chalendar et Lenhert.

Qcat	**Qclasse**	**Réponse attendue**	Exemple	**Focus**–*C*	**Focus**–*L*
F	PERS	*EN* type personne	*Who killed Lee Oswald ?*	Oswald	murderer
	LOC	*EN* type lieu	*Where is Inoco based ?*	Inoco	place
	TIME	expression temporelle	*When was the submarine invented ?*	submarine	epoch
	QTY	quantité numérique	*How fast can a Corvette go ?*	Corvette	speed
	ORG	*EN* type organisation	*What company has created DeepBlue ?*	DeepBlue	creator
	OBJ	entité générique	*What is Grenada's main commodity export ?*	Grenada	commodity
nonF	LIST	liste d'entités	*What were Columbus three ships ?*	ships	names
	DEF	définition/déscription	*What is platinum ?*	platinum	mineral
	HOW	explication	*How did Socrate die ?*	Socrate	death
	WHY	cause générique	*Why does the moon turn orange ?*	moon	reason
	WHY-F	raison (d'être connu)	*Who was Gandhi ?*	Gandhi	guide

TABLE 2.1 – Illustration des traits sémantiques clés extraits à l'analyse de la question via la taxonomie de questions du système YourQA. **Qcat** : catégorie de la question ; *factuelle* (F) ou *non-factuelle* (nonF). **Qclasse** : classe de la question (PERson, LOCation, TIME, QuantiTY, ORGanization, LIST, DEFinition, HOW, WHY et WHY-Famous. *EN* : entité nommée. **Focus** : focus de la question (*C* ou *L*, suivant la définition de *Chalendar* [23] ou de *Lenhert* [83], respectivement).

8. Réponse : IBM.

2.3.3 Recherche d'information

La recherche d'information a pour objectif de réduire l'espace de recherche au moment de l'étape d'extraction des réponses. Elle s'articule en deux fois deux temps : recherche et sélection de documents pertinents, puis recherche et sélection de passages pertinents extraits de ces premiers. Les traitements sont effectués grâce aux informations issues des documents eux-mêmes et de l'analyse de la question. Plus le degré d'information partagée entre le document (ou le passage) et la question est fort, plus les documents (et passages) sélectionnés à l'étape de Recherche d'Information seront pertinents pour l'extraction des réponses.

La recherche et la sélection des documents pertinents s'appuient sur les traits extraits à l'analyse de la question associés le plus souvent à des techniques d'expansion de requêtes. Par exemple, Magnini [91], tout comme QRISTAL [81], WEBCLOPEDIA [59] et QALC [23], se sert des mots-clés de la question accompagnés de termes synonymes pour interroger un moteur le recherche booléen MG [7] qui sonde une collection fermée de documents pré-découpés en passages d'au plus 20 lignes. D'autres systèmes, tournés vers l'interrogation du web (p. ex. RITEL [36], YourQA [121] et MULDER [79], FALCON [158]) font de l'expansion de requêtes afin d'augmenter le spectre de leur recherche, en particulier, avec l'ajout de lemmes. À cette fin, MULDER utilise WordNet, là où RITEL se sert de lexiques et de dictionnaires linguistiques comme le DELAS [21].

Contrairement à MULDER et YourQA qui utilisent un moteur pré-existant, RITEL utilise son propre moteur d'indexation et d'interrogation afin de bénéficier d'une analyse textuelle fine pour sa recherche de documents [36]. QALC utilise une double indexation pour ses recherches. Ainsi, il se sert dans un premier temps de MG pour trouver un grand nombre de passages (circa 1000) à l'aide de stems [9] obtenus à partir des mots de la question, les ré-indexe avec Fastr [62] et procède à une série de requêtes étendues sur cette réindexation. Les requêtes faites à Fastr se basent sur des variantes complexes (morphologiques, syntaxiques, sémantiques) des termes de la question [32]. Les passages trouvés par Faster les plus proches de la question sont sélectionnés comme candidats pour l'étape d'extraction de réponses de QALC.

En général, les systèmes attribuent des scores de pertinence aux documents (ou passages extraits des index) selon leur proximité avec la question, par exemple selon la pondération des termes du document en fonction de ceux trouvés au sein de la question.

La recherche et la sélection des passages pertinents se basent aussi sur l'analyse de la question et le contenu informationnel des textes. Ici, le type attendu de la réponse joue encore un rôle crucial et sert souvent de critère d'exclusion des passages quand ces derniers ne la couvrent pas.

Pour ces traitements, les systèmes de QR se différencient des systèmes de RI. Contrairement aux analyses de « surface » réalisées en RI pour la sélection des documents selon des mesures de pertinence globale (à l'échelle des documents uniquement), la recherche et la sélection des passages pertinents en QR se base sur l'analyse minutieuse du contenu textuel des documents à un niveau linguistique, associée à des mesures de pertinence globale mais aussi locale, afin de mieux rendre compte de l'importance des informations sélectionnées à l'étape d'extraction de réponses.

L'objectif de la sélection consiste à préserver les meilleurs passages pour l'extraction des réponses. Par exemple [116, 36], des stratégies de sélection basées sur la densité commune des termes associés à la question et issus des passages peuvent être appliquées pour décider si un passage est pertinent ou non pour l'extraction de réponses. Certains systèmes extraient des phrases au lieu de passages (p. ex. QALC, FIDJI [106] et YourQA).

9. Version édulcorée du procédé de lemmatisation, cependant le stemming concerne la totalité d'un mot, pas seulement la flexion.

Chalendar (QALC) et Magnini pré-découpent les documents en blocs de texte au cours des pré-traitements QR avant leur indexation. Dans ce cas, on parle d'*index-time passaging* [10] [124]. Galibert avec RITEL procède différemment : il extrait des passages pertinents à la sélection des documents. Dans ce cas, on parle de *search-time passaging* [11]. Dans le cadre du découpage des documents en passage, on peut aussi distinguer les systèmes qui extraient un unique passage depuis les documents pertinents comme le système de Robertson [125] et ceux qui en extraient plusieurs comme chez Galibert et Moldovan [101].

Les systèmes index-time passaging utilisent souvent des mesures de pertinence orientées RI dans leur tâche (p. ex. TF-IDF [65], OKAPI BM25 [125] disponibles notamment dans Lemur [12] et Lucene [51] ou des variantes qui se fondent sur des graphes de dépendances basés sur les termes de la requête [161]) et des modèles vectoriels pour assurer la représentation des documents.

Les systèmes search-time passaging se fondent plutôt sur les traits d'information extraits à l'analyse de la question et au sein des documents, et de modèles linguistiques pour assurer la représentation des documents. Cependant, on peut noter quelques approches statistiques récentes pour la sélection de passages pertinents en QR, comme les approches de Ganesh [37], Khalid [73] et Deveau [27] basées sur des modèles de langues [63, 128], des mesures de divergence de Kullback-Leibler [78, 77] (similaire à celle de Laffartey et Zhai [80] pour la sélection de documents en RI) et des représentations N-grams du contenu textuel des documents.

De même que pour les travaux de segmentation textuelle [130, 53] en blocs thématiques, le travail d'extraction de passages en QR se fonde sur l'idée d'analyse de documents par *fenêtre glissante* [73] pour définir quelles zones du texte examiner et identifier des passages. Cette fenêtre délimite la taille des passages qui seront extraits s'ils sont jugés pertinents vis-à-vis de la question. Les passages extraits à l'aide d'une telle fenêtre sont *disjoints* (ou non chevauchants) quand elle admet des frontières de passages *strictes* (c.-à-d. chaque segment d'extraction contient du texte qui lui est unique) et *chevauchants* quand celle-ci admet des frontières de passages *floues* (c.-à-d. quand les segments d'extraction peuvent être redondants) [53, 73]. La fenêtre d'analyse opère soit sur la représentation du texte d'origine (p. ex. basée sur les paragraphes d'origine [73]) soit sur des blocs générés à partir des ces derniers. Par exemple, suivant l'algorithme de TextTiling de Hearst, les phrases de chaque paragraphe de texte à segmenter sont regroupées en blocs de pseudo-phrases au préalable.

Nous avons dit que Chalendar avec QALC ou bien Magnini pré-découpent les documents à l'indexation selon une taille fixe de 20 lignes maximum choisie arbitrairement. Ceci revient à fixer la taille de la fenêtre d'analyse des documents à l'avance et fait entrer les passages obtenus suivant ce genre d'approche dans la catégorie des passages disjoints. Cette façon de faire est souvent employée en QR, bien que certains systèmes comme GuruQA [116] et RITEL utilisent des fenêtres glissantes qui autorisent des extractions de passages chevauchant de taille variable définie par apprentissage. Chez RITEL, la taille des passages est variable au sens où le système se sert de plusieurs fenêtres d'analyse de taille fixe. Tiedemann [146], Khalid [73] et Hearst [53] définissent la taille des passages de manière empirique. Tiedemann [146] propose de segmenter un même document plusieurs fois, à l'aide d'une fenêtre glissante de taille variable. Dans son travail, il explore la possibilité de faire se chevaucher les passages entre eux plusieurs fois, afin d'amplifier l'effet de redondance des documents sur le système QR Joost [14].

10. Étant donné que les passages sont extraits avant les recherches, au cours de l'indexation des documents.
11. Étant donné que les passages sont extraits au cours des recherches.
12. http://www.lemurproject.org

2.3.4 Extraction de réponses

L'extraction de réponses a pour but de fouiller les passages pertinents fournis par la Recherche d'Information, d'en extraire des réponses, puis de sélectionner parmi ces dernières les plus pertinentes. Cette étape comme les précédentes se fonde sur les traits d'information extraits lors de l'analyse de la question (en particulier le type attendu de réponse) et sur le contenu des passages.

L'extraction de réponses se fait au niveau de la phrase. Les méthodes utilisées pour y arriver proviennent en majorité du domaine de l'Extraction d'Information. Elles s'appuient sur des patrons d'extraction centrés autour d'entités nommées et sur des scores de pertinence associés aux termes de la question et des passages où trouver les réponses, ainsi qu'à leurs termes dérivés (p. ex. lemmes, synonymes, concepts). Ces scores de pertinence s'appuient souvent sur ceux qui ont été attribués aux documents correspondants à l'étape de RI.

Une fois extraites des passages fournis par le module de RI, les réponses sont évaluées, classées puis soit soumises à l'utilisateur, soit transformées avant de lui être présentées. Le premier cas est courant dans le cadre de réponses à des questions factuelles [106, 120]. Le second cas se produit dans le cadre de réponses à des questions plus complexes ou quand on veut personnaliser les réponses faites à l'utilisateur. On peut alors utiliser un moteur de génération dédié à la formulation de réponses [38].

Pour extraire les réponses des passages, l'approche classique comprend deux étapes [59, 23, 36] : d'abord, repérer si on trouve dans un passage donné une EN du bon type (c.-à-d. du type ou d'un type compatible avec celui de la réponse) et voir s'il en existe plusieurs dans la même phrase, puis extraire les mots de la phrase les plus proches de ceux de la question.

Prenons pour exemple, la question (Q) et le passage (P) suivant et leurs analyses en EN :

(Q) Quand est né Christophe Colomb ?
→ Type attendu : DATE

(P) *Christophe Colomb (en italien, Cristofo Colombo) (né entre le 25 août et le 31 août 1451 à Gênes ...*
→ *<PERS> (en italien, <PERS>) (né entre le <DATE1> et le <DATE2> à <LOC> ...*

Dans cet exemple, le candidat-réponse pourra être *<DATE1>* (même si celui-ci est incomplet). Une réponse possible fournie à l'utilisateur serait : *25 août* (sans aucun traitement additionnel) ou *Christophe Colomb est né le 25 août.* (via l'utilisation d'un moteur de génération de réponses).

En l'absence d'EN du bon type (c.-à-d. du type attendu par la question) dans un passage, une approche par *pattern matching* est appliquée afin d'associer à une question donnée des formulations de réponses possibles [43], en catégorisant le type d'information recherchée. Par exemple, l'information recherchée porte-t-elle sur une cause, une conséquence, une explication, etc. L'idée ici est de couvrir le maximum de variations possibles dans la formulation des réponses, tant sur leur forme que sur le sens qu'elles véhiculent, sur la base de la forme et du sens véhiculés par la question.

La sélection des réponses consiste à déterminer les n candidats-réponses les plus pertinents, parmi l'ensemble des réponses extraites précédemment. n peut être fixé au préalable par apprentissage, arbitrairement comme dans QALC [23] ou encore sur la base d'étude du comportement utilisateur comme dans YourQA [120].

Cette sélection s'appuie souvent sur des *mesures de similarité* proches de celles employées en sélection de documents/passages en RI et par l'application de méthode de ré-ordonnancement et de validation de réponses.

Ces mesures se distinguent par le niveau linguistique auquel elles font appel. Les plus simples agissent à un niveau lexical (p. ex. comparaison entre les termes de la question et de la réponse par *sac de mots*), là où d'autres agissent à un niveau plus syntaxique, par exemple, les mesures basées sur des *N-grams* ou des *distances d'édition* [13]. Les plus abstraites agissent à un niveau sémantique. Par exemple, les mesures de type *Jian-Conrath* [14] [64] permettent un calcul de similarité entre les concepts exprimés dans la question et ceux issus de la réponse. Il est possible de combiner les différentes mesures obtenues en une seule mesure [120].

2.3.5 Évaluation des systèmes

En QR, et plus généralement en RI, le processus d'évalutation d'un système vise à évaluer sa qualité. Les mesures développées à cette fin ont mûri au fil du temps et, notamment, au cours des campagnes d'évaluation : historiquement TREC (USA) puis NTCIR (Asie) et CLEF (Europe) et, plus récemment, EQuer et QUAERO (France). En QR, ces mesures se basent sur les réponses fournies par les systèmes en fonction des questions qui leur sont posées et de critères caractéristiques de la tâche d'évaluation QR. Ces derniers sont nombreux, trop nombreux pour pouvoir être exhaustif. Nous n'en donnons ici qu'un aperçu (cf. Galibert [36] pour un panel plus large et détaillé de ces critères) et l'illustration des défis QR de la campagne TREC de 1999 [150] à 2007 [151] proposée à la figure 2.7 avec le détail des corpus associés présenté au tableau 2.2.

Parmi les critères relatifs aux questions, on a : la catégorie des questions (p. ex. factuelle, définition), la manière de les créer (p. ex. artificiellement, à partir de requêtes utilisateurs, par reformulation d'autres questions), leur style (questions enchaînées, question avec cible, question avec support), ainsi que leur nombre et leur difficulté.

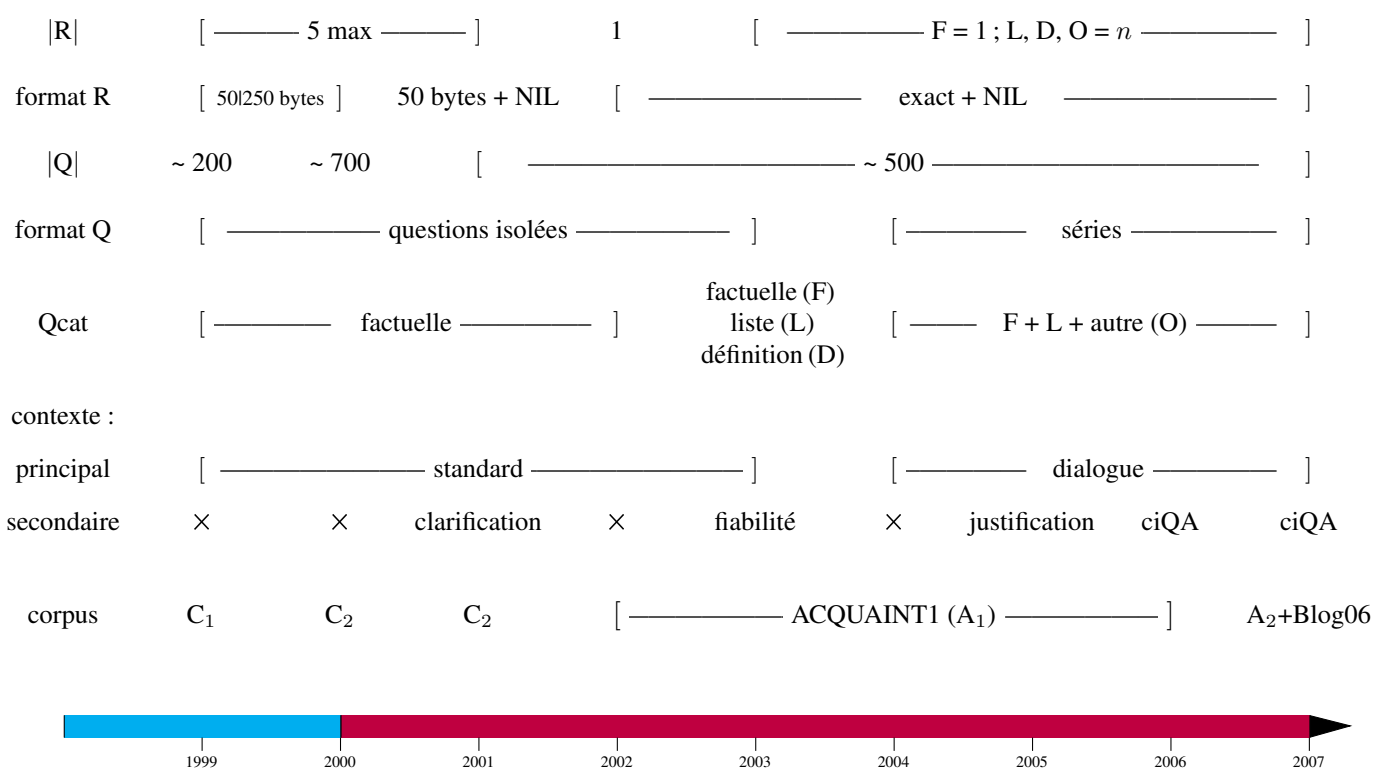
Parmi les critères relatifs aux recherches, on a : l'orientation des recherches (collection fermée/Internet), le type de documents utilisés (p. ex. multimédia, journaux), la nature de ces derniers (p. ex. style, sujet), des mesures quantitatives sur les corpus utilisés (p. ex. taille, nombre de mots, nombre de phrases) et l'échelle de ces derniers (restreint, large ou infini dans le cas du Web).

Parmi les critères relatifs aux réponses, on a : l'exactitude des réponses fournies (incorrecte, inexacte, correcte, non justifiée), le nombre de réponses autorisées par question (p. ex. une seule ou plusieurs), leur format (seule ou accompagnée du document réponse, d'un passage justificatif et d'un score de confiance [15]) et enfin, la taille des réponses (p. ex. court passage, chaîne de caractères limitée en taille).

Finalement, le contexte dans lequel évoluent les systèmes pour une campagne donnée constitue un dernier critère d'évaluation (p. ex. QR dans un contexte de clarification, d'interaction ou de recherche multilingue).

13. Ces distances sont établies à partir des phrases de textes ou des représentations syntaxiques de la question et des réponses.
14. Ce genre de mesure a été développé pour fonctionner sur des bases de données lexicales type *WordNet* par Jiang et Conrath.
15. Ce score sert à estimer la probabilité que la réponse soit juste d'après le système.

(a)



(b)

	1999	2000	2001	2002	2003	2004	2005	2006	2007
Référence : (LCC)	.55 MRR #2	.58 MRR #1 **Falcon**	.57 MRR #2	.85 CWS #1 PA1 [100]	.70 P #1 PA2 [50]	.77 P #1 PA2 [99]	.71 P #1 PA2 [49]	.57 P #1 PA3 [102]	.49 P #2 Chaucer-2 [54]
Autres :	.28 MRR #8 **GuruQA** (IBM)	.32 MRR #2 **Webclopedia**	.75 R #HC **Mulder**			.31 P #4 **Piquant II** (IBM)			.71 P #1 **PA4** (Lymba)
|P|	26	28	36	34	25	28	30	27	28

FIGURE 2.7 – Exemples de critères d'évaluation utilisés dans les campagnes QR (TREC 1999-2007). **(a) Critères**. |Q| et |R| : nombre total de questions et de réponses (distinctes) authorisées. format R : format de réponses authorisé ; en bytes (50 ou 250) ou en terme d'exactitude : correcte, incorrecte, etc. ou NIL (aucune réponse n'existe dans le corpus de travail). format Q : format de la question. contexte : challenges QR principal et secondaire (× : pas de challenge). Qcat : catégorie de question. corpus : corpus de travail. **(b) Performances de systèmes QR sur les campagnes TREC 1999-2007**. Le format des performances systèmes fournies est : score $_{\text{M \#rang}}$. M : la mesure QR considérée pour l'année. MRR : moyenne des rangs réciproques, P : précision, R : rappel et CWS : Combined Weighted Score. #rang : position des paraticipants dans le classement TREC de l'année (HC : hors compétition). Les résultats fournies de 1999 à 2001 ne concernent que les challenges avec réponses au format 50 bytes. Les résultats fournies de 2003 à 2007 ne concernent que les questions factuelles. En gras, les systèmes dont nous avons parlé dans le chapitre. PA : le système Power Answer du LCC (v1 à v3) puis de Lymba (v4). |P| : nombre de participants.

Corpus	**Création**	période	**Taille**	nb doc	**Source**	type	format
C1	Trec-qr99	[89-96]	1.9	0.528	FR, LAT+FT+FBIS (A)	news	article
C2	Trec-qr00	[88-96]	3.0	0.979	(A), AP, WSJ, SJMN	news	article
A1	Trec-qr02	[96-00]	3.0	1.033	AP + NYT + XNA (B)	news	article
A2	Trec-qr07	[04-06]	2.5	0.907	(B), AFP, CNA, LAT	news	article
Blog06	Trec-bg06	[05-06]	148	3.215	RSS et Atom (feeds)	blog	p + fu*

TABLE 2.2 – **Ci** : corpus TREC (NIST). **Ai** : corpus ACQUAINT (LDC). **Taille** : taille des corpus (Gigabytes). nb doc : nombre de documents (Million). **Source** et type : origine et genre des documents (blog : XML feeds, pages d'acceuils et permalinks correspondants, suivant différentes thématiques : news, sport, politique, santé, etc. ; intègre des spams). Format : format des documents (p + fu* : posts + follows-up, c.-à-d. des commentaires sous forme de permalinks). **Création** : date de création (Trec-qr et Trec-bg : campagne TREC, challenges QR et BLOG) et période de temps couverte

Mesures classiques versus mesures avancées

Les mesures d'évaluation habituellement utilisées en QR sont : la *précision*, le *top-n* [16] et la moyenne des rangs réciproques, définies équations (2.1), (2.2) et (2.3), respectivement. Ce sont les mesures que nous utilisons dans cette thèse. Les calculs de scores associés à ces 3 mesures reposent exclusivement sur des critères d'évaluation relatifs à la qualité des réponses fournies par les systèmes, en particulier, leur exactitude (voir précédemment). Nous présentons ces 3 mesures successivement, dans lesquelles on note par CR_i le rang de la première réponse correcte pour la question i. CR_i prend pour valeur $+\infty$ si aucune réponse correcte n'a été trouvée :

– la *précision* (ou P@1) est le ratio entre le nombre de réponses correctes et le nombre total de questions. Si le système est capable de fournir plusieurs réponses par question, on ne considère que la première :

$$\text{P@1} \quad = \quad \frac{\#\text{CR}_i = 1}{\#\text{questions}} \tag{2.1}$$

– le *top-n* (ou P@n) correspond à la précision définie sur les rangs 1 à n. Autrement dit, cette mesure nous permet de connaître la précision du système au travers du classement des n premières réponses apportées par le système relativement à chaque question évaluée, par exemple, avec $n = 10$ (fixée comme valeur standard dans nos expérimentations), on considère la densité de réponses justes selon les 10 premières réponses pour chaque question. La définition du *top-n* est :

$$\text{P@n} \quad = \quad \frac{\#\text{CR}_i \leq n}{\#\text{questions}} \tag{2.2}$$

– le *Mean Reciprocal Rank* (Moyenne des Rangs Réciproques ou MRR) permet de mesurer la qualité du classement des hypothèses (c.-à-d. réponses) effectuées par le système. La réponse correcte la mieux classée est pondérée par l'inverse de son rang initial. Une absence de réponse correcte entraîne une contribution nulle. Le score final est la moyenne de ces contributions :

$$\text{MRR} \quad = \quad \frac{\sum \frac{1}{\text{CR}_i}}{\#\text{questions}} \tag{2.3}$$

16. Nous emprunterons parfois le terme *rappel* comme terme synonyme de top-n dans ce manuscrit.

D'autres mesures existent, comme les mesures de *F-score* (combinaison moyennée précision/top-n) ou des mesures basées sur des calculs de scores plus complexes. Par exemple, les scores de type *K* qui visent à estimer le niveau de confiance qu'un système attribue à ses propres réponses. Il existe aussi des scores spécifiques à certaines catégories de questions (p. ex. les questions dites *complexes* comme les questions de *définition*) ou certaines approches (p. ex. à partir du τ de Kendall [72]). D'autres mesures combinent plusieurs scores d'évaluation et estiment des moyennes avancées (p. ex. *Confidence Weighted Score* [17]).

2.4 Présentation du système RITEL

Le système de questions-réponses (QR) que nous utilisons dans cette thèse s'appelle RITEL. Les propositions que nous allons détaillé dans cette thèse portent sur l'aspect Recherche d'Information de ce système.

Ce système est décrit en détails dans [36] et de façon compacte dans [10]. RITEL est né dans le cadre du projet Ritel. Il est conçu comme un *système de dialogue †* en domaine ouvert [147].

Nous présentons dans un premier temps la philosophie de ce système et son originalité à la section qui suit ainsi que son architecture. Puis, aux sections 2.4.2 et 2.4.3, nous décrivons le fonctionnement de RITEL en surface et en profondeur respectivement, vis-à-vis de ses étapes de traitements-clés.

2.4.1 Philosophie et originalité

La philosophie sous-jacente à RITEL est d'avoir une maîtrise totale des performances du système et notamment de la vitesse d'exécution, le système étant utilisé dans un cadre interactif et oral. Pour cela RITEL utilise plusieurs paramètres de contrôle de performance (dont les valeurs sont fixées en partie par apprentissage et en partie de manière empirique). Ces paramètres (parfois appelés *paramètres de tuning*) permettent de contrôler les temps de calcul nécessaires pour la réalisation des étapes critiques de la chaîne QR (c.-à-d. les différentes phases de sélection pertinente de documents, passages et réponses).

L'idée est que la quantité de travail réalisée par le système ne doit dépendre d'aucun parti pris théorique figé, mais du meilleur compromis possible entre rapidité d'exécution et performance du système (en terme de précision et rappel) dans le but de garantir sa robustesse en temps de traitement et en performance.

L'utilisation d'une **analyse unifiée multi-niveaux** et d'un **Descripteur De Recherche** (DDR) distingue RITEL des autres systèmes de QR et en a fait son originalité. L'analyse de RITEL structure l'ensemble des textes et questions fournis au système en entités individuelles ayant la forme d'arbres de type (les mots d'origine se trouvent au niveau des feuilles). La construction de ces arbres se base sur un moteur de création de règles (*Wmatch* [36]) dédié aux traitements de telles structures.

Le DDR est une structure représentant ce que le système a compris de la question utilisateur. Cette structure est éventuellement accompagnée d'entités complémentaires issues du contexte dans lequel la demande a été formulée (p. ex. un dialogue dans lequel s'inscrit la question). Cette structure est une représentation abstraite qui s'affranchit de la forme des entrées et qui décrit la recherche à effectuer dans le but de répondre à une question donnée. Elle est lisible et compréhensible par l'humain (ce qui facilite les expérimentations), complète et structurée (toutes les informations nécessaires pour les recherches y sont résumées) et joue un rôle clé dans l'attribution de scores de pertinence aux différentes phases de sélection d'information de la chaîne de traitement QR (sélection de documents, passages et réponses en fonction du contexte de la question).

17. Cf. Galibert [36] pour une présentation de toutes ces mesures, excepté le τ de Kendall.

On peut voir le DDR comme un *select* sur une base de données correspondant aux documents, qui servirait au système à trouver des réponses [36]. Dans une perspective de QR coopératif, avec l'intention de mutualiser les capacités spécifiques de recherche de différents systèmes, le DDR est une représentation pivot d'échange d'informations, dédiée à la communication des systèmes.

2.4.2 Généralités

RITEL s'appuit sur une analyse multi-niveaux appliquée à l'identique sur les questions et sur les documents. Cette analyse permet de repérer et typer des éléments d'information pertinents qui peuvent prendre la forme d'*entités nommées*, complexes et structurées, de chunks morpho-syntaxiques etc. Les documents sont d'abord analysés en totalité, puis indexés d'après les sorties de l'analyse. Les documents n'étant pas pré-découpés en passages à l'indexation, RITEL se situe dans la catégorie des systèmes à *search-time passaging*. Les recherches sont faites dans l'index complet.

À partir de l'analyse de la question, RITEL génère un *descripteur de recherche* (DDR) contenant toutes les informations utiles pour la recherche de documents et l'extraction de passages pertinents ainsi que des candidats réponses. Ces informations correspondent aux éléments de la question, leurs transformations possibles (p. ex. dérivations morphologiques) et les types de réponses attendus (p. ex. types d'EN). À chaque élément du DDR est associé un poids.

L'étape de Recherche d'Information de RITEL consiste à chercher les documents pertinents pour l'étape d'extraction de réponses. Dans un premier temps, RITEL interroge son index, sur la base des éléments d'information du DDR. Puis, dans un second temps, il sélectionne les n documents susceptibles de contenir une réponse, sur la base des éléments du DDR, de leur poids et d'un calcul fusionnant ces informations en un score de pertinence attribué à chaque document trouvé dans l'index. Ensuite, des passages sont extraits de chaque document pertinent via une *fenêtre d'analyse glissante à taille variable* fonction de la classe de la question, de paramètres de tuning et d'un algorithme de filtrage de l'information. Les passages obtenus sont donc de taille variable potentiellement *chevauchant*. Une fois extraits, ces passages sont scorés, selon le même principe que celui utilisé pour la sélection des documents pertinents. L'extraction et l'évaluation des candidats réponses s'appuient, de façon classique, sur la redondance de ces candidats, au sein des documents et des passages pertinents. À chacun des candidats réponses, un score est associé [10] et les k meilleures réponses sont choisies et retournées à l'utilisateur (réordonnées ou non) ou au module de dialogue de RITEL, réordonnées ou non.

La figure 2.8 présente l'architecture standard de RITEL.

2.4.3 Spécificités

Dans cette section nous détaillons le fonctionnement des différents modules de traitements de RITEL.

2.4.3.1 Traitements d'analyse et d'indexation

RITEL applique la même analyse sur les documents et les questions. Dans chaque cas, l'analyse dérive des arbres de *chunks* (éléments) sémantiques utilisés pour la recherche et pour l'extraction des réponses. Les chunks d'information pertinents extraits de ces arbres peuvent prendre la forme de type d'EN simples ou complexes (structurées ou non), de types morpho-syntaxiques, thématiques ou dialogiques (l'un comme l'autre permettant de répondre à certains besoins issus de la composante dialogue de RITEL).

L'analyse s'appuie sur une hiérachie d'environ 300 types organisés autour de ces 4 classes.

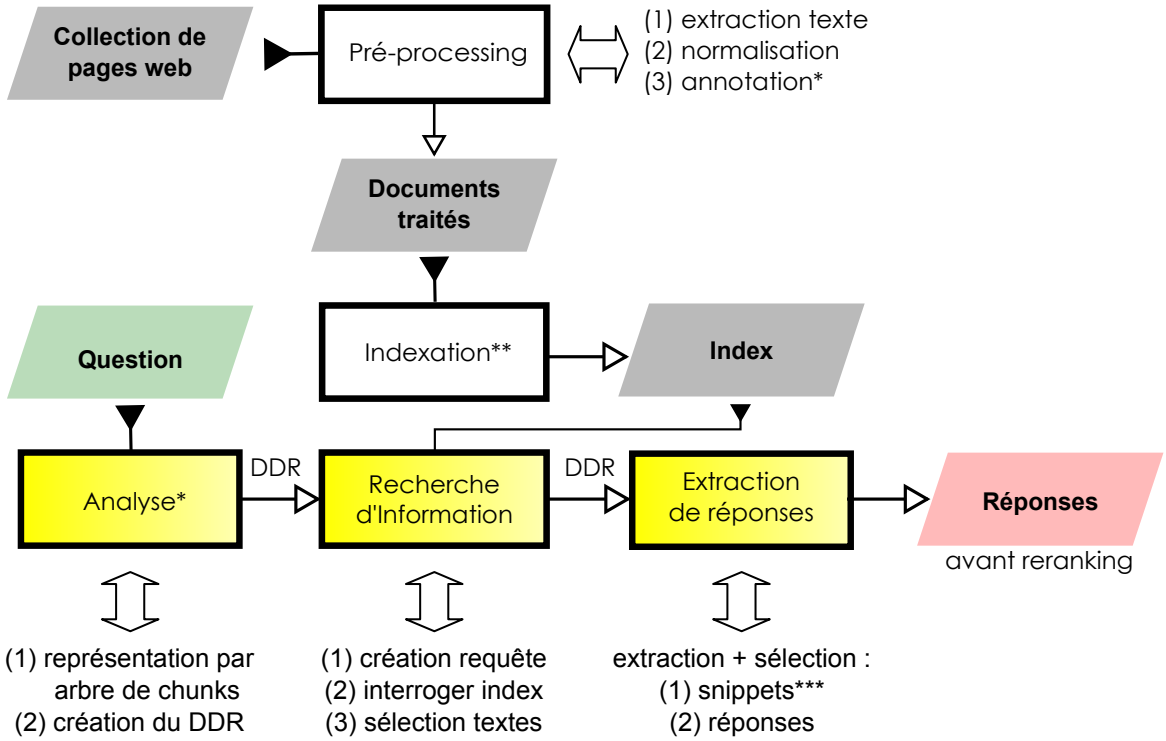


FIGURE 2.8 – Architecture fonctionnelle du système RITEL. Boîte jaune : module de traitements QR. Boîte blanche : module de prétraitements QR. Flèche noire : entrée de modules. Flèche blanche : sortie de module. Double flèche : phases de traitements. DDR : *Descripteur De Recherche*. * : signifie que l'analyse appliquée sur les documents au moment du pré-processing est la même que celle appliquée à l'analyse des questions avant les recherches. ** : indexation maison. *** : cette étape correspond à l'extraction de passages. Le reranking indiqué à l'étape d'extraction des réponses sur le schéma (boîte rose) n'est pas opérationnel dans la version standard de RITEL.

Les résultats d'analyse des documents sont indexés, afin d'accélérer les temps de traitement, à l'aide de tables de hashage sur des paires *(type, valeur)*. Des versions transformées de chacune des paires originales issues des documents sont associées à ces dernières au sein de l'index. Elles prennent la forme de procédures pré-traitées qui servent à guider et faciliter les dérivations linguistiques qui sont réalisées lors des recherches et à l'extraction des réponses. Pour chaque paire (transformée ou non) le nombre d'occurrences associées est lui aussi stocké dans l'index.

L'analyse de RITEL structure l'ensemble des questions et des textes en entités individuelles sous forme d'arbres typés qui s'apparentent aux représentations syntaxiques habituelles, mais à un niveau sémantique. La figure 2.9 donne un exemple d'arbre obtenu à l'issue de l'analyse. Chaque chunk est typé hiérarchiquement dans l'arbre de représentation d'une phrase ou d'une question. L'exemple de la figure 2.9 contient comme paires (type,valeur) la paire `(_pays,palestinien)` elle-même contenue dans la paire de plus haut niveau `(_orig,palestinien)` [18]. Ce sont toutes ces paires qui sont indexées.

18. Le type _orig signifie *origine* au sens large du terme : provenance, nationalité et origine de quelque chose ou bien de quelqu'un.

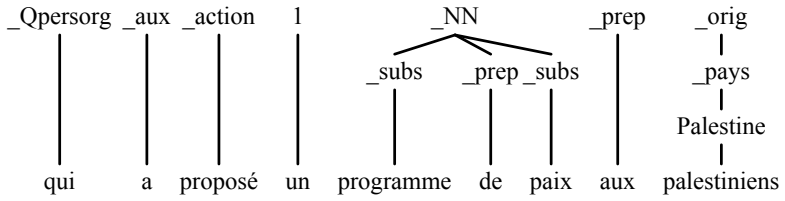


FIGURE 2.9 – Représentation arborée des hiérarchies de chunks produites par RITEL durant l'analyse.

2.4.3.2 Génération du *Descripteur De Recherche*

À partir de l'analyse des questions, RITEL crée une représentation abstraite, compréhensible par l'humain, de ce qu'il a saisi de la demande de l'utilisateur. Cette représentation est un *Descripteur De Recherche* (DDR).

Le DDR structure et organise, selon un ordre de priorité, l'ensemble des informations utiles aux différentes étapes de la chaîne de traitement QR. Il contient notamment les éléments importants de la question et leurs transformations possibles (dérivations de lemmes, de synonymes, de substantifs, etc.), la classe de la question et les types possibles de réponse pour la question posée (c.-à-d. correspondant à la classe de la question). Les types de la question et de la réponse sont déterminés par l'application d'un filtre contenant des patrons. Chaque élément du DDR reçoit un poids qui lui est propre. Les poids sont définis par une classification empirique du type d'information extrait selon son degré de pertinence pour les recherches. En plus des poids, certains éléments sont marqués *critiques* et d'autres *secondaires*, en fonction de leur nécessité pour la tâche QR. Tous les éléments du DDR peuvent se traduire sous la forme de paire *(type,valeur)* comme celles utilisées à l'indexation. Ceci permet, d'une part, de faciliter la comparaison entre les éléments issus du DDR et ceux issus de l'index au moment des recherches (documents, passages, réponses) et, d'autre part, d'obtenir un calcul direct des scores de pertinence lors des phases de sélection, sur la base des comptes de co-occurrences accompagnant chaque paire dans l'index.

Le DDR se construit en deux temps : *instanciation*, puis *expansion*. Au cours de la première phase, seuls les éléments critiques, secondaires, type(s) possible(s) de la réponse et leurs paramètres de tuning (c.-à-d. poids associés) sont définis, sur la base des éléments de la question. Le DDR une fois instancié est qualifié de *brut*. Au cours de la seconde phase, le DDR est enrichi par l'ajout de nouveaux éléments critiques et secondaires obtenus selon des jeux de transformations à partir des éléments du DDR brut. Ces nouveaux éléments sont soit juxtaposés aux anciens, soit attachés sous ces derniers ou bien sous certains des éléments nouvellement ajoutés. Le DDR après expansion est qualifié d'*étendu*. Sous cette forme, le DDR est structuré hiérarchiquement selon des niveaux de pertinence qui fixe le degré d'importance à leur accorder par la suite : des éléments les moins importants placés au plus bas niveau de la hiérarchie (c.-à-d. les éléments *subalternes* ou *sous-éléments*) aux éléments les plus importants placés au plus haut niveau de cette hiérarchie (c.-à-d. les éléments *principaux*). La position d'un élément dans cette hiérarchie module le degré de pertinence déjà associé au caractère critique ou secondaire que revêtent les éléments du DDR, ainsi que le poid associé à chacun.

2.4.3.3 Recherche et Extraction d'Information

Recherche d'Information La recherche des documents pertinents consiste à utiliser tous les éléments qui se trouvent dans le DDR comme des mots-clés afin de former une requête et interroger l'index de RITEL. Tous les documents renvoyés contiennent potentiellement des réponses. L'idée est ensuite de sélectionner parmi cet ensemble de documents les n documents les plus pertinents. L'approche de RITEL consiste à sélectionner ces n documents via un score de pertinence attribué à chacun, en fonction de leur contenu et des éléments du DDR. Le score de pertinence d'un document se calcule à l'aide des comptes d'occurrences associés à chaque paire *(type,valeur)* issue de l'index pour ce document et des paires du DDR. Pour ce faire, des séries de comptes et scores intermédiaires sont attribués à chaque élément du DDR, au fur et à mesure d'un parcours ascendant de la hiérarchie d'éléments du DDR. Le score intermédiaire de chaque élément dépend de son niveau dans la hiérarchie, des scores propres à chacun de ses sous-éléments (s'il en a), de son poids, et du caractère (critique ou secondaire) qui lui a été accordé à la création du DDR.

Extraction de réponses Dans cette étape, des passages pertinents pour l'extraction de réponses sont extraits des n documents sélectionnés à l'étape de Recherche d'Information. RITEL extrait des passages de taille variable dépendant de la catégorie de la question. À chaque passage est associé un score de pertinence suivant le même principe de scoring que celui présenté pour la sélection des documents pertinents à l'étape de RI. L'extraction et l'évaluation des candidats réponses s'appuient sur la redondance de ces candidats dans les documents et les passages. Tout élément trouvé dans un passage qui correspond au type attendu de réponse et qui ne correspond à aucun élément du DDR est considéré comme un candidat réponse potentiel [19]. À chacun d'eux, RITEL associe un score de pertinence $S(A)$:

$$S(A) = \frac{[w(A) \sum_E \max_{e=E} \frac{w(E)}{(1+d(e,A))^\alpha}]^{1-\gamma} \times S_{snip}^{\gamma}}{C_d(A)^\beta C_s(A)^\delta} \tag{2.4}$$

où :
- $d(e, A)$ = distance entre la réponse (A) et l'ensemble des éléments (E) du DDR
- C_s = nombre d'occurences de A dans l'ensemble des passages
- C_d = nombre d'occurences de A dans la collection de documents
- S_{snip} = score du passage
- $w(A)$ = poids du type de réponse et $w(E)$ poids de E du DDR
- α, β, γ et δ paramètres de tuning

Ceci revient à dire que chaque élément E du DDR s'additionne au score du candidat réponse ($\sum_E$) selon un rapport proportionnel à son poids ($w(E)$) et inversement proportionnel à sa distance avec le candidat réponse ($d(e, A)$). Cette distance correspond au nombre d'entités d'un passage qui se trouvent entre E et A. Si plusieurs instances du même élément se trouvent dans un passage, seule la meilleure est conservée ($\max_{e=E}$). Ce score est ensuite pondéré par le score du passage (S_{snip}) et normalisé par les fréquences du candidat réponse dans tous les documents (C_d) et les passages (C_s). Les scores de paires *(type,valeur)* identiques sont additionnés et amènent au score final de chaque candidat réponse.

19. En effet, le contraire reviendrait à dire qu'on cherche à extraire des passages des réponses exprimées sous une forme très proche de la question. Hors il est très rare qu'une question contienne elle-même sa réponse, à l'exception de questions comme *Quelle est la couleur du cheval blanc d'Henri IV ?*

« To be, or not to be : that is the question. » [137]

WILLIAM SHAKESPEARE

Chapitre 3

Modélisation du langage et pertinence de documents en QR

Sommaire

3.1 Introduction

NOUS avons vu à la section 2.3.3 qu'en RI les documents sont sélectionnés dans la plupart des cas sur la base des clés de requêtes, indépendamment de tout contexte associé à la requête correspondante, et que pour cette sélection, il existe diverses façon de procéder, parmi lesquelles l'utilisation de modèles de langue. Les modèles de langue, qui modélisent de quelle manière les mots de la langue s'agencent les uns avec les autres, sont utilisés en RI depuis de nombreuses années [80].
En contexte de RI pour du QR, nous avons vu à la section 1.3.3 du chapitre 1 que les modèles de langue sont aussi utilisés (voir par exemple [138]). En QR, leur utilisation sert surtout à évaluer la pertinence des passages extraits étant donnée une question [90, 37] dans le but de réordonner ces passages. Toutefois, il ne semble pas y avoir eu de travaux qui chercheraient à estimer la qualité d'un document pour une tâche QR indépendamment d'une question.

C'est ce que nous avons proposé et étudié dans ce chapitre. Nous décrivons une approche s'appuyant sur des modèles de langue pour aider un système QR orienté web dans sa tâche de sélection de documents.

Nous avons analysé des documents sélectionnés par le système RITEL lors de son étape de RI. On a pu constater lors de cette étude que certains documents a priori pertinents (car contenant en effet la réponse à une question) sont éliminés dès cette étape de la chaîne QR.

Nous présentons trois extraits de pages web (voir figures 3.1, 3.2 et 3.3) tirés de notre corpus d'évaluation sur lesquelles la sélection de documents de RITEL est en défaut. Ces extraits sont fournis pour montrer qu'il semble utile de filtrer les documents (a priori) pertinents sélectionnés par un système QR. Ils contiennent tous les trois la réponse à une question de test sur notre corpus d'évaluation. À chaque extrait, nous avons associé dans la légende qui lui correspond, la question à laquelle ce dernier est censé pouvoir répondre.

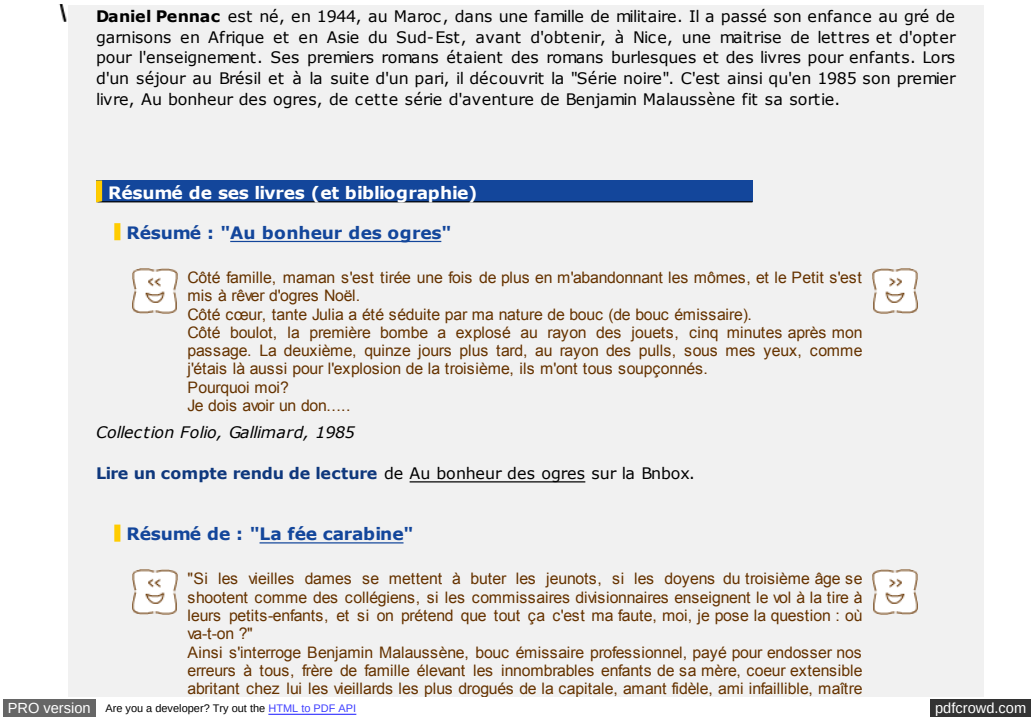
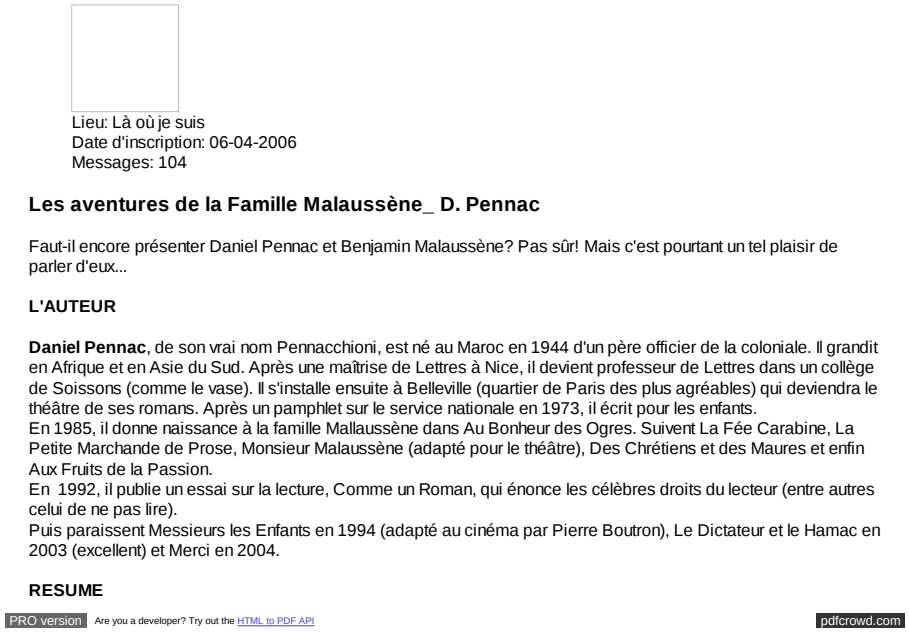


FIGURE 3.1 – Exemple de page non sélectionnée par RITEL pour la question *Quand Daniel Pennac est-il né ?*.

Il s'avère aussi que le système RI de RITEL sélectionne des documents totalement inadaptés à la tâche QR.

Les documents, notamment issus du web, peuvent contenir des informations pertinentes certes mais aussi des informations non pertinentes, qui s'apparentent pour un système à du bruit (*scories* issues des différents pré-traitements dont l'extraction de texte). Notre hypothèse est qu'utiliser des modèles de langue pour estimer l'adéquation entre un document et la tâche QR peut aider le système QR à ne sélectionner que des documents pertinents pour la tâche et donc à effectuer sa recherche de réponses dans de meilleures conditions. En effet, en filtrant a priori les documents, on évite de choisir une réponse (fausse) dans des documents non pertinents ; inversement en éliminant des documents non pertinents, on peut permettre à RITEL de conserver plus de documents dont certains auparavant rejetés contenaient une réponse valide.



Lieu: Là où je suis
Date d'inscription: 06-04-2006
Messages: 104

Les aventures de la Famille Malaussène_ D. Pennac

Faut-il encore présenter Daniel Pennac et Benjamin Malaussène? Pas sûr! Mais c'est pourtant un tel plaisir de parler d'eux...

L'AUTEUR

Daniel Pennac, de son vrai nom Pennacchioni, est né au Maroc en 1944 d'un père officier de la coloniale. Il grandit en Afrique et en Asie du Sud. Après une maîtrise de Lettres à Nice, il devient professeur de Lettres dans un collège de Soissons (comme le vase). Il s'installe ensuite à Belleville (quartier de Paris des plus agréables) qui deviendra le théâtre de ses romans. Après un pamphlet sur le service nationale en 1973, il écrit pour les enfants.
En 1985, il donne naissance à la famille Mallaussène dans Au Bonheur des Ogres. Suivent La Fée Carabine, La Petite Marchande de Prose, Monsieur Malaussène (adapté pour le théâtre), Des Chrétiens et des Maures et enfin Aux Fruits de la Passion.
En 1992, il publie un essai sur la lecture, Comme un Roman, qui énonce les célèbres droits du lecteur (entre autres celui de ne pas lire).
Puis paraissent Messieurs les Enfants en 1994 (adapté au cinéma par Pierre Boutron), Le Dictateur et le Hamac en 2003 (excellent) et Merci en 2004.

RESUME

FIGURE 3.2 – Autre exemple de page non sélectionnée par RITEL pour la question *Quand Daniel Pennac est-il né ?*.

Navigation : « ‹ › » [tous les artistes] [une fiche au hasard]

Claude François

Autres alias : Claude François et Frédérique Barkoff
Claude François et Martine Clemenceau

Nationalité : française

Date de naissance : 01/02/1939 (décès le 11/03/1978)

Liens : Le site officiel de Claude François

Se procurer ses disques :

Discographie :

On peut entendre sur Bide&Musique...

par années de sortie | par ordre alphabétique

1963 En rêvant à Noël

1967 L'Homme au traineau

1967 Mais quand le matin

FIGURE 3.3 – Exemple de page non sélectionnée par RITEL pour la question *À quelle date Claude François est-il mort ?*.

Ce chapitre s'articule en quatre sections. À la section 3.2, nous présentons une méthode statistique capable d'évaluer la pertinence intrinsèque d'une page web pour la sélection de documents en QR. La méthode s'appuie sur un modèle de langue et un système de classification spécifiques. La section 3.3 présente une adaptation de la méthode décrite en 3.2, où les paramètres du modèle sont choisis en fonction de la question. À la section 3.4, nous présentons l'évaluation de la méthode et deux expériences visant à contrôler les résultats obtenus. Les évaluations comme les contrôles sont effectués sur un corpus de 500k pages web en français, à l'aide du système RITEL. Enfin, en section 3.5, nous discutons les résultats d'évaluation obtenus et concluons à propos du travail mené.

3.2 Méthode d'évaluation de la pertinence intrinsèque d'un document

Nous présentons dans cette section une méthode permettant d'évaluer la pertinence *a priori* (ou intrinsèque) d'un document pour la sélection de documents en QR. Notre objectif est de décider a priori si un document est intéressant pour la tâche QR. Cette méthode repose sur l'utilisation d'un modèle de langue et d'un système de classification binaire de pages web en catégorie *pertinent* ou *non pertinent* pour les recherches en QR. Elle est conçue pour le filtrage des documents candidats à l'extraction de passages. La méthode devant être appliquée à des documents issus du web, il est nécessaire de commencer par extraire le contenu textuel de ces documents. Toutefois, dans cette section nous ne présentons pas la partie extraction qui fait l'objet d'une présentation dans le chapitre suivant, à la section 4.2.1.

L'articulation entre la méthode d'évaluation de la pertinence intrinsèque d'un document (méthode EPID) et le filtrage des documents en QR est expliqué à la section 3.2.1. La présentation générale de la méthode est donnée à la section 3.2.2 qui détaille le rôle du modèle de langue et du système de classification de pages web sur lesquels repose la méthode EPID. La section 3.2.3 détaille la création et les paramètres du modèle de langue. La section 3.2.4 détaille le système de classification des documents.

3.2.1 La méthode EPID pour le filtrage des documents en QR

La méthode EPID permet de filtrer les documents sélectionnés par un système QR, dans le but de ne retenir *in fine* que les documents réellement pertinents et de faciliter ainsi les phases d'extraction de passages puis d'extraction de candidats réponses. Un autre effet bénéfique de la méthode est d'accélérer les traitements QR à l'exécution en raison d'une quantité de données moins importante à traiter. Un autre avantage de notre méthode de filtrage est de pouvoir s'adapter à tout système QR.

Par facilité d'usage, nous emploierons donc souvent les termes *filtrage* et *méthode de filtrage* en référence à l'usage de la méthode EPID pour la sélection de documents en QR.

Intégration d'un module de filtrage à la chaîne QR Différentes approches étaient possibles pour mettre en œuvre la méthode proposée, par exemple intégrer le calcul de pertinence dans l'étape de sélection de documents. Nous avons plutôt opté, pour une raison de facilité expérimentale, pour l'ajout d'un module supplémentaire de filtrage des documents sélectionnés. Cette étape, que nous nommons **étape de filtrage**, intervient à l'issue de la phase de sélection de documents et avant les phases d'extraction de passages et de candidats réponses. Nous montrons son intégration sur la chaîne de traitement QR de RITEL à la figure 3.4.

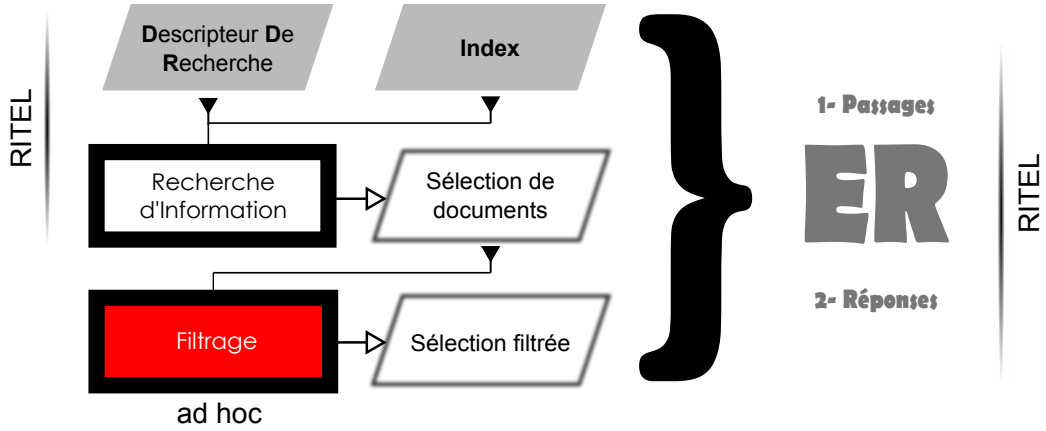


FIGURE 3.4 – Intégration du module de filtrage à la chaîne de traitements standard de RITEL.

3.2.2 Présentation générale de la méthode

But de la méthode La méthode EPID a pour but de déterminer si un document *d* est pertinent ou non pour les recherches en QR. La notion de pertinence est définie par la proximité entre le document à évaluer *d* et deux corpus : un corpus de documents « généraux » (GEN) et un corpus de pages web « spécifiques » à la tâche de QR (DEV509). Ce dernier corpus est constitué de documents supports de réponses correctes fournies par les systèmes ayant participé aux campagnes d'évaluation QR du programme Quaero de 2008 et 2009. Plus *d* sera proche des documents issus des corpus GEN et DEV509, plus celui-ci sera considéré comme pertinent pour les recherches en QR.

Application de la méthode L'application de la méthode EPID à un document *d* suit 2 étapes :

1. Estimation de la qualité intrinsèque de *d* par le biais d'un modèle de langue sous forme de scores. Les scores obtenus correspondent à différentes valeurs de paramètres du modèle de langue. Ce dernier est construit sur la base du corpus GEN.
2. Classification (binaire) de *d* comme document *pertinent* ou *non pertinent* pour la recherche en QR. Cette classification est automatique. Elle dépend des scores attribués à *d* à l'étape 1, et de seuils sur ces scores. Ces seuils sont déterminés par le modèle de langue utilisé à l'étape 1 et reposent sur les valeurs des scores calculées sur le corpus DEV509. Ce dernier sert de modèle de documents *a priori* pertinents pour les recherches en QR.

Seuls les documents jugés pertinents à l'issue de cette procédure servent de documents candidats à l'extraction des passages.

3.2.3 Scoring par modèle de langue

3.2.3.1 Construction du modèle

Corpus de documents généraux (GEN) Le modèle de langue est appris à l'aide d'une collection de 2Giga-mots d'articles de presse journalistique, dans un souci de couverture de vocabulaire. Le dictionnaire de mots extraits à partir de ces articles est de 500k mots. Tous ces articles sont en français. Pour 85%, ils proviennent de journaux d'information au format numérique tels que *Le Monde* et l'AFP. Le reste de ces articles est composé d'articles en provenance de sites de presse web tels que *Google news* et *Yahoo !*.

Propriété du modèle Le modèle est un modèle de langue 3-grams centré sur les mots, donnant la probabilité qu'un mot apparaisse à la suite de deux mots le précédant. Les probabilités du modèle sont obtenues sur des fréquences de co-occurrence de triplets de mots issus des documents du corpus GEN. Ce modèle est donc surfacique (il ne s'intéresse qu'aux mots).

Il est possible d'introduire plus d'abstraction dans le modèle si on intègre d'autres informations, telles que les parties du discours (POS) ou des catégories sémantiques. Cependant, nous avons préféré dans un premier temps nous focaliser sur l'étude d'un modèle surfacique, afin de mieux comprendre l'influence des paramètres sur le comportement de RITEL, avant d'envisager un modèle plus complexe.

3.2.3.2 Paramètres du modèle

Des expériences préliminaires nous ont permis de constater que les paramètres du modèle de langue les plus utiles pour l'évaluation de la qualité intrinsèque des documents sont la **perplexité** et le **ratio de mots hors vocabulaire**. En effet, des paramètres comme le nombre de mots par phrase ou le nombre total de phrases par document ne se sont pas révélés satisfaisants d'un point de vue QR (c.-à-d. que l'utilisation de ces paramètres pour le filtrage des documents ne permet pas à RITEL d'obtenir de meilleures performances).

La perplexité La perplexité (PPX) d'un document *d* d'après un modèle de langage (*Language Model, LM*) est définie selon l'équation (3.1).

$$PPX(d) = P_{LM}(d)^{\frac{1}{|d|}} \tag{3.1}$$

où $P_{LM}(d)$ est une probabilité estimée par le modèle de langue qui correspond à l'adéquation de d avec le modèle de langue. $|d|$ est le nombre total de mots qui se trouvent dans d. La perplexité permet de mesurer la ressemblance entre d et les textes d'apprentissage du modèle de langue (issus du corpus GEN dans notre cas).

Le ratio de mots hors vocabulaire Le ratio de mots hors vocabulaire (*Out Of Vocabulary, OOV*) d'un document *d* est défini suivant l'équation (3.2).

$$OOV(d) = \frac{\left|\overline{d \cap LM}\right|}{|d|} \tag{3.2}$$

où $|d \cap LM|$ correspond au nombre de mots de d appartenant au vocabulaire connu du modèle de langue (le vocabulaire constitué à partir des documents d'apprentissage du modèle). Au contraire, $\left|\overline{d \cap LM}\right|$ correspond au nombre de mots hors vocabulaire. Le ratio de mots hors vocabulaire correspond au nombre de mots inconnus du modèle de langue, divisé par le nombre total de mots se trouvant dans d.

3.2.4 Classification par seuils de pertinence

La méthode de classification que nous avons développée catégorise automatiquement un document en tant que document *pertinent* ou *non pertinent* pour les recherches en QR. Elle s'appuie sur les 2 mesures OOV et PPX définies précédemment.

Cette méthode utilise des fonctions de sélection qui permettent de calculer des seuils limites de pertinence au-delà desquels un document est jugé non pertinent. Elles sont définies sur la base des moyennes et des écarts types calculés à partir des distributions relatives à chacun de ces paramètres. Les distributions sont estimées sur l'ensemble des documents qui nous sert de modèle de pertinence pour la sélection des documents en QR (le corpus DEV509).

Les prédicteurs obtenus s'apparentent à des modèles de mélange de gaussiennes, mais n'en sont pas (nous n'avons pas implémenté stricto sensu de tels modèles dans cette thèse).

La fonction de sélection et les seuils limites de pertinence sont présentés à la section 3.2.4.3. Le calcul des distributions moyennes et des écarts types est présenté à la section 3.2.4.2. Le modèle servant à leur calcul est présenté à la section 3.2.4.1.

3.2.4.1 Modèle de pertinence

Le modèle de pertinence utilisé pour la classification est un ensemble de documents *a priori* pertinents pour la recherche en QR. Ce corpus est constitué de 509 pages web. Il a été constitué lors des campagnes d'évaluation QR du projet Quaero de 2008 et 2009. Ces pages contiennent des réponses correctes aux questions du corpus de développement de ces campagnes. Chaque document a été validé par un adjudicateur humain en tant que document-réponse pertinent (c.-à-d. comme un document contenant une réponse valide du point de vue d'une des questions d'évaluation).

Pour contrôler la pertinence des documents du corpus DEV509, nous avons pris soin de vérifier que moins de 10% de ces derniers étaient rejetés par notre système de classification.

3.2.4.2 Distributions, moyennes et écarts types

Les distributions, moyennes et écarts types utilisés pour la classification des documents, sont définis ainsi : les distributions sont estimées en attribuant à chacun des documents du corpus DEV509 un score de PPX et d'OOV via le modèle de langue (ces distributions sont considérées comme étant des gaussiennes) et les valeurs des moyennes et des écarts types relatives à OOV et PPX sont calculées à l'aide de leur distribution respective.

Variante d'estimation des distributions Nous avons défini une variante pour estimer les distributions OOV et PPX. Cette variante consiste à retirer les documents qui s'avèrent les moins pertinents du corpus DEV509, avant d'estimer les distributions relatives à chacun de ces paramètres. Les documents retirés sont en réalité des faux positifs (c.-à-d. des documents en marge du corpus DEV509), peu pertinents pour la sélection de documents en QR d'après nos vérifications manuelles (p. ex. des documents erronés).

L'identification des faux positifs dans le corpus DEV509 se fait grâce aux scores d'OOV et PPX fournis par le modèle de langue et des valeurs moyennes et d'écarts types estimées pour les deux paramètres sur l'ensemble des documents du corpus. Tout document qui présente des scores d'OOV et PPX soit trop inférieurs, soit trop supérieurs à la moyenne (c.-à-d. inférieurs ou supérieurs à 3 fois l'écart type correspondant), est considéré comme marginal et est exclu du corpus.

La méthode d'estimation des distributions par exclusion préalable du corpus DEV509 des documents marginaux est dite *restreinte* et dans le cas contraire, *normale*. On donne au tableau 3.1 les moyennes et écarts type calculés sur le corpus DEV509 pour les paramètres OOV et PPX en fonction de ces deux méthodes.

Méthode	**normale**		**restreinte**	
Param/Stat	M	SD	M	SD
OOV	1.74	1.98	1.46	1.12
PPX	210.2	252.9	187.6	106.1

TABLE 3.1 – Moyennes (M) et écarts types (SD) estimés pour les paramètres OOV et PPX sur les documents du corpus DEV509, en fonction des méthodes d'estimations des distributions OOV et PPX **normale** et **restreinte**.

3.2.4.3 Fonction de sélection et seuillage

Le calcul des seuils limites de pertinence au-delà desquels un document est classé *non pertinent* pour l'extraction de passage se fonde sur l'équation (3.3). Celle-ci suit une loi *normale*, puisque nous avons postulé que les distributions d'OOV et PPX sur lesquelles elle s'appuie suivent cette même loi (ces distributions sont gaussiennes). La fonction de sélection générique utilisée pour classer un document est définie à l'équation (3.4)

$$S_p = M_p + c \times SD_p \tag{3.3}$$

où $p \in \{OOV, PPX\}$; M_p et SD_p correspondent respectivement aux valeurs de moyenne et d'écart type liées à p (p fixé à OOV ou PPX) comme indiquées par le tableau 3.1 ; et $c \in \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$ [1]. La constante c permet d'influer sur le degré de tolérance du filtrage des documents. S_p est une valeur qui représente un seuil de pertinence au-delà duquel un document d est jugé *non pertinent* pour l'extraction de passages. Ainsi, un document sera retenu comme pertinent si :

$$d_p < S_p \tag{3.4}$$

où d_p est le score attribué par le modèle de langue au document *d* au cours de son évaluation (p fixé).

Plus $c \times SD_p$ est grand, plus la sélection sera permissive (c.-à-d. la fonction de sélection aura tendance à classer un document comme *pertinent*) et moins le filtrage des documents sera actif. Dans ce cas, la quantité de documents acceptés sera importante, et la plupart des documents sélectionnés initialement par le système lors des recherches seront conservés pour l'extraction de passages.

Combinaison entre fonction de sélection Nous avons testé une seconde fonction de sélection qui combine les seuils de pertinence d'OOV (S_{OOV}) et de PPX (S_{PPX}) pour réaliser la classification d'un document d. Cette fonction de sélection est définie à l'équation (3.5).

$$d_{OOV} < S_{OOV} \wedge d_{PPX} < S_{PPX} \tag{3.5}$$

On indiquera ce type de fonction sélection avec la notation OOV+PPX (oov+ppx ou plus simplement op) dans le reste de cette thèse.

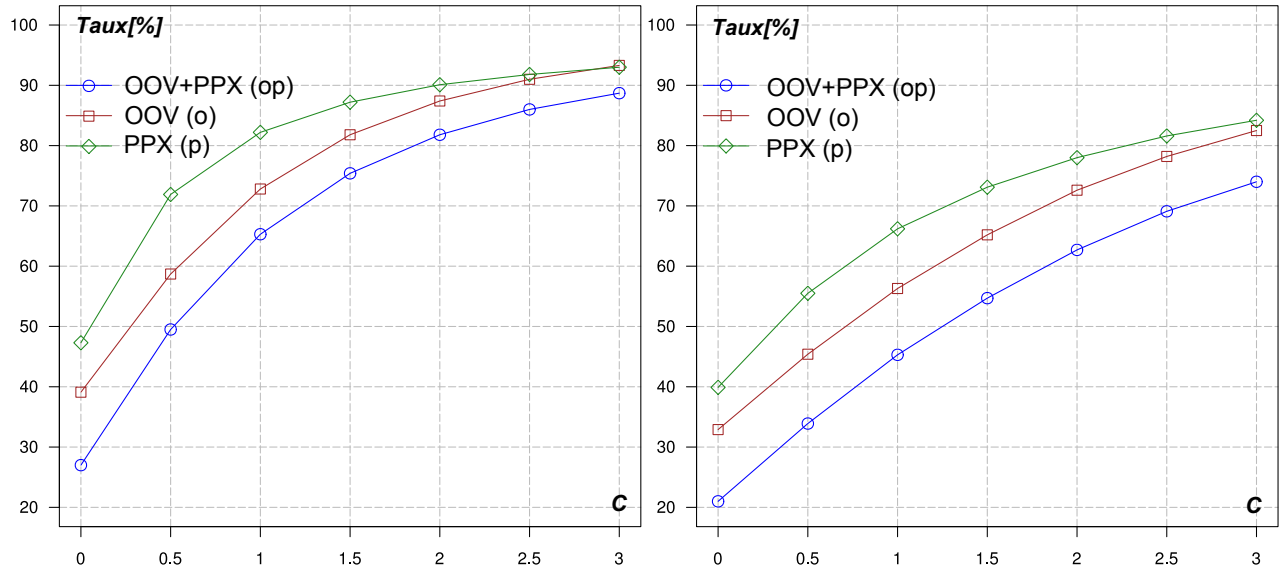
1. Ces valeurs ont été choisies au cours de tests préliminaires.

Dans les autres cas (c.-à-d. quand un document d est classé à l'aide de la fonction de sélection indiquée à l'équation (3.4) (c.-à-d. basée sur OOV ou PPX de façon non combinée), on utilisera les notations OOV (alternativement oov ou o) et PPX alternativement ppx ou p).

Variantes de sélection À chaque triplet $\langle f, m, c\rangle$ où f est une fonction de sélection (3 au total : oov, ppx et oov+ppx), m est une méthode d'estimation des distributions OOV et PPX sur les documents du corpus DEV509 (2 au total : méthode *normale* ou *restreinte*), et c est la constante de variation associée à SD_p dans f (7 valeurs possibles au total) correspond une variante de sélection différente. Il y a au total 42 (3*2*7) variantes de sélection possibles pour la classification des documents.

3.2.4.4 Classifieurs et listes de documents pertinents

Les classifieurs que nous utilisons dans la méthode EPID correspondent aux 42 variantes de sélection définies précédemment pour la classification des documents. Nous avons ainsi généré 42 listes de documents *a priori* pertinents d'un point de vue QR. La figure 3.5 donne le taux de sélection de documents *pertinents* par classifieur, après application de la méthode EPID sur le corpus de 499 734 pages web servant dans nos évaluations, en fonction des différentes composantes de sélection définies par le triplet $\langle f, m, c\rangle$.



(a) **[N]** (b) **[R]**

FIGURE 3.5 – Taux de sélection de documents (en pourcentage) à l'application de la méthode d'évaluation de la pertinence intrinsèque de documents sur le corpus d'évaluation Quaero *Q07fr* de 499 734 pages web. c : constante de sélection. oov, ppx et oov+ppx : fonction de sélection. **N** et **R** : méthode d'estimation *normale* en 3.5a versus *restreinte* en 3.5b des distributions OOV et PPX servant à établir les seuils de pertinence des fonctions de sélection.

Constat 1 On constate d'après la figure 3.5 que les taux de sélection de documents indiqués aux graphes 3.5a et 3.5b suivent bien le comportement des fonctions de sélection que nous avons décrites précédemment. En effet, on peut apprécier que plus la valeur de c est grande, plus les taux de sélection des classifieurs sont forts, quelle que soit la méthode suivie pour estimer les seuils limites de sélection des documents à partir du corpus DEV509. Ces taux oscillent **entre 25% et 80% pour la méthode normale** (N) et **entre 20% et 65% pour la méthode restreinte** (R) dans le premier intervalle de valeurs de c ($c \in [0, 1]$). Les taux correspondant aux autres valeurs de c ($c \in [1.5, 3]$) oscillent respectivement **entre 75% et 90%** et **entre 55% et 85%** pour chacune de ces méthodes. Ces valeurs sont résumés en table 3.2.

Les taux de sélection plus élevés dans le cas de N versus R sont dus aux valeurs de moyennes et écarts types relatifs à OOV et PPX indiquées au tableau 3.1 pour la construction des classifieurs. Ces valeurs étant plus fortes dans le cas de la méthode *normale*, elles rehaussent les seuils de pertinence qui servent pour la classification et la rendent plus lâche. En retour, un classifieur « normal » (ou de type **N**) a tendance à sélectionner plus de documents qu'un classifieur « restreint » (ou de type **R**).

Constat 2 On constate aussi d'après la figure 3.5 que l'impact de c sur les taux de sélection est plus important en général suivant la méthode *normale* que suivant la méthode *restreinte* : les taux de sélection observés aboutissent plus rapidement à des taux élevés dans le cas de la première méthode que dans le cas de la seconde. Il en va de même pour les taux de sélection en fonction du type de classification (oov+ppx, oov et ppx). Les écarts globaux en fonction des méthodes, du type de classification et des deux intervalles de valeurs de c mentionnées plus haut sont indiqués dans la table 3.2.

Il ne semble donc pas négligeable du point de vue de la classification d'avoir écarté du corpus DEV509 les documents marginaux avant d'établir les seuils de pertinence définissant les fonctions de sélection. Nous verrons ce qu'il en est du point de vue de la sélection des documents en QR au moment des évaluations.

c	$c \in [0, 1]$				$c \in [1.5, 3]$			
f	op	p	op	p	op	o	op	p
m	**N**		**R**		**N**		**R**	
	min	max	min	max	min	max	min	max
Taux [%]	27.0	82.2	21.0	66.2	75.4	93.3	54.7	84.2
intervalle	25.0	80.0	20.0	65.0	75.0	90.0	55.0	85.0
max-min	∼55		∼45		∼15		∼30	

TABLE 3.2 – Complément d'information sur les taux de sélection de documents (en pourcent) indiqués à la figure 3.5.

Afin de mesurer l'impact de la méthode EPID sur la sélection des documents en QR, nous avons intégré les 42 listes de documents *a priori* pertinents que nous avons créées précédemment au module de filtrage de RITEL. Les listes de documents les plus appropriées pour le filtrage sont confrontées aux documents sélectionnés par RITEL suite à l'étape de Recherche d'Information. Les documents communs sont sélectionnés comme candidats pertinents pour l'extraction de passages. Dans la section qui suit, nous expliquons comment RITEL détermine quelles listes sont les plus appropriées pour le filtrage.

	Loc	Nombre	Pers	Qdef_pers	Single	Subsnn	Time	Total
# Question	119	160	139	56	57	83	108	*722*

TABLE 3.3 – Répartition des questions utilisées pour l'estimation des paramètres de RITEL.

3.3 Méthodes d'appariements classe-liste

RITEL applique différents paramètres de tuning pour l'extraction des passages et des réponses en fonction des classes de questions. Une hypothèse vraisemblable consiste à penser qu'apparier les listes de documents de filtrage avec les différentes classes de questions utilisées par RITEL contribuerait à améliorer le système.

Entraînement de RITEL [36] RITEL utilise un corpus de 722 questions (décrit au tableau 3.3) en phase de tuning de ses paramètres de recherche et d'extraction d'information. Chacune des questions est accompagnée de ses réponses de référence. L'entraînement consiste à évaluer l'impact de chaque configuration de paramètres de tuning possible sur les performances QR (à la fois du point de vue des résultats QR finaux et intermédiaires) en fonction des différentes classes de questions. Dans chaque cas, les paramètres de tuning qui mènent aux meilleures performances deviennent les paramètres de référence utilisés par RITEL au moment des recherches.

Les performances du système dépendent des documents disponibles à l'étape de RI. Ainsi, il a semblé judicieux d'entraîner RITEL sur chaque liste de documents indépendamment des autres listes et d'associer aux diverses classes de questions les listes menant aux meilleurs résultats (en terme de précision, de MRR, puis de top-10) d'entraînement afin de garantir un appariement automatique (listes, classe de question) optimal.

Nous avons testé deux approches pour apparier les listes de documents pertinents aux classes de question (ou **appariement classe-liste**). Celles-ci sont décrites aux paragraphes qui suivent.

Méthode d'appariement 1 (Ma1) La même liste de documents est associée à chaque classe de question. Cette liste est celle qui mène RITEL aux meilleures performances globales à l'entraînement (c.-à-d. sans tenir compte d'une classe de questions particulière).

Méthode d'appariement 2 (Ma2) À chaque classe de question est associée une liste de documents. Une même liste peut être appariée à plusieurs classes à la fois. Chaque classe est appariée à la liste qui mène RITEL aux meilleures performances sur cette classe à l'entraînement.

Dans le cas de la méthode Ma2, seules les classes de questions dominantes au sein du corpus d'entraînement sont considérées. Ces classes sont :

- **Pers** (personne) ; **Loc** (lieu) ;
- **Qdef_pers** (une question à propos de quelqu'un) ;
- **Time** (temps, date, horaire) ;
- **Nombre** (une quantité numérique) ;
- et **Subsnn** (une question dont la réponse n'est ni une définition ni une EN étendue ; p. ex. le groupe nominal dans *Que mange une poule ?*).

À toute autre classe de question est associée la meilleure liste de documents en général.

Qclasse	#q	Question
Loc	66	*Où se situe Pondichéry ?*
Nombre	45	*Combien pèse la tour Eiffel ?*
Pers	55	*Qui est le chef de l'Akatsuki ?*
Qdef_pers	45	*Qui est Chuck Norris ?*
Subsnn	39	*Quels sont les secteurs qui recrutent ?*
Time	48	*Quand est sorti l'Amiga ?*

TABLE 3.4 – Répartition et exemples de questions à travers les classes dominantes du corpus d'évaluation de RITEL (*309* questions au total). **Qclasse** : classe de question. **#q** : nombre total de questions par classe de question. **Question** : exemple de question.

3.4 Évaluation

L'évaluation a pour but de mesurer l'apport de la méthode EPID en QR en étudiant l'impact du filtrage des documents sélectionnés par un système QR. Les évaluations se font avec le système RITEL sur un corpus de 500k pages web et un jeu de 309 questions fournis par le programme Quaero. Ces corpus sont présentés à la section 3.4.1. Le filtrage est réalisé à l'aide des listes de documents et des stratégies d'appariement **Ma1** et **Ma2** présentées à la section précédente.

La section 3.4.2 présente les résultats d'évaluation de la méthode EPID sur le système RITEL, en fonction des stratégies Ma1 et Ma2. La section 3.4.3 présente une série d'expériences qui visent à contrôler ces résultats. À travers ces deux sections, nous observons les résultats d'évaluation sous différents angles : du point de vue des performances, des listes de filtrage et des classes de question.

3.4.1 Corpus d'évaluation

Nous avons nommé le corpus de pages web utilisé dans nos expérimentations *Q07fr*. Ce dernier est composé de 499 734 pages web (5Gbytes) tout venant (blogues, forums, news, tutoriaux, articles d'encyclopédies du type Wikipedia, journaux, *e*-boutiques, etc.) en français. Ce corpus est utilisé pour les campagnes d'évaluation QR du projet Quaero [122]. Les jeux de questions de test et d'entraînement utilisés dans nos évaluations sont les mêmes que ceux utilisées lors de la campagne QR de 2010. Ils sont composées respectivement de 309 et 722 questions *factuelles*. Le tableau 3.4 présente un exemple de question pour chacune des 6 classes dominantes de questions dans ce corpus.

Le corpus de documents *Q07fr* a été collecté grâce au moteur de recherche Exalead [2] (par Exalead) sur la base de requêtes utilisateurs faites à ce moteur dans le courant de l'année 2007 [122]. Les questions utilisées lors de l'apprentissage et de l'évaluation de 2010 ont été constituées partiellement sur ces requêtes, par une équipe du LNE en charge de créer les jeux d'évaluation QR du projet Quaero [122]. Elles ne correspondent pas aux questions accompagnant les documents du corpus DEV509 qui nous sert de modèle de pertinence QR dans la méthode EPID.

2. http://www.exalead.com

3.4.2 Résultats

Dans cette section, nous présentons les résultats d'évaluation obtenus par le système RITEL, en lien avec les 2 méthodes d'appariement et les fonctions de sélection présentées dans ce chapitre. Ici, nous cherchons à vérifier le bien-fondé des méthodes d'appariements Ma1/Ma2 et à poser les bases d'une analyse profonde de la méthode EPID.

Dans un premier temps, nous interprétons les résultats du point de vue des performances d'évaluation en mettant l'accent sur les performances de rappel (top-10) [3] du système. Ensuite, nous interprétons les résultats du point de vue des classes de questions, puis des listes de filtrage. En fin de section, nous donnons un récapitulatif des résultats obtenus selon ces 3 niveaux d'interprétation et les discutons.

Résultats généraux Le tableau 3.5a présente les résultats obtenus par le système RITEL en terme de précision (**P@1**), **MRR** et top-10 (**P@10**), en fonction des méthodes d'appariement Ma1 et Ma2, et des différentes fonctions de sélection (o, p et op) utilisées pour la création des listes de documents pertinents. RITEL a servi à apparier les classes de questions aux listes de filtrage, c'est-à-dire aux 14 listes pour les 3 fonctions de sélection (o, p, et op). On a ainsi une condition de référence (bsln) sans filtrage et 6 conditions d'expérimentation différentes en croisant les 2 méthodes d'appariement aux 3 familles de listes possibles (colonne **Cond**). Par exemple, la condition **Ma1**.p correspond aux résultats obtenus par RITEL suivant Ma1 et une liste de filtrage de type p (c.-à-d. définie uniquement selon la perplexité).

Précision, MRR et rappel (top-10) Comme on peut le voir dans le tableau 3.5a, les résultats sont proches pour les 7 conditions testées avec cependant une très légère amélioration pour les conditions utilisant le filtrage. Ceci suggère à première vue que l'influence du filtrage sur la sélection des documents pertinents est faible. RITEL est toujours plus précis en condition **Ma1** qu'en condition **Ma2** et sans filtrage (**bsln**). C'est en condition **Ma1**.op qu'on observe les meilleures performances (**Ma1**.op : **33%** versus **bsln** : 31.7%). Mais on constate en terme de rappel (top-10), que c'est en condition **Ma2**.op qu'on observe les meilleures performances (**Ma2**.op : **56.0%** versus **bsln** : 53.4%).

Ces résultats suggèrent d'une part que l'impact de la méthode de filtrage sur la sélection de RITEL est plus important en terme de rappel que de précision. D'autre part, ils suggèrent que l'utilisation combinée des paramètres OOV et PPX (par l'utilisation d'une fonction de sélection de type op) prévaut sur leur utilisation séparée OOV (o) ou PPX (p). Nous avons vu précédemment que les taux de sélection des listes de filtrage de type op étaient les plus bas. Il semble donc que l'emploi de listes restrictives en filtrage (c.-à-d. contenant peu de documents), assure à RITEL de meilleures performances.

Focus sur le top-10 Le tableau 3.5b présente une vue détaillée du top-10 des réponses indiqué au tableau 3.5a. Si l'on se focalise sur le top-3, on constate que rechercher le meilleur équilibre entre quantité de réponses extraites et pertinence de ces réponses semble passer par l'utilisation d'un appariement classe-liste selon la méthode **Ma2**.

En effet, en regardant le classement des réponses de RITEL dans le top-3, on constate que les conditions de type **Ma2** permettent systématiquement de trouver un plus grand nombre de réponses pertinentes que les conditions **Ma1** équivalentes. La meilleure condition à cet égard est la condition **Ma2**.op (148 réponses dans le top-3 à comparer à 143 réponses pour la condition **Ma1**.op).

3. Nous approximons ici le rappel par la précision au rang 10.

Cond	P@1	MRR	P@10	#q
Ma1.op	**33.0**	40.6	55.0	309
Ma1.o	31.7	39.5	53.4	309
Ma1.p	31.4	38.7	53.7	309
bsln	**31.7**	39.5	**53.4**	309
Ma2.p	31.1	39.4	54.7	309
Ma2.op	30.4	39.7	**56.0**	309
Ma2.o	29.4	38.3	53.1	309

(a)

Cond	Ma1			bsln	Ma2		
rang	o	op	p	-	o	op	p
1	98	**102**	97	**98**	91	94	96
2	32	31	26	32	38	39	32
3	11	10	13	11	12	15	17
4-10	22	27	30	22	23	25	24
Total	165	170	166	**165**	164	**173**	169
1-3	141	143	136	141	141	148	145

(b)

TABLE 3.5 – (a) Résultats QR globaux par condition expérimentale (**Cond**). **P@1** : précision ; **MRR** : moyenne des rangs réciproques ; **P@10** : précision sur 10 rangs (ou top-10). **#q** : nombre total de questions évaluées. (b) Focus sur les résultats du top-10 donnés en (a), par position (**rang**) dans le top-10 (réponses justes uniquement). pos : classement des conditions.

Résultats par classe de question Le tableau 3.6a présente les résultats obtenus pour les 7 conditions présentées auparavant, en terme de précision (**P@1**) en croisant les méthodes d'appariement (**Ma1** et **Ma2**) et les fonctions de sélection (o, p et op). Ces résultats sont donnés en fonction des classes de questions dominantes du jeu de test et de la classe **glb** qui couvre toutes les classes.

Le tableau 3.6b présente les appariements classe-liste fait par RITEL pour chaque condition expérimentale, en fonction des différentes classes de questions considérées. Par exemple, dans le cas de **loc** et de **Ma1**.p, le tabeau indique que la liste de filtrage utilisé est de type « 2.5n » (c.-à-d. une liste de type normale « n », avec une constante de variation de « 2.5 »), le nom complet de la liste de filtrage étant « p2.5n ». Quand aucune liste n'apparaît (p. ex. pour **nbr** condition **Ma2**.p), on parlera d'appariement nul. Ceci se produit quand aucune liste ne permet à RITEL à l'apprentissage d'obtenir de meilleures performances avec filtrage que sans. Dans ce cas, le filtrage est inactif à l'exécution et le système se comporte comme en condition **bsln**.

D'après les résultats du tableau 3.6a, les conditions **Ma2** se distinguent peu de la condition **bsln**. En effet, jamais RITEL n'extrait plus de réponses pertinentes en condition **Ma2**. Le seul cas où les résultats sont équivalents à ceux obtenus en condition **Ma1** (**loc**), correspond au cas où le filtrage est inactif pour **Ma2** d'après le tableau 3.6b (par exemple sélection **Ma2**.p pour **nbr**). En revanche, on constate que la condition **Ma1** permet une légère amélioration des résultats.

En croisant les performances observées au tableau 3.6a avec les appariements donnés au tableau 3.6b, on remarque tout d'abord que les listes menant aux performances globales les plus élevées **op2.5n**, o2.5n et p2.5n, sont peu restrictives, même si l'une d'elle **op2.5n** combine les deux paramètres o et p et sélectionne plus que les deux autres. Leur taux de sélection est supérieur à 80% de documents. Ce résultat suggère que l'hypothèse émise sur l'usage de listes restrictives en filtrage pour améliorer les résultats QR est mitigée, si on regarde la précision seulement. On remarque par ailleurs que ces listes sont également celles qui mènent aux meilleures performances par classe de question. C'est par exemple le cas pour les classes **qpers** et **loc** en condition **Ma1**.p et **Ma1**.op respectivement. Ceci est contraire à nos heuristiques d'appariements et signifie que la liste de documents pertinents la plus appropriée pour le filtrage d'une classe donnée en contexte d'apprentissage ne semble pas la plus appropriée en contexte d'évaluation.

Cond$_{P@1}$	**glb**	**loc**	**nbr**	**pers**	**qpers**	**sub**	**time**
Ma1.op	**33.0**	54.5	33.3	**32.7**	**31.1**	2.6	**31.2**
Ma1.o	31.7	50.0	**35.6**	**32.7**	24.4	5.1	**31.2**
Ma1.p	31.4	**57.6**	**35.6**	27.3	24.4	2.6	27.1
bsln	31.7	50.0	**35.6**	**32.7**	24.4	**5.1**	**31.2**
Ma2.p	31.1	54.5	**35.6**	27.3	24.4	2.6	29.2
Ma2.op	30.4	51.5	**35.6**	27.3	24.4	2.6	29.2
Ma2.o	29.4	50.0	31.1	**32.7**	24.4	0.0	25.0
#q	309	66	45	55	45	39	48

(a)

Cond	**glb**	**loc**	**nbr**	**pers**	**qpers**	**sub**	**time**
Ma1.op	**2.5n**	2.5n	2.5n	2.5n	2.5n	2.5n	2.5n
Ma1.o	2.5n	2.5n	2.5n	2.5n	2.5n	2.5n	2.5n
Ma1.p	2.5n	2.5n	2.5n	2.5n	2.5n	2.5n	2.5n
bsln				-			
Ma2.p	2.5n	2n	-	2.5n	-	3n	1.5r
Ma2.op	2.5n	2n	-	3n	-	3n	1r
Ma2.o	3n	3r	2.5n	3n	3r	3n	1r

(b)

TABLE 3.6 – (a) Résultats QR globaux (**glb**) et par classe de question (**loc**, **nbr**, etc.) pour chaque condition expérimentale (**Cond**) en terme de précision ($P@1$). (b) Focus sur les appariements classe-liste obtenus d'après les résultats d'entraînement de RITEL pour chaque condition expérimentale en (a).

Ce dernier résultat soulève plusieurs hypothèses d'explication. La première hypothèse est que définir les appariements classe-liste en fonction des performances d'apprentissage n'est pas un bon critère pour garantir des performances optimales en condition d'évaluation. La deuxième hypothèse est que le jeu de questions d'apprentissage utilisé pour définir les appariements classe-liste automatiquement lors de la phase de tuning des différents paramètres de RITEL n'est pas de taille suffisante. Dans ce cas, on se trouve en situation de surapprentissage. La troisième hypothèse est que le choix d'appariement sur la catégorie des questions n'est pas suffisamment pertinent vis-à-vis du filtrage. Dans ce cas, un appariement classe-liste fondé sur le type de réponse attendue par la question pourrait peut-être mieux répondre à notre problème. Néanmoins comme à chaque classe de questions peut correspondre plusieurs types de réponse attendue, nous aurions toujours le même problème lié à la quantité de données d'apprentissage. Une autre possibilité serait de s'appuyer sur une classification différente des questions, par exemple thématique.

Récapitulatif et discussion Les performances globales sont relativement proches avec et sans filtrage. Le bénéfice apporté par le filtrage sur les performances de RITEL est plus important en rappel qu'en précision. Le filtrage a tendance à faciliter l'extraction de réponses pertinentes (c.-à-d. dans le top-3). Par ailleurs, les listes utilisées dans les appariements gagnants classe-liste (c.-à-d. les appariements menant aux meilleurs résultats QR) sont des listes peu restrictives (c.-à-d. de type « n », avec une constante de sélection élevée $c \geq 2$), dont le taux de sélection est supérieur à 80%. Les meilleurs résultats de RITEL sont obtenus à l'aide d'une liste de type op (**op2.5n**). Du point de vue du filtrage, ces résultats suggèrent que seuls les documents les plus bruités (c.-à-d. les moins pertinents pour les recherches comme des pages classées X et les exemples illustrés par les extraits de pages aux figures 3.6 à 3.9) sont écartés de la sélection des documents de RITEL et que l'hypothèse selon laquelle l'usage de listes restrictives améliorerait les performances QR était infondée. Du point de vue de la méthode EPID, ces résultats suggèrent que la combinaison des paramètres OOV et PPX prévaut sur leur utilisation séparée. Le corollaire en terme de classification est de penser qu'il est préférable d'utiliser des prédicteurs fondés sur une fonction de sélection de type op plutôt que o ou p.

Les méthodes d'appariement de type **Ma1** favorisent plutôt la précision et celles de type **Ma2** le rappel. Il est fréquent que pour le top-1 les résultats obtenus selon **Ma2** soient moins bons que ceux obtenus suivant Ma1 ou sans filtrage (**bsln**). Par ailleurs, les meilleures listes de filtrage globalement sont aussi les meilleures par classe de question. Ces résultats signifient que les meilleures listes en contexte d'apprentissage ne sont pas les plus appropriées en contexte d'évaluation.

Ainsi, définir les appariements classe-liste à partir des performances QR d'apprentissage n'aide pas la tâche du système en contexte d'évaluation. Cette conclusion nous a amené à penser que RITEL se trouvait peut-être en situation de surapprentissage et que le corpus d'entraînement utilisé durant l'entraînement du système n'était pas suffisant. Il se pourrait également que définir les appariements classe-liste sur la base d'un critère de performances comme nous l'avons fait ne soit pas le bon choix ou qu'il serait préférable que les appariements se fassent sur d'autres critères comme le type de réponse attendue des questions ou encore leur thème, plutôt que sur leur classe.

Il n'en reste pas moins que soit la précision soit le rappel peuvent être globalement améliorés par une méthode de filtrage, et que nous pouvons ainsi choisir la méthode la plus appropriée selon que nous voulons favoriser le rappel ou la précision. Par exemple, pour appliquer un module de ré-ordonnancement des candidats réponses, nous pourrions choisir une approche qui favorise le rappel.


AbsolutFlashJeux

AbsolutFlashJeux

! - ap - Defiscalisation - PretImmobilier - RachatCredit - Annecy - Astro - Chat - Abs - Keyword - Blog - Ben Blog - Pages - Sex - News

Â« Prev [abs] Â» Next Â»

AbsolutFlashJeux

RÃ©sultats Recherche via Google ™

AbsoluFlash jeux gratuit en ligne jeux en ligne, Tous les jeux

* Tous les jeux vidÃ©o en flash gratuits, jeu video en freeware,jeux en ligne, jeux gratuit,jeux vidÃ©o d'adresse, jeux de stratÃ©gie, jeux d'arcade,

Tous les jeux vidÃ©o en flash gratuits, jeu video en freeware drole

* Tous les jeux vidÃ©o en flash gratuits, jeu video en freeware,jeu en ligne, jeu gratuit,jeux vidÃ©o d'adresse, jeu de stratÃ©gie, jeux d'arcade, jeux

Jeux absolutflash

* Vous avez recherchÃ© absolutflash sur Jeux Video Flash, un site de jeux flash en ligne qui regroupe plus de 4000 jeux gratuit en ligne rÃ©alisÃ©s en Flash.

Jeux absolutflash com

* Vous avez recherchÃ© absolutflash sur Jeux Video Flash, un site de jeux flash en ligne qui regroupe plus de 4000 jeux gratuit en ligne rÃ©alisÃ©s en Flash.

Jeux Cherche Recherche de absolut flash

* Bref, on aura donc droit Ã plusieurs minijeux histoire de voir si vous avez le re Faire une recherche de absolut flash en video

Jeux Cherche Top recherche page 1

* Gun jeux avec cheval 1774 Games naruto 1748 Codes pour gta st andreas sur ordinateur 1740 Mahjong titans gratuits 1737 Absolut flash 1693

absolut flash jeux de pere

* absolut flash jeux de pere Sites. a la recherche de tof [color=red:296b339e94]ben voila je vais me metter au dual et je vais acheter le commencal absolut

Absolut Flash Games

* Fiche rÃ©fÃ©rencement » bensworld/Absolut/Flash/Games. Informations. Absolut Flash Games · Absolut Flash Games. RubriqueJeux > Jeux en ligne

Jeux en ligne Nicoland

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

FIGURE 3.6 – Exemple de page web bruitée, écartée par la méthode EPID ; PPX=974.97, OOV=6.04% (220/3738 mots).

FIGURE 3.7 – Exemple de page web bruitée, écartée par la méthode EPID ; PPX=1795.11, OOV=5.69% (60/1164 mots).

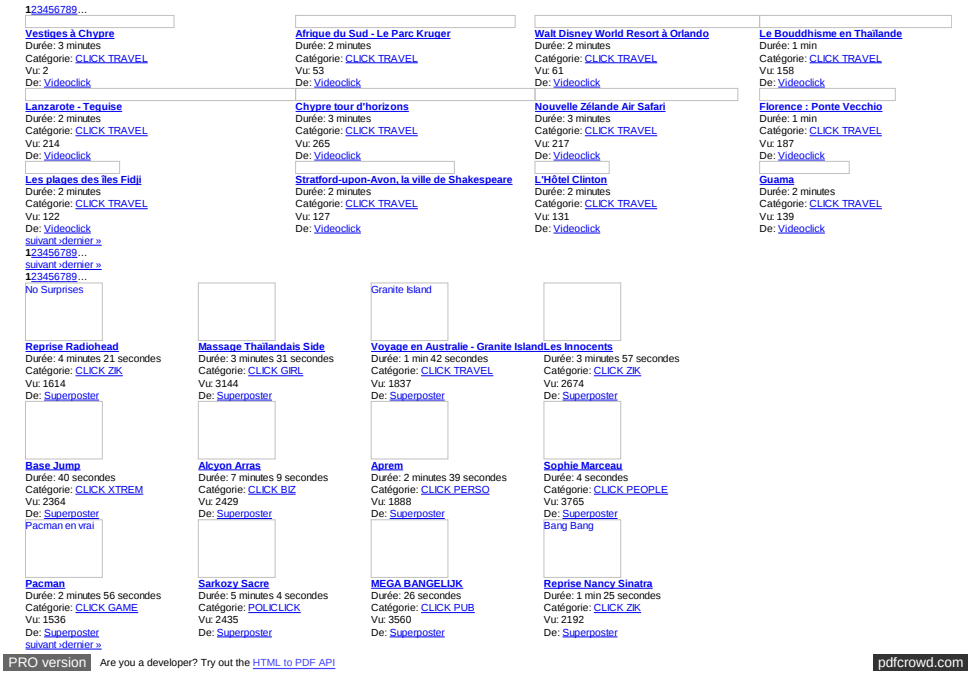
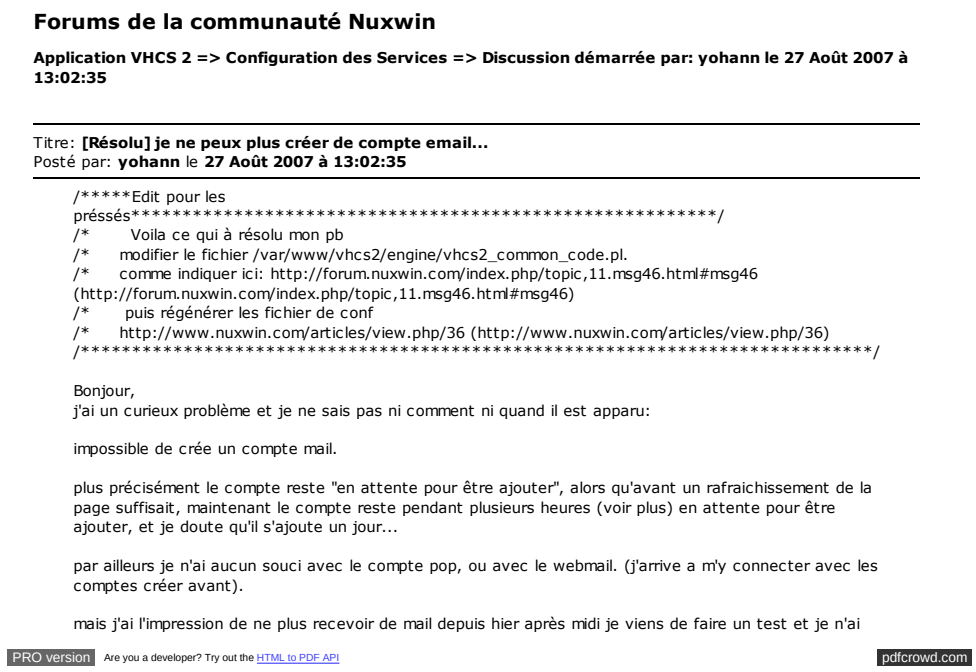


FIGURE 3.8 – Exemple de page web bruitée, écartée par la méthode EPID ; PPX=2897.18, OOV=9.52% (79/783 mots).



Forums de la communauté Nuxwin

Application VHCS 2 => Configuration des Services => Discussion démarrée par: yohann le 27 Août 2007 à 13:02:35

Titre: **[Résolu] je ne peux plus créer de compte email...**
Posté par: **yohann** le **27 Août 2007 à 13:02:35**

/*****Edit pour les
préssés**/
/* Voila ce qui à résolu mon pb
/* modifier le fichier /var/www/vhcs2/engine/vhcs2_common_code.pl.
/* comme indiquer ici: http://forum.nuxwin.com/index.php/topic,11.msg46.html#msg46
(http://forum.nuxwin.com/index.php/topic,11.msg46.html#msg46)
/* puis régénérer les fichier de conf
/* http://www.nuxwin.com/articles/view.php/36 (http://www.nuxwin.com/articles/view.php/36)
/***/

Bonjour,
j'ai un curieux problème et je ne sais pas ni comment ni quand il est apparu:

impossible de crée un compte mail.

plus précisément le compte reste "en attente pour être ajouter", alors qu'avant un rafraichissement de la page suffisait, maintenant le compte reste pendant plusieurs heures (voir plus) en attente pour être ajouter, et je doute qu'il s'ajoute un jour...

par ailleurs je n'ai aucun souci avec le compte pop, ou avec le webmail. (j'arrive a m'y connecter avec les comptes créer avant).

mais j'ai l'impression de ne plus recevoir de mail depuis hier après midi je viens de faire un test et je n'ai

FIGURE 3.9 – Exemple de page web bruitée, écartée par la méthode EPID ; PPX=1089.53, OOV=15.41% (868/5173 mots).

3.4.3 Contrôles

L'application du test de McNemar [96, 1] sur les résultats de la section précédente se sont tous révélés négatifs. Ce test établit la significativité des résultats obtenus entre 2 conditions A et B et une mesure M donnée selon des observations sur A et B. Un test (bilatéral) permet d'estimer une valeur Q pour un degré de liberté fixé et d'en dériver une valeur p qui, inférieure (ou égale) à un certain seuil critique α, rejette l'hypothèse nulle H_0 et rend les différences observées entre A et B significatives. Dans le cadre QR, une table de contingence comptabilise le total de questions pour lesquelles le système trouve une réponse d'exactitude identique (p. ex. réponse juste selon A et B) ou différente (p. ex. réponse juste selon A et fausse selon B) en conditions A et B. Ce test est repris plus en détail et illustré au chapitre suivant, dans les expériences de la section 4.4.2.

Il nous a donc semblé nécessaire de conduire quelques expériences de contrôles, afin de voir si les tendances que nous avons observées précédemment sont stables avec des conditions expérimentales différentes.

À la section 3.4.3.1, le contrôle consiste à reproduire les expériences de filtrage menées précédemment, à l'aide d'une version antérieure de RITEL (notée RITEL_{08}) aux résultats de référence inférieurs d'environ 10% (MRR) à ceux de RITEL, du à l'absence de certaines heuristiques utilisées par RITEL à l'analyse des questions. À la section 3.4.3.2, le contrôle consiste à vérifier la cohérence des listes de filtrage créées par le biais de la méthode EPID, en s'intéressant à l'effet d'un filtrage opéré avec des listes générées aléatoirement.

3.4.3.1 Contrôle 1 : réplication des résultats sur RITEL_{08}

Précision, MRR et rappel (top-10) Globalement, on constate d'après le tableau 3.7a que le comportement de RITEL_{08} suit celui de RITEL : **Ma1** est la condition d'appariement qui mène aux meilleurs résultats. Cependant, on voit que RITEL_{08} a de meilleurs résultats selon les conditions **Ma1** que selon les conditions de type **Ma2**, quelles que soient la mesure considérée ; rappelons qu'avec RITEL le rappel était favorisé par la condition **Ma2**. On voit aussi qu'en contexte de filtrage, RITEL_{08} a toujours de meilleurs résultats qu'avec la condition de référence (**bsln**). Enfin, on peut noter que l'impact du filtrage est légèrement plus prononcé avec RITEL_{08} que dans les expériences contrôlées et qu'il ne s'applique pas selon le même schéma. En effet, il y a quasiment 4 points de différence en terme de précision entre le moins bon résultat (**bsln** : **24.6%**) et le meilleur ici (**Ma1**.p : **28.2%**) contre environ 2 points sur RITEL, et à peine 1 point de différence en terme de rappel ici (conditions **Ma1**.op et **bsln** cette fois) contre presque 3 sur RITEL.

Ces résultats suggèrent d'une part que le bénéfice apporté par la méthode de filtrage pour la tâche QR est plus prononcé sur RITEL_{08} que sur RITEL et que, très clairement, la méthode **Ma1** prévaut sur la méthode d'appariement **Ma2** dans ces expériences de contrôles. Ceci confirme que la méthodologie d'appariement classe-liste que nous avons mise en place n'est pas satisfaisante. D'autre part, ces résultats amènent à tempérer la conclusion à laquelle nous avions abouti à propos de la préférence d'utilisation de fonction de sélection combinée relativement aux paramètres OOV et PPX. En effet, ici, l'utilisation séparée de ces paramètres, en l'occurrence PPX (p), mène le système à une plus grande précision en condition **Ma1**.p. Mais là aussi, globalement le filtrage permet une amélioration des résultats.

Focus sur le top-10 Le détail des résultats de top-10 (**P@10**) pour les 2 meilleures conditions (**Ma1**.o et **Ma1**.op) et la condition de référence (**bsln**) est donné au tableau 3.7b. On constate que le nombre total de réponses est similaire dans les trois conditions (environ 140), et représente une différence proche de 30 réponses par rapport au nombre de réponses équivalent extrait avec RITEL.

L'effet du filtrage est quantitativement moins important ici que dans les expériences contrôlées, bien qu'il permette à RITEL$_{08}$ d'améliorer ses performances.

Cond08	**P@1**	**MRR**	**P@10**	**#q**
Ma1.p	28.2	34.2	45.6	309
Ma1.o	27.8	34.3	46.0	309
Ma1.op	26.5	33.3	**46.3**	309
bsln	**24.6**	31.6	**45.6**	309
Ma2.p	27.5	33.7	45.6	309
Ma2.o	25.6	32.4	44.7	309
Ma2.op	25.2	32.0	44.0	309

(a)

	Cond08		
rang	**Ma1**.p	**bsln**	**Ma1**.op
1	87	**76**	82
2	23	21	27
3	7	19	9
4-10	24	25	25
Total	141	**141**	**143**
1-3	117	116	118

(b)

TABLE 3.7 – (a) Résultats QR globaux par condition expérimentale (**Cond08**) avec RITEL$_{08}$. **P@1** : précision. **MRR** : moyenne des rangs réciproques. **P@10** : précision sur 10 rangs (top-10). **#q** : nombre total de questions évaluées. (b) Focus sur les résultats du top-10 présenté en (a). À chaque position (**rang**), on indique le nombre de réponses pertinentes trouvées par le système.

Cond08$_{P@1}$	**glb**	**loc**	**nbr**	**pers**	**qpers**	**sub**	**time**
Ma1.p	28.2	54.4	13.6	23.6	26.7	2.9	35.4
Ma1.o	27.8	**54.4**	**18.2**	18.2	**26.7**	0.0	**37.5**
Ma1.op	26.5	50.0	13.6	20.0	**26.7**	**2.9**	**37.5**
bsln	24.6	52.9	**18.2**	16.4	17.8	0.0	29.2
#q	309	66	45	55	45	39	48

(a)

Cond08	**glb**	**loc**	**nbr**	**pers**	**qpers**	**sub**	**time**
Ma1.p	3r	3r	3r	3r	3r	3r	3r
Ma1.o	2.5n	2.5n	2.5n	2.5n	2.5n	2.5n	2.5n
Ma1.op	2.5n	2.5n	2.5n	2.5n	2.5n	2.5n	2.5n
bsln				-			
Ma2.p	2n	3n	1.5n	3n	3n	3n	2r
Ma2.o	2.5n	3n	3r	3r	3n	1r	0n
Ma2.op	3n	2n	1.5n	3n	2.5n	2r	1r

(b)

TABLE 3.8 – (a) Précision (P@1) QR globale (**gbl**) et par classe de question (**loc**, **nbr**, etc.) selon chaque condition expérimentale (**Cond08**) avec RITEL$_{08}$. (b) Focus sur les appariements classe-liste utilisé par le système en (a). **#q** : nombre total de questions par classe.

On remarque aussi que le filtrage maximise l'extraction de réponses en tête de classement (en top-1 ou top-2) alors que le comportement classique du système (**bsln**) tend à extraire des réponses à des rangs inférieurs. Ceci coïncide avec les observations que nous avons faites dans les expériences contrôlées.

Résultats par classe de question On présente au tableau 3.8 les mêmes informations que celles présentées pour l'analyse de RITEL par classe de questions. Pour ces analyses, nous nous focalisons sur les trois meilleures conditions du tableau 3.7a et la condition de référence.

En confrontant les tableaux 3.8a et 3.8b, on arrive à la même conclusion pour RITEL$_{08}$ que pour RITEL : utiliser des listes permissives (c.-à-d. contenant de nombreux documents) au moment du filtrage semble augmenter les performances QR. En effet, on voit au tableau 3.8b que les listes utilisées dans les conditions **Ma1** sont peu restrictives ($c > 2$).

Contrairement à nos observations sur RITEL, les appariements classe-liste gagnants en contrôle se fondent aussi bien sur des listes *restreintes* que *normales* (c.-à-d. des listes de type « r » ou « n »). En particulier, celle qui mènent RITEL$_{08}$ aux meilleurs résultats (**p3r**) est de type « r » (condition **Ma1**.p). Dans les autres cas, on retrouve les meilleures listes vues dans nos expériences de filtrage initiales (les listes op2.5n et o2.5n). Toutes ces listes ont des taux de sélection supérieurs à 80%.

Récapitulatif et discussion En contrôle, le système bénéficie toujours de l'utilisation du filtrage. Le bénéfice apporté par le filtrage sur les performances de RITEL$_{08}$ est plus important en précision qu'en rappel et a tendance à faciliter l'extraction de réponses pertinentes (c.-à-d. dans le top-3) tout en préservant le nombre de réponses justes extraites au total (top-10). Les performances sont systématiquement meilleures en condition **Ma1**. Les appariements classe-liste gagnants concernent des listes permissives de type « n » ou « r » ($c \geq 2$) avec des taux de sélection supérieurs à 80%. La meilleure liste globalement est aussi la meilleure liste par classe de question : elle s'appuie sur une fonction de sélection de type p (c.-à-d. **p3r**).

En conséquence, ce contrôle tempère l'idée selon laquelle il est préférable d'utiliser une fonction de sélection qui combine OOV et PPX afin de créer des listes de filtrage. S'il tempère également l'idée que les listes utiles pour le filtrage ne sont que les listes de type « n », il semble valider le fait que seules des listes ayant des taux de sélection supérieurs à 80% de documents s'avèrent utiles pour le filtrage. Enfin, ce contrôle nous permet de confirmer 2 conclusions essentielles tirées des expériences contrôlées : d'une part qu'il est utile de filtrer les documents sélectionnés par un système QR afin d'en extraire des passages et des réponses pertinentes (filtrer les documents sélectionnés par un système QR pourrait donc lui faciliter la tâche) et, d'autre part, que notre méthodologie d'appariement classe-liste n'est pas satisfaisante.

3.4.3.2 Contrôle 2 : tirage aléatoire de documents pertinents

Dans cette section, nous nous demandons dans quelle mesure les listes de filtrage que nous avons utilisées pour évaluer la méthode EPID sont cohérentes. Pour répondre à cette question, nous avons mis en place un filtrage basé sur un échantillonnage aléatoire simple [131]. Notre hypothèse consiste à penser qu'un tel filtrage devrait avoir un impact réduit sur les performances de RITEL par rapport à celui observé dans les expériences contrôlées.

Approche Nous avons généré 42 nouvelles listes de documents tirés aléatoirement sans remise parmi les documents du corpus *Q07fr*. Les proportions de documents de ces nouvelles listes respectent celles des 42 listes utilisées dans nos expériences initiales de filtrage.

Lors d'expériences préliminaires, nous avons testé la méthode d'appariement **Ma2** à l'aide des listes créées aléatoirement. Les résultats obtenus sont similaires à ceux présentés plus loin dans la section.

Le tableau 3.9a présente les résultats obtenus par RITEL en terme de précision (**P@1**), **MRR** et top-10 (**P@10**), pour 3 conditions différentes d'expérimentation. **La première condition** (**bsln**) est la condition de référence sans filtrage. Celle-ci correspond à la condition de référence présentée dans les expériences contrôlées.

La seconde condition (**Ma1**.op) est une condition avec filtrage. Celle-ci correspond au meilleur filtrage obtenu dans les expériences contrôlées. La liste de filtrage associée à toutes les classes de question ici est donc la liste **op2.5n**. **La dernière condition** (random ou **rdm**) est la condition contrôle. Ici, la liste **op2.5n** a été générée par échantillonnage aléatoire simple comme expliqué plus haut.

Cond	P@1	MRR	P@10	#q
Ma1.op	**33.0**	40.6	**55.0**	309
bsln	**31.7**	39.5	**53.4**	309
rdm	31.4	39.2	53.4	309

TABLE 3.9 – Résultats QR globaux par condition expérimentale (**Cond**). **P@1** : précision. **MRR** : moyenne des rangs réciproques. **P@10** : précision sur 10 rangs (top-10). **#q** : nombre total de questions évaluées. pos : classement des conditions.

On constate que les résultats obtenus par RITEL en condition **rdm** sont globalement moins bons que ceux obtenus dans les 2 autres conditions. Cependant, on peut noter que les écarts observés ne sont pas très prononcés et notamment, les conditions **rdm** et **bsln** sont très proches l'une de l'autre.

La liste utilisée par RITEL en condition **rdm** est très peu sélective (85% de documents environ). Elle couvre donc une très grande partie de l'index de documents utilisé par RITEL en condition **bsln**. Il n'est donc pas surprenant que, dans ces deux conditions, on aboutisse à des performances similaires (et même identiques en rappel). Après vérification, il s'avère qu'il en va de même pour les conditions **rdm** et **Ma1**.op, étant donné que le nombre de documents communs aux listes de filtrage utilisées dans ces 2 conditions est élevé (seulement 30% de leurs documents diffèrent). Cependant, on note que les écarts ici sont plus marqués que dans les conditions **rdm** et **bsln**, suggérant que les listes de filtrage créées via la méthode EPID semblent avoir plus de cohérence.

3.5 Conclusion

Dans ce chapitre nous avons présenté une approche statistique qui permet de filtrer la sélection des documents réalisée par un système de réponses à des questions. Nous avons appliqué cette approche sur le français avec le système de Questions-Réponses **RITEL**.

Cette approche évalue la pertinence intrinsèque des documents trouvés par le système et fonctionne en 2 temps : estimation de la pertinence intrinsèque des documents, puis classification de ces derniers en tant que documents *(non) pertinents* pour l'extraction de passages. Pour ce faire, d'un côté on utilise un **modèle de langue** construit à l'aide d'un large corpus de documents journalistiques (GEN) qui fournit *a priori* des mesures objectives sur le degré de normativité d'un texte en terme de **perplexité** (PPX) et de **ratio de mots hors vocabulaire** (OOV). De l'autre, on utilise un **modèle de classification binaire** qui repose sur un corpus de 509 documents de référence en QR (**DEV509**).

Ce modèle permet de juger de la pertinence QR d'un texte à l'aide des scores d'OOV et PPX fournis par le modèle de langue et d'un triplet de paramètres : $\langle f, m, c \rangle$, où f est une fonction de sélection (il y en a **3** au total : o, p et op), m est une méthode d'estimation des distributions OOV et PPX sur les documents du corpus DEV509 (il y en a **2** au total : méthode *normale* et méthode *restreinte*) et c est la constante de variation utilisée par la fonction de sélection pour influer sur l'écart type associé aux moyennes d'OOV et PPX estimées grâce au corpus DEV509 (qui prend **7** valeurs possibles : $c = \{0, 1, 1.5, 2, 2.5, 3\}$).

Sur la base des 3*2*7 combinaisons de valeurs possibles associées respectivement aux paramètres f, m, et c, nous avons établi 42 classifieurs et générés **42 listes de documents pertinents** pour les recherches en QR. Ceci a été fait en appliquant chaque classifieur au corpus d'évaluation **Quaero** français de 499 734 pages web *Q07fr*. Ces listes sont intégrées à un **module de filtrage** placé entre les modules de Recherche d'Information et d'extraction de réponses de la chaîne de traitements de RITEL. Au moment des recherches, les documents listés sont recoupés avec ceux sélectionnés par le système. Les documents communs sont préservés pour l'extraction de passages, tandis que les autres sont considérés comme non pertinents pour cette tâche et rejetés.

Nous avons testé **2 méthodes d'appariement** des classes de questions aux listes de filtrage. Dans les deux cas, ces dernières ont été créées sur la base des performances QR obtenues à l'entraînement par RITEL. La première méthode (**Ma1**) associe chaque classe de question à la même liste ; celle-ci est la liste qui mène le système aux meilleures performances d'apprentissage, indépendamment d'une classe de questions particulière. La deuxième méthode (**Ma2**) associe chaque classe de question à la liste pour laquelle RITEL présente les meilleures performances d'apprentissage pour la classe.

Les résultats d'évaluation obtenus par RITEL en contexte de filtrage sur le corpus *Q07fr* et le jeu de test de 309 questions factuelles de la campagne Quaero de 2009 révèlent que la méthode d'évaluation de la pertinence intrinsèque d'un document (**méthode EPID**) peut aider un système QR dans sa tâche : les performances QR obtenues par RITEL sont plus élevées en condition de filtrage qu'en condition référence. Apparemment, le filtrage permet d'éliminer les documents les plus bruités pour les étapes d'extraction de passages et de réponses (les listes de filtrage utilisées par le système au moment des recherches contiennent entre 80% et 90% des documents sélectionnés à l'étape de RI par RITEL). Cependant, ceci n'est qu'une tendance, les tests de significativité de ces résultats s'étant avérés négatifs.

Les résultats obtenus en contrôle corroborent globalement cette tendance. Ces derniers nous ont aussi permis de tempérer l'hypothèse de préférence d'utilisation de fonctions de sélection combinées (op) en filtrage. Utiliser des listes de type o, p ou op ne semble donc pas avoir un impact important sur le filtrage dans le cadre de la tâche QR. **Ces résultats sont assez éloignés de notre intuition de départ**. En général, nous nous attendions à ce que les listes utilisées en filtrage avec des taux de sélection plus faibles (c.-à-d. des listes restreintes de type op avec une constante de variation faible) mènent RITEL à des gains de performances plus marqués.

Bien que les tests de significativité soient négatifs, les tendances systématiques observées d'un gain avec un filtrage nous amènent à penser que **la méthode EPID telle que nous l'avons appliquée présente un intérêt, cependant limité, en QR, en particulier dans le cadre de la sélection pertinente de documents.**

Les résultats d'évaluation et de contrôle s'accordent pour conclure que définir des appariements classe-liste sur la base des résultats d'apprentissage n'est pas une solution satisfaisante. En effet, la méthode d'appariement **Ma2** devrait permettre à RITEL d'obtenir de meilleures performances que sa contrepartie **Ma1**, **Ma2** choisissant pour chaque classe de question la liste qui lui convient le mieux. Une première hypothèse d'explication consiste à penser que RITEL manque de données d'entraînement lors de l'estimation automatique de ses différents paramètres (**hypothèse H1**) : il y a dans ce cas surapprentissage. Une autre hypothèse consiste à penser que définir les appariements classe-liste en fonction des performances d'apprentissage n'est pas un bon critère pour garantir des performances optimales en évaluation (**hypothèse H2**) : dans ce cas, d'autres critères d'appariement, par exemple, la thématique des questions et des documents, pourraient être plus pertinents. À l'image de certains systèmes de dialogue, il serait avantageux pour un système QR de disposer de ce genre d'indice pour affiner les stratégies de recherche et d'extraction appliquées aux différentes étapes de traitements.

Un autre critère pourrait être d'apparier les listes au type de réponse attendue.

En dehors des hypothèses H1 et H2 mentionnées pour expliquer l'intérêt positif mais limité du filtrage des documents, nous nous interrogeons sur la pertinence même des listes que nous avons générées avec la méthode EPID. Sont-elles suffisamment élaborées pour servir un système QR dans sa tâche ?

Pour répondre à cela, il faut se pencher du côté des modèles sur lesquels s'appuie la méthode : les corpus GEN, DEV509 et le modèle de langue. A priori, le corpus d'articles GEN ayant servi à créer le modèle de langue semble de taille respectable pour assurer une bonne généricité. Le corpus DEV509 servant de modèle de documents pertinents à la création des classifieurs n'est peut-être pas assez conséquent (509 documents) pour produire des listes adaptées au filtrage (**hypothèse H3**). Pour vérifier cette hypothèse, on pourrait essayer de consolider ce corpus par l'ajout de nouveaux documents du même genre que ceux qui le composent déjà.

On peut aussi penser que l'usage d'un modèle de langue plus riche que celui que nous avons utilisé pour évaluer la pertinence intrinsèque d'un document pourrait mener à la création de listes de filtrage plus adaptées pour la sélection de documents en QR (**hypothèse H4**). Nous tenons à rappeler ici que l'utilisation initiale d'un modèle de langue simple est un choix délibéré, le but étant de faciliter la compréhension des phénomènes observés à son usage. Comme nous l'avons dit, enrichir les paramètres du modèle de langue passe par l'intégration de nouveaux paramètres au niveau de la représentation des documents. Par exemple, les parties du discours, des informations d'ordre syntaxique et/ou sémantique, etc. qui interviennent à divers niveaux de représentation de la langue plus complexes que le niveau surfacique d'un texte auquel nous avons opéré.

Les 4 hypothèses que nous avons proposées dans cette conclusion pour expliquer l'effet mitigé du filtrage sur la sélection pertinente de documents en QR sont relativement naturelles dès lors que l'on travaille en apprentissage automatique. Les hypothèses H1 et H3 relèvent de l'apprentissage automatique pur et présentent moins d'enjeux que les 2 autres hypothèses du point de vue de la langue. L'hypothèse H4 nous semble présenter un intérêt majeur du point de vue de la langue, de même que l'hypothèse H2, pour laquelle nous avons mené des expériences préliminaires. Nous n'avons cependant pu, faute de temps, les explorer pour les valider.

Nous avons jugé plus intéressant dans un premier temps de pousser plus loin l'introspection de la méthode EPID pour la sélection de documents en QR, en s'intéressant de plus près à la variation des performances QR en fonction de l'échelle de filtrage des documents considérée. Que se passe-t-il quand le filtrage s'opère à une échelle plus fine, sur des sous-parties de documents ? En effet, les pages web sur lesquelles nous travaillons présentent un contenu textuel relativement long (en moyenne 300 lignes par page) et vraisemblablement multi-thématiques d'après les analyses manuelles que nous avons faites des documents couverts par les listes de filtrage qui semble pénaliser le système dans sa tâche (**hypothèse H0**). En effet, il apparaît clairement à travers ces analyses que certaines pages sont rejetées au moment du filtrage alors qu'elles contiennent certaines zones d'informations pouvant s'avérer utiles pour le système QR dans sa tâche.

Au prochain chapitre, nous présentons le système de segmentation de pages web que nous avons développé afin de suivre cette piste.

« Science is a way of talking about the universe in words that bind it to a common reality. Magic is a method of talking to the universe in words that it cannot ignore. The two are rarely compatible. » [35]

NEIL GAIMAN

Chapitre 4

Pré-segmentation et sélection pertinente de documents en QR

Sommaire

4.1 Introduction

DANS ce chapitre, nous présentons un système de segmentation que nous avons développé pour mettre à l'épreuve l'hypothèse **H0** émise à l'issue du chapitre 3. Selon cette hypothèse, la variabilité naturelle des pages web en taille et en contenu (et notamment leur caractère multi-thématique) pourrait pénaliser un système QR dans sa tâche.
Cette hypothèse fait suite à l'analyse des sorties du module de filtrage parallèlement aux sorties des modules d'extraction de passages et de réponses de RITEL. En effet, il s'avère que si le filtrage permet au système la sélection de documents pertinents pour l'extraction de passages et de réponses, les modules associés ne permettaient pas soit d'extraire les bons passages, soit les bonnes réponses. Dans certains cas, le filtrage rejette aussi des documents à tort ; nous donnons des exemples au paragraphe intitulé : « motivation ». Ce dernier nous sert aussi à motiver nos choix de segmentation.

La segmentation de pages nous permettra au chapitre 5 d'appliquer la méthode d'évaluation de la pertinence intrinsèque d'un document (méthode EPID) sur des parties de documents (c.-à-d. à l'échelle de segments), et de poursuivre l'étude de son influence sur la sélection de documents en QR à une échelle d'analyse plus fine que celle des documents suivie au chapitre 3.

Motivation Nous présentons deux extraits de pages web (voir les figures 4.1 et 4.2) tirées de notre corpus d'évaluation. Ces extraits sont fournis pour montrer qu'il semble opportun, au moins dans certains cas, de segmenter les documents afin d'aider le système à trouver des réponses. Pour le second extrait, nous donnons un exemple de paragraphe normalisé tel que nous l'avons utilisé dans nos évaluations à la figure 4.3.

Le premier extrait de pages web que nous donnons à la figure 4.1 illustre une faiblesse de la méthode EPID. En effet, le document est rejeté à tort au filtrage alors même que celui-ci contient plusieurs réponses à la question *Quel est le prix d'un banc de musculation ?*. On voit dans l'extrait que l'essentiel du contenu textuel de la page web associée a un style très télégraphique, avec une syntaxe dépourvue de ponctuation, formée d'une simple juxtaposition de mots, dont beaucoup de marques commerciales et parfois en anglais. L'ensemble de ces caractéristiques fait de cette page un document fortement éloigné de ceux connus du modèle de langue qui, sans surprise, lui attribue des valeurs de PPX et OOV élevées (c.-à-d. 1983.5 et 7.85% respectivement) qui mènent à son rejet au moment du filtrage. Dans ce cas, la segmentation pourrait permettre de réduire les scores d'OOV et PPX localement sur des parties de la page et permettre à certaines d'être sélectionnées lors du filtrage.

Le deuxième extrait que nous donnons à la figure 4.2 illustre le cas où une page web est catégorisée à raison comme étant *pertinente* au moment du filtrage, mais pour laquelle RITEL est en défaut. La version normalisée d'une partie de cet extrait est donné à la figure 4.3. Ici, les valeurs de PPX et OOV fournies par le modèle de langue au cours de la procédure de filtrage sont dans la moyenne des documents (c.-à-d. 147.4 et 1.79% respectivement), et le document est sélectionné. Celui-ci est pertinent et peut permettre de répondre à la question *Quand la coupe de l'America a-t-elle lieu ?*. Pourtant, RITEL n'arrive pas à extraire un passage qui contient une réponse bien que celle-ci soit présente dans le premier paragraphe de son contenu principal, comme on le voit clairement à la figure 4.3. Dans ce cas, la segmentation devrait faciliter le système dans sa tâche de découpage de documents en passages et permettre d'en extraire un passage pertinent.

Stratégie de segmentation haut niveau Nous avons vu au chapitre 2 qu'il était de coutume en QR de découper les documents en passages soit au moment de l'indexation (c.-à-d. **index-time passaging** [124, 73]), soit au moment des recherches (c.-à-d. **search-time passaging** [124, 73]). La figure 4.4a illustre le premier type de découpage, la figure 4.4b le second.

Un intérêt de segmenter les documents en passages est d'accélérer globalement le système à l'exécution en soulageant les étapes d'extraction de passages et de réponses dédiées au traitement fin de la langue. D'un point de vue QR, l'intérêt de segmenter les documents en passages est de travailler avec un contenu informationnel (corrélé au contenu linguistique, en particulier lexical et sémantique) plus cohérent afin de permettre l'extraction de réponses plus pertinentes que celles issues de la globalité du texte et va dans le sens de l'hypothèse **H0**.

À notre connaissance, personne n'a tenté de segmenter les documents préalablement à leur indexation tout en découpant les segments obtenus en passages lors des recherches afin de renforcer l'effet de cohérence locale des pages web, et voir si cela peut faciliter le travail d'un système QR dans sa tâche.

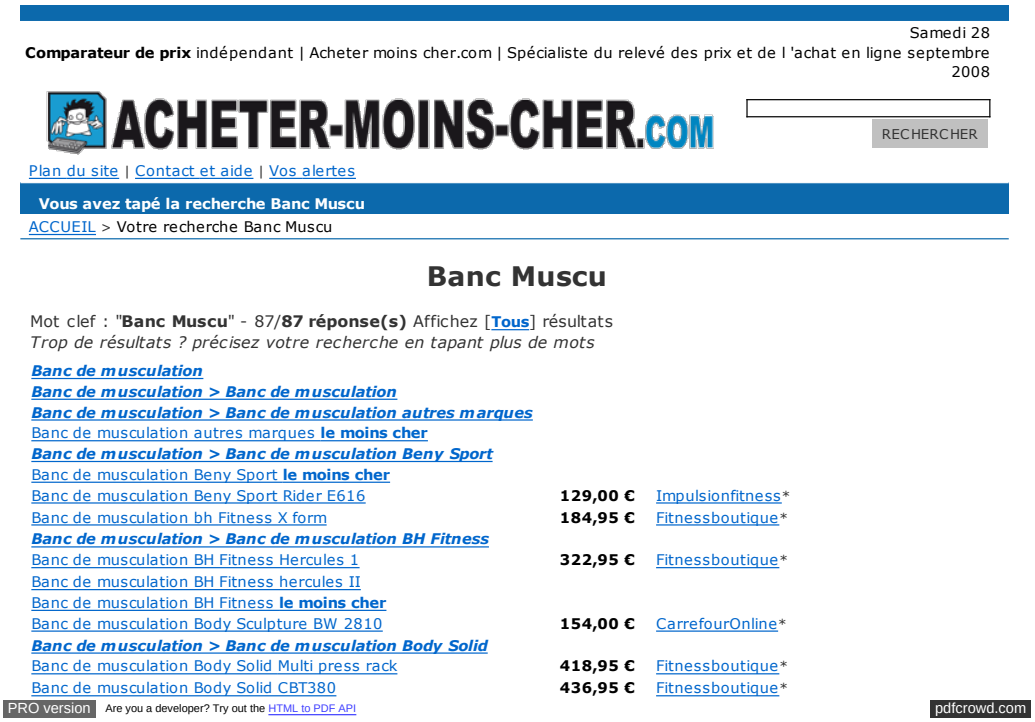

Samedi 28
Comparateur de prix indépendant | Acheter moins cher.com | Spécialiste du relevé des prix et de l 'achat en ligne septembre 2008
ACHETER-MOINS-CHER.COM
RECHERCHER
Plan du site | Contact et aide | Vos alertes
Vous avez tapé la recherche Banc Muscu
ACCUEIL > Votre recherche Banc Muscu
Banc Muscu
Mot clef : "**Banc Muscu**" - 87/**87 réponse(s)** Affichez [**Tous**] résultats
Trop de résultats ? précisez votre recherche en tapant plus de mots
Banc de musculation
Banc de musculation > Banc de musculation
Banc de musculation > Banc de musculation autres marques
Banc de musculation autres marques **le moins cher**
Banc de musculation > Banc de musculation Beny Sport
Banc de musculation Beny Sport **le moins cher**
Banc de musculation Beny Sport Rider E616 **129,00 €** Impulsionfitness*
Banc de musculation bh Fitness X form **184,95 €** Fitnessboutique*
Banc de musculation > Banc de musculation BH Fitness
Banc de musculation BH Fitness Hercules 1 **322,95 €** Fitnessboutique*
Banc de musculation BH Fitness hercules II
Banc de musculation BH Fitness **le moins cher**
Banc de musculation Body Sculpture BW 2810 **154,00 €** CarrefourOnline*
Banc de musculation > Banc de musculation Body Solid
Banc de musculation Body Solid Multi press rack **418,95 €** Fitnessboutique*
Banc de musculation Body Solid CBT380 **436,95 €** Fitnessboutique*
PRO version Are you a developer? Try out the HTML to PDF API
pdfcrowd.com

FIGURE 4.1 – Exemple de page (non segmentée) filtrée à tort par la méthode EPID.

Nous avons profité du découpage habituel des documents en passage opéré par RITEL au moment des recherches pour ajouter une étape de pré-segmentation des documents au cours des pré-traitements du système au moment de l'indexation. Notre objectif ici est de répondre à la question suivante : *Segmenter les documents à l'indexation en plus de leur découpage habituel en passages au moment des recherches, peut-il aider un système QR dans sa tâche ?*. De plus, la réponse à cette question permettra de valider ou non l'hypothèse **H0**.

Critère de segmentation Nos observations nous ont conduit à chercher à appliquer une méthode de segmentation capable d'identifier les sous-parties d'un document et d'en extraire les blocs correspondants.

L'idée de segmenter des documents (article de journaux, livres etc.) en blocs est un axe de recherche activement exploré dans les années 90 en RI textuelle.

Un type d'approche, employée notamment par Salton [130] et Hearst [53], consiste à opérer une segmentation en blocs thématiques sur le texte des documents, les frontières de blocs étant les endroits du texte où on détecte un changement de thème. Un calcul de proximité lexicale entre blocs de pseudo-phrases adjacentes permet de découper le texte d'origine en segments plus homogènes de taille variable.

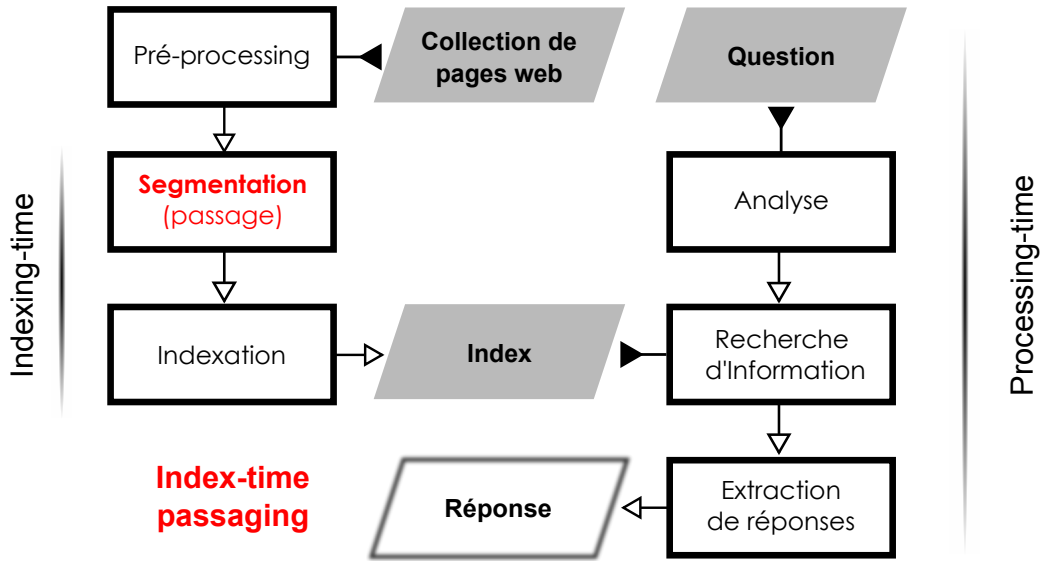
Chez Salton, la proximité lexicale est obtenue à l'aide de mesures de distances vectorielles, chaque bloc étant représenté par un vecteur lexical. En fonction de valeurs seuils sur ces distances, des fusions entre paragraphes sont opérées jusqu'à ce que les blocs ne contiennent plus que du texte thématiquement homogène.

FIGURE 4.2 – Exemple de page (non segmentée) accepté au filtrage face à laquelle RITEL est en échec.

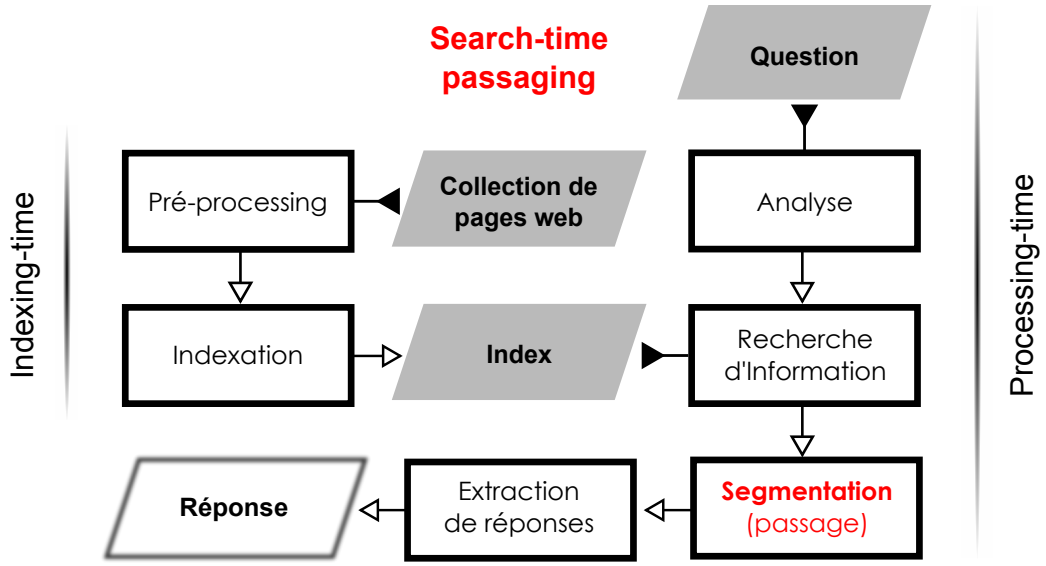
```
CQFD numéro 004 .
feuilleton .
coupe de l' América : Marseille boit la tasse .
mis à jour le : 15 septembre 2003 .
auteur : Lionel Raymond , Olivier Cyran .
depuis juillet , la mairie de Marseille " hisse les couleurs de la France " en  \
faisant battage pour l' accueil en 2007 de la coupe de l' America , une course  \
à voiles pour milliardaires .
si le jury suisse lui donne satisfaction , les bétonneurs ne verront plus aucun \
frein à l' accomplissement de leur rêve : transformer une ville prolo en marina \
de luxe .
déjà , la candidature marseillaise fait sentir ses dégâts .
et son appareil de propagande .
```

FIGURE 4.3 – Version normalisée (1ier paragraphe) de l'extrait de page donné à la figure 4.2.

La démarche de segmentation opérée par Hearst [53] opère dans le même esprit que celle de Salton à l'aide d'heuristiques linguistiques plus fines : son algorithme de segmentation (*TextTiling*) s'appuie sur la théorie de la cohésion lexicale [48] et une mesure de distance basée sur des statistiques de co-occurrences établies à partir d'une représentation textuelle à base d'unités lexicales élémentaires (*tokens*) groupées en pseudo-phrases.



(a) Index-time passaging



(b) Search-time passaging

FIGURE 4.4 – Illustration des 2 stratégies de segmentation majeures pour le découpage des documents en passages en QR.

Plus récemment, des travaux dans le contexte de la RI dans des documents multimédia [119] et dans ceux de l'OCR et du traitement de l'image (p. ex. avec des algorithmes de découpage tels que X-Y cut [108, 97] et Docstrum [111]) sur des pages web ou des documents numériques (numérisation de pages web incluses) [114] ont proposé de nouvelles approches pour la segmentation de documents. Nous en listons ci-dessous quelques-unes.

Faessel [30] propose une nouvelle méthode d'indexation des documents, basée sur l'information des représentations DOM et CSS des pages web pour segmenter ces dernières avant indexation. Qi et Davison en classification automatique de pages web en thème [119] utilisent également la représentation DOM des pages pour réaliser leur segmentation et classer les pages sur la base de la thématique des blocs obtenus.

Asirvatham *et al.* [3], utilisent l'échantillonnage des couleurs des images pour catégoriser leur page hôte en genre. Dans [75], pour la même tâche, les auteurs utilisent le rendu visuel des pages.

Guo et al. [47] utilisent des indices visuels (rendu des pages fourni par le moteur de Mozilla [1]), géométriques (coordonnées des éléments DOM au sein du rendu des pages) et de style (répétition d'information) pour définir les blocs d'information pertinents trouvés dans les pages.

En web sémantique, Vadrevu *et al.* [149] utilisent des critères de découpage fondés sur l'homogénéité locale du contenu informationnel des pages web basés sur le modèle de « path entropy », et sur les indices visuels dérivés de leur arbre DOM.

Choix de segmentation À notre connaissance, aucune tentative d'application des techniques récentes que nous venons de présenter pour la segmentation de documents web n'a été faite dans un contexte de segmentation de pages en QR et, notamment, celle guidée par la thématique des documents. Nous avons choisi une méthode de segmentation thématique car connaître la thématique des différentes parties constitutives d'un texte (ou une page web) pourrait permettre de renforcer l'analyse sémantique des documents pour un système QR. Nous donnons à titre d'exemple à la figure 4.5 l'extrait d'une page web d'actualité politique tiré de notre corpus d'évaluation qui montre l'intérêt d'appliquer une segmentation thématique.

Cependant, préalablement à l'utilisation de techniques avancées, nous avons voulu obtenir des résultats de référence, en utilisant en premier lieu un algorithme de première génération dans notre système de segmentation de pages web, avec le TextTiling de Hearst. En effet, plus fin que l'algorithme de Salton, l'algorithme de Hearst a déjà fait ses preuves en sélection de documents en RI et se fonde sur une philosophie TAL qui répond à notre problématique. Par ailleurs, dans la perspective de travaux futurs orientés sur la segmentation visuelle de pages web en QR, la segmentation par TextTiling nous permettra de bénéficier d'une segmentation TAL de référence, à comparer à des approches moins textuelles (thématiques ou non).

C'est dans ce cadre que le travail présenté dans ce chapitre se place.

Ce chapitre s'articule en quatre sections. À la section 4.2 nous présentons globalement puis étape par étape le système de segmentation mis en place. La section 4.3 décrit l'effet de la segmentation sur le corpus de référence. À la section 4.4 nous évaluons ce système en contexte QR. Enfin, nous concluons à propos du travail de segmentation réalisé dans ce chapitre en section 4.5.

1. www.mozilla.org

FIGURE 4.5 – Exemple de page web multi-thématiques.

4.2 Procédure de pré-segmentation

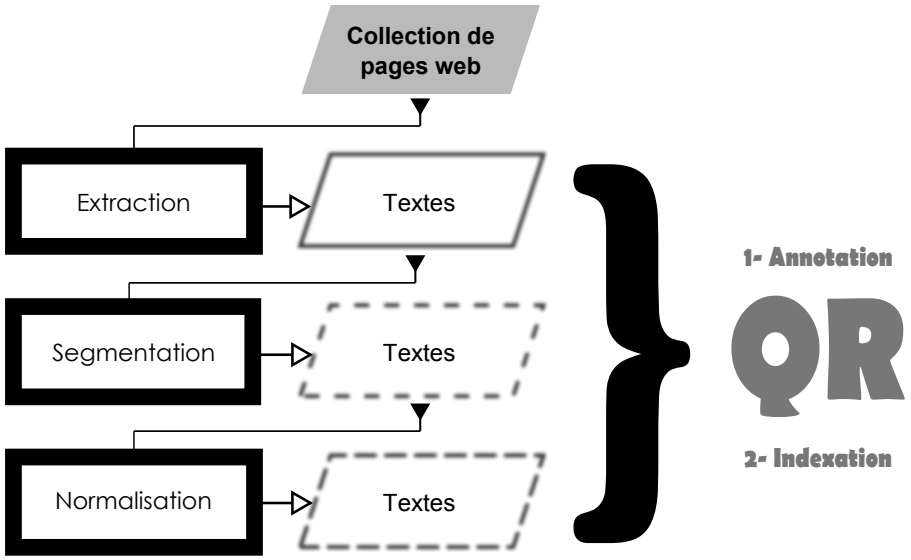


FIGURE 4.6 – Notre procédure de segmentation de pages web en lien avec les pré-traitements QR.

La figure 4.6 présente les étapes-clés de la chaîne de traitement correspondant à la segmentation des documents que nous opérons préalablement à leur indexation. Celles-ci vont de l'extraction du contenu textuel des pages web de recherche à l'obtention de blocs de texte normalisés. Cette pré-segmentation se place en amont de l'annotation des documents sur la chaîne de pré-traitement de RITEL.

Dans cette section, nous détaillons le système de pré-segmentation correspondant à la procédure que nous venons brièvement d'introduire. Les étapes d'extraction, de segmentation et de normalisation sont décrites successivement aux sections 4.2.1, 4.2.2 et 4.2.3 respectivement.

4.2.1 Extraction textuelle

Cette procédure d'extraction se déroule en deux temps : **pré-traitement** (section 4.2.1.1), puis **représentation** et **extraction** du contenu textuel des pages web (section 4.2.1.2).

La phase de pré-traitement des pages web est prise en charge par *Kitten* [31], un outil de traitement de documents web développé au LIMSI. La représentation et l'extraction du contenu textuel des pages web se fait sur les versions des pages pré-traitées par Kitten à l'aide du navigateur textuel de pages web *Lynx*[2].

4.2.1.1 Pré-traitement des pages web

Le pré-traitement des documents web est réalisé à l'aide de Kitten. Ce choix est motivé par les performances état de l'art que ce dernier a obtenues en qualité d'extracteur textuel [31] dans le cadre d'évaluations QR sur le système FIDJI [106].

2. lynx.browser.org

Kitten est un outil développé au LIMSI, dédié aux traitements et à la normalisation de données Html. Les pages web fournies en entrée sont traitées et de nouvelles pages web au format Xhtml valide W3C (encodées en UTF8) sont produites en sortie. Ces pages sont bien formées (correction de leur squelette Html via *jTidy* [3]), sans erreurs d'encodage (correction de leur encodage via *jChardet* [4] et conversion des caractères Html spéciaux dans une base Unicode via *HTMLCleaner* [5]).

Kitten produit des pages web exploitables en Extraction d'Information (EI) [5] sans appliquer d'heuristiques de nettoyage prédéfinies contrairement à des outils classiques de nettoyage de contenu comme *Boilerpipe* [74] ou *Ncleaner* [29]. Par exemple Ncleaner préserve pour l'essentiel le texte des balises `<title>`, `<h1>`, `<h2>`, `<h3>`, `<div>` et `<p>` contenues dans le corps des pages, toute autre balise étant jugée non pertinente pour l'extraction, bien que les informations textuelles véhiculées par les attributs Html de n'importe quelle balise pourraient fournir une information pertinente pour un système d'extraction. Kitten quant à lui dispose de nombreuses fonctions et filtres configurables qui le rendent flexible. Ainsi, il est possible de conserver le contenu des attributs `<title>` associé à un lien tout en supprimant le lien ou au contraire conserver ce lien tout en supprimant les attributs `<title>` qui lui sont associés. Kitten se rapproche donc plutôt d'outils de développement populaires dans le domaine de l'EI web comme la librairie Python *Beautiful Soup* [6] et le framework de crawling web *Scrapy* [7].

Kitten dispose de son propre module d'extraction de pages web et d'un système d'extraction *back-off* basé sur Lynx. Ce dernier sert d'extracteur principal dans notre système de segmentation.

4.2.1.2 Représentation des pages web et extraction textuelle

La représentation des pages utilisée pour l'extraction se fait grâce à Lynx. Ce dernier est un navigateur d'informations distribuées à portée générale pour Internet. Il permet de naviguer sur le web depuis une console, en mode textuel. C'est un outil libre intégré nativement sur la plupart des distributions Linux grand public comme Ubuntu. Il intègre de nombreuses fonctionnalités web, dont l'extraction du contenu textuel de pages web.

Nous avons retenu Lynx pour deux raisons. La première raison est qu'il fournit une extraction textuelle de pages web qui reflète leur rendu visuel. La seconde raison est que Lynx peut fournir une décomposition linéaire du contenu des pages web en blocs de texte, adaptée à la plupart des traitements QR d'analyse de documents.

Si Lynx produit des extractions textuelles fidèles au rendu visuel des pages web, l'agencement des blocs d'extraction diffère de celui observé dans un navigateur web classique du type *Firefox* [8]. En effet, Lynx effectue une traversée gauche-droite descendante des pages web. En conséquence, les blocs d'information textuelle rencontrés le long du parcours sont mis bout à bout dans le fichier d'extraction résultant. Ainsi, on trouve souvent dans les extractions textuelles de Lynx la suite de blocs suivante, donnés ici selon leur contenu visuel : bandeau, menus, colonne gauche, bloc de contenu principal, colonne droite, puis pied de page. On peut aussi trouver des agencements moins stéréotypés selon le *design* des pages et trouver des séries de blocs de contenu principal qui s'enchaînent.

L'étape d'extraction textuelle est réalisée par Lynx via le système d'extraction *back-off* de Kitten.

3. http://jtidy.sourceforge.net
4. Portage Java de la détection automatique d'encodage d'une page du moteur de Mozilla.
5. http://htmlcleaner.sourceforge.net
6. http://www.crummy.com/software/BeautifulSoup
7. scrapy.org
8. http://www.mozilla.org

4.2.2 Stratégie de segmentation

Nous avons utilisé deux stratégies de segmentation en blocs de texte. La première stratégie consiste à segmenter les textes extraits par Lynx en blocs thématiques de taille variable. La deuxième stratégie segmente les textes extraits par Lynx de façon uniforme en blocs de taille fixe.

4.2.2.1 Segmentation par TextTiling

L'algorithme de segmentation thématique que nous avons utilisé correspond à une version de l'algorithme TextTiling créé par Hearst en 1997 [53]. Ce dernier segmente un texte en unités appelées *multi-paragraphes*, en fonction des thématiques abordées dans le texte. Cet algorithme est prévu initialement pour fonctionner sur des textes structurés (e.g. articles de journaux, textes de livres . . .) et de grande taille (c.-à-d. plusieurs pages).

Une des questions sous-jacentes aux expériences que nous avons menées est de savoir si cet algorithme peut s'avérer adéquat à la segmentation de pages web.

L'algorithme original, présenté en détail dans [53], s'articule autour des 3 étapes suivantes :
- **tokenisation** ;
- **calcul de scores lexicaux** ;
- **identification de frontières**.

La tokenisation La procédure de segmentation démarre par une étape de tokenization du texte qui lui est fourni en entrée. Les mots qui sont des *stopwords* ne sont pas tokenisés et sont écartés. Les autres subissent une étape de racinisation basée sur leur *analyse morphologique* fine de la langue. Le texte est ensuite découpé en pseudo-phrases de tokens. Cette dernière étape dépend de la structure initiale du texte. En effet, les paragraphes d'origine servent de point d'ancrage à la constitution des pseudo-phrases. Les paragraphes sont détectés grâce à l'indentation du texte (différents types d'indentation sont reconnus).

La taille des pseudo-phrases (w) est fixée par tuning sur des textes d'entraînement ou bien à l'aide de la configuration standard de l'algorithme de TextTiling, où $w = 20$ tokens.

La proximité lexicale L'algorithme évalue la proximité lexicale qui existe entre les différentes paires (p1,p2) de pseudo-phrases consécutives produites à la tokenization et leur attribue un score i. Ce score est fondé sur les tokens constitutifs de p1 et de p2 ainsi que des tokens trouvés dans leur voisinage respectif v1 et v2 comme indiqué à la figure 4.7. Selon la stratégie utilisée pour le calcul des scores de proximité, ces voisinages varient. Il y a deux stratégies principales associées à l'algorithme de TextTiling pour le calcul de i. Celles-ci sont [9] : *blocks* et *vocabulary introduction* dans les termes de Hearst.

Score lexicaux par « comparaison de blocs » Ici, v1 forme un bloc de pseudo-phrases qui englobe les k pseudo-phrases qui précèdent p1 alors que le bloc formé par v2 englobe les k pseudo-phrases qui suivent p2. Dans ce cas, le score de proximité lexicale i attribué à une paire (p1,p2) donnée se fonde sur le nombre de tokens communs entre v1 complété de p1 d'un côté et v2 complété de p2 de l'autre.

9. Cf. [52] pour une présentation du troisième type de stratégie appelée *chains*, proches des méthodes développées par Morris et Hirst [107] afin d'extraire la structure d'un texte à l'aide de chaînes lexicales [48].

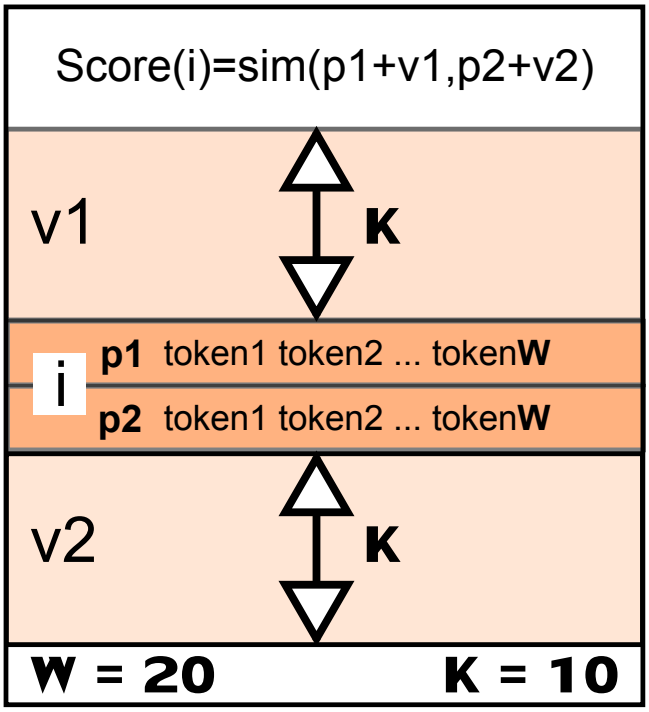


FIGURE 4.7 – Illustration du calcul de scores lexicaux par comparaison de blocs suivi par l'algorithme de TextTiling.

k est fixé par tuning sur des textes d'entraînement ou suivant la valeur standard définie dans l'algorithme de TextTiling, c'est-à-dire $k = 10$ pseudo-phrases [10].

Score lexicaux par « apparition de termes » : v1 correspond à la pseudo-phrase qui précède p1 et v2 à celle qui suit p2. Dans ce cas, le calcul du score lexical attribué à une paire (p1,p2) se fonde sur le nombre de tokens uniques à v1 et à v2 vis-à-vis des tokens déjà rencontrés dans le texte auparavant. En d'autres termes, on cherche ici le nombre de tokens absents du vocabulaire de tokens constitué au fur et à mesure des calculs de scores lexicaux sur les paires de pseudo-phrases ayant précédé la paire (p1,p2) courante.

Le marquage des frontières L'algorithme procède au marquage des frontières pertinentes de paragraphes vis-à-vis des changements thématiques. Ces frontières sont déterminées sur la base des scores lexicaux mesurés précédemment. Ceci est fait à l'aide d'une fenêtre glissante sur ces scores et d'un algorithme de *smoothing* qui permet de gommer les minima locaux jugés non pertinents pour le marquage des frontières. Les minima locaux jugés pertinents servent ensuite de points de coupe à l'algorithme de TextTiling qui extrait tout le texte (blocs de textes ou *multi-paragraphes*) se trouvant entre 2 points de coupe successifs dans le document d'origine. Chaque bloc de texte ainsi extrait est supposé relever d'une thématique différente de celle de ces voisins immédiats.

10. Cette valeur est censée représenter la longeur moyenne d'un paragraphe [53].

L'algorithme de *smoothing* s'appuie sur deux paramètres de tuning s [11] et n [12]. Ces derniers sont déterminés par entraînement ou via les valeurs standard du TextTiling. Dans ce cas, on a $n = 1$ et $s = 2$.

L'implémentation que nous avons utilisée du TextTiling est fournie par le package Python NLTK sans la fonction d'*analyse morphologique*. Le calcul des scores lexicaux se fait par *comparaison de blocs*. Les valeurs des paramètres de tuning que nous avons utilisées sont celles de la configuration standard du TextTiling ($w = 20$, $k = 10$, $n = 1$ et $s = 2$).

4.2.2.2 Segmentation uniforme

Cette segmentation permet de contrôler la segmentation par TextTiling. Elle se contente de segmenter chaque fichier texte qui lui est présenté en 8 blocs, c'est-à-dire la moyenne du nombre de blocs de segmentation obtenus par TextTiling sur notre corpus d'expérimentation au cours de tests préliminaires. On a pu s'apercevoir que ceci revenait finalement à fixer la taille moyenne des blocs en nombre de lignes (voir tableau 4.1).

La segmentation se fait selon un parcours linéaire du texte d'entrée, du début jusqu'à la fin. Les points de coupe sont déterminés à l'avance selon le nombre total de lignes dans le texte et le nombre maximum de blocs fixé en sortie (8). Les textes trop petits (ceux de moins de 8 lignes) ne sont pas segmentés et sont considérés comme des blocs uniques.

4.2.3 Normalisation

Notre normalisation réalise entre autres une séparation des mots et des nombres de la ponctuation, reconstruit la casse sur les mots, ajoute de la ponctuation le cas échéant et sépare en phrases ou pseudo-phrases le texte d'entrée. Elle s'appuie sur des lexiques, des dictionnaires de règles et des modèles de langue [25].

Nous avons vu à la figure 4.3 un exemple de cette normalisation sur le contenu textuel d'un des documents utilisé dans nos évaluations.

4.3 Application de la pré-segmentation pour les évaluations

Dans cette section, nous présentons des statistiques sur les corpus obtenus en appliquant le système de pré-segmentation décrit à la section précédente.

Le tableau 4.1a présente les résultats des traitements (en nombre de fichiers) de chacune des étapes de la chaîne de segmentation et de celles des pré-traitements QR (annotation et indexation) sur le corpus d'évaluation *Q07fr*, pour chaque condition expérimentale : sans segmentation (**bsln**), avec segmentation TextTilling (**TT**) et avec segmentation contrôle (**ctrl**),

On peut noter que le nombre de fichiers issus de la segmentation dans les conditions **ctrl** et **TT** sont proches par construction (environ 20K blocs de différence). En effet, la segmentation de contrôle a été conçue pour fournir les nombres moyens de blocs que fournit la segmentation TextTiling. Les blocs indexés dans ces 2 conditions correspondent aux 484 060 textes indexés en condition **bsln**.

11. Taille de la fenêtre de *smoothing* autour d'un score lexical donné.
12. Nombre de fois où est répétée l'opération de *smoothing* autour d'un score lexical donné.

(a)

Étape/Cond	bsln		ctrl		TT	
extraction	**497 734**	16h	**497 734**	16h	**497 734**	16h
segmentation	-	-	3 686 749	2h	3 857 585	68h
normalisation	485 037	6,5h	3 660 264	15h	3 686 875	15,5h
annotation	485 037	4h	3 660 264	4,3h	3 686 875	4,3h
indexation	**484 060**	4h	**3 658 988**	20h	**3 686 857**	20h
durée totale		∼**30h**		∼**58h**		∼**123h**

(b)

Stat/Index	nbB	nbL	nbB	nbL	nbB	nbL
Min	1	1	1	1	1	1
Max	1	461 075	8	1 262	186	9 277
Sd	0	7 646,5	0,01	18,7	8,4	32,61
Mean	1	295,4	8	19,4	7,3	20,9

TABLE 4.1 – (a) Nombre de blocs par condition expérimentale (**Cond**) selon leur type : segmenté (**ctrl** et **TT**) ou non (**bsln**), selon les étapes nécessaires à les créer (**Etape**) et la durée de leur traitement. (b) Statistiques (**Stat**) du nombre de blocs (**nbB**) et de lignes (**nbL**) moyens (**Mean**), minimum (**Min**), maximum (**Max**) et déviation standard (**Std**) des index (**Index**) relatifs à chaque condition expérimentale.

La durée totale des traitements suivant **ctrl** et **TT** est respectivement 2 à 4 fois plus longue que pour **bsln** [13]. Mis à part en segmentation, les temps de traitement sont identiques entre **ctrl** et **TT**.

Le tableau 4.1b présente le nombre moyen de blocs (**nbB**) et de lignes par bloc (**nbL**) obtenu en conditions **ctrl** et **TT**. Ces informations sont aussi données pour la condition **bsln** à titre indicatif (où un fichier est vu comme un bloc).

On constate que les algorithmes de TextTiling et de contrôle se comportent de façon très similaire : en moyenne, le nombre de blocs produits (**ctrl** : 8 et **TT** : 7,3) ainsi que leur taille (**ctrl** : 19,4 et **TT** : 20,9) sont semblables. Compte tenu des critères de segmentation que nous avions fixés pour l'algorithme de contrôle (8 blocs par fichier, sur la base des résultats de segmentation obtenu via le TextTiling dans nos expériences préliminaires sur 10% du corpus *Q07fr*), ce constat était prévisible. Cependant, on peut remarquer que le TextTiling produit quelques fois plus de segments que la segmentation uniforme : la déviation standard et le nombre maximum d'unités sont plus élevés pour **TT** que pour **ctrl** sur le nombre de lignes et le nombre de blocs.

Dans le cas du TextTiling, le nombre de lignes maximal très élevé est dû à quelques documents particulièrement longs et très peu segmentés (moins de 8 blocs).

4.4 Évaluation

L'évaluation a pour but de mesurer l'impact de notre stratégie de segmentation des documents sur les performances QR. Pour ce faire, nous avons testé notre approche à l'aide de RITEL sur les mêmes corpus de documents (*Q07fr*) et de questions que pour les évaluations du chapitre précédent.

À la section 4.4.1, nous présentons les 3 conditions expérimentales évaluées. Les résultats d'évaluation correspondant sont présentés à la section 4.4.2. Enfin, à la section 4.4.3, nous procédons à deux analyses approfondies autour de la segmentation des documents, en lien avec les performances de RITEL.

13. Tous les traitements sont distribués et exécutés en parallèle sur des serveurs Intel/Xeon L5520 à 2.27GHz, 16 CPUs, 65Go MEM en 64bits, excepté pour les traitements de segmentation suivant **TT**, exécutés sur un Intel/Dual core à 2.80GHz, 20Go MEM en 32bits.

4.4.1 Conditions expérimentales

Les trois conditions expérimentales que nous avons testées sont les suivantes :
- **condition 1** : condition sans segmentation ou baseline (**bsln**) ;
- **condition 2** : condition en segmentation par TextTiling (**TT**) ;
- **condition 3** : condition en segmentation contrôle (**ctrl**).

4.4.2 Résultats

Les résultats sont donnés aux tableaux 4.2a et 4.2b. Nous avons utilisé le test de McNemar [96, 1], implémenté dans R [14], pour juger de la significativité de ces résultats. Les résultats de ces tests sont donnés au tableau 4.3 [15].

En général On constate, au tableau 4.2a, que les deux conditions de segmentation testées obtiennent des résultats proches de la condition baseline (**bsln**), suggérant ainsi que la segmentation n'apporte pas de réels bénéfices au système QR. Les performances de RITEL sont très proches en terme de **précision** (**P@1**) : il y a moins d'1 point de différence entre les conditions **bsln** et **TT**. Mais le **MRR** présente un écart plus important entre les conditions (**bsln** versus **ctrl** : 2 points de différence et **bsln** versus **TT** : 1 point de différence). D'après cette mesure, la segmentation **ctrl** semble permettre au système de trouver de meilleures réponses (c.-à-d. des réponses plus pertinentes) qu'en condition baseline ou TextTiling. Toutefois, d'après les tests statistiques des performances QR présentés au tableau 4.3, ceci n'est qu'une tendance.

En effet, ce tableau nous indique que l'hypothèse nulle, selon laquelle la différence observée entre les conditions **bsln** et **ctrl** n'est pas significative en regard du top-10, est à peine rejetée, la valeur de p obtenue par le test dans ces conditions n'étant pas inférieure mais tout juste alignée sur le seuil critique de significativité α [16].

Test de McNemar Le test de McNemar [96] utilisé dans nos expériences établit la significativité des résultats observés entre 2 conditions A et B et une mesure M donnée, selon des variations observées entre A et B, synthétisées dans une table de contingence 2×2. De là, le test (bilatéral) estime une valeur Q (c.-à-d. le χ^2 de McNemar) pour un degré de liberté df donné et dérive une valeur p (ou p-value). Si p est inférieure (ou égale) au seuil critique α, l'hypothèse nulle H_0 est rejetée et la différence observée entre A et B est jugée significative.

Dans notre cas, une table de contingence comptabilise le total de questions (#q) pour lesquelles RITEL trouve une réponse de même exactitude en conditions A et B. Il y a 4 types de compte, nombre total de questions avec une réponse : correcte (r) selon A et selon B (#rr), fausse (w, xs ou xl) selon A et selon B (#WW), correcte selon A et fausse selon B (#rW) et inversement (#Wr).

Focus sur les résultats du top-10 L'étude du nombre total de bonnes réponses fournies par RITEL selon leur position dans le **top-10** tableau 4.2b (**bsln** : 178, **TT** : 183 et **ctrl** : 190) confirme la tendance déjà mentionnée.

On voit également d'après ce tableau que les réponses apportées par le système en condition **ctrl** (jusqu'à 12 réponses supplémentaires, soit 3,9% de réponses en plus) se trouvent majoritairement dans le top-3, là où la segmentation par TextTiling a tendance à apporter de nouvelles réponses à des rangs inférieurs.

14. http://www.r-project.org
15. Ce dernier est le même sur le top-10 et le MRR, car le test de McNemar ne permet pas de distinguer les réponses selon leur rang.
16. p=0,059 si on augmente la précision du test de McNemar fournie au tableau 4.3 selon **bsln** et **ctrl** en top-10.

Cond	P@1	MRR	P@10	#q
bsln	31.4	39.6	**57.6**	309
ctrl	31.7	**41.6**	**61.5**	309
TT	**32.0**	40.5	59.2	309
Cond	**#r**	**#xs**	**#xl**	**#w**
bsln	97	6	8	198
ctrl	98	11	5	195
TT	**99**	11	2	197

(a)

	Cond		
rang	**bsln**	**ctrl**	**TT**
1	97	98	**99**
2	26	**32**	26
3	17	**26**	19
4-10	38	34	39
Total	**178**	**190**	183
1-3	140	156	144

(b)

TABLE 4.2 – (a) **P@1** : précision. **MRR** : moyenne des rangs réciproques. **P@10** : précision sur 10 rangs (top-10). **#q** : nombre total de questions évaluées. #r, #xs, #xl et #w : nombre total de réponses *justes*, *trop courtes*, *trop longues* et *fausses*, selon le top-1). (b) Focus sur les résultats du top-10 présentés en (a). Pour chaque position (**rang**), est indiqué le nombre de réponse pertinentes trouvées par RITEL (c.-à-d. réponse de type *justes*).

df=1,α=.05		**ctrl** (A) / **bsln** (B)					**bsln** (A) / **TT** (B)					**ctrl** (A) / **TT** (B)				
mesure	#q	**r**	**W**	Q	p	nul	**r**	**W**	Q	p	nul	**r**	**W**	Q	p	nul
P@1	**r**	77	21	0	1	×	80	17	.02	.86	×	80	18	0	1	×
	W	20	191				19	193				19	192			
P@10	**r**	167	23	3.55	.05	✓	164	14	.48	.48	×	174	16	1.44	.23	×
(MRR)	**W**	11	108				19	112				9	110			

TABLE 4.3 – Résultats de significativité du test (bilatéral) de McNemar pour les résultats QR du tableau 4.2a. Q : χ^2 de McNemar. df : degré de liberté. p : p-valeur. α : seuil critique. nul : hypothèse nulle (rejet : ✓). #q : nombre total de questions pour lesquelles on a une réponse de même nature ou non entre 2 conditions A et B. **r** : réponse *juste*. **W** : réponse *fausse*. Les rectangles **rW/rW** représentent des tables de contingence.

Remarque à propos des temps d'exécution Nous avons pu constater que la segmentation des documents accélérait les traitements QR. Ceci est un point particulièrement intéressant pour des systèmes QR avec de fortes contraintes de fonctionnement temps réel (p. ex. dans le cas de systèmes QR interfacés à un système de dialogue comme RITEL). Ce point contre-balance positivement le surcoût des traitements pré-QR induit par la segmentation des documents comme nous avons pu le voir à la section 4.3.

4.4.3 Analyses

Dans la mesure où les résultats dans les différentes conditions présentées précédemment sont très proches, il nous a paru nécessaire de procéder à des analyses supplémentaires afin de conclure sur l'intérêt d'un segmentation. Les analyses présentées dans cette section ont pour but d'étendre les résultats précédents et de nous permettre de mieux comprendre l'impact de la segmentation des documents sur le comportement de RITEL.

À la section 4.4.3.1, nous nous intéressons aux taux de couverture de questions que représentent les réponses trouvées par le système pour les 3 conditions expérimentales évaluées. À la section 4.4.3.2, nous étudions finement l'effet de la segmentation uniforme sur les différentes composantes de RITEL.

4.4.3.1 Analyse 1 : couverture intra- et -inter-système

On considérera par la suite que $\text{RITEL}_{\text{cond}}$ où cond = bsln, ctrl ou TT correspondent à 3 instances différentes de RITEL, une pour chaque condition expérimentale.

Préambule Nous avons présenté à la section précédente les résultats obtenus par RITEL en conditions classiques (**bsln**) et avec segmentation (**TT** et **ctrl**). Dans cette section, nous nous intéressons à la complémentarité entre ces différentes approches. Notre objectif est d'observer comment ces différentes approches opèrent pour aboutir à une bonne réponse. En effet, l'idée ici est de combiner les sorties de RITEL fournies dans chaque condition afin d'appliquer une procédure de réordonnancement sur la combinaison obtenue et d'évaluer théoriquement le potentiel d'une telle démarche préalablement à son implémentation. On examinera donc, pour chaque condition expérimentale, les réponses fournies par le système $\text{RITEL}_{\text{cond}}$.

Le tableau 4.4a présente la couverture de RITEL aux rangs 1 (**top-1**) et 10 (**top-10**). Les comptes indiqués sous chaque condition aux lignes *all* et *uniq* donnent le nombre de questions auxquelles RITEL a su répondre correctement au total (all) et parmi celles-ci, le nombre de questions auxquelles seul ce système a fourni une réponse correcte (uniq). Nous appellerons ce compte (uniq) la *couverture intra-système*. Par exemple, en condition de référence (**bsln**) les réponses de rang 1 trouvées par RITEL couvrent 97 des 309 questions de test ; nous dirons que la couverture intra-système est de 10, c'est-à-dire que $\text{RITEL}_{\text{bsln}}$ et seul $\text{RITEL}_{\text{bsln}}$ a su trouver et placer en rang 1 (top-1) une réponse correcte à 10 des questions évaluées.

La colonne de droite dans ce tableau indique pour les différentes classes de compte (all et uniq) le nombre de questions communes auxquelles RITEL a su répondre, toutes conditions confondues. Par exemple, en (**top-1**), on voit que le nombre de questions communes couvertes par RITEL à travers les 3 conditions est 70. Ce nombre représente la *couverture inter-systèmes* en top-1. Sans surprise, on constate (par construction) que la couverture est nulle ($|\textbf{bsln} \cap \textbf{ctrl} \cap \textbf{TT}| = 0$) lorsqu'on considère l'intersection des couvertures intra-systèmes (ligne top-1.uniq), **bsln**.uniq, **TT**.uniq et **ctrl**.uniq [17].

On peut noter que les taux de couverture inter-systèmes sont légèrement inférieurs dans le cas du **top-10** (chiffres en vert et noir à la ligne **top-10**.uniq). En effet, plus on considère de réponses, plus il y a de chances que plusieurs systèmes aient trouvé des réponses équivalentes. Ainsi, plus on considère de réponses moins la diversité des questions couvertes est large et plus la couverture intra-système relative à chaque condition diminue, et le taux de couverture inter-systèmes augmente (environ 70% en top-1 versus 75% en top-10).

Potentiel du réordonnancement des sorties combinées de RITEL en conditions bsln, ctrl et TT Dans ce paragraphe, nous estimons théoriquement le potentiel d'un réordonnancement de réponses sur les sorties combinées de $\text{RITEL}_{\text{bsln}}$, $\text{RITEL}_{\text{ctrl}}$ et RITEL_{TT}, sur la base des analyses présentées plus tôt. Les résultats de cette estimation sont données au tableau 4.4b.

Ici, on se place dans le contexte d'un réordonnancement parfait (réponse oracle), c'est-à-dire qui placerait en tête de classement toutes les réponses aux questions couvertes spécifiquement dans chaque condition.

Dans ces conditions, on atteint 117 réponses en top-1 (c.-à-d. la somme des 98 réponses de rang 1 trouvées par $\text{RITEL}_{\text{ctrl}}$ avec les 10 réponses aux questions couvertes spécifiquement par $\text{RITEL}_{\text{bsln}}$ et les 9 correspondantes pour RITEL_{TT}), comme indiqué à la première ligne dans le tableau 4.4b.

17. En effet, ces couvertures représentent des ensembles de questions disjoints et donc, leur intersection est l'ensemble vide.

(a)

	Couverture			
	intra-système			inter-système
top-1	**bsln**	**ctrl**	**TT**	**bsln ∩ ctrl ∩ TT**
all	97	98	99	70
uniq	10	11	9	0
top-10	**bsln**	**ctrl**	**TT**	**bsln ∩ ctrl ∩ TT**
all	178	190	183	159
uniq	6	8	4	0

(b)

Mesure	∑	total	#q	oracle	**ctrl**	g-ctrl	**bsln**	g-bsln
P@1	98+10+9	117	309	**37.8**	31.7	+6.1	31.4	+6.4
P@10	190+6+4	200	309	**64.7**	61.5	+3.2	57.6	+7.1

TABLE 4.4 – **(a)** Couvertures intra-système et inter-système de RITEL, en condition expérimentale avec (**ctrl** : segmentation uniforme, **TT** : segmentation par TextTiling) et sans segmentation (**bsln**), en fonction du **top-1** et du **top-10**. all : nombre total de questions couvertes par des réponses correctes. uniq : nombre total de questions auxquelles sont fournies des réponses correctes, uniquement par la condition spécifiée. La colonne "inter-système" indique le nombre de questions communes aux 3 conditions pour les 2 types de mesure considérés (all et uniq). **(b)** Précision au rang 1 (**P@1**.oracle) et 10 (**P@10**.oracle) dans le cas d'un réordonnancement parfait à partir des sorties combinées trouvées par RITEL dans chaque condition vis-à-vis des conditions **ctrl** et **bsln** (respectivement g-ctrl et g-bsln).

De cette somme, si on recalcule la précision (P@1) correspondante (colonne *oracle*, même ligne) sur la base des 309 questions de test, on obtient un score de **37.8%**. Comme indiqué dans le tableau aux colonnes *g-bsln* et *g-ctrl*, les gains correspondant comparé aux scores de précision de $\text{RITEL}_{\text{bsln}}$ et $\text{RITEL}_{\text{ctrl}}$ sont respectivement de +6.4% et +6.1%.

Les informations relatives au **top-10** (deuxième ligne de la table 4.4b) permettent de juger du maximum de performances atteignable en appliquant une procédure de réordonnancement sur les sorties combinées de RITEL dans les 3 conditions sur le corpus *Q07fr*. Dans ce cas, on voit qu'il est possible d'opérer un gain de +3.2% de rappel comparé aux performances de RITEL en condition **ctrl** (**P@10**.oracle : 64.7% versus **P@10**.ctrl : 61.5%) et +7.1% vis-à-vis de la condition de référence (**P@10**.bsln : 57.6%).

4.4.3.2 Analyse 2 : impact de la pré-segmentation par module QR

Dans cette section, nous procédons à un diagnostic fin de l'impact de la segmentation uniforme face au comportement habituel de RITEL. L'analyse concerne successivement l'étape de détection des entités nommées de RITEL, la sélection des documents, l'extraction de passages et de réponses. En effet, bien que globalement les gains obtenus à l'aide d'une segmentation soient réduits, nous avons voulu voir l'effet de la segmentation sur chacune de ces étapes, afin de valider l'hypothèse **H0**.

Support du diagnostic Les statistiques de la table 4.5 réfèrent à des taux de couverture ($\%c$ et le taux complémentaire $\%\bar{c}$) d'une part, et à des quantités de réponses ($\#r_{in}$ et $\#r_{out}$) d'autre part. Les taux s'expriment en pourcents, les quantités en nombre de réponses. Ces statistiques sont données sur la base des 309 questions de test utilisées dans nos évaluations et des réponses de référence dont nous diposons pour chacune d'elle.

Contenu du diagnostic En bleu foncé à la table 4.5, on indique les étapes de traitements analysées avant l'extraction de réponses, par ordre d'exécution : le repérage d'entités nommées (**Entity**), la sélection de documents pertinents (**Document**) et l'extraction de passages pertinents (**Snippet**).

	bsln				**ctrl**			
Statistiques	$\%c$	$\%\overline{c}$	$\#r_{in}$	$\#r_{out}$	$\%c$	$\%\overline{c}$	$\#r_{in}$	$\#r_{out}$
Questions	100	0.0	**309**	0	100.0	0.0	**309**	0
Entity (E)	**84.8**	15.2	262	47	**85.4**	14.6	264	45
E+T	80.3	19.7	248	61	80.9	19.1	250	59
Document (D)	**86.1**	13.9	266	43	**83.5**	16.5	258	51
D+E	78.6	21.4	243	66	76.4	23.6	236	73
D+E+T	74.4	25.6	230	79	**71.8**	28.2	222	87
Snippet (S)	**77.7**	22.3	240	69	**78.0**	22.0	241	68
S+E	72.2	27.8	223	86	72.5	27.5	224	85
S+E+T	69.3	30.7	**214**	95	68.6	31.4	**212**	97
Answer (P@10)	**57.6**	42.4	**178**	131	**61.5**	38.2	**190**	119

TABLE 4.5 – Diagnostic des performances de RITEL par étape de traitement (analyse de la question exclue).

Chaque ligne de la table a la signification suivante : Questions : nombre total de questions évaluées. **Entity** (E) : combien de fois une réponse de référence se trouve à l'intérieur de l'une des entitées nommées détectées par RITEL dans l'ensemble des documents de l'index. Par exemple, en condition **ctrl** on voit que le taux de couverture correspondant est de **85.4%**. E + Type (T) : combien de fois une réponse est couverte par une EN correctemement détectée et typée [18] par RITEL dans les documents de l'index. **Document** (D) : combien de fois une réponse se trouve parmi l'un des documents sélectionnés par RITEL. Par exemple, en condition **ctrl** on voit que le taux de couverture correspondant est de **83.5%**. D+E : combien de fois une réponse correctement couverte par une EN se trouve parmi les documents pertinents sélectionnées par RITEL à l'étape de Recherche d'Information. D+E+T : similaire à D+E, mais en incluant le typage des EN. **Snippets** (S) : même idée que pour D, mais concernant les passages. En condition **ctrl**, on voit que le taux de couverture associé est de **78.0%**. S+E et S+E+T : même idée que pour D+E et D+E+T respectivement, mais concernant les passages. Answer : performances QR de RITEL relativement au rang 10 (top-10 ou P@10). On retrouve ici les résultats d'évaluation de notre procédure de segmentation présentés aux tableaux 4.2a et 4.2b.

Diagnostic D'après le tableau 4.5, on constate globalement que les performances de RITEL aux différentes étapes de la chaîne QR sont meilleures en condition de segmentation (**ctrl**) qu'en condition baseline (**bsln**). En condition **ctrl**, on voit que RITEL a extrait **190** des **212** réponses de référence contenues dans les passages sélectionnés. Cette quantité de réponses correspond à une précision au rang 10 de **61.5%**. De même, on voit qu'en condition **bsln** la précision correspondante est de **57.6%** (alternativement 42.4% d'échec, comme indiqué à la colonne $\%\overline{c}$; même ligne).

18. Le type ici est le type attendu de la réponse, comme défini à l'analyse de la question à partir de sa classe.

Entity Dans la section « Entity » du tableau 4.5, on constate qu'il y a plus de réponses couvertes par les EN détectées par RITEL (E) en condition **ctrl** qu'en condition **bsln** (environ 1 point), mais un peu moins suivant la couverture correcte des réponses en EN. Ce constat s'estompe quand on considère ces EN correctement typées (E+T) (**bsln**.%c.E+T : 80.3% versus **ctrl**.%c.E+T : 80.9%).

Ces résultats suggèrent que la détection en EN de RITEL est (légèrement) facilitée dans un contexte réduit à une partie du texte (c.-à-d. à l'échelle des segments), plutôt que dans un contexte englobant tout le texte (c.-à-d. à l'échelle des documents). Le corrolaire de ce constat serait que l'analyse linguistique opérée par RITEL en détection d'EN bénéficie d'un contenu linguistique, en particulier lexical et sémantique, plus cohérent à l'échelle des segments. Ceci va dans le sens de l'hypothèse **H0**.

Document Dans la section « Document » du tableau 4.5, on constate que la segmentation semble avoir un impact inverse à celui observé sur la détection en EN. En effet, on constate que les documents sélectionnés par RITEL ont une meilleure couverture en condition **bsln** (**86.1%**) qu'en condition **ctrl** (**83.5%**). **Une hypothèse possible** consiste à penser que plus les documents extraits sont longs, plus leur couverture est large et plus il est probable d'y trouver de réponses. Il en va de même quand on considère le marquage (D+E) et le typage (D+E+T) des EN dans les documents. Cette perte démontre que RITEL laisse de côté des documents pertinents, lorsqu'ils ont été segmentés.

Par ailleurs, on observe une chute totale de 12% en terme de couverture entre D et D+E+T : en condition **ctrl**, on perd 22 réponses en passant de D à D+E, puis 14 en passant de D+E à D+E+T. **Ceci reflète l'impact plus prononcé des erreurs et/ou absence de marquage des EN sur la couverture des documents comparé à celui des erreurs de typage**. Ici, nettement, l'hypothèse **H0** est en défaut.

Snippet De la même manière que pour la sélection de documents, on voit dans la section « Snippet » du tableau 4.5 que l'extraction de passages pertinents fait chuter graduellement les seuils maximum de performances envisageables en sortie QR. En effet, on passe de **78.0%** (S), à 72.5% (S+E) et jusqu'à 68.6% (S+E+T) de couverture en condition **ctrl** (c.-à-d. une chute d'environ 10% de couverture).

On notera que la couverture des passages (S) est légèrement supérieure en condition **ctrl** (**78.0%**) que **bsln** (**77.7%**). Cependant, on notera également que l'impact du marquage en EN (S+E) et de leur typage (S+E+T) est plus sévère sur l'extraction de passages en condition **ctrl** qu'en condition **bsln** (p. ex. **ctrl**.%$\overline{c}$.S+E+T : 30.7% versus **ctrl**.%$\overline{c}$.S+E+T : 31.4%).

Ceci pourrait s'expliquer de façon similaire aux différences de couvertures relevées en conditions **bsln** et **ctrl** à l'extraction des documents (D) : la plus faible quantité d'information disponible dans un segment comparé à celle d'un document complet fait que la probabilité d'extraire des passages contenant des EN accordées avec le type de réponse attendu par la question dans le cas de segments est plus faible que pour des documents complets.

Answer Concernant l'étape d'extraction des réponses, en condition **ctrl** on voit que RITEL a réussi à extraire **190** des **212** réponses couvertes par les passages sélectionnés (S+E+T). Ce nombre de réponses correspond à une précision au rang 10 (P@10) de **61.5%** de succès, le plafond maximum étant 68.6% (**ctrl**.%c.S+E+T). Ce plafond représente le seuil de rappel (top-10 ou P@10) maximum atteignable par RITEL, si les 212 réponses couvertes par les passages étaient effectivement extraites dans le top-10 à la sortie du module d'extraction de réponses.

Or, il est intéressant de remarquer que si $\text{RITEL}_{\text{bsln}}$ peut atteindre un seuil de performances maximales légèrement plus élevé que $\text{RITEL}_{\text{ctrl}}$ (respectivement seuil optimal en rappel de 69.3% et de 68.6%), le premier présente de moins bonnes performances finales que le second : en condition **bsln** le système extrait 12 réponses de moins qu'en condition **ctrl** (**178** réponses extraites au total), pour un taux de couverture final en sortie QR environ 4 points inférieur à celui obtenu en condition de segmentation (c.-à-d. **57.6%**). Une explication possible peut être que, bien que l'extraction de passages moins longs en condition **ctrl** (20 lignes) qu'en condition **bsln** (300 lignes) conduit à une perte d'informations, en retour la petite taille des passages semble faciliter la tâche d'extraction des réponses.

Récapitulatif **Les analyses entreprises dans cette section suggèrent d'une part** que la qualité du travail de détection en EN joue un rôle primordial pour la recherche des documents et l'extraction de passages pertinents en QR, la perte totale de couverture relative à ces 2 étapes avoisinant les 20% (à peu près 12% dans le permier cas et 10% dans le second), avec un impact plus important selon le marquage en EN que celui de leur typage. Ceci suggère que le système d'annotation en EN de RITEL mérite d'être plus travaillé à court terme que celui du système de typage des EN. **D'autre part, ces analyses montrent que** la segmentation des documents a un impact positif sur l'extraction de passages (S) et de réponses, la détection et le typage des EN dans les document de l'index (E). Cependant, l'impact de la segmentation est négatif sur la sélection de documents (D).

4.5 Conclusion

La question à laquelle nous avons tenté de répondre dans cette section est : segmenter les documents à l'indexation (**index-time passaging**) en plus de leur découpage habituel en passages au moment des recherches (**search-time passaging**) aide-t-il un système de Question-Réponses dans sa tâche ?

Pour cela, nous avons préservé la segmentation classique des documents en passages de RITEL au moment des recherches et testé deux types de pré-segmentation des documents avant indexation. Ces pré-segmentations sont supportées par une extraction du contenu textuel des pages web à l'aide de Kitten et de Lynx. L'une segmente les textes extraits en blocs lexicalement cohérents de taille variable par TextTiling, l'autre les segmente uniformément en blocs de taille fixe par un algorithme dédié.

Les résultats obtenus ne nous permettent pas de trancher nettement en faveur de l'une ou l'autre de ces approches, les tests de significativité (McNemar) que nous avons opérés sur les résultats QR de RITEL avec et sans segmentation ne révèlent qu'une tendance. Ainsi, pré-segmenter les pages web selon notre procédure semble aider un système de QR dans sa tâche, les performances de RITEL étant globalement meilleures en contexte de segmentation qu'en contexte de référence. Cette tendance est plus marquée pour la segmentation uniforme de pages web que pour une segmentation plus « intelligente » par TextTiling. Les performances de RITEL s'améliorent d'environ 0.5 point en terme de précision et 4 points en terme de rappel avec une segmentation uniforme comparé au fonctionnement habituel du système.

Ce constat est contradictoire avec d'autres travaux (p. ex. ceux de Salton [130]), mais confirme certaines conclusions apportées par Hearst dans ses applications de la segmentation thématique en Recherche d'Information textuelle [53]. Plusieurs hypothèse s'offrent à nous pour tenter d'expliquer ce constat.

Une première hypothèse concerne notre procédure de segmentation de pages web. En effet, en théorie nous aurions dû inverser les étapes de normalisation et de segmentation, afin de fournir au TextTiling des extractions textuelles correctement normalisées. Une telle inversion nécessite de modifier notre chaîne de normalisation.

Pour cela, il faut désactiver le retrait de l'indentation des textes, utile à l'algorithme de TextTiling, ce que nous n'avons pu faire, faute de temps.

Un autre facteur expérimental qui pourrait expliquer les résultats obtenus concerne l'implémentation même du TextTiling que nous avons utilisée. Nous rappelons ici que l'implémentation NLTK du TextTiling n'intègre pas la composante d'*analyse morphologique* utilisée par Hearst dans la version d'origine du TextTiling en tokenization, et que le calcul des scores lexicaux se fait par « comparaison de blocs » (méthode *blocks*). On peut supposer que la segmentation opérée dans cette configuration est assez primitive et ne répond que peu à notre volonté d'effectuer une segmentation thématique. Il serait intéressant de tester dans quelle mesure un calcul des scores lexicaux basé sur la théorie de la cohésion lexicale (méthode *chains*) ou par « apparition de termes » (méthode *vocabulary introduction*) n'est pas plus adapté que la segmentation par la méthode *blocks*.

Une autre hypothèse vraisemblable concerne la nature des documents avec lesquels nous avons travaillé. Nous utilisons le TextTiling sur des textes issus de pages web, par définition non structurés et relativement courts dans le cas de segments, alors que ce dernier est paramétré pour fonctionner sur des documents textuels longs et bien structurés. Il pourrait être nécessaire de reparamétrer l'algorithme pour les données que nous avons évaluées, mais aussi de les répliquer sur un autre corpus afin de vérifier dans quelle mesure ce constat dépend de l'utilisation du corpus *Q07fr*.

Pour nous aider dans ces investigations, des analyses manuelles de corpus sont nécessaires, afin de repérer dans quels cas le TextTiling permet d'extraire des réponses que la segmentation uniforme ne permet pas.

D'autres méthodes de segmentation devraient être envisagées et, notamment, des méthodes de segmentation textuelle en thème afin d'évaluer les résultats obtenus par le biais du textTiling mais aussi des méthodes de segmentation « visuelle », telles que celles dont nous avons parlé en introduction de chapitre. En effet, ce dernier type de segmentation nous semble fortement adapté à la segmentation de pages web, la structuration du contenu textuel des pages sur le web étant pensée autour de leur rendu visuel.

Dans l'optique de continuer à explorer la segmentation uniforme des documents, il serait intéressant de déterminer la fenêtre optimale de segmentation des documents (≥ 20 ou ≤ 20 lignes de texte), en prenant en référence les travaux initiés par Tiedemann *et al*. [146] ainsi que par Khalid *et al*. [73].

La première analyse détaillée que nous avons menée pour étudier l'impact de la segmentation des documents sur le comportement de RITEL a confirmé que l'emploi d'une procédure de réordonnancement de réponses sur les sorties combinées du système avec (**ctrl** et **TT**) et sans segmentation (**bsln**) présentait un potentiel intéressant pour améliorer les performances QR finales.

Nous avons pu montrer théoriquement qu'il était possible d'obtenir, dans le contexte d'un réordonnancement parfait de réponses (c.-à-d. qui permettrait de placer en tête de classement toutes les réponses aux questions couvertes spécifiquement dans chaque condition), jusqu'à 6.4% de gain en précision (P@1) et 7.1% en rappel (P@10) comparé aux performances de $\text{RITEL}_{\text{bsln}}$, et 6.1% de gain en précision et 3.2% en rappel comparé aux performances de $\text{RITEL}_{\text{ctlr}}$.

Ceci est un résultat plutôt positif qui devrait mener à des tests en condition réelle d'une procédure de réordonnancement combinant les différentes sorties de RITEL avec et sans segmentation. Un exemple de procédure de réordonnancement développée par Guillaume Bernard [9] au cours de sa thèse sur RITEL est brièvement présentée au chapitre 2. L'utilisation d'une procédure d'aggrégation des sorties de plusieurs réordonnanceurs telle que celle proposée récemment par Hong et Si [56] pour éviter un réordonnancement sous optimal du à l'emploi d'un unique réordonnanceur est une voie à explorer dans ce travail.

La seconde analyse détaillée que nous avons menée pour étudier l'impact de la segmentation des documents sur le comportement de RITEL montre globalement que la segmentation des documents a un impact positif sur l'extraction de passages, la détection et le typage des EN dans les documents de l'index et l'extraction des réponses. De ce point de vue, il semblerait que nous puissions valider l'hypothèse **H0**. Cependant, le seuil maximum de réponses atteignable après extraction des passages est supérieur en condition de référence. Pour expliquer ce phénomène, nous avons émis l'hypothèse que RITEL produit moins de passages à partir de documents segmentés (environ 20 lignes) qu'à partir de documents non segmentés (environ 300 lignes) mais qu'en retour, la tâche d'extraction de réponses en est facilitée. Cette hypothèse reste à vérifier. Par ailleurs, l'effet de la segmentation est négatif sur la phase de sélection des documents de RITEL (les taux de couverture des documents sélectionnés par le système en sortie de l'étape RI sont moins bons avec que sans segmentation) ; ceci pourrait s'expliquer par la taille relativement faible d'un segment par rapport à celle d'un document entier, un document contenant plus probablement une réponse qu'un segment d'un point de vue statistique. Cette hypothèse reste également à verifier.

La pré-segmentation permet à RITEL d'obtenir de meilleures couvertures en EN correctement marquées et typées. En revanche, le marquage et le typage corrects des EN dans les passages/documents segmentés trouvés par le système au moment des recherches sont moins bons. Notamment, si les taux de couverture des réponses dans les passages extraits par RITEL sont (légèrement) plus élevés en contexte de segmentation, une fois pris en compte le marquage et le typage des documents en EN, ces taux deviennent meilleurs en contexte de référence. Ce point souligne le besoin d'amélioration du système de reconnaissance en EN de RITEL (bien que celui-ci présente déjà 85% de précision en contexte de segmentation). En conséquence, nous dirons que **H0** n'est validée que partiellement par nos expériences de segmentation.

Au prochain chapitre, nous utilisons notre système de segmentation pour appliquer la méthode d'évaluation de la pertinence intrinsèque d'un document à l'échelle des segments.

« Science is a way of trying not to fool yourself. The first principle is that you must not fool yourself, and you are the easiest person to fool. » [34]

RICHARD FEYNMAN

Chapitre 5

Méthode d'évaluation de la pertinence intrinsèque sur des documents pré-segmentés en QR

Sommaire

5.1 Introduction

DANS ce chapitre, nous appliquons la méthode d'évaluation de la pertinence intrinsèque d'un document (méthode EPID) sur des parties de documents (segments), issues du système de segmentation de pages web présenté au chapitre précédent. Nous pourrons ainsi contrôler l'hypothèse **H0**, émise suite aux évaluations de la méthode EPID en conclusion de chapitre 3 (cf. page 78), et qui est à l'origine de la création du système de segmentation. Ce changement de paradigme (soit échelle des documents versus échelle des segments) nous permet d'étudier de manière approfondie les liens de dépendance existant entre les performances de RITEL et les paramètres principaux qui définissent la méthode EPID.

Ce chapitre s'articule autour de 3 sections. À la section 5.2 nous présentons la démarche que nous avons suivie pour appliquer la méthode EPID à l'échelle des segments. Nous y comparons également la classification en *pertinent* ou *non pertinent* des pages complètes ou des segments, sur le corpus *Q07fr*.

À la section 5.3, nous évaluons d'abord l'impact de la méthode EPID sur la sélection des documents de RITEL à l'échelle des segments, puis comparons ces résultats de filtrage à ceux obtenus précédemment à l'échelle des documents. Ensuite, nous présentons une étude des dépendances entre les performances QR de RITEL et les paramètres de filtrage de la méthode EPID à l'échelle des segments. À la section 5.4, nous concluons à propos des travaux présentés dans ce chapitre.

5.2 Méthode EPID appliquée à l'échelle des segments

Nous appliquons la méthode EPID à l'échelle des segments selon la même procédure utilisée à l'échelle des documents. Nous avons pour cela besoin de listes de filtrage composées de segments. Nous devons donc appliquer la méthode EPID sur une version segmentée du corpus *Q07fr*, à l'aide du même modèle de langue que celui utilisé à l'échelle des documents et de prédicteurs adaptés pour classer des segments.

La méthode de création des classifieurs reste la même. Cependant le modèle de documents utilisé pour leur création (c.-à-d. le corpus **DEV509**) a besoin d'être également segmenté, à l'image du corpus *Q07fr*, utilisé dans les évaluations du chapitre précédent. Par ailleurs, les segments obtenus à partir du corpus DEV509 qui ne sont pas pertinents du point de vue QR doivent être éliminés. Un segment non pertinent est un segment qui ne contient pas de réponses aux questions de référence qui accompagnent le corpus DEV509. Faisant cela, nous aboutissons à la création du corpus **DEV698**.

La section 5.2.1 présente nos choix de segmentation pour l'application de la méthode EPID à l'échelle des segments, et le corpus DEV698 utilisé comme modèle de documents pertinents pour la sélection des documents en QR à cette échelle. La section 5.2.2 présente les distributions, moyennes et écarts types OOV et PPX obtenus sur la base du corpus DEV698, afin de générer les classifieurs utilisés pour générer les listes de segments pertinents utilisées en filtrage. La section 5.2.3 présente les taux de sélection relatifs à chaque classifieur, obtenus en appliquant la méthode EPID à l'échelle des segments sur le corpus de documents *Q07fr*.

5.2.1 Segmentation des données

Suite aux évaluations QR présentées au chapitre précédent, nous avons opté pour la segmentation uniforme de pages web des corpus *Q07fr* et DEV509, qui a démontré une plus grande simplicité et de meilleures performances que la méthode de segmentation par TextTiling.

Corpus d'évaluation L'ensemble des segments uniformes relatifs aux pages web du corpus *Q07fr*.

Modèle de documents pertinents DEV698 Le corpus de développement doit inclure des documents pertinents, c'est-à-dire des documents contenant une réponse appartenant au corpus de questions-réponses de développement utilisé. Pour cela, la procédure est : (i) nous avons extrait du corpus segmenté *Q07fr* les segments correspondant aux documents du corpus DEV509 et (ii) nous avons classé l'ensemble de ces segments en *pertinents* ou *non pertinents* pour les recherches, selon la présence ou l'absence respectivement d'une réponse de référence à l'intérieur des segments. Ce tri a été fait automatiquement, puis complété et validé à la main.

Nous avons donc composé un nouveau corpus de documents appelé **DEV698** qui sert de modèle de documents pertinents pour la méthode EPID appliquée en contexte de segmentation. Ce corpus est composé de 698 segments en exacte correspondance avec les documents du corpus DEV509 : tous les segments qui le composent sont couverts par les documents du corpus DEV509.

Méthode	**N**		**R**	
Param/Stat	M	SD	M	SD
OOV$_{\text{DEV698}}$	1.89	2.88	1.45	1.61
PPX$_{\text{DEV698}}$	226.3	296.3	184.6	119.3
OOV$_{\text{DEV509}}$	1.74	1.98	1.46	1.12
PPX$_{\text{DEV509}}$	210.2	252.9	187.6	106.1

TABLE 5.1 – Moyennes (M) et écarts types (SD) estimés pour les paramètres OOV et PPX en fonction des corpus **DEV698** (segments) et **DEV509**, et des méthodes *normale* (**N**) et *restreinte* (**R**) d'estimation des distributions OOV/PPX.

5.2.2 Distributions, moyennes et écarts type OOV et PPX à l'échelle des segments

On donne au tableau 5.1 les moyennes et écarts types calculés sur le corpus DEV698 comme nous l'avions fait sur le corpus DEV509, en terme de ratio de mots hors vocabulaire (OOV) et perplexité (PPX) suivant chacune des deux méthodes d'estimation des distributions relatives à OOV et PPX (méthode *normale* ou **N** versus méthode *restreinte* ou **R**). Les résultats obtenus sur le corpus DEV509 ont été reportés directement dans ce tableau.

Moyennes OOV et PPX pour les corpus DEV698 et DEV509 On peut noter que si les résultats moyens en terme d'OOV et de PPX sont très proches entre les 2 corpus selon la méthode restreinte, ils iffèrent selon la méthode normale. En effet, le calcul des distributions relatives aux paramètres OOV et PPX a tendance à lisser les moyennes et les écarts types calculés avec le retrait des documents/blocs marginaux suivant la méthode restreinte.

Nous observons que l'utilisation de la méthode normale aboutit à des valeurs de OOV et PPX légèrement supérieures pour le corpus DEV698 (OOV : 1.89 et PPX : 226.3) que pour le corpus DEV509 (OOV : 1.74 et PPX : 210.3). L'hypothèse **H0** nous incitait pourtant à envisager l'inverse. Ainsi, l'application de la méthode EPID à l'échelle des segments, ne semble pas faire une grande différence du point de vue du calcul des distributions relatives aux paramètres OOV et PPX.

Il est vraisemblable que les différences constatées soient provoquées par les changements de normalisation dus aux traitements d'extraction textuelle opérés par la procédure de segmentation de pages web. De premières analyses permettent en effet de constater que Lynx ajoute certaines informations lors de ses extractions, en lien notamment avec la structure visuelle des pages ou des contenus multi-média. Par exemple il ajoute les symboles ,-,# et $\star$ en tête de phrase pour marquer l'indentation du texte dans les menus, ou encore des expressions textuelles telles que `::happy::` pour représenter les balises d'émoticônes.

Écarts types OOV et PPX pour les corpus DEV698 et DEV509 Contrairement aux valeurs moyennes dont nous venons de discuter, les valeurs relatives aux écarts types associées aux paramètres OOV et PPX sont toujours plus importantes pour le corpus DEV698 que pour le corpus DEV509.

Deux hypothèses ici peuvent expliquer ces résultats. La première est que l'effet de lissage des erreurs de normalisation et l'effet lié aux formes inconnues du modèle de langue que nous avions constaté à l'échelle des documents ne se produisent pas à l'échelle des segments. En effet, la quantité d'information contenue dans les pages segmentées est en moyenne huit fois plus faible que dans les pages complètes. La seconde hypothèse est que l'ajout d'informations textuelles opéré par Lynx en pré-segmentation joue aussi un rôle dans les écarts observés.

5.2.3 Classifieurs

Les graphiques 5.1a et 5.1b donnent les taux de sélection obtenus par application de la méthode EPID à l'échelle des segments sur la version segmentée uniformément du corpus *Q07fr*. Ici, le corpus DEV698 nous a servi à créer des classifieurs adaptés pour la catégorisation de segments textuels, et chacun d'eux a été employé pour évaluer les segments de ce corpus. Ces taux représentent donc les taux de sélection des 42 listes de segments. On a rappelé à la figure 5.1 les graphiques correspondants obtenus au cours de nos expériences de filtrage à l'échelle des documents (graphiques 5.1c et 5.1d). Dans tous les cas, les taux de sélection sont exprimés en fonction du triplet de paramètres de sélection utilisé pour définir les classifieurs : $\langle f, m, c \rangle$.

Constat 1 Globalement, on voit que les taux de sélection obtenus suivant la méthode d'estimation *normale* (5.1a) des distributions d'OOV et PPX sont plus élevés que les taux correspondants obtenus selon la méthode *restreinte* (5.1b). Par exemple, on passe brutalement de 5% de différence à $c = 0$, à 15% à $c = 0.5$, puis 20% dans l'intervalle de valeurs restant $c = [1, 3]$ avec la fonction de sélection combinée op (courbe bleue). Avec l'application de la méthode R et la fonction de sélection combinée op (courbe bleue), la progression des taux de sélection en fonction de c est quasiment linéaire. La progression des taux de sélection relatifs aux fonctions de sélection non combinées o et p suivent cette tendance dans une moindre mesure : les différences de sélection entre o/p d'un côté et op de l'autre sont d'environ 10%.

Constat 2 Pour les deux méthodes N et R, les taux de sélection relatifs à o et p convergent vers le même point : 80% dans le cas de R (5.1b) et 90% dans le cas de N (5.1a). Dans le cas de l'application de la méthode normale, les courbes correspondant aux fonctions o et p sont très proches et s'entrelacent alors que dans le cas de l'application de la méthode restreinte, elles sont parallèles. Ceci suggère qu'avec la méthode restreinte le paramètre OOV est moins discriminant que PPX pour la sélection des documents à l'échelle des segments, ce qui est l'inverse de ce que nous avions constaté avec les expériences sur les documents complets (voir le graphique 5.1d), qui suggéraient que le paramètre OOV est plus discriminant que PPX à l'échelle globale des documents. Par contre avec l'application de la méthode normale, ces mêmes courbes (5.1a) s'entrelacent quelle que soit la valeur de c. Ceci suggère qu'ici aucun des 2 paramètres OOV et PPX n'est plus discriminant que l'autre. Cela n'était vrai que pour les valeurs les plus élevées de c dans nos expériences initiales (voir le graphique 5.1c).

Constat 3 Si on s'intéresse globalement aux différences de taux de sélection de la méthode EPID aux 2 échelles étudiées (comparaisons des graphiques 5.1a-5.1c et 5.1b-5.1d), on constate que le comportement de la méthode est proche dans les deux cas. En effet, les taux de sélection sont plus élevées avec l'application de la méthode N qu'avec la méthode R et toujours plus bas avec la fonction de sélection op qu'avec les fonctions o ou p. Le pouvoir de discrimination de op sur o/p est plus marqué dans le cas de la méthode R, et plus encore à l'échelle des segments (DEV698 versus DEV509). Cependant, l'écart entre les taux correspondants, tous paramètres confondus (f, m ou c), n'est que de 5%. Ceci est plus particulièrement vrai aux extrémités des courbes (intervalles [0,0.5] et [2.5,3] selon c) qu'au centre (intervalle [1,2]).

Récapitulatif et conclusion En conclusion, il semble que la méthode EPID se comporte de façon similaire à l'échelle des segments et des documents (même évolution des taux de sélection dans les deux cas).

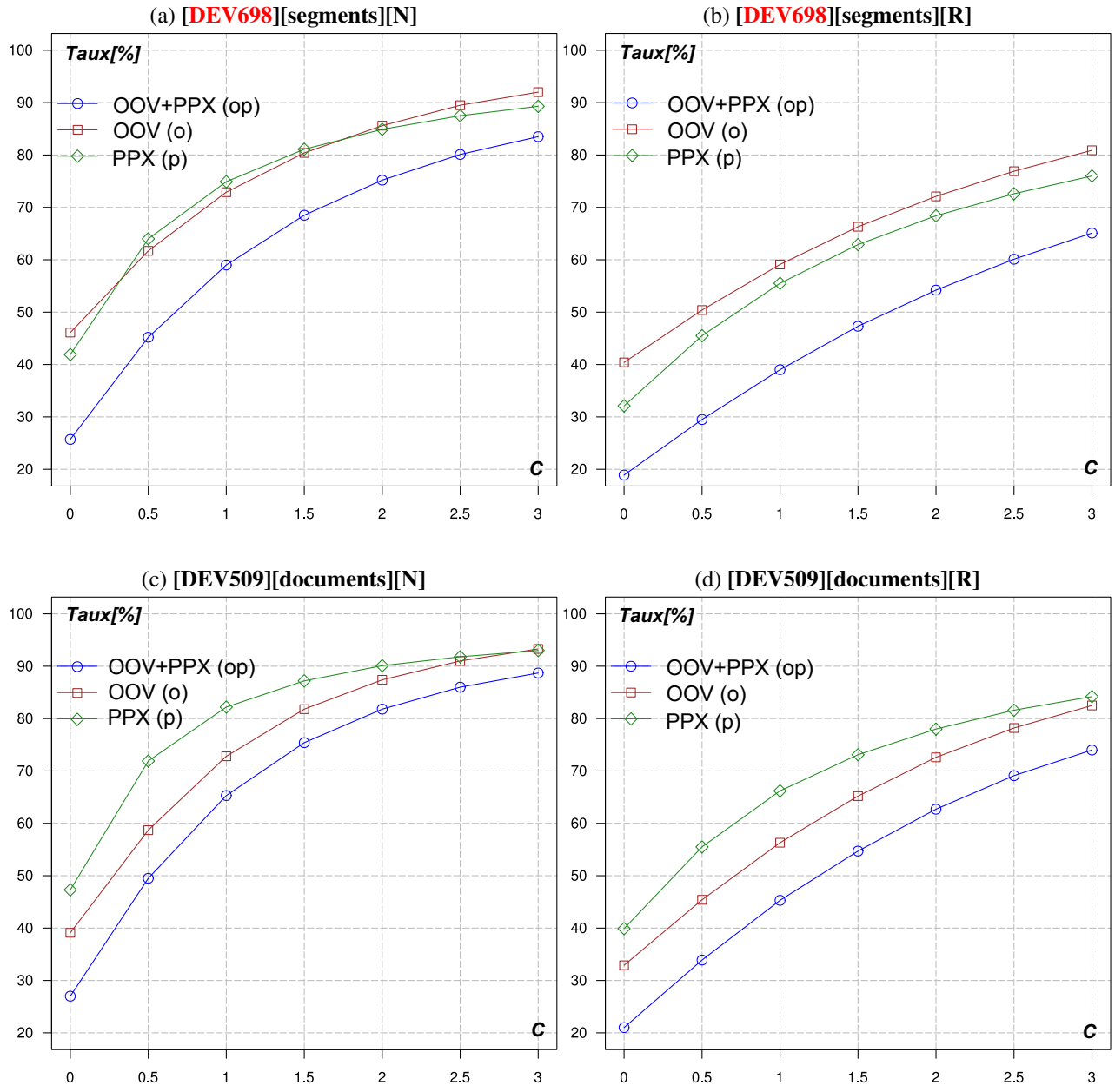


FIGURE 5.1 – Taux de sélection de documents (en pourcents) obtenus à l'application de la méthode EPID sur le corpus d'évaluation Quaero *Q07fr* de 499 734 pages web suivant les corpus de pages segmentées **DEV698** (haut) et non-segmentées **DEV509** (bas), en fonction du triplet de paramètres EPID $\langle f, m, c \rangle$. f : fonction de sélection (o, p ou op). m : stratégie d'estimation des distributions OOV et PPX (**[N]** ou **[R]**). c : constante de variation ($c = \{0, 1, 1.5, 2, 2.5, 3\}$).

Cependant, le changement d'échelle inverse le pouvoir de sélection des paramètres OOV et PPX. En effet, à l'échelle des segments, PPX est plus discrimant qu'OOV (et de façon plus prononcée suivant la méthode R).

Pour expliquer ce changement, une première hypothèse est que l'impact des modifications textuelles induites par Lynx au cours de la pré-segmentation est plus fort sur PPX que sur OOV. En retour, ceci rend l'évaluation de la qualité d'un document plus difficile pour le calcul de PPX que celui d'OOV. En effet, nous avons pu constater au cours d'analyses préliminaires à l'échelle des documents, que des modifications, même légères, de ponctuation, d'abbréviation, de casse, ou l'ajout de certains caractères au texte des documents tels que ceux ajoutés par Lynx au contenu textuel des pages web lors du pré-traitement précédant la segmentation, affectent de manière significative le calcul des scores de PPX.

Une hypothèse complémentaire consiste à penser qu'à l'échelle des segments, le contenu informationnel restreint du segment pénalise le calcul de PPX et favorise celui d'OOV. En effet, à cette échelle le ratio de mots par document est nettement supérieur au ratio équivalent par phrase et, par conséquent, l'effet de lissage statistique opéré par le nombre de mots présents dans un segment sur le calcul d'OOV est plus fort à cette échelle que celui des phrases sur le calcul de PPX.

5.3 Évaluation

Cette section s'articule autour de 3 sections principales. À la section 5.3.1, nous présentons les différentes conditions d'expérimentations utilisées pour évaluer la méthode EPID appliquée à des segments. À la section 5.3.2, nous présentons les résultats d'évaluation correspondant et les comparons aux résultats équivalents obtenus par le système à l'échelle des documents. À la section 5.3.3, nous étudions spécifiquement les résultats QR obtenus par RITEL à l'échelle des segments.

Nous reprenons dans cette section les mêmes corpus d'évaluation que ceux utilisés à travers les chapitres précédents. L'index de documents considéré dans les expériences de filtrage à l'échelle des segments est celui utilisé en condition **ctrl** dans les expériences de pré-segmentation présentées au chapitre 4 (voir section 4.4.1 pour la présentation de cette condition).

5.3.1 Conditions expérimentales

Nous avons testé 43 conditions expérimentales réparties en 2 catégories : *référence* et *filtrage*. Dans le premier cas, qui correspond à la condition **ctrl** mentionnée précédemment, RITEL n'utilise pas le filtrage des documents. Dans le second cas, qui regroupe les 42 conditions liées aux listes, RITEL utilise l'une des 42 listes de filtrages produites par application de la méthodes EPID sur le corpus *Q07fr*, à l'échelle des segments. Dans ce cas, une seule liste de segments est utilisée au cours des recherches pour filtrer la sélection des documents pertinents réalisée par RITEL, indépandemment des classes de questions. On teste alors le potentiel de chaque liste de filtrage en contexte d'évaluation. Par exemple, en condition **op3n**, la liste utilisée par RITEL pour le filtrage est la liste de segments définie par le triplet de paramètres de sélection $\langle f, m, c\rangle = \langle oov + ppx, N, 3.0\rangle$.

Nous ne reprenons pas ici la présentation des conditions relatives aux méthodes d'appariements **Ma1** et **Ma2** du chapitre 3. En effet, leur application à l'échelle des segments aboutissent à des résultats similaires à ceux obtenus à l'échelle des documents, et confirme que ces méthodes ne permettent pas l'obtention d'appariements classe-listes suffisamment génériques pour être utilisés convenablement en contexte d'évaluation.

5.3.2 Résultats

Dans cette section, nous évaluons tout d'abord l'impact du filtrage opéré par la méthode EPID à l'échelle des segments sur les performances QR de RITEL présentés en section 5.3.2.1. Nous comparons ensuite les résultats de filtrage correspondant obtenus par le système à l'échelle des documents en section 5.3.2.2.

5.3.2.1 Impact du filtrage à l'échelle des segments

Les résultats discutés ici concernent les tableaux 5.2a (ou tableau de référence), 5.2b (ou tableau de filtrage) et 5.2c (ou tableau des gains).

Si on s'intéresse aux performances obtenues par RITEL en général (ligne **global**), on voit que les gains sont positifs en terme de précision (**P@1** : **+2.6**), non significatifs en terme de **MRR** (+0.1) et négatifs en terme de rappel (**P@10** : -3.6). Le gain négatif en rappel semble suggérer que la pré-segmentation des documents nuit globalement au filtrage. Autrement dit, à l'échelle des segments, le filtrage n'est pas approprié si on cherche à tirer le meilleur parti des capacités de rappel du système. En s'intéressant aux performances de RITEL obtenues par classe de questions en terme de précision et de MRR, il est clair que le couplage de la segmentation et du filtrage est gagnant-gagnant pour les classes principales. Notamment, les gains apportés par le filtrage sur la segmentation suivant le MRR se situent dans un intervalle allant de +1.8 (classe **nbr**) à +8.7 (classe **pers**). On nomme cet intervalle de gains **g-I**$_{\text{MRR}}$ et la moyenne associée (+4.0 points environ) **mg-I**$_{\text{MRR}}$.

Ceci signifie que la détérioration des performances causée par le filtrage sur les capacités de rappel du système n'est pas suffisamment importante pour annihiler les effets positifs du couple segmentation/filtrage. Les gains perçus en terme de précision compensent positivement la détérioration des performances en rappel. Il suffit de parcourir rapidement les colonnes de précision et de rappel dans le tableau des gains pour s'en convaincre : les améliorations en précision sont beaucoup plus importantes que les pertes en rappel. Notamment, le pic de gains atteint pour la classe **pers** (**+14.5**) compense presque à lui seul le pic de perte atteint en rappel pour la classe **qpers** (-15.6).

On notera pour la suite que les pics de gains perçus en précision à l'aide du filtrage se produisent sur les classes de questions pour lesquelles RITEL est le plus en difficulté en segmentation seule (voir les tableaux de référence et de gains), à savoir : les classes **pers**, **qpers** et **subsnn**.

Les intervalles de gains de performances (ainsi que les classes de question qui les bornent) et moyennes associées aux 3 mesures d'évaluation des performances de RITEL sont synthétisées ici :

- **g-I**$_{\text{P@1}}$ $= [nbr, pers]$ $= [+0.0, +14.5]$ où **mg-I**$_{\text{P@1}}$ $= +7$
- **g-I**$_{\text{MRR}}$ $= [nbr, pers]$ $= [+1.8, +8.7]$ où **mg-I**$_{\text{MRR}}$ $= +4$
- **g-I**$_{\text{P@10}}$ $= [qpers, nbr]$ $= [-15.6, +2.2]$ où **mg-I**$_{\text{P@10}}$ $= -2$

5.3.2.2 Comparaison de performances QR aux différentes échelles de filtrage

Dans cette section, nous comparons les performances de RITEL obtenus en contexte de filtrage entre l'application de la méthode aux documents complets et aux segments de documents.

Préambule : comparer ce qui est comparable On tient ici à préciser que les résultats mentionnés dans le tableau 5.2 ne sont pas tous strictement comparables.

En effet, les conditions expérimentales ne pas exactement équivalentes. Dans les tableaux portant sur l'application de la méthode EPID sur les segments (5.2a, 5.2b et 5.2c), les résultats portent sur l'utilisation d'une seule liste de filtrage quelle que soit la catégorie de la question. On utilise la liste donnant les meilleurs résultats sur le corpus de développement, toutes catégories de questions confondues. Les tableaux qui portent sur l'application de la méthode EPID sur les documents complets (5.2d, 5.2e et 5.2f) contiennent le meilleur résultat obtenu selon les conditions Ma1 ou Ma2, liées aux catégories de questions.

La normalisation utilisée pour évaluer la méthode EPID appliquée aux documents complets est légèrement différente de celle utilisée lors de l'application de la méthode aux documents segmentés. Ces différences concernent majoritairement les ajouts textuels faits par Lynx lors de la pré-segmentation des pages web.

	ctrl				ctrl+fltr			g-ctrl+fltr		
Qcat	**P@1**	**MRR**	**P@10**	#q	**P@1**	**MRR**	**P@10**	**P@1**	**MRR**	**P@10**
global	31.7	41.6	**61.5**	309	**34.3**	41.7	57.9	**+2.6**	+0.1	-3.6
loc	50.0	58.6	**72.7**	66	**56.1**	61.3	71.2	**+6.1**	+2.7	-1.5
nbr	**35.6**	46.2	68.9	45	**35.6**	48.0	**71.1**	+0.0	+1.8	**+2.2**
pers	27.3	39.7	60.0	55	**41.8**	48.4	**61.8**	**+14.5**	+8.7	**+1.8**
qpers	28.9	42.0	**75.6**	45	**37.8**	45.2	60.0	**+8.9**	+3.2	-15.6
subsnn	7.7	14.6	**28.2**	39	**15.4**	19.4	**28.2**	**+7.7**	+4.8	+0.0
time	37.5	44.7	**62.5**	48	**41.7**	47.6	**62.5**	**+4.2**	+2.9	+0.0

(a) (b) (c)

	bsln				fltr			g-fltr		
Qcat	**P@1**	**MRR**	**P@10**	#q	**P@1**	**MRR**	**P@10**	**P@1**	**MRR**	**P@10**
global	31.7	39.5	53.4	309	**33.0**	40.6	**55.0**	**+1.3**	+1.1	+1.6
loc	50.0	58.5	**75.8**	66	**57.6**	64.5	**75.8**	**+7.6**	+6.0	+0.0
nbr	**35.6**	48.6	**75.6**	45	**35.6**	48.6	**75.6**	+0.0	+0.0	+0.0
pers	**32.7**	40.9	52.7	55	**32.7**	41.3	**54.5**	+0.0	+0.4	**+1.8**
qpers	24.4	34.8	**48.9**	45	**31.1**	37.9	**48.9**	**+6.7**	+3.1	+0.0
subsnn	**5.1**	5.6	7.7	39	2.6	3.9	**10.3**	-2.5	-1.7	**+2.5**
time	**31.2**	37.8	50.0	48	**31.2**	39.8	**60.4**	+0.0	+2.0	**+10.4**

(d) (e) (f)

TABLE 5.2 – (a) Impact du filtrage sur les performances QR de RITEL à l'échelle des segments. **P@1** : précision. **MRR** : moyenne des rangs réciproques. **P@10** : précision au top-10. (b) Impact du filtrage sur les performances QR de RITEL à l'échelle des documents. (c) Table des gains correspondants à (b) relativement à (a).
(d, e et f) Mêmes informations que celles données respectivement en (a), (b) et (c), mais sans pré-segmentation des documents. Les performances (en pourcents) sont données en général (**global**), et par catégorie de question (pour les catégories dominantes du corpus d'entraînement et de test de RITEL **loc**, **nbr**, etc.).

Les lignes **global** du tableau 5.2 nous indiquent qu'en terme de précision (**P@1**), le gain en appliquant le filtrage à l'échelle des segments (**ctrl+fltr**) est 2 fois plus élevé (**+2.6%**) que le gain correspondant à l'échelle des documents (**+1.3%**). De plus, le rappel **global** est plus élevé à l'échelle des segments (57.9%) qu'à celle des documents (**55.0%**). **On peut globalement conclure que le filtrage est plus efficace à l'échelle des segments qu'à l'échelle des documents.**

Par classe de question, on constate qu'à l'échelle des documents, le filtrage influence plus fréquemment le rappel que la précision (résultats en rouge versus résultats en vert au tableau 5.2f). On constate l'effet inverse à l'échelle des segments (tableau 5.2c). Dans les deux cas, le filtrage favorise la précision plus qu'il ne favorise le rappel. Les bénéfices perçus par le système en précision sont en effet généralement plus élevés à l'échelle des segments qu'à l'échelle des documents. **Ces résultats confirment que le filtrage est plus efficace à l'échelle des segments.**

5.3.3 Étude

Dans cette section, nous étudions dans un premier temps à un niveau très général les résultats obtenus par RITEL à l'échelle des segments pour les 43 conditions de test, à l'aide d'analyses multi-variées, en terme de précision et de rappel (section 5.3.3.1). Dans un deuxième temps, nous étudions ces mêmes résultats à un niveau plus spécifique, du point de vue des listes de filtrage et des classes de questions (section 5.3.3.2). Cette seconde étude a pour but de mieux saisir les liens de dépendances qui existent entre les paramètres de la méthode EPID et les performances QR de RITEL à l'échelle des segments. En effet, bien que, globalement, on observe des gains, définir avec exactitude les raisons à l'origine de ces gains est délicat.

5.3.3.1 Dépendances globales entre performances QR et paramètres de filtrage

Les paramètres de filtrage sont les paramètres de sélection qui caractérisent la méthode d'évaluation intrinsèque des documents, c'est-à-dire les paramètres f, m et c [1].

Afin de faciliter l'étude des 43 conditions testées en fonction de ces paramètres, nous avons mené 3 analyses multi-variées séparées (une par paramètre) sous forme de nuages de points. L'influence de chaque paramètre sur les performances globales de RITEL est illustrée à travers les graphiques (5.2b, 5.2c et 5.2d) de la figure 5.2.

Les analyses multi-variées que nous avons faites sont des **analyses bi-variées**, c'est-à-dire que les performances de RITEL pour chaque condition sont projetées dans un espace à 2 dimensions dans un référentiel de coordonnées (x,y) centrées au point d'origine (24,45), en référence aux moins bons résultats obtenus par le système. Dans ces analyses, la projection des performances se fait en fonction des résultats QR obtenus par RITEL en terme de précision (**P@1** ou top-1) et rappel (**P@10** ou top-10), portés respectivement par les axes des abscisses et des ordonnées.

Préambule Le graphique 5.2a présente les performances de RITEL. Il rend compte de l'amplitude des performances obtenues par le système.

On voit dans ce graphique que la majeure partie des conditions produisent des résultats semblables, compris dans un intervalle de précision 30-34% et un intervalle de rappel 55-60%. Ces intervalles sont plus ou moins bornés par les seuils minimum et maximum de performances observées en contexte de filtrage à l'échelle des documents et dans nos expériences de pré-segmentation (cf. tableaux 3.5 à la page 70, et 4.2 à la page 95).

Les 12 meilleures conditions testées (en orange) et la condition de référence (**ctrl**) sont indiquées par des chiffres précédés du symbole #. Ces derniers donne le rang correspondant à chaque condition à travers le classement complet des 43 conditions test (ou TOP_{43}). Par exemple, on peut voir que la condition **ctrl** se place $25^{\text{ième}}$ dans le TOP_{43}.

1. f : fonction de sélection (oov, ppx ou oov+ppx) ; m : méthode d'estimation des distributions OOV et PPX (*normale* ou **N** et *restreinte* ou **R**) ; et c : constante de variation ($c = \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$).

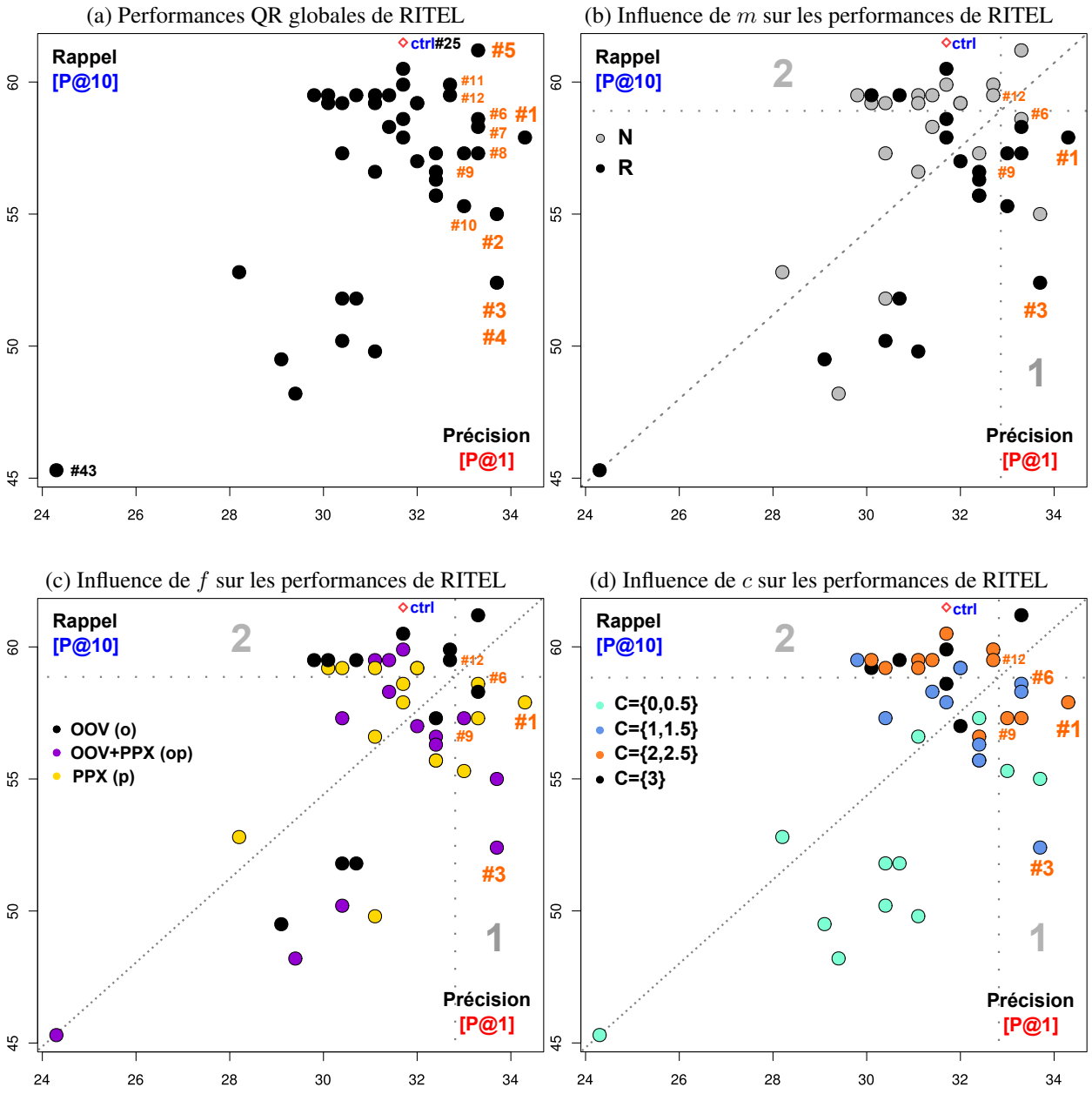


FIGURE 5.2 – Analyse bi-variées (**P@1**/**P@10**) des performances QR globale de RITEL pour les 43 conditions de test. En **orange** ou **Zone 2** : les 10 meilleures résultats en terme de précision. **Zone 1** : les meilleurs résultats en terme de rappel. ◇ : résultats de référence (c.-à-d. **ctrl**). (a) Analyses indépendantes des 3 paramètres de sélection : f, m et c (performances QR gobales de RITEL). m : méthodes *normale* (**N**) ou *restreinte* (**R**) d'estimation des distributions OOV et PPX à partir du corpus **DEV698**. f : fonction de sélection (o, p ou encore op). c : constante de variation ($c = \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$).

Influence du paramètre de sélection m D'après le graphique 5.2b, on peut voir que **les conditions testées se séparent en 2 groupes**, situés de chaque côté de la ligne marquée en pointillé, côté précision (**P@1**) et côté rappel (**P@10**) respectivement.

Côté rappel, on trouve majoritairement les conditions définies à l'aide de la méthode normale (**N**). On voit que la condition (**ctrl**) se situe de ce côté et permet à RITEL d'obtenir les meilleures performances en rappel (aucune autre condition n'est placée plus haut que (**ctrl**) dans le graphique).

Côté précision, on trouve majoritairement les conditions définies selon l'autre méthode (**R**). La condition qui permet à RITEL d'obtenir les meilleures performances (**#1**) est de ce côté. Cette condition est proche de la condition de référence, suggérant que le filtrage à l'échelle des segments permet d'augmenter la précision du système sans pour autant dégrader fortement son rappel.

D'après ces résultats, on constate de façon générale, qu'à l'échelle des segments, le paramètre de sélection m rend le système plus précis quand il prend la valeur R et meilleur en rappel pour N. Cela fait écho à certaines conclusions apportées par nos expérience initiales de filtrage : l'emploi de listes *normales* entraîne un filtrage plus souple qui laisse passer plus de documents et favorise le rappel du système.

Influence du type de sélection f Ici et dans les paragraphes qui suivent, nous nous appuierons sur les distributions et les comptes relatifs aux différents paramètres de sélection, donnés au tableau 5.3.

D'après le graphique 5.2c, on peut voir que **les conditions testées se séparent en 3 groupes**. Un groupe est situé côté précision (**P@1**), à droite de la ligne verticale en pointillé (rectangle 1). On y fera référence comme étant la **zone 1**. Les conditions de cette zone sont celles qui rendent RITEL le plus précis. De manière similaire, un autre groupe se situe côté rappel (**P@10**), à l'intérieur de la **zone 2**. Les conditions de cette zone sont celles qui mènent RITEL aux meilleurs taux de rappel. Le dernier groupe est situé à l'exterieur des **zones 1** et **2**.

Côté précision (**zone 1**), visuellement on remarque que les sélections de type op (en violet) et p (en jaune) sont prédominantes sur les sélections de type o (en noir). Ceci est cohérent avec le fait que RITEL semble tirer un meilleur parti en terme de précision en filtrant les documents à l'aide de listes de type op. En effet, dans les 3 cas où il utilise une liste de ce type, les résultats QR qui correspondent sont bien situés dans le TOP_{43}, rangs (**#2**, **#3** et **#9**). Seule la condition en tête de classement (**#1**), liée à p, se place devant. Ici on voit que plus on se rapproche de la zone de rappel **zone 2**, plus la prédominance des listes liées aux fonctions de type o et p est forte. **Dans ce contexte, le meilleur compromis P@1/P@10 passe par l'emploi de listes de type p**.

Côté rappel (**zone 2**), on voit que les sélections de type o et p sont prédominantes par rapport à celles de type op. Ici, RITEL obtient les meilleurs résultats en utilisant des listes de type o, comme le prouve la présence des 3 points situés à l'extrémité droite de la zone (c.-à-d. à la jonction avec la **zone 1**). Par ailleurs, on constate d'après le tableau 5.3 que la moitié (**7**) des 14 conditions caractérisées par une fonction de sélection de type o se trouvent dans cette zone. **Il semble donc qu'à l'échelle des segments, le filtrage permet d'extraire un plus grand nombre de réponses** (après la condition **ctrl**), **avec des listes de type o et ensuite p**.

En considérant les distributions z1 et z2 en fonction de f indiquées dans le tableau 5.3, on voit que les taux de couverture des conditions étudiées dans les 2 zones analysées plus haut sont équitablement réparties entre o (9 conditions) et p (8 conditions) et représentent 17/23 conditions se trouvant réunies dans ces 2 zones, **soit 80% des conditions testées qui participent aux meilleures performances de RITEL**.

Enfin, si on s'intéresse aux performances obtenues par RITEL dans les conditions restantes (carré extérieur aux zones **zone 1** et **zone 2**), on voit que la majorité des conditions correspondantes s'appuient sur des fonctions de sélection de type op (8 conditions). Ici, RITEL obtient les moins bonnes performances.

analyse	AnB-m			AnB-f			AnB-c		
paramètre	**z1**	**z2**	total	**z1**	**z2**	total	**z1**	**z2**	total
N	3	**11**	14						
R	**6**	3	9						
total	9	14	**23**						
o				2	**7**	**9**			
p				**4**	4	**8**			
op				3	3	6			
total				9	14	23			
$\mathbf{c_0}$							2	0	2
$\mathbf{c_1}$							**3**	2	5
$\mathbf{c_2}$							3	**8**	**11**
$\mathbf{c_3}$							1	4	5
total							9	14	23

TABLE 5.3 – Distributions (**z1** et **z2** en correspondance respectivement avec les zones Zone 1 et Zone 2 marquées sur les graphes de la figure 5.2) et comptes (total) des meilleures conditions en terme de précision (**P@1**) et rappel (**P@10**), pour les paramètres de sélection m, f et c et chaque analyse bi-variée (**P@1**/**P@10**) des performances globales de RITEL. Les analyses correspondant aux paramètres de sélection sont respectivement : AnB-m, AnB-f et AnB-c. m : méthodes d'estimation des distributions OOV et PPX (**N** : normale ou **R** : restreinte), à partir du corpus **DEV698**. f : fonction de sélection (**o**, **p** et **op**). c : constante de variation telle que $\mathbf{c_0} : c = \{0, 0.5\}$, $\mathbf{c_1} : c = \{1, 1.5\}$, $\mathbf{c_2} : c = \{2, 2.5\}$ et $\mathbf{c_3} : c = 3$.

Influence de la constante de variation c D'après le graphique 5.2d, on peut voir que **les conditions testées se séparent en 3 groupes**. Ces groupes sont les mêmes que ceux que nous avons définis précédemment (zone 1, zone 2 et carré extérieur à ces 2 zones).

On voit que la majeure partie des conditions avec une constante de variation c inférieure à 2 (les points en bleu clair et bleu ciel sur le graphique) sont dans le carré extérieur zone 1 et zone 2. **Ceci suggère que l'utilisation de listes peu permissives ($c < 2$) n'est globalement pas satisfaisant d'un point de vue QR.**

En revanche, les conditions qui reflètent l'utilisation de listes caractérisées par une constante de variation plus élévées ($c \geq 2$) se trouvent assez logiquement **côté rappel**, zones 2 [2]. On voit au tableau 5.3 que c'est le cas pour 12 des 14 conditions couvertes dans cette zone (réunion c_2 et c_3). Ces résultats confirment le constat fait au cours de nos expériences de filtrage à l'échelle des documents selon lequel, les meilleures listes en général sont celles avec une forte constante de sélection ($c > 2$).

Côté précision, le constat est plus nuancé. En effet, mis à part le point noir à l'intersection des zones 1 et zones 2, les conditions qui permettent à RITEL d'obtenir les meilleurs résultats à l'échelle des segments (#1 à #10) sont telles que $c < 1$ (2/9 conditions au tableau 5.3, #2 et #3) et $c < 2$ (**3**/9 conditions, #1, #8 et #9). Ceci concerne donc plus de la moitié des conditions (5/9 conditions). **En conséquence, à l'échelle des segments**, si on veut une grande précision (**P@1**), il semble plus intéressant d'employer des listes de filtrage caractérisées par une faible constante de variation ($c < 2$). Toutefois, **le meilleur compromis précision/rappel (P@1/P@10) possible reste d'utiliser des listes caractérisées par une constante de variation intermédiaire** ($1.5 \leq c \leq 2.5$) ; comme le suggère le petit groupe de conditions limitrophes à la zones 1 et incluant #1, #8, #9, #11 et #12.

2. Les taux de sélection des listes définies avec une valeur de c élevée sont également élevés (cf. figure 5.1 pour rappel). Donc, de telles listes contiennent plus de documents que des listes définies avec une constante de sélection plus faible.

Influence croisée des paramètres m, f et c Après avoir étudié séparément l'influence de chaque paramètre de sélection sur les performances QR, nous étudions leur influence mutuelle en lien avec les taux de sélection des listes de filtrage (cf. figure 5.1 à la page 107).

Ici, les analyses consistent à mettre en relation les informations (N, R, o, p, op et les différentes valeurs de c) apportées par les analyses bi-variées avec les taux de sélection associés aux listes de filtrage correspondantes.

Visuellement, cela revient : (i) à superposer les nuages de points correspondant à chaque paramètre de sélection : 5.2b (m), 5.2c (f) et 5.2d (c), puis (ii) à recomposer l'information complète relative à chaque point via le code couleurs qui lui est associé à chaque graphique, pour finalement (iii) identifier le taux de sélection correspondant donnés aux graphiques 5.1a ou 5.1b.

En recoupant les clusters de conditions définis par les **zones 1** et **2** des graphes 5.2c et 5.2d avec les informations supplémentaires apportées par le graphe 5.2b, **on constate que les conditions qui permettent à RITEL d'obtenir les meilleures performances en terme de rappel (P@10) sont en général caractérisées par un triplet de sélection $T = \langle f, m, c\rangle$ tels que $T = \langle o/p, N, c_2/c_3\rangle$ ($\mathbf{c_2} : c = \{2, 2.5\}$ et $\mathbf{c_3} : c = 3$)**. En d'autres termes, cela signifie que les listes de filtrage qui mènent RITEL vers le meilleur rappel, sont des listes normales (N), avec un facteur de sélection élevé ($c \geq 2$) et une fonction de sélection qui ne combine pas les paramètres OOV et PPX (o/p). D'après le graphique 5.1a, de telles listes présentent des taux de sélection au delà des 80% (p. ex. les listes o2.5n, o3n et p2n). **Ces résultats s'accordent avec les conclusions émises à propos des taux de sélection menant RITEL aux meilleures performances à l'échelle globale des documents**.

De la même manière, on constate que **les conditions menant RITEL aux meilleures performances en terme de précision (P@1), sont en général caractérisées par un triplet de sélection $T = \langle f, m, c\rangle$ tels que $T = \langle o/p, R, c_1/c_2\rangle$ ($\mathbf{c_1} : c = \{1, 1.5\}$)**. En d'autre terme, cela signifie que les listes de filtrage qui rendent en général RITEL précis, sont des listes restreintes (R), avec un facteur de sélection intermédiaire ($1 \leq c \leq 2.5$) et une fonction de sélection qui ne combine pas les paramètres OOV et PPX (o/p). Le graphique 5.1b, nous indique cette fois que ce genre de liste a un taux de sélection compris entre 55% et 70%. **Ces résultats sont en fort désaccord avec les conclusions émises à propos des taux de sélection menant RITEL aux meilleures performances à l'échelle des documents** (voir plus haut).

Pour un rappel optimum, le filtrage des documents n'est pas opératoire. En effet, on constate que la meilleur performance de RITEL en rappel se produit en condition de référence **ctrl**.

Pour une précision optimale, le filtrage des documents n'est pas caduc. En effet, il permet d'apporter un gain de plus d'un point vis-à-vis du fonctionnement habituel du système. Dans ce cas, on voit qu'appliquer des listes avec un taux de sélection très faible (c.-à-d. entre 25% et 55%) peut être la solution. Abstraction faite de la meilleure performance obtenue par RITEL, le genre de listes dont on parle ici est caractérisée par un triplet de sélection $T = \langle f, m, c\rangle$ tel que $T = \langle op, N, c_0\rangle$ ($\mathbf{c_0} : c = \{0, 0.5\}$) ou $T = \langle op, R, c_1\rangle$. On notera que ce genre de combinaison couvre parfois des catégories de listes hors-**zones 1** et **2**.

Le meilleur compromis QR possible P@1/P@10 passe par l'emploi de listes caractérisées par un triplet de sélection $T = \langle f, m, c\rangle$ tel que $T = \langle p, R, c_2\rangle$ (c.-à-d. avec un taux de sélection proche des 70%).

Récapitulatif et conclusion Contrairement au constat réalisé à l'échelle des documents, les listes les plus appropriées pour le filtrage à l'échelle des segments (TOP_{10}) sont moins permissives (majoritairement de type R avec une fonction de sélection f de type op ou p), leur taux de sélection oscillant entre 25% et 70% de documents. Ceci suggère qu'à l'échelle des segments, le filtrage est plus efficace qu'à l'échelle des documents complets et va dans le sens de l'hypothèse **H0**.

Suite à l'étude que nous avons menée, nous donnons au tableau 5.4 les combinaisons de paramètres de sélection qui semblent les plus à même d'aider la tâche QR avec un filtrage à l'échelle des segments en fonction du type de performance recherchée :

- performance globale moyenne en terme de précision (**P@1**) ou de rappel (**P@10**) ;
- performance globale optimale en terme de précision (**P@1**-b) ou de rappel (**P@10**-b) ;
- meilleur compromis en terme de précision et de rappel (**P@1/P@10**).

L'ensemble de ces combinaisons couvrent 1/3 des 43 conditions expérimentales testées. Nous rappellons que si on cherche à avoir le plus de réponses possibles en **top-10** (**P@10**) indépendamment de la précision, il est préférable de désactiver le filtrage des documents qui ne permet jamais de meilleurs résultats que la condition de référence (**ctrl**).

Contexte : segmentation		**Filtrage**								**Sélection**			
Performance recherchée		N	R	**o**	**p**	**op**	c_0	c_1	c_2	c_3	T	c	**Taux** [%]	**c-Taux**
	P@1-b	+	+			+	+	+			$\langle op, N, c_0 \rangle \vert \langle op, R, c_1 \rangle$	$c < 2$	25 à 55	1
Précision	**P@1**		+	+	+			+	+		$\langle o/p, R, c_1/c_2 \rangle$	$1 \leq c \leq 2.5$	55 à 70	2
	P@1/P@10		+		+				+		$\langle p, R, c_2 \rangle$	$c = \{2, 2.5\}$	c. 70	3
Rappel	**P@10**	+		+	+				+	+	$\langle o/p, N, c_2/c_3 \rangle$	$c \geq 2$	> 80	4
	P@10-b										**ctrl**		100	5

TABLE 5.4 – Paramétrage du module de filtrage en fonction des performances QR globales recherchées : optimales (**P@1**-b, **P@10**-b), moyennes (**P@1**, **P@10**) et mixte (**P@1/P@10**), pour la sélection de documents pertinents segmentés uniformément (c. 8 × 20 lignes par document). $c_0 : c = \{0, 0.5\}$, $c_1 : c = \{1, 1.5\}$, $c_2 : c = \{2, 2.5\}$ et $c_3 : c = \{3\}$.

5.3.3.2 Dépendances fines entre performances QR et paramètres de filtrage

Dans cette section, les points de vues adoptés pour l'étude des performances de RITEL sont ceux des listes de filtrage (et taux de sélection associés) ainsi que des classes de questions. Les analyses correspondantes se veulent complémentaires de celles présentées à la section précédente et visent à poursuivre l'introspection de la méthode EPID initiée au chapitre 3. Cette introspection est très poussée, plus peut-être que ne le permet la significativité des résultats, étant donné le nombre de questions par classe de questions lors de l'évaluation. Cependant, il nous a paru utile de la présenter, malgré sa complexité, car la méthodologie qui la sous-tend, qui a pour but de donner une **explication** des résultats, peut être appliquée dans d'autres situations semblables, où des systèmes complexes produisent des résultats complexes impliquant de nombreux paramètres.

Les résultats discutés ici réfèrent majoritairement au tableau 5.5. Il contient des informations relatives aux listes de filtrage (comme les taux de sélection correspondant à chaque liste) et aux gains correspondant perçus par RITEL en filtrage à l'échelle des segments, en fonction des classes prédominantes de questions dans le corpus d'évaluation. Les informations à propos des gains ont été reprises du tableau 5.2 (donné à la page 107), celles relatives aux triplets de sélection et aux classes de taux de sélection (c-Taux) ont été reprises du tableau 5.4.

Pour expliciter le lien avec les analyses bi-variées menées précédemment, nous avons éclaté le nom des listes en fonction des paramètres de filtrage f, m et c correspondant à chaque liste. Le nombre de questions par classe (#q) n'est pas reporté dans cette table, vu son incidence a priori nulle sur les résultats de RITEL [3].

3. On peut se fier à la variation des résultats de référence (**ctrl**) et de filtrage en fonction de #q au tableau 5.2 pour s'en assurer.

	Listes et taux							g-ctrl+fltr	
Qcat	listes	f	c	m	T	Taux	c-Taux	**P@1**	**P@10**
global	**p2.5r**	p	2.5	R	$\langle p, R, c_2 \rangle$	72.6	3	**+2.6**	-3.6
loc	**p2.5r**	p	2.5	R	$\langle p, R, c_2 \rangle$	**72.6**	3	**+6.1**	-1.5
nbr	**o3n**	o	3	N	$\langle o, N, c_3 \rangle$	92.0	4	+0.0	**+2.2**
pers	**op1r**	op	1	R	$\langle op, R, c_1 \rangle$	39.0	1	**+14.5**	**+1.8**
qpers	**o0r**	o	0	R	$\langle o, R, c_0 \rangle$	40.4	1	**+8.9**	-15.6
subsnn	**o2.5n**	o	2.5	N	$\langle o, N, c_2 \rangle$	89.5	4	**+7.7**	+0.0
time	**op3r**	op	3	R	$\langle op, R, c_3 \rangle$	**65.1**	2	**+4.2**	+0.0

TABLE 5.5 – Performances de RITEL par classe de question, du point de vue des listes de filtrage. Les taux (Taux) comme les gains (**g-ctrl+fltr**) sont présentés en pourcents (%). Pour le reste des informations, successivement de gauche à droite : classe de questions (**Qcat**), meilleures listes pour le filtrage en fonction des 6 classes prédominantes du corpus de questions d'évaluation Quaero, paramètres de sélection utilisés à la création des listes (f, c, m), triplet de sélection correspondant (T), classes de taux de sélection associé à chaque liste (c-Taux). Gains en précision (**P@1**) et en rappel (**P@10**) perçus en filtrage sur le fonctionnement habituel de RITEL en contexte de segmentation (**ctrl**).

Analyse globale **Les six listes les plus appropriées parmi les 42 testées en appariements fixes sont : p2.5r, o3n, op1r, o0r, o2.5r et op3r** (chacune associée à la condition du même nom).

On constate au tableau 5.5 que les taux de sélection de ces listes (colonne Taux) sont compris entre 39.0% pour **op1r** (**pers**) et 92.0% pour **o3n** (**nbr**), que 2/3 des listes sont de type restreintes (R), que les valeurs de c sont soit très élevées ($c > 2$ pour 2/3 des listes), soit très basses ($c < 1.5$ pour le 1/3 restant) et que sur les 6 listes concernées, une est de type p (**loc**), deux sont de types op (**pers** et **time**), les trois restantes étant de type o. Les conditions de filtrage sont donc très diverses.

Les listes de types R influencent en général à la fois la précision et le rappel. Celles de types N n'influencent que l'une ou l'autre des mesures. Les gains (en précision) et les pertes majeurs (en rappel) sont perçus à l'utilisation des listes de type R, ceux perçus à l'utilisation de leurs homologues de type N étant globalement plus légers. En revanche, ces dernières ne pénalisent jamais le système (au pire elles ne lui apportent aucun gain). **Ici, les résultats vont dans le sens inverse de ceux observés à l'échelle des documents, à savoir que les listes les plus restrictives mènent aux meilleures performances**. Ici, les listes de type R dominent (2/3).

Deux tiers des listes sont couvertes par les patrons de sélection définis au tableau 5.4 [4]. Ainsi, ces patrons génériques, définis à partir des analyses bi-variées des performances QR globales de RITEL, sont représentatifs des performances du système par classe. Ceci veut dire qu'à l'échelle des segments le comportement du filtrage observé par classe est similaire à celui observé en général (**global**). Les critères d'optimalité définis suite à ces analyses [5] semblent bien s'accorder avec l'influence concrète du type de listes sur les performances du système. Par exemple, le triplet qui définit la liste **op1r** (**pers**) $\langle op, R, c_1 \rangle$ entre dans la catégorie des patrons $\langle op, N, c_0 \rangle | \langle op, R, c_1 \rangle$. Comme indiqué au tableau 5.4, ces derniers visent l'optimalité des performances en précision (**P@1**-b), et on voit que le gain apporté par cette liste selon **P@1** est de **+14.5**. Il en va de même pour la liste **o3n** (**nbr**) associée à $\langle o, N, c_3 \rangle$ et couverte par $\langle o/p, N, c_2/c_3 \rangle$, en rappel (**+2.2**). On voit qu'une adaptation dans un cas et l'ajout d'un patron dans l'autre cas suffisent pour capturer le 1/3 des listes restantes,

4. Les triplets correspondant à ces listes au tableau 5.5, couverts par les patrons indiqués au tableau 5.4, sont marqués en orange. Ces triplets sont sous spécifiques comparés aux patrons du tableau 5.4, car ils définissent une unique liste.

5. Recherche de précision (**P@1**) ou rappel (**P@10**) en général, recherche de précision (**P@1**-b) ou rappel (**P@10**-b) optimum, et recherche du meilleur compromis entre précision et rappel (**P@1**/**P@10**).

c'est-à-dire les listes non couvertes par les patrons génériques spécifiés au tableau 5.5 qui correspondent aux triplets de sélection marqués en noir dans ce tableau. À partir de là, **un appariement classe-liste spécifié par le biais des patrons devient envisageable.**

Analyse spécifique **On peut diviser les listes étudiées ici en trois groupes selon les taux de sélection des listes associées :**

- les listes associées à **pers** et **qpers** ($c\_Taux = 1$: taux compris entre 25% et 55%) ;
- les listes associées à **loc** et **time** ($c\_Taux = \{2, 3\}$: taux compris entre 55% et 70%) ;
- les listes associées à **nbr** et **subsnn** ($c\_Taux = 4$: taux compris entre 80% et 90%).

Groupe 1 $c\_Taux = 1$: op1r/pers et o0r/qpers. Ces listes sont les moins permissives. Elles sont toutes les deux de type R et ont une très faible constante de sélection ($c < 1.5$). Ce sont les listes qui mènent aux pics de performance les plus forts en précision (**pers.P@1** : **+14.5** et **qpers.P@1** : **+8.9**) et à la fois à l'un des rares gains et à la chute de performances la plus importante en rappel (**pers.P@10** : **+1.8** et **qpers.P@10** : -15.6). Comme nous l'avions fait remarquer plus tôt, les classes de ce groupe tirent le plus avantage du filtrage. Sans lui, elles font parties toutes les deux des trois classes de questions (avec **subsnn**) ayant la précision la moins élevée (cf. tableau 5.2). Notamment, avec le filtrage, la classe **pers** passe de la 5ième position à la 2ième dans le TOP_6 des classes de question en précision (ou $\text{TOP}_{6P@1}$). **D'après ces résultats, nous pouvons dire que par classe de question, un couplage $m = R$ et $c \leq 1$ joue en faveur d'une précision optimale, éventuellement au détriment du rappel.**

Il est intéressant de noter que les deux classes de questions concernées ici sont proches l'une de l'autre d'un point de vue QR. En effet, les questions de type **pers** ont pour focus une personne (p. ex. une personnalité politique, un écrivain, un sportif ou un personnage de fiction) et attendent en réponse un nom propre. Celles de type **qpers** s'appuient sur les noms propres afin de répondre à des questions à propos de la personne en focus (p. ex. *Qui est Mike Jagger ? Qu'a fait Carl Marx dans sa vie ? Qui a succédé à Jean-Paul II ?*). Sous cet angle, il n'est pas surprenant que des listes similaires (paramètres de sélection et taux) permettent de trouver des réponses basées sur des mécanismes de recherche et d'extraction d'information similaires (ici, la recherche et l'extraction d'entités nommées relatives aux noms de personnes). La préférence de précision, au détriment du rappel, peut refléter la nécessité, lorsque l'on cherche des noms propres, d'écarter les documents contenant beaucoup de mots hors-vocabulaire ; ces documents contiennent peut-être beaucoup de noms propres inconnus qui perturbent l'extraction de réponses, comme par exemple des listes de noms.

Au-delà de ces considérations, il devient moins évident de savoir pourquoi ces listes ne sont pas de type p : les conditions de ce type sont généralement très mal placées pour cette classe et, par exemple, la condition **p1r** est 38ième/42 pour la classe **qpers**. Il est également difficile de savoir ce que les documents qui se cachent derrière ces listes ont en commun. Y a-t-il ici un lien entre le fait que pour les classes **pers** et **qpers**, il est primordial que les mots ne soit pas abîmés sous peine de perdre des noms propres, et le fait de choisir des listes basées à chaque fois sur le ratio OOV pour ces 2 classes ? Pour répondre à ces questions, il faut aller voir à la source, dans les documents de ces listes, mais aussi s'intéresser aux entitées nommées qu'on y trouve. Notamment, y a-t-il un lien particulier entre les erreurs d'annotation en noms propres et la nécessité d'avoir un filtrage particulièrement restreint ici ? Un très grand nombre de noms propres annotés dans les documents entraîne-t-il un plus grand besoin de filtrage ?

Groupe 2 $c\_Taux = \{2, 3\}$ **: p2.5r/loc et op3r/time.** Ces listes sont relativement peu permissives. Elles sont de type R, avec une constante de sélection très forte ($c > 2$). Ces listes mènent à des gains de performance alignés ou en dessous de la moyenne en précision (**mg-I**$_{\text{P@1}} = +7$, **loc.P@1** : **+6.1** et **time.P@1** : **+4.2**), sans nuire à RITEL en rappel (**loc.P@10** : **-1.5** et **time.P@10** : +0.0). Ces deux classes (**loc** et **time**), après celles du groupe 1 et la classe **subsnn** (dont nous discuterons ensuite), tirent le plus avantage du filtrage (tableau 5.2) tout en laissant les performances stationnaires vis-à-vis des TOP$_6$ à la fois en précision et en rappel (**loc** reste en tête dans les deux cas, **time** ne perd qu'une place en précision et passe 3$^{\text{ième}}$). **D'après ces résultats, nous pouvons dire que, par classe de question, un couplage** $m = R$ **et** $c \geq 2$ **jouera en faveur d'une meilleure précision, tout en préservant le rappel.**

Ici, comme pour le groupe précédent, on peut s'interroger sur le fait que des listes de type p et op permettent au système de tirer un meilleur avantage du filtrage que des listes aux paramètres de sélection similaires (re-treintes, à forte constante c), mais de type o.

L'effet des paramètres m et c sur le filtrage des documents sélectionnés par RITEL consiste essentiellement à réduire l'espace de recherche E$_R$. L'analyse des résultats par classe de question (tableau 5.5) semble laisser penser que l'effet du paramètre f consiste à influer sur la nature de cet espace. Du point de vue du modèle de langue, on dira que f influe sur la nature de E$_R$ en terme d'information. Du point de vue QR, on dira que ce paramètre influe sur la nature linguistique de E$_R$. Des deux points de vue, la question que les résultats soulèvent est de découvrir quelle réalité (informationnelle ou linguistique) sommeille au sein des paramètres OOV et PPX, en lien avec les classes de questions.

Groupe 3 $c\_Taux = 4$ **: o3n/nbr et o2.5n/subsnn.** Ces listes sont les plus permissives. Elles sont de type N, avec une constante de sélection très forte ($c > 2$). Ces listes mènent à des gains au-dessus de la moyenne, à la fois en terme de précision (**mg-I**$_{\text{P@1}} = +7$, **nbr.P@1** : +0.0 et **subsnn.P@1** : **+7.7**), et en terme de rappel (**mg-I**$_{\text{P@10}} = -2$, **nbr.P@10** : **+2.2** et **subsnn.P@10** : +0.0). Elles ne nuisent jamais au système. Néanmois, après filtrage, les 2 classes de questions qui leur correspondent présentent les taux de précision les plus bas (tableau 5.2), bien que **subsnn** double sa précision. La raison à cela semble due à la marginalité de ces deux classes de question vis-à-vis de celles des groupes 1 et 2. En effet, **nbr** est la classe présentant les performances qui restent le plus stationnaires avec et sans filtrage, **subsnn** étant la classe pour laquelle le système présente le plus de difficultés de traitement à l'origine. Vu les taux de sélection opérés par les listes **o3n** (89.5%) et **o2.5n** (92%) qui leur correspondent, il semblerait que l'élimination de documents fortement bruités [6] permette au système d'extraire des réponses pertinentes en top-1 dans le cas de **subsnn**, mais à des rangs inférieurs uniquement dans le cas de **nbr**, dans une quantité infinitésimale. En effet, le gain perçu en rappel pour cette classe (+2.2%), sur la base du nombre de questions évaluées de ce type (45), représente une seule réponse en plus que le système a trouvé via le filtrage à l'échelle de segments par rapport au nombre de réponses qu'il a su extraire sans. Ceci suggère que l'effet du filtrage sur la classe **nbr** est sans influence sur le comportement habituel du système. **D'après ces résultats, nous pouvons dire que par classe de question, un couplage** $m = N$ **et** $c > 2$ **jouera en faveur d'une meilleure précision ou d'un meilleur rappel, sans jamais pénaliser les performances de RITEL.** Ici encore, on peut s'interroger sur le lien qui existe entre les deux types de classes étudiées et l'intérêt apparent dans l'application d'une fonction de sélection de type o pour le filtrage de ces classes plutôt que de type p ou op.

6. On rappelle que c'est l'hypothèse qu'on avait avancée pour expliquer l'impact relativement faible du filtrage sur le comportement habituel du système à l'échelle des documents au chapitre 3.

Récapitulatif et conclusion À l'échelle des segments, le comportement du filtrage observé par classe est proche de celui observé en général (global). Ainsi, les patrons de sélection que nous avons définis sur la base des performances QR globales de RITEL correspondent majoritairement aux triplets de sélection spécifiques des meilleures listes par classe de question. Il est donc possible de créer un appariement classe-liste spécifié par le biais de tels patrons pour la sélection de documents pertinents en QR. Le tableau 5.6 présente de tels appariements. Dans ce cas, nous avons repris les patrons issus de la table 5.4 (voir p. 116) et les avons étendus afin de capturer les types de listes non couvertes par classe de question. Les patrons obtenus couvrent 1/3 des 42 listes que nous avons évaluées dans nos expérimentations.

Cependant, nous avons relevé un manque de stabilité de certains résultats, par exemple dans le choix de la fonction f. Il est donc nécessaire de voir si un appariement ainsi optimisé généralise bien ou non, pour d'autes systèmes que RITEL et d'autre corpus d'évaluation que ceux que nous avons utilisés dans cette thèse.

Qcat	T
global	$\langle p, R, c_2 \rangle$
loc	$\langle p, R, c_2 \rangle$
nbr	$\langle o/p, N, c_2/c_3 \rangle$
pers	$\langle o/op, R, c_0 \rangle \vert \langle op, R, c_1 \rangle$
qpers	$\langle o/op, R, c_0 \rangle \vert \langle op, R, c_1 \rangle$
subsnn	$\langle o/p, N, c_2/c_3 \rangle$
time	$\langle op, R, c_3 \rangle$

TABLE 5.6 – Proposition d'appariement classe-liste spécifié par patron de sélection (T) pour le filtrage de documents pertinents pré-segmentés en QR. En orange, les (parties de) patrons définis à partir des analyses bi-variées relatives aux performances globales de RITEL (Cf. table 5.5). En noir, ceux définis à partir de l'analyse des performances de RITEL par classe de questions (**Qcat**). $c_0 : c = \{0, 0.5\}$, $c_1 : c = \{1, 1.5\}$, $c_2 : c = \{2, 2.5\}$ et $c_3 : c = \{3\}$.

Si les analyses par classe de questions nous ont permis l'extension des patrons de sélection pour la construction d'un nouveau type d'appariement classe-liste, elles nous ont également amené à nous interroger sur la nature des documents sélectionnés par le filtrage (Quel genre d'information contiennent-ils ? Qu'ont-ils en commun ?) et sur l'apparente prédisposition de certaines listes de filtrage à aider RITEL dans sa tâche pour certaines classes de questions et pas d'autres, à l'échelle des segments. En effet, la répartition des listes à travers les classes de question ne semble pas être fortuite (p. ex. des classes similaires d'un point de vue QR comme **pers** et **qpers** semblent filtrées de manière similaire).

Pour répondre à ces questions, **il faut** :

- **mener des investigations plus poussées sur le lien entre entités nommées, classes de questions et paramètres de filtrage (m, c et f) suivant les performances QR** ;
- se pencher sur la façon dont les erreurs de marquage en entités nommées affectent le filtrage ;
- et s'intéresser à la composition des listes de filtrage.

Le dernier point vise spécifiquement à répondre aux questions que nous venons d'énoncer à propos de la nature des documents sélectionnés par le filtrage. Pour nous guider dans cette tâche, il pourrait être utile de faire appel à des algorithmes de clustering guidés par le contenu des documents.

5.4 Conclusion

Dans ce chapitre, nous avons appliqué la méthode de filtrage introduite au chapitre 3 sur le corpus de documents segmentés uniformément, présenté au chapitre 4.

Pour cela, nous avons d'abord créé un corpus de développement contenant des segments a priori pertinents pour les recherches d'un point de vue QR (le corpus **DEV698**). Ce corpus a été créé à partir du modèle que nous avions utilisé dans nos expériences initiales de filtrage (c.-à-d. le corpus de documents **DEV509**).

À partir du corpus **DEV698**, nous avons créé 42 nouveaux classifieurs adaptés pour la catégorisation de documents segmentés, que nous avons appliqués sur le corpus Quaero de 500k pages web (*Q07fr*). Nous avons ainsi généré 42 listes de segments. De là, nous avons pu tester la validité de notre méthode d'évaluation d'un texte, à l'échelle des segments.

Les classifieurs générés à partir du corpus de segments **DEV698** se comportent de façon similaire à ceux obtenus à partir du corpus de documents **DEV509** : même phase de progression des taux de sélection dans les deux cas selon les trois paramètres de filtrage : f, m et c (les taux moyens varient d'environ 5%).

Néanmoins, on a pu remarquer que le changement d'échelle inversait le pouvoir discriminant des paramètres OOV et PPX (ce dernier étant plus discrimant à l'échelle des segments qu'il ne l'était à l'échelle des documents). Cet effet se trouve plus marqué avec la méthode restreinte ($m = R$) d'estimation des distributions OOV et PPX qu'avec la méthode normale ($m = N$). Nous avons vu à la table 5.1 que les valeurs de moyennes et d'écarts types des paramètres OOV et PPX calculées à partir des corpus **DEV509** et **DEV698** étaient très proches (quelle que soit la valeur de m).

Cet effet pourrait provenir des modifications du contenu textuel des pages web opérées par Lynx à leur extraction. Il peut aussi s'expliquer par le contenu informationnel restreint des segments, qui rend le modèle de langue plus sensible envers PPX qu'envers OOV.

Le travail que nous avons mené ensuite a consisté à étudier l'impact du filtrage à l'échelle des segments avec les 42 listes générées suivant la méthode EPID et à comparer les résultats correspondant à ceux obtenus en filtrage à l'échelle des documents au chapitre 3.

D'après ces études nous pouvons dire que, globalement, le filtrage en contexte de segmentation aide un système QR dans sa tâche et de manière plus efficace à l'échelle des segments qu'à l'échelle des documents.

L'analyse de l'impact du filtrage sur les performances QR globales de RITEL suggère que le couplage segmentation-filtrage n'est pas approprié en terme de rappel. Dans ce cas, il vaut mieux désactiver le filtrage et suivre le fonctionnement habituel du système. Cependant, l'ampleur des gains perçus en précision et MRR par classe de questions suggère que (en moyenne +7% et +4% respectivement) la détérioration des performances du système causé par le filtrage en rappel est compensée par les bénéfices en précision.

L'étude comparative du comportement de RITEL en contexte de filtrage aux deux échelles de représentation étudiées suggère que le filtrage favorise plus fréquemment la précision que le rappel, et de façon plus conséquente. Par ailleurs, à l'une comme à l'autre de ces échelles, le filtrage est sans effet sur la classe **nbr**. En effet, pour cette classe, les performances QR de RITEL sont stationnaires, quelque soit l'échelle de filtrage considérée.

Enfin, nous avons mené une étude qui vise à mieux saisir les rapports de dépendances qui existent entre les paramètres m, f et c de la méthode EPID et les performances QR de RITEL, à partir des résultats d'évaluation de filtrage de RITEL en contexte de segmentation. L'étude s'est faite à deux niveaux de granularité différent : à un niveau global sans tenir compte des classes de question, à un niveau plus fin en en tenant compte.

Au niveau global, nous avons réalisé des analyses bi-variées en terme de précision et de rappel des performances de RITEL sur chacune des 42 listes de filtrage que nous avons générées et du comportement habituel du système sans filtrage (en contexte de segmentation) en fonction des paramètres de sélection. Ces analyses nous ont permis de mieux comprendre l'influence mutuelle et respective de ces paramètres sur les performances QR. À l'issue de ce travail, nous avons proposé différents calibrages-types de filtrage pour la sélection de documents pertinents en QR, suivant le niveau de performances recherché : optimalité des performances en terme de précision ou de rappel, performances moyennes en général (centrées sur la précision ou le rappel) et meilleur compromis entre précision et rappel. Ces paramétrages sont détaillés au tableau 5.4.

Spécifiquement, nous avons étudié les performances de RITEL à l'aide des mêmes méthodes que celles utilisées au niveau global, en terme de précision, MRR et rappel. Ici, nous avons adopté différents points de vues : point de vue des performances, des listes de filtrage et des classes de questions, avec un intérêt particulier pour les deux derniers points de vues. Cela nous a permis de faire le lien avec les résultats d'analyses multivariées et de définir empiriquement de nouveaux appariements classe-liste basés sur les patrons de sélection de la table 5.4. De tels patrons permettent de couvrir plusieurs listes de filtrage à la fois, en fonction des paramètres m, f, et c, en adéquation avec chaque classe de question. Ces patrons sont détaillés dans la tableau 5.6.

L'analyse des dépendances entre performances QR globales et paramètres de filtrage aux deux niveaux étudiés nous ont appris que le changement de discrimination de PPX mentionné plus tôt dans cette conclusion, n'a d'emprise sur le filtrage qu'à un niveau global : choisir des listes de filtrage avec une fonction de sélection basé sur ce paramètre (c.-à-d. $f = p$), tend à orienter les performances du système vers un meilleur compromis précision-rappel, mais n'assure pas les meilleures performances ni en précision, ni en rappel, ni globalement ni par classe de question.

Les analyses menées à un niveau global nous ont permis de confirmer l'hypothèse **H0** : à l'échelle des segments, le filtrage est plus efficace qu'à l'échelle des documents. À cette échelle, les listes amenant RITEL aux meilleures performances ont des taux de sélection qui oscillent entre 25% et 70%. Cela montre que le filtrage appliqué à cet échelle fait plus que seulement écarter les documents très bruités, comme nous l'avions conclu au chapitre 3 en étudiant l'impact du filtrage sur des documents complets.

Spécifiquement, les analyses de dépendances ont montré que les listes de filtrage menant aux meilleures performances pourraient être dépendantes des classes de question. On a notamment constaté que des classes similaires d'un point de vue QR étaient filtrées de manière similaire, c'est-à-dire que leurs paramètres de sélection m, f et c optimaux ont des caractéristiques proches. En particulier, on a conclu de ces analyses que si m et c ont une influence marquée vis-à-vis du nombre de documents filtrés, f semblait influer sur la nature de ces derniers. Ces analyses nous ont conduit à nous poser un certain nombre de questions vis-à-vis du lien apparent entre les classes de questions et le type de listes utilisées en filtrage. Par exemple, des classes sensibles aux erreurs de vocabulaire, comme les classes **pers** et **qpers**, semblent prédisposées à l'utilisation de listes avec une fonction de sélection basée sur OOV plutôt que PPX. Sans analyses supplémentaires du lien probable entre les entités nommées couvertes dans nos listes, les classes de question et les paramètres de filtrage, il est difficile d'aller plus loin dans les interprétations et l'influence des paramètres m, f et c sur le comportement du système.

Les analyses de dépendances que nous avons menées appuient en général les conclusions apportées par les résultats d'évaluation à propos de l'impact du fitrage sur les performances QR. En effet, le comportement du système par classe est similaire à son comportement global, exception faite des classes **subsnn** et **nbr** qui semblent nettement en marge des autres classes de question et pour lesquelles le filtrage semble n'avoir aucune influence.

« Que m'est-il permis d'espérer ? » [67]

EMMANUEL KANT

Chapitre 6

Conclusion du travail de thèse et perspectives de recherche

6.1 Conclusion

NOTRE travail de thèse s'est déroulé dans le cadre des systèmes de recherche de réponses précises à des questions, autrement dit les systèmes de questions-réponses. L'une des idées majeures à l'origine de nos travaux dans cette thèse est qu'utiliser des techniques de RI pour l'évaluation de la pertinence des documents dans un contexte QR pouvait améliorer les résultats d'un système QR. En effet, les travaux présentés dans le cadre de campagnes d'évaluation telles que TREC et surtout NTCIR dans les domaines de la RI et du QR, et en particulier du challenge IR4QA de la campagne NTCIR-7, mettent en avant le potentiel qu'il y a à intégrer des approches RI dans un système QR.

Cette volonté d'intégrer RI et QR nous a motivé dans cette thèse à utiliser et étudier rigoureusement un modèle basé sur une approche de RI en QR orientée web, pour la sélection pertinente de documents ; nous avons choisi pour cela l'emploi d'un modèle de langue, technique largement utilisée en RI.

Dans cette thèse, nous avons parcouru l'état de l'art des systèmes de Questions-Réponses. Nous avons vu quels étaient leurs fondements, leurs mécaniques et leurs composantes. Nous avons pu constater que parmi les étapes-clés de la chaîne QR (analyse de question, recherche d'information et extraction de réponse), une phase de traitement importante consistait à sélectionner les documents pertinents dans le contexte d'une question posée par un utilisateur au système.

À cette occasion, nous avons vu que cette sélection dépendait de la classe de la question et du type de réponse qui lui correspondait, ces types prenant la forme d'entités nommées (EN) utilisées pour guider le système dans ses recherches à l'intérieur des documents.

Nos travaux se sont orientés essentiellement dans deux directions : tout d'abord l'utilisation d'un modèle de langue pour évaluer la pertinence d'un document et ensuite la segmentation des documents issus du web en documents plus homogènes.

Étant données les techniques d'extraction d'information utilisées en QR, nous considérons qu'un document pertinent pour une recherche QR est tout d'abord un document proche, dans sa forme, d'un document bien écrit. Le présupposé est que les documents trop bruités issus du web conduisent à une extraction de réponse non pertinente, et ne sont donc pas pertinents.

Nous avons donc mis en place une méthode statistique capable d'évaluer la pertinence intrinsèque (ou a priori) d'un document (méthode EPID), pour la sélection des documents en QR. Cette méthode s'appuie sur un modèle de langue et un système de classification (binaire) de documents en classe de pertinence (document *pertinent* ou document *non pertinent* pour les recherches en QR). Le modèle que nous avons développé s'adapte en QR en filtrant la sélection des documents pertinents faite par le système sur la base de la classe des questions.

Des études préliminaires sur les corpus à notre disposition ont mis en avant le fait que les documents issus du web pouvaient être longs et surtout hétérogènes. Nous avons émis l'hypothèse que la variabilité naturelle des pages web en taille et en contenu pénalise vraisemblablement un système QR dans sa tâche. Si le découpage des documents en passages est très étudié en QR (par exemple recherche de la taille optimale des passages, recherche de la meilleure fenêtre d'analyse des documents pour la découpe, etc.), il intervient soit lors d'une étape de pré-traitement au moment de l'indexation des documents (c.-à-d. **index-time passaging**) soit lors d'une étape ultérieure, au moment de la recherche de candidats-réponses (c.-à-d. **search-time passaging**). Jamais ce découpage n'intervient lors de ces deux étapes à la fois. Nous avons émis l'hypothèse que ce découpage pratiqué aux deux extrémités de la chaîne de traitement peut améliorer les résultats en renforçant l'effet de cohérence locale des documents.

Dans ce contexte, nous avons mis en place un système de pré-segmentation qui consiste à segmenter le contenu textuel des pages utilisées par un système de QR orienté web à l'indexation, puis à redécouper les segments en passages au moment des recherches. L'idée était de vérifier si une telle approche peut effectivement aider un système QR dans sa tâche et de valider (ou invalider) l'hypothèse sous-jacente émise lors de nos analyses préliminaires. Ce système repose sur le navigateur textuel de pages web Lynx (pour l'extraction du contenu textuel des pages), un outil de normalisation de pages web (Kitten) développé au LIMSI et un algorithme dédié pour la segmentation de documents textuels.

Les corpus et le système que nous avons utilisés dans nos évaluations en contexte QR sont restés fixes à travers toutes nos expérimentations. Toutes ces évaluations ont été faites dans le cadre du programme Quaero en utilisant RITEL, un système QR développé au LIMSI. Les jeux de questions (309 questions de test et 722 d'entraînement) que nous avons utilisés portent exclusivement sur des questions d'ordre factuel. Le corpus de documents *Q07fr* est composé de 500k pages web en français.

La méthode EPID a été évaluée à travers les performances de RITEL. Nous avons produit des listes de documents a priori pertinents pour les recherches en QR. Le module de filtrage mis en place intègre ces listes de documents pertinents. Lorsque le module de RI de RITEL retourne une liste de documents pertinents étant donnée la question, le module de filtrage ne conserve que les documents a priori pertinents. Seuls les documents communs aux listes les plus appropriées pour le filtrage et aux documents sélectionnés par RITEL seront candidats pour l'extraction de passages. Nous avons d'abord appliqué ce filtrage à l'échelle des documents (c.-à-d. sur des documents complets), puis à l'échelle des segments (c.-à-d. sur des sous parties de documents) grâce au système de pré-segmentation de pages web.

Les résultats d'évaluation QR obtenus suggèrent que le filtrage des documents permet d'améliorer *en général* les résultats. **À l'échelle des documents**, le filtrage aide de façon substantielle le système dans sa tâche sans jamais le pénaliser. Cependant, les gains perçus restent faibles (environ 1% moyen en terme de précision, MRR et rappel) en comparaison des performances habituelles de RITEL (sans filtrage). **À l'échelle des segments**, le filtrage semble avoir plus d'impact. En effet, les résultats obtenus par RITEL dans ce cadre sont généralment plus élevés que sans filtrage (circa 2.5% de gains supplémentaires avec filtrage que sans en précision globale).

Par ailleurs, ces résultats sont clairement meilleurs par classe de question qu'ils ne l'étaient à l'échelle des documents, même s'il arrive que le filtrage pénalise le système en rappel (-2% en terme de rappel moyen par classe). Les analyses montrent que la détérioration des performances de RITEL dans ce cas est largement compensée par les gains perçus en précision (7% de gains moyen en terme de précision pour un total de 4% de gains en terme de MRR au final). Il n'en reste pas moins que si le but recherché à cette échelle est d'avoir le plus de réponses pertinentes possible (p. ex. dans l'idée d'appliquer une procédure de réordonnancement de réponses sur les sorties du système), il est préférable de désactiver le filtrage.

À l'échelle des documents, les résultats d'évaluation obtenus suggèrent que le filtrage permet d'éliminer les documents les plus bruités. En effet, à cette échelle, les listes utilisées par le système au moment du filtrage contiennent entre 80% et 90% des documents sélectionnés à l'étape de RI par RITEL. Cependant, ceci n'est qu'une tendance, les tests de significativité (McNemar) de ces résultats ne sont pas concluants. Nous avons également testé deux méthodes d'appariements (liste, classe de question) : **Ma1** où on considère une seule liste quelle que soit la classe de la question et **Ma2** où on considère la meilleure liste pour chacune des classes de questions. Cet appariement est appris sur le corpus de questions d'entraînement. À l'échelle des documents, aucun résultat (que ce soit avec la méthode proposée ou la méthode contrôle) n'a permis de conclure en l'intérêt d'un appariement de ce type.

Ce constat nous a amené à penser que RITEL était en situation de surapprentissage, et que les données d'apprentissage utilisées à l'entraînement du système pour l'estimation automatique de ses paramètres de recherche et d'extraction d'information pour définir de tels appariements, n'étaient pas suffisantes.

À l'échelle des segments, le filtrage semble permettre une élimination d'autres documents que les plus bruités. En effet, les listes utilisées par le système au moment du filtrage contiennent entre 25% et 70% des segments sélectionnés à l'étape de RI par RITEL. Ceci nous a permis de confirmer notre hypothèse selon laquelle la variabilité naturelle des pages web en taille et en contenu pénalise vraisemblablement un système QR dans sa tâche. En effet, à l'échelle des segments, la cohérence locale du texte rend le filtrage moins permissif et favorise l'usage de listes de documents plutôt restreinte pour filtrer la sélection de RITEL.

Les analyses de dépendances entre les différents paramètres utilisés (fonction de sélection m, méthode d'estimation des distributions f et constante de variation c) de la méthode EPID et les performances QR, que nous avons menées à un **niveau global**, à l'échelle des segments, nous ont permis à l'aide de patrons de sélection, de proposer différents calibrages-types de filtrage pour la sélection de documents en QR, en fonction d'un niveau de performances recherché. Ces patrons couvrent plusieurs listes de filtrage à la fois.

Ces analyses de dépendances nous ont permis non seulement de proposer de **nouveaux appariements classe-liste** fondés sur les patrons de sélection définis par les différents calibrages-types de filtrage pour la sélection de documents en QR mais aussi de mieux comprendre l'influence des paramètres de la méthode EPID sur les performances QR, et notamment des paramètres OOV (mots hors vocabulaire) et PPX (perplexité) du modèle de langue sur lequel elle s'appuie. Ainsi, nous avons pu comprendre que si le changement d'échelle inversait le pouvoir discriminant de ces paramètres (PPX étant plus discrimant à l'échelle des segments qu'il ne l'était à l'échelle des documents), cela n'avait d'emprise sur le filtrage qu'à un niveau global : choisir des listes avec une fonction de sélection basée sur ce paramètre tend à orienter les performances du système vers un meilleur compromis précision-rappel, mais n'assure pas les meilleures performances ni en précision ni en rappel, ni par classe de question, ni globalement. En particulier, nous avons pu conclure que si la fonction de sélection m et la constante de variation c ont une influence marquée vis-à-vis du nombre de documents filtrés, la méthode d'estimation des distributions f semblait, elle, influer sur la nature des documents sélectionnés.

Ces analyses ont montré que les listes de filtrage menant aux meilleures performances étaient dépendantes des classes de question. Par exemple, des classes sensibles aux erreurs de vocabulaire, comme les classes *pers* et *qpers*, semblent prédisposées à l'utilisation de listes avec une fonction de sélection basée sur OOV (o) plutôt que PPX (p). Néanmoins, on atteint les limites explicatives du modèle lors de l'étude des résultats de filtrage par classe de questions. À ce niveau de granularité, le besoin se fait sentir de se pencher sur une analyse plus linguistique des résultats pour continuer l'introspection de la méthode EPID. Notamment, l'idée serait de savoir si un lien de dépendance existe entre les classes de question, le type de listes prépondérantes pour le filtrage et le type d'entités nommées couvertes dans les deux cas.

Lors de l'évaluation de la procédure de pré-segmentation des documents sur la tâche QR, les résultats obtenus nous ont permis de valider partiellement notre hypothèse de départ. Partiellement car les tests statistiques (McNemar) opérés sur ces résultats sont tout juste significatifs. Mais validation malgré tout, car pré-segmenter les documents à l'indexation en plus de leur découpage habituel en passages au moment des recherches amène un gain de performances d'environ 4% en rappel (0.3% en précision) par rapport aux performances du système sans segmentation. La segmentation la plus efficace s'est révélée être une segmentation uniforme (c.-à-d. quand on découpe les documents en segments de taille fixe) et pas une segmentation thématique (par TextTiling), comme nous le pensions. Nous avons vu que la segmentation uniforme permettait l'extraction de réponses plus pertinentes (majoritairement dans le top-3) et en plus grand nombre que la segmentation par TextTiling.

Nous avons procédé à une évaluation modulaire de la chaîne de traitement QR. Celle-ci nous a permis de constater que la pré-segmentation permet d'extraire davantage de bonnes réponses. De ce point de vue, nous avons validé notre hypothèse. Or la sélection de document donne de meilleurs résultats, c'est-à-dire qu'elle retourne davantage de documents contenant la réponse, lorsque le système travaille sur des documents complets que lorsqu'il travaille sur des segments. Notre hypothèse est que cela peut s'expliquer par le fait que les segments étant plus courts que les documents complets, ils pouvaient contenir moins d'informations pertinentes que ceux-ci.

Si l'extraction de passages contenant une réponse de référence a une bonne précision, en revanche ces derniers contiennent plus rarement la réponse bien délimitée et typée que lorsque la sélection se fait sur des documents complets. Autrement dit, le module d'extraction des candidats-réponses a moins de chance d'extraire la bonne réponse. Cependant, le taux de bonnes réponses est meilleur. Pour expliquer ce phénomène, nous avons émis l'hypothèse que RITEL produit moins de passages pertinents à partir de documents segmentés qu'à partir des documents non segmentés mais qu'en retour, la tâche d'extraction de réponses en est facilitée. Par ailleurs, ceci nous a pointé du doigt la nécessité d'améliorer le système de reconnaissance en EN de RITEL.

En s'intéressant à la couverture des réponses trouvées par RITEL en terme de questions pour les différentes conditions de segmentation testées (uniforme et thématique) et la condition de référence (sans segmentation), nous avons pu confirmer que l'emploi d'une procédure de réordonnancement de réponses sur les sorties combinées du système dans les 3 conditions présentait un potentiel intéressant pour améliorer les performances QR finales. En effet, nous avons montré théoriquement qu'il était possible d'obtenir dans le contexte d'un réordonnancement parfait de réponses (c.-à-d. qui permettrait de placer en tête de classement toutes les réponses aux questions couvertes spécifiquement dans chaque condition) jusqu'à 6.4% de gain en précision et 7.1% en rappel, comparé aux performances habituelles de RITEL sans segmentation.

6.2 Perspectives

Si l'approche expérimentale que nous avons suivie (contrôles, analyses et étude approfondie des résultats d'évaluation) nous a permis de mieux comprendre les tenants et les aboutissants du filtrage des documents aux différents niveaux de granularité considérés (échelle des documents et échelle des segments), celle-ci ne change pas fondamentalement la réalité des résultats obtenus : les bénéfices que le filtrage peut apporter par le biais de la méthode EPID pour un système de QR dans sa tâche restent mitigés. En effet, les tests statistiques que nous avons pu mener nous ont seulement permis de parler de tendance.

Il nous semble donc nécessaire de sortir du cadre d'expérimentation que nous avions fixé, en testant les méthodes développées dans cette thèse toujours en contexte QR, mais de façon détachée des données que nous avons utilisées. Cela nécessite d'un point de vue général, de répéter les expériences que nous avons menées dans cette thèse à l'aide d'un corpus de documents différent du corpus Quaero *Q07fr* et de jeux de questions d'entraînement et d'évaluation autres que ceux que nous avons utilisés. Dans cette perspective, il nous semble intéressant de faire varier la taille de ces corpus afin d'étudier leur impact sur la méthode EPID.

Dans le cadre du projet Quaero, nous disposons d'un corpus de pages web de 20G (2.5M de pages web), incluant le corpus de 5G *Q07fr*. Nous projetons donc un passage à l'échelle de nos index en contexte de filtrage (avec et sans segmentation) sur RITEL pour connaître l'effet de notre méthode à grande échelle.

Du point de vue des données, deux autres facteurs à tester concernent la nature des questions (orale ou écrite) et la langue. RITEL a la capacité de travailler sur les deux natures de question, en plusieurs langues (français, anglais et espagnol). Nous comptons également tester ce genre de variabilité avec un corpus de 2000 questions orales/écrites, sur le français et portant sur une collection de documents oraux et écrits.

Nous avons émis plusieurs hypothèses concernant les résultats peu concluants de la méthode EPID à l'échelle des documents. L'une d'elle suggère d'enrichir le modèle de langue sur lequel repose la méthode. En effet, il est vrai que les paramètres OOV et PPX que nous avons utilisés sont relativement primitifs. Il serait donc intéressant de questionner le pouvoir de discrimination de paramètres plus complexes, en se plaçant à un niveau de représentation de la langue plus abstrait que le simple niveau lexical que nous avons étudié. À un niveau syntaxique, on pourrait par exemple enrichir le modèle en utilisant des n-grams basés sur les catégories morphosyntaxiques des mots.

Une autre idée concernant l'utilisation de modèles de langues plus complexes serait tout simplement d'en utiliser plusieurs, si possible spécialisés selon différentes thématiques. Cette idée est moins facilement réalisable que la précédente au sens où il est plus facile d'adapter la représentation linguistique d'un corpus de textes en domaine général que d'acquérir en quantité suffisante de nouvelles données pour la création de modèles en domaine spécifique (c.-à-d. adaptés à chaque thématique désirée).

L'autre hypothèse qu'il nous semble intéressant de tester concerne la robustesse du modèle de documents pertinents sur lequel repose la méthode. À l'échelle des documents, nous avons utilisé un corpus de 509 documents (DEV509). À l'échelle des segments nous avons utilisé un corpus de 698 documents pré-segmentés (DEV698). Nous projetons d'utiliser à cette échelle un corpus plus grand, d'au moins 1000 segments, couvrant partiellement les documents du corpus DEV509, afin de mesurer son impact sur les performances de filtrage.

Lorsque nous avons proposé les méthodes d'appariements, nous nous sommes appuyé sur le type attendu de la réponse pour déterminer les classes de questions. Il serait intéressant, pour ne pas dire primordial, de tester les appariements en s'appuyant sur des classes de questions fondées sur le thème des questions, leur forme ou encore leur contenu lexical.

Concernant nos travaux de pré-segmentation des documents, il serait nécessaire d'évaluer la pertinence du système de segmentation que nous avons mis en place face à d'autres systèmes état de l'art en segmentation de pages web textuelles. Dans ce contexte, nous pensons poursuivre les travaux entamés avec le TextTiling en segmentation thématique en employant des implémentations plus poussées que celle que nous avons employée avec NLTK, et en particulier, tester un calcul des scores lexicaux basé sur la théorie des chaînes lexicales ou bien par « apparition de termes ». Ces scores sont peut-être plus appropriés pour la tâche de segmentation de textes issus de pages web que celui de la méthode par comparaison de « blocs » . Il nous semble également pertinent d'examiner d'autres types de segmentation, par exemple basés sur la représentation visuelle des pages web. Cette piste nous semble prometteuse quand on considère les travaux récents en segmentation de pages web (ou en classification de pages web) et quand on considère que la structure et le design des pages web guident la présentation et l'organisation de la lecture des pages et donc de leur contenu informationnel.

Nous sommes en train d'étudier l'impact du module d'extraction textuelle du contenu des pages web utilisant Lynx et Kitten, sur les performances QR. En effet, nous avons dit à plusieurs reprises que les ajouts textuels de Lynx aux sorties d'extraction pouvaient jouer un rôle sur les résultats obtenus par RITEL. L'idée retenue pour vérifier cette hypothèse consiste à filtrer les sorties de segmentation produites par notre système sur la base des ajouts textuels de Lynx et d'en étudier l'impact sur les sorties QR.

Nous pourrions vérifier, à l'aide d'une segmentation de référence manuelle, le caractère réellement thématique de la segmentation par TextTiling. Afin de mieux comprendre les différences de comportement entre cette segmentation et la segmentation uniforme, nous pourrions de plus examiner manuellement les cas où ces segmentations diffèrent quant aux réponses apportées. Ces expériences pourraient expliquer les résultats décevants de la segmentation par TextTiling.

Nous pourrions déterminer l'influence du nombre de blocs de segmentation et de leur taille sur les performances QR, notamment, déterminer la taille optimale de pré-segmentation des pages web à l'indexation. Ce dernier point rejoint de nombreux travaux menés dans le domaine de la sélection pertinente de passages. Dans ce cadre, nous pourrions comparer les méthodes standards de segmentation pour trouver la fenêtre d'analyse textuelle optimale d'un système QR à la stratégie d'adaptation automatique de la taille de cette fenêtre développée sur le système RITEL. De tels travaux pourraient nous permettre en particulier de tester l'hypothèse que nous avons émise à propos de la taille des segments pour expliquer les moins bons résultats de RITEL à l'étape de recherche de documents en contexte de segmentation vis-à-vis de son fonctionnement habituel.

Enfin, une étude fine du comportement d'extraction de réponses de RITEL devrait être menée afin de vérifier l'hypothèse selon laquelle la tâche d'extraction de RITEL est facilitée quand RITEL agit sur un nombre réduit de passages pertinents issus de documents segmentés (dont la taille moyenne est d'environ 20 lignes sur le corpus *Q07fr*) qu'à partir de documents non segmentés (d'environ 300 lignes en moyenne sur ce corpus).

6.3 Ouverture

Durant ces années de thèse, nous nous sommes beaucoup intéressé aux travaux de segmentation visuelle dans les domaines de la classification de pages web, de la Recherche d'Information dans des images, des vidéos et des représentation numériques de documents web et en analyse numérique d'images.

L'étude de la littérature associée à ces domaines nous a guidé vers des travaux de recherche menés en psychologie cognitive et, en particulier, en oculométrie [1], appliquée à la navigation de pages web chez l'humain.

1. L'oculométrie regroupe des techniques permettant d'enregistrer les mouvements oculaires et suivre le regard d'une personne.

Suite à ces investigations, il nous semblerait intéressant d'introduire la dimension questions-réponses dans le cadre d'études de la navigation de pages web, afin d'étudier la tâche QR, non pas à travers un système, mais à travers l'utilisateur en tant que sujet expérimental. Ainsi, en posant une question à un sujet, il serait possible de suivre les stratégies de recherche mises en place par ce dernier, à la fois d'un point de vue visuel et linguistique, dans le but de trouver une réponse en naviguant librement sur Internet. Nous pensons qu'appliquer le paradigme QR dans ce cadre expérimental permettrait de renouveler l'idée qu'on se fait de la tâche de RI en TAL en général et en QR en particulier, de mieux comprendre comment instancier cette tâche au sein de systèmes qui en sont dépendants et de mieux appréhender les critères de pertinence à appliquer afin de mettre en place des moteurs de recherche et des systèmes QR qui répondraient mieux à la demande des utilisateurs. À n'en pas douter, dans une telle démarche, les progrès et connaissances accumulés en QR, en RI et plus généralement en TAL ont clairement à apporter aux sciences cognitives, dans le cadre d'étude de la navigation de pages web en perception visuelle chez l'humain.

L'oculométrie est utilisée comme outil de mesure pour la recherche en psychologie cognitive (étude de l'attention), en psycholinguistique (étude de la lecture) ou en ergonomie (étude de la mémoire).

Bibliographie

[1] AGRESTI, A. *Categorical data analysis*. Wiley, New York, 1990. 3.4.3, 4.4.2

[2] ANDROUTSOPOULOS, I., AND MALAKASIOTIS, P. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research 38*, 1 (2010), 135–187. 2.3.1

[3] ASIRVATHAM, A. P., RAVI, K. K., PRAKASH, A., KRANTHI, A., AND RAVI, K. Web Page Classification based on Document Structure, 2001. 4.1

[4] BAEZA-YATES, R. A., AND RIBEIRO-NETO, B. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999, ch. 5, p. 122. 1.3.3

[5] BARONI, M., CHANTREE, F., KILGARRIFF, A., AND SHAROFF, S. Cleaneval : a Competition for Cleaning Web Pages. In *Proceedings of the 6th International Conference on Language Resources and Evaluation* (Marrakech, Morocco, 2008), LREC'08, European Language Resources Association (ELRA). 2.3.1, 4.2.1.1

[6] BARR, A., COHEN, P. R., AND FEIGENBAUM, E. A., Eds. *The Handbook of Artificial Intelligence, Volumes 1*. William Kaufmann, Inc., Los Altos, CA, USA, 1982. 2.2

[7] BELL, T. C., MOFFAT, A., WITTEN, I. H., AND ZOBEL, J. The MG retrieval system : compressing for space and speed. *Communications of the ACM 38*, 4 (1995), 41–42. 2.3.3

[8] BENDERSKY, M., AND CROFT, W. B. Modeling higher-order term dependencies in information retrieval using query hypergraphs. In *Proceedings of the 35th International ACM SIGIR conference on Research and Development in Information Retrieval* (Portland, OR, USA, 2012), SIGIR'12, ACM, pp. 941–950. 2.3.1

[9] BERNARD, G. *Réordonnancement d'hypothèses dans un système de questions-réponses.* PhD thesis, Paris-Sud, LIMSI/CNRS, 2011. 2.3.1, 4.5

[10] BERNARD, G., ROSSET, S., GALIBERT, O., BILINSKI, E., AND ADDA, G. The LIMSI participation in the QAst 2009 track : experimentating on answer scoring. In *Proceedings of the Multilingual Information Access Evaluation I. Text Retrieval Experiments, 10th Workshop of the Cross-Language Evaluation Forum* (Corfu, Grece, 2009), CLEF'09, Springer (Lecture Notes in Computer Science). 2.4, 2.4.2

[11] BLEI, D. M., NG, A. Y., AND JORDAN, M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research 3* (2003), 993–1022. 1.3.3

[12] BOBROW, D. G. *Natural language input for a computer problem solving system*. PhD thesis, MIT, 1964. 2.2

[13] BOBROW, D. G., KAPLAN, R. M., KAY, M., NORMAN, D. A., THOMPSON, H. S., AND WINOGRAD, T. GUS, A Frame-Driven Dialog System. *Artificial Intelligence 8*, 2 (1977), 155–173. 2.2

[14] BOUMA, G., FAHMI, I., MUR, J., VAN NOORD, G., VAN DER PLAS, L., , AND TIEDEMANN, J. Linguistic Knowledge and Question Answering. *Traitement Automatique des Langues (TAL) 46*, 3 (2005), 15–39. 2.3.3

[15] BRIN, S., AND PAGE, L. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems 30*, 1-7 (1998), 107–117. 1.3.2

[16] BURKE, R. D., HAMMOND, K. J., KULYUKIN, V. A., LYTINEN, S. L., TOMURO, N., AND SCHOENBERG, S. Question Answering from Frequently Asked Question Files : Experiences with the FAQ FINDER System. *AI Magazine 18*, 2 (1997), 57–66. 2.2

[17] CALLISON-BURCH, C., AND OSBORNE, M. Statistical Natural Language Processing. In *A Handbook for Language Engineers*, A. Farghaly, Ed. CSLI, 2003. 1.2.1

[18] CHU-CARROLL, J., AVERBOCH, G. A., DUBOUÉ, P. A., GONDEK, D., MURDOCK, J. W., PRAGER, J. M., HOFFMANN, P., AND WIEBE, J. IBM in TREC 2006 Enterprise Track. In *Text REtrieval Conference* (Gaithersburg, MD, USA, 2006), TREC15, National Institute of Standards and Technology (NIST). 2.2

[19] CHU-CARROLL, J., CZUBA, K., PRAGER, J. M., ITTYCHERIAH, A., AND BLAIR-GOLDENSOHN, S. IBM's PIQUANT II in TREC 2004. In *Text REtrieval Conference* (Gaithersburg, MD, USA, 2004), vol. Special Publication 500-261 of *TREC13*, National Institute of Standards and Technology (NIST). 2.2

[20] CHU-CARROLL, J., DUBOUÉ, P. A., PRAGER, J. M., AND CZUBA, K. IBM's PIQUANT II in TREC 2005. In *Text REtrieval Conference* (Gaithersburg, MD, USA, 2005), vol. Special Publication 500-266 of *TREC14*, National Institute of Standards and Technology (NIST). 2.2

[21] COURTOIS, B. Un système de dictionnaires électroniques pour les mots simples du français in Dictionnaires électroniques du français. *Langue française 87* (1990), 11–22. 2.3.3

[22] CURTIS, J., MATTHEWS, G., AND BAXTER, D. On the Effective Use of Cyc in a Question Answering System. In *Proceedings of the IJCAI Workshop on Knowledge and Reasoning for Answering Questions* (Edinburgh, Scotland, UK, 2005), IJAI'05 (KRAQ), Professional Book Center, pp. 61–70. 2.2

[23] DE CHALENDAR, G., DALMAS, T., ELKATEB-GARA, F., FERRET, O., GRAU, B., HURAULT-PLANTET, M., ILLOUZ, G., MONCEAUX, L., ROBBA, I., AND VILNAT, A. The Question Answering System QALC at LIMSI, Experiments in Using Web and WordNet. In *Text REtrieval conference* (Gaithersburg, MD, USA, 2002), TREC11, National Institute of Standards and Technology (NIST). 2.3.2, 2.1, 2.3.3, 2.3.4

[24] DE SAUSSURE, F. *Cours de Linguistique Générale*. Payot, Paris, 1995 (orig. 1916). 2.3.2

[25] DÉCHELOTTE, D., SCHWENK, H., ADDA, G., AND GAUVAIN, J.-L. Improved Machine Translation of Speech-to-Text outputs. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association* (Antwerp, Belgium, 2007), Interspeech'07, ISCA, pp. 2441–2444. 4.2.3

[26] DEVEAUD, R., BONNEFOY, L., AND BELLOT, P. Quantification et identification des concepts implicites d'une requête. In *Conférence en Recherche d'Infomations et Applications - Proceedings of the 10th French Information Retrieval Conference* (2013), CORIA'13, UNINE, pp. 341–356. 2.3.1

[27] DEVEAUD, R., SANJUAN, E., AND BELLOT, P. Are Semantically Coherent Topic Models Useful for Ad Hoc Information Retrieval ? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (Sofia, Bulgaria, 2013), vol. 2 of *ACL'13*, ACL, pp. 148–152. 2.3.3

[28] DYER, M. G. *In-Depth Understanding : A Computer Model of Integrated Processing for Narrative Comprehension*. MIT Press, Cambridge, MA, USA, 1983. 2.2

[29] EVERT, S. A Lightweight and Efficient Tool for Cleaning Web Pages. In *Proceedings of the 6th International Conference on Language Resources and Evaluation* (Marrakech, Morocco, 2008), LREC'08, European Language Resources Association (ELRA). 2.3.1, 4.2.1.1

[30] FAESSEL, N. *Indexation et interrogation de pages Web décomposées en blocs visuels*. PhD thesis, Université Paul Cézanne, LSIS/CNRS, 2011. 4.1

[31] FALCO, M.-H., MORICEAU, V., AND VILNAT, A. Kitten : a tool for normalizing HTML and extracting its textual content. In *Proceedings of the 8th International Conference on Language Resources and Evaluation* (Istanbul, Turkey, 2012), LREC'12, European Language Resources Association (ELRA). 4.2.1, 4.2.1.1

[32] FERRET, O., GRAU, B., HURAULT-PLANTET, M., ILLOUZ, G., AND JACQUEMIN, C. Quand la réponse se trouve dans un grand corpus. *Ingénierie des Systèmes d'Information 7*, 1-2 (2002), 95–123. 2.3.3

[33] FERRUCCI, D. A., BROWN, E. W., CHU-CARROLL, J., FAN, J., GONDEK, D., KALYANPUR, A., LALLY, A., MURDOCK, J. W., NYBERG, E., PRAGER, J. M., SCHLAEFER, N., AND WELTY, C. A. Building Watson : An Overview of the DeepQA Project. *AI Magazine 31*, 3 (2010), 59–79. 2.2

[34] FEYNMAN, R. P. *« What Is and What Should Be the Role of Scientific Culture in Modern Society » (talk delivered at the « Galileo Symposium », 1964). The Pleasure of Finding Things Out : The Best Short Works of Richard P. Feynman*. Perseus Publishing, New York, 1999. pp. 97–115. 5.1

[35] GAIMAN, N. *The Books of Magic : The Invisible Labyrinth*. DC Comics, Burbank, CA, USA, 1990–1991. 4.1

[36] GALIBERT, O. *Approches et méthodologies pour la réponse automatique à des questions adaptées à un cadre intéractif en domaine ouvert*. PhD thesis, Paris-Sud, LIMSI/CNRS, 2009. 1.2.3, 2.3.1, 2.3.2, 2.3.3, 2.3.4, 2.3.5, 2.4, 2.4.1, 17, 3.3

[37] GANESH, S., AND VARMA, V. Exploiting the Use of Prior Probabilities for Passage Retrieval in Question Answering. In *Recent Advances in Natural Language Processing* (Borovets, Bulgaria, 2009), RANLP'09, ACL, pp. 99–102. 2.3.3, 3.1

[38] GARCIA-FERNANDEZ, A. *Génération de réponses en langue naturelle orales et écrites pour les systèmes de question-réponse en domaine ouvert.* PhD thesis, Paris-Sud, LIMSI/CNRS, 2010. 2.3.4

[39] GENTZEN, G. *The collected papers of Gerhard Gentzen*. Szabo edition, Amsterdam, NH, Netherlands, 1969, ch. Investigations into logical deduction, pp. 68–131. 2.3.1

[40] GENTZEN, G. *The collected works of Gerhard Gentzen*. Szabo edition, Amsterdam, NH, Netherlands, 1969, ch. The consistency of elementary number theory, pp. 132–213. 2.3.1

[41] GRAPPY, A. *Validation de réponses dans un système de question réponses.* PhD thesis, Paris-Sud, LIMSI/CNRS, 2011. 2.3.1

[42] GRAPPY, A., GRAU, B., FALCO, M.-H., LIGOZAT, A.-L., ROBBA, I., AND VILNAT, A. Selecting Answers to Questions from Web Documents by a Robust Validation Process. In *Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence* (Lyon, France, 2011), WI'11, IEEE Computer Society, pp. 55–62. 2.3.1, 2.3.1

[43] GRAU, B. Systèmes de question-réponse. In *Méthodes Avancées pour les Systèmes de Recherche d'Informations*. Hermès, New Castle, PA, USA, 2004, ch. 10, pp. 189–218. Dir. M. Ihadjadene. 2.3.1, 2.3.2, 2.3.4

[44] GRAU, B., SABAH, G., AND VILNAT, A. Pragmatique et dialogue homme-machine. *Technique et Science Informatiques (1) 13* (1994), 9–30. 2.2

[45] GREEN, B. F., WOLF, A. K., CHOMSKY, C., AND LAUGHERY, K. BASEBALL : an automatic question answerer. In *Readings in Natural Language Processing*, K. S.-J. B. J. Grosz and B. L. Webber, Eds. Morgan Kaufmann, Los Altos, CA, USA, 1961, pp. 545–549. 2.2

[46] GULLI, A., AND SIGNORINI, A. The indexable web is more than 11.5 billion pages. In *Proceedings of the 14th international conference on World Wide Web* (Chiba, Japan, 2005), WWW'05, ACM, pp. 902–903. 1.1

[47] GUO, H., MAHMUD, J., BORODIN, Y., STENT, A., AND RAMAKRISHNAN, I. A General Approach for Partitioning Web Page Content Based on Geometric and Style Information. In *Proceedings of the 9th International Conference on Document Analysis and Recognition* (Curitiba, Paraná, Brazil, 2007), vol. 2 of *ICDAR'07*, IEEE Computer Society, pp. 929–933. 4.1

[48] HALLIDAY, M. A., AND HASAN, R. *Cohesion in English*. Longman, London, 1976. 4.1, 9

[49] HARABAGIU, S. M., MOLDOVAN, D. I., CLARK, C., BOWDEN, M., HICKL, A., AND WANG, P. Employing Two Question Answering Systems in TREC 2005. In *Text REtrieval Conference* (Gaithersburg, Maryland, USA, 2005), vol. Special Publication 500-266 of *TREC14*, National Institute of Standards and Technology (NIST). 2.3.5

[50] HARABAGIU, S. M., MOLDOVAN, D. I., CLARK, C., BOWDEN, M., WILLIAMS, J., AND BENSLEY, J. Answer Mining by Combining Extraction Techniques with Abductive Reasoning. In *Text REtrieval Conference* (Gaithersburg, Maryland, USA, 2003), TREC12, National Institute of Standards and Technology (NIST), pp. 375–382. 2.3.5

[51] HATCHER, E., AND GOSPODNETIC, O. *Lucene in Action*. Manning, Greenwich, CT, USA, 2004. 2.3.3

[52] HEARST, M. A. Multi-Paragraph Segmentation of Expository Text. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics* (Las Cruces, New Mexico, USA, 1994), ACL'94, Morgan Kaufmann / ACL, pp. 9–16. 9

[53] HEARST, M. A. TextTiling : Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics 23*, 1 (1997), 33–64. 2.3.3, 4.1, 4.2.2.1, 10, 4.5

[54] HICKL, A., ROBERTS, K., RINK, B., BENSLEY, J., JUNGEN, T., SHI, Y., AND WILLIAMS, J. Question Answering with LCC's CHAUCER-2 at TREC 2007. In *Text REtrieval Conference* (Gaithersburg, Maryland, USA, 2007), TREC16, National Institute of Standards and Technology (NIST). 2.3.5

[55] HIGASHINAKA, R., AND ISOZAKI, H. Automatically Acquiring Causal Expression Patterns from Relation-annotated Corpora to Improve Question Answering for why-Questions. *ACM Transaction on Asian Language Information Processing 7*, 2 (2008), 6 :1–6 :29. 2.3.1

[56] HONG, D., AND SI, L. Mixture model with multiple centralized retrieval algorithms for result merging in federated search. In *Proceedings of the 35th International ACM SIGIR conference on Research and Development in Information Retrieval* (Portland, OR, USA, 2012), SIGIR'12, ACM, pp. 821–830. 4.5

[57] HORI, C., HORI, T., TSUKADA, H., ISOZAKI, H., SASAKI, Y., AND MAEDA, E. Spoken Interactive ODQA System : SPIQA. In *Companion volume to the proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (Sapporo, Japan, 2003), ACL, pp. 153–156. 2.2

[58] HOVY, E., GERBER, L., HERMJAKOB, U., LIN, C.-Y., AND RAVICHANDRAN, D. Toward Semantics-based Answer Pinpointing. In *Proceedings of the 1st International Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (San Diego, CA, USA, 2001), HLT'01, ACM, pp. 1–7. 2.3.2

[59] HOVY, E. H., GERBER, L., HERMJAKOB, U., JUNK, M., AND LIN, C.-Y. Question Answering in Webclopedia. In *Text REtrieval conference* (Gaithersburg, MD, USA, 2000), TREC9, National Institute of Standards and Technology (NIST). 2.3.2, 2.3.3, 2.3.4

[60] HUANG, H.-H., CHANG, K.-C., AND CHEN, H.-H. Modeling Human Inference Process for Textual Entailment Recognition. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (Sofia, Bulgaria, 2013), vol. 2 of *ACL'13*, ACL, pp. 446–450. 2.3.1

[61] ITTYCHERIAH, A., FRANZ, M., AND ROUKOS, S. IBM's Statistical Question Answering System - TREC-10. In *Text REtrieval Conference* (Gaithersburg, MD, USA, 2001), TREC10, National Institute of Standards and Technology (NIST). 2.3.2

[62] JACQUEMIN, C. Syntagmatic and Paradigmatic Representations of Term Variation. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics* (College Park, MA, USA, 1999), ACL'99, ACL. 2.3.3

[63] JELINEK, F., MERIALDO, B., ROUKOS, S., AND I, M. S. *Readings in Speech Recognition*. Morgan Kaufmann, Burlington, MA, USA, 1990, ch. Self-organized language modeling for speech recognition, pp. 450–506. 1.3.3, 2.3.3

[64] JIANG, J., AND CONRATH, D. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics* (Tapei, Taiwan, 1997), ROCLING X, ACM, pp. 19–33. 2.3.4

[65] JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation 28*, 1 (1972). 1.3.2, 2.3.3

[66] JUÁREZ-GONZALEZ, A., TÉLLEZ-VALERO, A., DENICIA-CARRAL, C., Y GÓMEZ, M. M., NOR PINEDA, L. V., AND LENGUAJE, L. D. T. D. INAOE at CLEF 2006 : Experiments in Spanish Question Answering. In *Evaluation of Multilingual and Multi-modal Information Retrieval, 7th Workshop of the Cross-Language Evaluation Forum* (Alicante, Spain, 2006), CLEF'06, Springer. 2.3.1

[67] KANT, I. *Critique de la faculté de juger [Kritik der Urth eilskraft, 1790]*. Gallimard, Paris, 1985. pp. 845–1299 (Œuvres philosophiques, tome II, La Pleiade). 6.1

[68] KAPLAN, S. J. Designing a Portable Natural Language Database Query System. *ACM Transactions on Database Systems 9*, 1 (1984), 1–19. 2.2

[69] KATZ, B. Using English for Indexing and Retrieving. In *Proceedings of the 2nd International Conference in Computer-Assisted IR* (Cambridge, MA, USA, 1988), RIAO'88, CID, pp. 313–333. 1.3.1

[70] KATZ, B. Annotating the World Wide Web using Natural Language. In *Proceedings of the 5th International Conference in Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications)* (Montreal, Canada, 1997), RIAO'97, CID, pp. 136–159. 1.3.1

[71] KATZ, B., MARTON, G., BORCHARDT, G. C., BROWNELL, A., FELSHIN, S., LORETO, D., LOUIS-ROSENBERG, J., LU, B., MORA, F., STILLER, S., UZUNER, O., AND WILCOX, A. External Knowledge Sources for Question Answering. In *Text REtrieval conference* (Gaithersburg, MD, USA, 2005), vol. Special Publication 500-266, National Institute of Standards and Technology (NIST). 2.2

[72] KENDALL, M. G. A New Measure of Rank Correlation. *Biometrika 30*, 1/2 (1938), 81–93. 2.3.5

[73] KHALID, M. A., AND VERBERNE, S. Passage retrieval for question answering using sliding windows. In *Coling 2008 : Proceedings of the 22nd international workshop on Information Retrieval for Question Answering* (Manchester, UK, 2008), IRQA'08, ACL, pp. 26–33. 2.3.3, 4.1, 4.5

[74] KOHLSCHÜTTER, C., FANKHAUSER, P., AND NEJDL, W. Boilerplate Detection using Shallow Text Features. In *Proceedings of 3rd ACM International Conference on Web Search and Data Mining* (New York City, NY, USA, 2010), WSDM'10, ACM. 2.3.1, 4.2.1.1

[75] KOVACEVIC1, M., DILIGENTI, M., GORI, M., AND MILUTINOVIC1, V. Visual Adjacency Multigraphs : a Novel Approach for a Web Page Classification. In *Proceedings of the Workshop on Statistical Approaches to Web Mining* (Pisa, Italy, 2004), SAWM'04, pp. 38–49. 4.1

[76] KRIPKE, S. Naming and necessity. In *Semantics of natural language*, D. Davidson and G. Harman, Eds., Synthese Library. Reidel, Dordrecht, SH, Netherlands, 1972, pp. 253–355. 2.3.1

[77] KULLBACK, S. *Information Theory and Statistics*. Wiley, New York, 1959. 2.3.3

[78] KULLBACK, S., AND LEIBLER, R. A. On Information and Sufficiency. *Annals of Mathematical Statistics 22*, 1 (1951), 79–86. 1.3.3, 2.3.3

[79] KWOK, C., ETZIONI, O., AND WELD, D. S. Scaling question answering to the web. *Transactions on Information Systems 19*, 3 (2001), 242–262. 2.3.1, 2.3.2, 2.3.3

[80] LAFFERTY, J. D., AND ZHAI, C. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, Louisiana, USA, 2001), SIGIR'01, ACM, pp. 111–119. 2.3.3, 3.1

[81] LAURENT, D., SÉGUÉLA, P., AND NÈGRE, S. Cross Lingual Question Answering using QRISTAL for CLEF 2006. In *Evaluation of Multilingual and Multi-modal Information Retrieval, 7th Workshop of the Cross-Language Evaluation Forum* (Alicante, Spain, 2006), CLEF'06, Springer. 2.3.1, 2.3.3

[82] LENAT, D. B. CYC : a large-scale investment in knowledge infrastructure. *Communications of the ACM 38*, 11 (1995), 33–38. 2.2

[83] LENHERT, W. Human and computational question answering. *Cognitive Sciences 1*, 1 (1977), 47–63. 2.2, 2.3.2, 2.1

[84] LEVENSHTEIN, V. I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady 10* (1966), 707–710. 2.3.1

[85] LI, X., AND ROTH, D. Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics* (Taipei, Taiwan, 2002), T.-E. Chen and Y.-F. Liu, Eds., COLING'02, Morgan-Kaufman, pp. 556–562. 2.3.2

[86] LI, X., AND ROTH, D. Learning question classifiers : the role of semantic information. *Natural Language Engineering 12*, 3 (2006), 229–249. 2.3.2

[87] LIGOZAT, A. L. *Exploitation et fusion de connaissances locales pour la recherche d'informations précises*. PhD thesis, Paris-Sud, LIMSI/CNRS, 2006. 2.3.1, 2.3.2

[88] LINDSEY, R. Inferential Memory as the Basis of Machines which Understand Natural Language. In *Computers and Thought*, E. Feigenbaum and J. Feldman, Eds. McGraw Hill, New York, 1963, pp. 217–233. 2.2

[89] LIU, Q., AGICHTEIN, E., DROR, G., MAAREK, Y., AND SZPEKTOR, I. When web search fails, searchers become askers : understanding the transition. In *Proceedings of the 35th International ACM SIGIR conference on Research and Development in Information Retrieval* (Portland, OR, USA, 2012), SIGIR'12, ACM, pp. 801–810. 1.3.2

[90] LIU, X., AND CROFT, W. B. Passage Retrieval Based On Language Models. In *Proceedings of the 11th International Conference on Information and Knowledge Management* (McLean, VA, USA, 2002), CIKM'02, ACM, pp. 375–382. 3.1

[91] MAGNINI, B., NEGRI, M., PREVETE, R., AND TANEV, H. Is is the Right Answer ? Exploiting web redundancy for answer validation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (Philadelphia, PA, USA, 2002), ACL'02, ACL, pp. 425–432. 2.3.3

[92] MANNING, C. D., AND SCHÜTZE, H. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA, 1999. 1.2.1

[93] MARSH, E., AND PERZANOWSKI, D. MUC-7 Evaluation of IE Technology : Overview of Results. In *Proceedings of the 7th Message Understanding Conference* (Fairfax, VA, USA, 1998), MUC-7, ACL. 2.2

[94] MCCOY, K. F. Correcting misconceptions : What to say when the user is mistaken. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, Massachusetts, USA, 1983), CHI'83, ACM, pp. 197–201. 2.2

[95] MCKEOWN, K. R. *Text generation : using discourse strategies and focus constraints to generate natural language text*. Cambridge University Press, New York, NY, USA, 1985. 2.2

[96] MCNEMAR, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika 12*, 2 (1947), 153–157. 3.4.3, 4.4.2, 4.4.2

[97] MEUNIER, J.-L. Optimized XY-Cut for Determining a Page Reading Order. In *Proceedings of the 8th International Conference on Document Analysis and Recognition* (Seoul, Korea, 2005), ICDAR'05, IEEE Computer Society, pp. 347–351. 4.1

[98] MILLER, G. WordNet : A Lexical Database for English. *Communications of the ACM 38*, 11 (1995), 39–41. 2.2

[99] MOLDOVAN, D., HARABAGIU, S., CLARK, C., BOWDEN, M., LEHMANN, J., AND WILLIAMS, J. Experiments and Analysis of LCC's two QA Systems over TREC 2004. 2004. 2.3.5

[100] MOLDOVAN, D., HARABAGIU, S., GIRJU, R., MORARESCU, P., LACATUSU, F., NOVISCHI, A., BADULESCU, A., AND BOLOHAN, O. LCC Tools for Question Answering. In *Text REtrieval Conference* (Gaithersburg, Maryland, USA, 2002), TREC11, Department of Commerce, National Institute of. 2.3.5

[101] MOLDOVAN, D., HARABAGIU, S., HARABAGIU, A., PASCA, M., MIHALCEA, R., GIRJU, R., GOODRUM, R., RUS, V., AND BACKGROUND, I. The Structure and Performance of an Open-Domain Question Answering System. In *In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics* (Hong Kong, China, 2000), ACL'00, ACL, pp. 563–570. 2.3.2, 2.3.3

[102] MOLDOVAN, D. I., BOWDEN, M., AND TATU, M. A Temporally-Enhanced PowerAnswer in TREC 2006. In *Text REtrieval Conference* (2006), vol. Special Publication 500-272 of *TREC15*, National Institute of Standards and Technology (NIST). 2.3.5

[103] MOLDOVAN, D. I., CLARK, C., AND BOWDEN, M. Lymba's PowerAnswer 4 in TREC 2007. In *Text REtrieval Conference* (Gaithersburg, Maryland, USA, 2007), National Institute of Standards and Technology (NIST). 2.3.1

[104] MOOERS, C. N. *Application of Random Codes to the Gathering of Statistical Information*. PhD thesis, MIT, 1948. 1.1

[105] MOOERS, C. N. Zatocoding applied to mechanical organization of knowledge. *American Documentation 2*, 1 (1951), 20–32. 2.2

[106] MORICEAU, V., AND TANNIER, X. FIDJI : using syntax for validating answers in multiple documents. *Information Retrieval 13*, 5 (2010), 507–533. 2.3.1, 2.3.3, 2.3.4, 4.2.1.1

[107] MORRIS, J., AND HIRST, G. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. *Computational Linguistics 17*, 1 (1991), 21–48. 9

[108] NAGY, G., AND SETH, S. Hierarchical representation of optically scanned documents. *Proceedings of International Conference on Pattern Recognition 1* (1984), 347–349. 4.1

[109] NAS, A. P., RODRIGO, A., SAMA, V., AND VERDEJO, F. Overview of the Answer Validation Exercise 2006. In *Proceeding of the Evaluation of Multilingual and Multi-modal Information Retrieval, 7th Workshop of the Cross-Language Evaluation Forum* (Alicante, Spain, 2006), CLEF'06, Springer, pp. 257–264. 2.3.1

[110] NICOLLE, A., PIERREL, J.-M., ROMARY, L., VILNAT, A., SABAH, G., AND VIVIER, J. Quels processus pour les dialogues homme-machine ? In *Machine, langage et dialogue*. l'Harmattan, Paris, 1998, ch. 4, p. 129. 2.2

[111] O'GORMAN, L. The Document Spectrum for Page Layout Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 11 (1993), 1162–1173. 4.1

[112] PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. The PageRank Citation Ranking : Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999. 1.3.2

[113] PEÑAS, A., RODRIGO, A., AND VERDEJO, F. Overview of the Answer Validation Exercise 2007. In *Advances in Multilingual and Multimodal Information Retrieval*. Springer-Verlag, Berlin, Heidelberg, 2008, pp. 237–248. 2.3.1

[114] POLLAK, B. *Functional Semantic Analysis of Web Pages on the Visual Layer*. PhD thesis, Vienna University of Thechnology, 2007. 4.1

[115] POMIKÁLEK, J. *Removing Boilerplate and Duplicate Content From Web Corpora*. PhD thesis, Masaryk University, Faculty of Informatics, 2011. 2.3.1

[116] PRAGER, J., BROWN, E., CODEN, A., AND RADEV, D. Question-answering by predictive annotation. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (Athens, Greece, 2000), SIGIR'00, ACM, pp. 184–191. 2.3.3

[117] PRAGER, J. M., BROWN, E. W., RADEV, D. R., AND CZUBA, K. One Search Engine or Two for Question-Answering. In *Text REtrieval Conference* (Gaithersburg, Maryland, 2000), TREC9, National Institute of Standards and Technology (NIST). 2.3.2

[118] PRAGER, J. M., CHU-CARROLL, J., CZUBA, K., WELTY, C. A., ITTYCHERIAH, A., AND MAHINDRU, R. IBM's PIQUANT in TREC2003. In *Text REtrieval Conference* (Gaithersburg, MD, USA, 2003), TREC12, National Institute of Standards and Technology (NIST), pp. 283–292. 2.2

[119] QI, X., AND DAVISON, B. D. Web page classification : Features and algorithms. *ACM Computing Surveys 41*, 2 (2009), 12 :1–12 :31. 4.1

[120] QUARTERONI, S. *Advanced Techniques for Personalized, Interactive Question Answering*. PhD thesis, University of York, Department of Computer Science, 2007. 2.3.2, 2.3.4

[121] QUARTERONI, S., AND MANANDHAR, S. Designing an Interactive Open-domain Question Answering System. *Natural Language Engineering 15* (2009), 73–95. 1.3.2, 2.3.1, 2.3.2, 2.3.3

[122] QUINTARD, L., GALIBERT, O., ADDA, G., GRAU, B., LAURENT, D., MORICEAU, V., ROSSET, S., TANNIER, X., AND VILNAT, A. Question Answering on Web Data : The QA Evaluation in Quæro. In *Proceedings of the 7th International Conference on Language Resources and Evaluation* (Valletta, Malta, 2010), LREC'10, European Language Resources Association (ELRA). 3.4.1

[123] RAPHAEL, B. SIR : A computer program for semantic information retrieval. In *Semantic Information Processing*, M. Minsky, Ed. MIT Press, Cambridge, MA, USA, 1968, pp. 33–145. 2.2

[124] ROBERTS, I., AND GAIZAUSKAS, R. J. Evaluating Passage Retrieval Approaches for Question Answering. In *Advances in Information Retrieval, 2th European Conference on IR Research* (Sunderland, UK, 2004), ECIR'04, Springer, pp. 72–84. 2.3.3, 4.1

[125] ROBERTSON, S. E., AND HANCOCK-BEAULIEU, M. On the Evaluation of IR Systems. *Information Processing and Management 28*, 4 (1992), 457–466. 2.3.3

[126] ROBERTSON, S. E., WALKER, S., AND HANCOCK-BEAULIEU, M. Experimentation as a way of life : Okapi at TREC. *Information Processing and Management 36*, 1 (2000), 95–108. 1.3.3

[127] RODRIGO, A., PEÑAS, A., AND VERDEJO, F. Overview of the answer validation exercise 2008. In *Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access* (Aarhus, Denmark, 2009), CLEF'08, Springer-Verlag, pp. 296–313. 2.3.1

[128] ROSENFELD, R. Two decades of statistical language modeling : Where do we go from here. *Proceedings of the IEEE 88*, 8 (2000), 1270–1278. 2.3.3

[129] SAKAI, T., KANDO, N., LIN, C.-J., MITAMURA, T., SHIMA, H., JI, D., CHEN, K.-H., AND NYBERG, E. Overview of the NTCIR-7 ACLIA IR4QA Task. In *In Proceedings of NTCIR-7 Workshop Meeting* (Tokyo, Japan, 2008), NTCIR-7. 1.3.2

[130] SALTON, G., SINGHAL, A., BUCKLEY, C., AND MITRA, M. Automatic text decomposition using text segments and text themes. In *Proceedings of the the 7th ACM conference on Hypertext* (Washington DC, USA, 1996), HYPERTEXT'96, ACM, pp. 53–65. 2.3.3, 4.1, 4.5

[131] SAPORTA, G. *Probabilité, Analyse des données et Statistique*, 2nd ed. Technip, Paris, Fance, 2006. 3.4.3.2

[132] SCHANK, R. C. Conceptual Dependency : A Theory of Natural Language Understanding. *Cognitive Psychology 3*, 4 (1972), 532–631. 2.2

[133] SCHANK, R. C., AND ABELSON, R. P. Scripts, plans, and knowledge. In *Proceedings of the 4th international joint conference on Artificial intelligence* (Tblisi, USSR, 1975), vol. 1 of *IJCAI'75*, Morgan Kaufmann Publishers Inc., pp. 151–157. 2.2

[134] SCHANK, R. C., AND ABELSON, R. P. *Scripts, Plans, Goals and Understanding : an Inquiry into Human Knowledge Structures*. L. Erlbaum, Hillsdale, NJ, USA, 1977. 2.2

[135] SCHANK, R. C., GOLDMAN, N., RIAGER, C. J., AND RISSBECK, C. Margie : memory, analysis, response generation, and inference on English. In *Proceedings of the 3rd international joint conference on Artificial intelligence* (Stanford, USA, 1973), IJCAI'73, Morgan Kaufmann, pp. 255–261. 2.2

[136] SEBASTIANI, F. Machine Learning in Automated Text Categorization. *ACM Computing Surveys 34*, 1 (2002), 1–47. 1.2.1

[137] SHAKESPEARE, W. *The Tragedy of Hamlet, Prince of Denmark*. Crofts, New York, 1946 (orig. 1599–1602). Act I, Scene 3, Lines 70–72. 3.1

[138] SHI, L., NIE, J.-Y., AND CAO, G. RALI Experiments in IR4QA at NTCIR-7. *Proceedings of the 7th NTCIR Workshop Meeting* (2008). 3.1

[139] SIMMONS, R. F. Synthex. *j-CACM 4*, 3 (1961), 140–140. 2.2

[140] SIMMONS, R. F. Answering English questions by computer : a survey. *Communications of the ACM 8*, 1 (1965), 53–70. 2.2

[141] SIMMONS, R. F. Semantic Networks : Their Computation and Use for Understanding English Sentences. In *Computer Models of Thought and Language*, R. C. Schank and K. M. Colby, Eds. W.H. Freeman and Co., San Francisco, 1973, pp. 61–113. 2.2

[142] SUNDBLAD, H. A Re-examination of Question Classification. In *Proceedings of the 16th Nordic Conference of Computational Linguistics* (Tartu, Estonia, 2007), NODALIDA'07, Kluwer Academic, pp. 394–397. 2.3.2

[143] SUNDHEIM, B. M. Overview of the third message understanding evaluation and conference. In *Proceedings of the 3rd Message Understanding Conference* (San Diego, CA, USA, 1991), MUC-3, ACL, pp. 3–16. 2.2

[144] SUNDHEIM, B. M., AND CHINCHOR, N. A. Survey of the Message Understanding Conferences. In *Proceedings of the workshop on Human Language Technology* (Princeton, New Jersey, USA, 1993), HLT'93, ACL, pp. 56–60. 2.2

[145] TATU, M., ILES, B., AND MOLDOVAN, D. I. Automatic Answer Validation Using COGEX. In *Evaluation of Multilingual and Multi-modal Information Retrieval, 7th Workshop of the Cross-Language Evaluation Forum* (Alicante, Spain, 2006), vol. 4730 of *CLEF'06*, Springer, pp. 494–501. 2.3.1, 2.3.2

[146] TIEDEMANN, J. Comparing Document Segmentation Strategies for Passage Retrieval in Question Answering. In *Proceedings of the Conference on Recent Advances in Natural Language Processing* (Borovets, Bulgaria, 2007), RANLP'07, RANLP 2007 Organising Committee. 2.3.3, 4.5

[147] TONEY, D., ROSSET, S., MAX, A., GALIBERT, O., AND BILINSKI, E. An Evaluation of Spoken and Textual Interaction in the RITEL Interactive Question Answering System. In *Proceedings of the 6th International Conference on Language Resources and Evaluation* (Marrakech, Morocco, 2008), LREC'08, European Language Resources Association (ELRA). 1.2.3, 2.4

[148] TURING, A. M. Computing machinery and intelligence. *Mind 59*, 236 (1950), 433–460. 2.2

[149] VADREVU, S., GELGI, F., AND DAVULCU, H. Semantic partitioning of web pages. In *Proceedings of the 6th international conference on Web Information Systems Engineering* (New York, NY, USA, 2005), WISE'05, Springer-Verlag, pp. 107–118. 4.1

[150] VOORHEES, E. M. The TREC-8 Question Answering Track Report. In *Text REtrieval Conference* (Gaithersburg, Maryland, USA, 1999), TREC8, National Institute of Standards and Technology (NIST). 2.3.5

[151] VOORHEES, E. M. Overview of the 16th Text REtrieval Conference. In *Text REtrieval Conference* (Gaithersburg, Maryland, USA, 2007), vol. Special Publication 500-274 of *TREC16*, National Institute of Standards and Technology (NIST). 2.3.5

[152] VOORHEES, E. M., AND HARMAN, D. Overview of the 9th Text REtrieval Conference. In *Text REtrieval Conference* (Gaithersburg, MD, USA, 2000), TREC9. 2.2

[153] WAHLSTER, W., MARBURGER, H., JAMESON, A., AND BUSEMANN, S. Over-Answering Yes-No Questions : Extended Responses in a NL Interface to a Vision System. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence* (Karlsruhe, West Germany, 1983), IJCAI'83, William Kaufmann, pp. 643–646. 2.2

[154] WEBBER, B. L. Questions, Answers and Responses : Interacting with Knowledge-Base Systems. In *On Knowledge Base Management Systems : Integrating Artificial Intelligence and Database Technologies* (Islamorada, FL, USA, 1985), Springer, pp. 366–402. 2.2

[155] WILENSKY, R. PAM - A Program That Infers Intentions. In *IJCAI* (Cambridge, MA, USA, 1977), William Kaufmann, p. 15. 2.2

[156] WINOGRAD, T. *Understanding Natural Language*. Academic Press, New York, NY, USA, 1972. 2.2

[157] WOODS, W. A. Progress in natural language understanding : an application to lunar geology. In *Proceedings of the national computer conference and exposition* (New York, 1973), AFIPS '73, ACM, pp. 441–450. 2.2

[158] WU, L., FALOUTSOS, C., SYCARA, K. P., AND PAYNE, T. R. FALCON : Feedback Adaptive Loop for Content-Based Retrieval. In *Proceedings of the 26th International Conference on Very Large Data Bases* (Cairo, Egypt, 2000), VLDB'00, Morgan Kaufmann Publishers Inc., pp. 297–306. 2.3.3

[159] XU, J., AND CROFT, W. B. Query Expansion Using Local and Global Document Analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Zurich, Switzerland, 1996), SIGIR'96, ACM, pp. 4–11. 2.3.1

[160] ZHAI, C., AND LAFFERTY, J. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems 22*, 2 (2004), 179–214. 1.3.3

[161] ZHANG, W., MING, Z., ZHANG, Y., NIE, L., LIU, T., AND CHUA, T.-S. The Use of Dependency Relation Graph to Enhance the Term Weighting in Question Retrieval. In *Proceedings of the 24th International Conference on Computational Linguistics : Technical Papers* (Mumbai, India, 2012), COLING'12, The COLING 2012 Organizing Committee, pp. 3105–3120. 2.3.3

Glossaire

A

ACLIA Advanced Cross-lingual Information Access 25

AFP Agence France Presse (journal d'information) 50

AP Associated Press (newswire) 50

AVE Answer Validation Evaluation 41

C

CCLQA Complex Cross-Lingual Question Answering 25

CLEF Conference and Labs of the Evaluation Forum http://www.clef-initiative.eu 41

CLIR Cross-Lingual Information Retrieval 26

CNA Central News Agency (newswire) 50

CSS Cascading Style Sheets www.w3.org/Style/CSS 86

D

DARPA Defense Advanced Research Projects Agency http://www.darpa.mil 34

DOM Document Object Model www.w3.org/DOM 86

domaine fermé Les systèmes de QR en domaine fermé répondent à des questions à sujet spécifique 32

domaine ouvert Les systèmes de QR en domaine ouvert répondent à des questions à sujet non spécifique 32

E

ETI Entreprises de Taille Intermédiaire 20

F

FAQ Foire Aux Questions (ou *Frequently Asked Questions* en anglais) 35

FBIS Foreign Broadcast Information Service (newswire) 50

FOL Langage du Premier Ordre (ou *First Order Logic* en anglais) 42

FR Federal Register (newswire) 50

FT Financial Times (newswire) 50

H

Html Hypertext markup langage http://www.w3.org/TR/REC-html40 39

I

IA Intelligence Artificielle 32

IDF Fréquence Inverse du Document (ou *Inverse Document Frequency* en anglais) 24

IMDB Internet Movie Database http://www.imdb.com 22, 34

L

LAT Los Angeles Times (newswire) 50

LCC Langage Computer Corporation http://www.langagecomputer.com 41

LDC Linguistic Data Consortium http://www.ldc.upenn.edu 50

LNE Laboratoire National de métrologie et d'Essais http://www.lne.fr 68

M

MUC Message Understanding Conference 34

N

NII National Institute of Informatics http://www.nii.ac.jp/en 144

NIST National Institute of Standards and Technology http://www.nist.gov 35, 50

NTCIR NII Test Collection for IR systems http://research.nii.ac.jp/ntcir/index-en.html 25

NYT New York Times (newswire) 50

O

OCR Optical Character Recognition 86

ODQA Open Domain Question Answering 34

P

PME Petites et Moyennes Entreprises 20

POS Parties du discours (ou *Part-Of-Speech* en anglais) 43

Q

QC Classification de questions (ou *Question Classification* en anglais) 33, 43

QR Questions-Réponses 21

R

RITEL Système de communication homme-machine du projet Ritel 21

Ritel Recherche d'information par téléphone 21, 144

S

SJMN San Jose Mercury News (newswire) 50

système de dialogue Un système de dialogue peut se voir comme un système de RI en modalité orale 51

T

TAL Traitement Automatique des Langues naturelles 20

TF Fréquence de Termes (ou *Term Frequency* en anglais) 24

TREC Text REtrieval Conference `http://trec.nist.gov` 35, 50

W

WSJ Wall Street Journal (newswire) 50

X

Xml Extensible markup langage `http://www.w3.org/XML` 39

XNA Xinhua News Agency (newswire) 50