Recherche de passages pertinents dans les fichiers logs par enrichissement de requêtes

Hassan Saneifar[*,**] — **Stéphane Bonniol**[**] — **Anne Laurent**[*] — **Pascal Poncelet**[*] — **Mathieu Roche**[*]

[*] LIRMM, Université Montpellier 2, CNRS, France
{saneifar, laurent, poncelet, mroche}@lirmm.fr
[**] Satin IP Technologies, France
stephane.bonniol@satin-ip.com

RÉSUMÉ. *Les systèmes de question réponse sont considérés comme la prochaine génération des moteurs de recherche. Notre article s'intéresse à la première étape d'un tel processus qui consiste à rechercher des passages pertinents possédant des réponses. Une telle tâche peut se révéler difficile en raison de la complexité des données traitées, des fichiers logs dans notre cas. Notre contribution repose sur un double enrichissement de requêtes primitives en utilisant une méthode d'apprentissage fondée sur la notion de "monde lexical", des connaissances morpho-syntaxiques et une nouvelle fonction de pondération des termes. Cette fonction originale a pour objectif d'attribuer un poids élevé aux termes susceptibles d'être corrélés au contexte de réponse. Les expérimentations menées sur des données réelles montrent que notre protocole d'enrichissement des requêtes permet d'extraire les passages pertinents par rapport aux questions.*

ABSTRACT. *The question answering systems are considered the next generation of search engines. This paper focuses on the first step of this process which is to search for relevant passages containing the responses. Such a task can be difficult because of the complexity of data, logs files in our case. Our contribution is based on enrichment of queries using a learning method based on the notion of "lexical world" and a novel term weighting function. This original weighting function, implemented within the enrichment process, aims to assign a high weight to terms that might be relative to the context of the answer. Experiments conducted on real data show that our protocol of primitive query enrichment allows to extract relevant passages.*

MOTS-CLÉS : *Recherche d'Information, Système question réponse, Recherche de passage, enrichissement de requête, apprentissage de contexte*

KEYWORDS: *Information Retrieval, Question Answering System, Passage Retrieval, Query Enrichment, Learning of Context*

1. Introduction

La recherche d'information a pour objectif de trouver les documents relatifs à un sujet spécifié par un utilisateur. Le sujet est normalement exprimé sous forme d'une liste de mots ou de termes spécifiques. Cependant, les besoins de certains domaines rendent les méthodes de Recherche d'Information (RI) inefficaces. En effet, lorsque l'objectif est de trouver des réponses précises, les systèmes de Recherche d'Information ne sont pas pertinents en raison du nombre considérable de documents retrouvés. En outre, toutes les informations se trouvant dans les documents ne sont pas toujours en corrélation avec la requête de l'utilisateur. C'est la raison pour laquelle les systèmes de Question Réponse (QR) représentent un enjeu important aujourd'hui.

Les systèmes de Question Réponse (QR) ont pour objectif de retrouver un fragment pertinent d'un document qui pourrait être considéré comme la meilleure réponse concise possible à une question de l'utilisateur. Il existe deux catégories principales de systèmes de QR : (1) domaine ouvert et (2) domaine restreint. Dans le premier cas, les questions se posent sur les domaines généraux et la source d'information sont de grands corpus constitués de documents de plusieurs domaines. En effet, le Web est la source la plus souvent utilisée dans les systèmes de QR en domaine ouvert. L'évaluation des systèmes QR en domaine ouvert a été intégrée depuis 1999 dans des campagnes d'évaluation américaines TREC (Text REtrieval Conference)[1].
Dans la deuxième catégorie, les systèmes de QR ont pour but de répondre à des questions dans un *domaine spécifique*. Les ressources sont également des documents *techniques* et *spécialisés*. Les domaines restreints ou autrement dit les domaines clos ont certaines caractéristiques qui restreignent l'application des méthodes de QR du domaine ouvert (Doan-Nguyen *et al.,* 2004). La faible fréquence d'apparition de réponse dans les ressources ou encore les particularités de certaines données textuelles spécialisées rendent la recherche de réponses plus difficile. Nous détaillons ces aspects dans la section 2.

La recherche de passages pertinents représente une phase importante des systèmes de QR. Le passage se définit comme une séquence de longueur fixe de mots commençant et se terminant n'importe où dans un document (Ofoghi *et al.,* 2006). La recherche de passages pertinents consiste à trouver des fragments de documents, susceptibles de contenir la réponse à une question. Les passages représentes un bon compromis entre une collection de documents et des réponses exactes.

Dans cet article, nous présentons nos travaux sur la recherche de passages dans un domaine spécialisé. Nous traitons un type particulier de données textuelles complexes : des fichiers logs générés par des outils de conception de circuits intégrés (Saneifar *et al.,* 2009a, Saneifar *et al.,* 2009b). Les logs représentent une source principale d'information sur la situation de conceptions, sur les produits ou même sur les causes des problèmes ayant pu se produire. Ces logs contiennent également les informations essentielles sur les conditions de production et sur les produits finaux. Dans ce do-

1. http ://trec.nist.gov/

maine, la vérification de la qualité des produits nécessite de répondre à un certain nombre de questions techniques et spécialisées. Nous avons pour objectif de trouver des segments des logs qui contiennent les réponses.

Nous proposons, dans cet article, un système de recherche de passage fondé sur une *nouvelle approche d'enrichissement de requêtes* initiales par un processus d'*apprentissage* et une *méthode de pondération des mots pertinents* du contexte. Le processus d'enrichissement s'effectue en deux phases. Premièrement, nous proposons une méthode d'apprentissage du contexte de questions, fondée sur la notion de "*monde lexical*" et l'utilisation de connaissances morpho-syntaxiques. Notre processus d'apprentissage permet d'enrichir la requête primitive en identifiant les termes représentatifs du contexte afin de les intégrer dans la requête. Deuxièmement, nous proposons une fonction originale de pondération des termes qui a pour objectif de donner un *poids élevé* aux termes qui ont une *probabilité importante* de faire partie des passages pertinents (recherchés). Les termes ayant les poids les plus élevés sont intégrés dans la requête afin de l'enrichir.
Notre approche est un système semi-automatique et interactif fondé sur les retours d'expérience des utilisateurs. Nous montrons que notre approche donne des résultats satisfaisants sur les *données réelles* avec une intervention moindre de l'utilisateur.

Dans la section 2 nous présentons les particularités des logs et les limites des systèmes QR en domaines restreints. Les travaux portant sur le domaine QR sont présentés dans la section 3. La section 4 développe notre approche de recherche de passage et l'enrichissement de requête. Les expérimentations sur des données réelles sont présentées en section 5.

2. Problématique

Les fichiers logs issus des outils de conception de circuits intégrés sont peu exploités de manière intelligente malgré le fait qu'ils contiennent des informations essentielles pour évaluer la qualité de la conception. Les caractéristiques particulières des données logs, décrites ci-après, rendent les techniques classiques de TAL (Traitement Automatique de Langue) et de Recherche d'Information inefficaces.

2.1. *Fichiers logs & RI*

Nous considérons les fichiers logs comme des données textuelles complexes car tout d'abord ils sont multi-sources. En effet, dans la conception des circuits intégrés, *plusieurs outils* de conception peuvent être utilisés. Malgré le fait que les logs issus du même niveau de conception contiennent *les mêmes informations*, les structures et le vocabulaire peuvent significativement différer en fonction de l'*outil* de conception utilisé. Plus précisément, pour *la même information*, *chaque outil* de conception possède souvent *son propre vocabulaire*. Pourtant les questions (*requêtes*) sont exprimées sous un seul format qui ne correspond pas forcement au vocabulaire propre à tous les

outils. Considérons le fragment "Capture the total fixed STD cell" comme une question de l'utilisateur. Nous faisons produire deux fichiers logs (*e.g.* log "A" et log "B") par deux outils différents. La réponse dans le log "A" s'exprime ainsi :

Std cell area: 77346 sites, (non-fixed:74974 fixed:2372)

Or, la réponse dans le log"B" est :

preplaced standard cell is: 24678

Tel que montré ci-dessus, la même information, dans deux fichiers logs produits par deux outils différents, est représentée par des structures et un vocabulaire différents. Les mots-clés de la question (i.e. Fixed, STD & cell) se trouvent dans la réponse extraite du log"A". Alors que, la réponse du log"B" ne contient que le mot "cell". Dans la mesure où il existe un dictionnaire qui associe le mot STD à standard, nous trouvons les deux mots "standard" et "cell". Pourtant en donnant ces deux mots dans une requête à un système de Recherche d'Information sur le log"B", des passages de log"B" contenant des phrases non pertinentes peuvent être retrouvées :

"standard cell seeds is : 4567"

"Total standard cell length = 0.4536".

Ceci s'explique par le fait que la question est exprimée en utilisant le vocabulaire du log"A". Autrement dit, pour une question donnée, les réponses trouvées dans les logs issus de certains outils ne contiennent pas nécessairement les mots-clés de la question. Par conséquent, la requête initiale peut être pertinente pour les logs générés par un outil mais non pertinente pour les logs issus d'un autre outil. Or, le partage des mots de question et leurs variantes syntaxiques entre la question et le passage est un facteur important pour évaluer la pertinence de passages. Les travaux fondés sur le partage des mots de la question et les passages sont détaillés dans la section 3.
De plus, la performance d'un système de QR dépend en grande partie de la redondance des occurrences de réponses dans le corpus (Brill *et al.,* 2001)(Lin, 2007). Les méthodes développées pour les systèmes de QR sont généralement fondées sur l'hypothèse qu'il existe plusieurs occurrences de réponses dans le corpus. Or, dans les fichiers logs issus des outils de la conception de circuits intégrés, les informations sont rarement répétées. Cela signifie que pour une question, il existe seulement un seul passage pertinent (contenant la réponse).

En outre, les outils de conception évoluent au cours du temps et cette évolution se produit souvent de manière imprévisible. Le format des données disponibles change, ce qui rend inefficace l'utilisation des méthodes statiques.
Par ailleurs, les méthodes de TAL et de RI développées pour les textes écrits en langue naturelle, ne sont pas forcement bien adaptées aux logs. En outre, beaucoup de notions dans ces fichiers sont des symboles, des abréviations, mots techniques ou mots alphanumériques, qui sont seulement compréhensibles en considérant la documentation du domaine.
De plus, le langage utilisé dans ces logs est une difficulté qui influence les méthodes d'Extraction d'Information. Bien que la langue utilisée dans ces logs soit l'anglais, les contenus de ces données textuelles n'en respectent pas la grammaire "classique".

Par conséquent, nous proposons l'enrichissement de requêtes initiales pour qu'elles soient pertinentes sur tous les types de logs et adaptées aux caractéristiques des logs. Dans le cadre de l'enrichissement, le système proposé apprend le contexte de la question en analysant la tendance d'apparitions des mots co-occurrents autour des mots-clés de la question. Ensuite, nous évaluons les mots qui pourraient être relatifs à la réponse grâce à notre nouvelle fonction de pondération de termes. La requête enrichie est finalement adaptée à l'hétérogénéité existant dans le corpus de logs.

2.2. *Recherche de passages dans les logs*

La phase d'extraction de passages influence significativement la performance des systèmes de QR car les réponses finales sont cherchées dans les passages retrouvés. La plupart des systèmes de QR, pour une question donnée, extraient un nombre important de passages susceptibles de contenir la réponse. Or, un point important dans un système de QR est de restreindre, le plus possible, le nombre de documents (*passages*) dans lesquels s'effectue la recherche de la réponse. Étant donné que nous nous situons dans un domaine très spécialisé, la précision élevée dans les réponses finales (c'est-à-dire le pourcentage de réponses correctes) est un enjeu très important. Cela signifie que le système d'extraction de passages doit classer le passage pertinent (*selon un score de pertinence*) dans les premières positions parmi les passages retrouvés.

3. Les travaux existants

La plupart des algorithmes de recherche de passages dépendent des occurrences des termes des requêtes à partir des documents textuels (Ofoghi *et al.,* 2006). Afin d'enrichir la requête, l'utilisation des variantes morphologiques et sémantiques des termes inclus dans la requête est étudiée dans (Chalendar *et al.,* 2002). Dans ce travail, la méthode utilisée reconnaît les termes extraits de la question dans les documents, ainsi que leurs variantes morphologiques, syntaxiques ou sémantiques.
La reformulation de questions (*c.-à-d. les schémas de surface ou les paraphrases*) est une méthode standard utilisée pour améliorer la performance des systèmes de QR. La technique est fondée sur l'identification de différentes manières pour exprimer la réponse donnée à une question en langage naturel (Kosseim *et al.,* 2008). Par exemple, pour une question du type "Qui a fondé la Croix-Rouge américaine ?", un système de QR fondé sur les schémas surfaciques cherche les reformulations comme "le fondateur de la Croix-Rouge américaine est X" ou comme "X, le fondateur de la Croix-Rouge américaine". La réécriture des questions utilisant des schémas de surfaces est exploitée aussi dans TREC9 et TREC10. (Mollá, 2006) et (Kosseim *et al.,* 2008) présentent différentes approches afin de prendre en compte les variantes sémantiques (les reformulations sémantiques) en complément des variantes syntaxiques. Afin de trouver des passages pertinents à une question, (Lamjiri *et al.,* 2007) évaluent chaque passage en utilisant une fonction de notation fondée sur la couverture des mots-clés de la requête qui se trouvent aussi dans le passage. Les systèmes de QR utilisent aussi des

techniques d'expansion de requêtes afin d'améliorer la performance. Ces méthodes peuvent utiliser les thésaurus (Jing *et al.,* 1994) ou être fondées sur l'incorporation des termes les plus fréquents dans les M documents pertinents.

Malgré les résultats satisfaisants obtenus par l'utilisation des schémas surfaciques et les variantes syntaxiques dans ces travaux, ces méthodes ne sont pas pertinentes dans le contexte des fichiers logs en raison des problèmes décrits dans la section 2. Les raisons principales de ces difficultés sont liées au fait qu'une réponse n'est pas reformulée de différentes manières dans un fichier logs et il existe un manque de redondance de réponses dans le corpus des logs. De plus, il existe plusieurs mots-clés techniques et alphanumériques dans les logs dont les variantes syntaxiques ou sémantiques se révèlent plus complexes à exploiter.

4. Extraction de passages fondée sur une méthode d'apprentissage et un enrichissement de requêtes

Nous proposons, dans cet article, une approche de recherche de passages fondée sur un nouveau processus interactif de l'enrichissement de requêtes par apprentissage et associé à un processus original de pondération (scoring) des termes du contexte. La figure 1 décrit le schéma du processus global de notre approche. Chaque étape de ce processus sera détaillée dans les sections suivantes. Notre protocole d'apprentissage

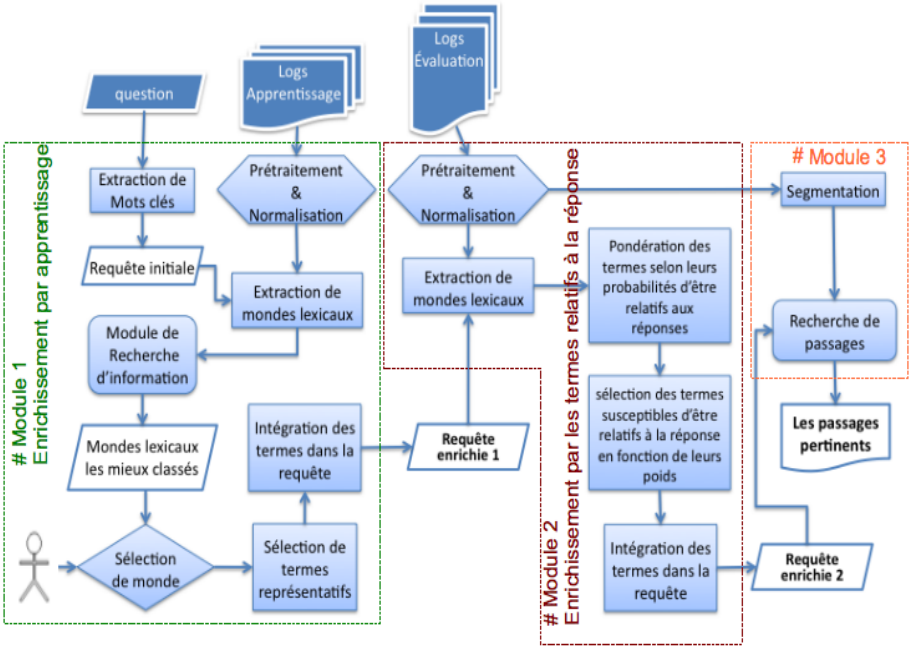

Figure 1 – Processus d'enrichissement de requête initiale

de contexte a pour but de mieux déterminer le contexte de la question en analysant les termes co-occurrents autour des mots-clés de la question. La nouvelle fonction de pondération des termes, proposée dans cet article, permet d'identifier les mots relatifs aux réponses. Dans ce système, le taux d'intervention des utilisateurs peut être variable et dépend de l'autonomie attendue du système. L'architecture de notre approche consiste en trois modules principaux :

– enrichissement de la requête initiale par apprentissage de contexte (mod. 1)

– enrichissement par les termes susceptibles d'être corrélés à la réponse (mod. 2)

– recherche de passages en utilisant la requête enrichie (mod. 3)

Le premier module enrichit la requête initiale (ci-après $R_{initiale}$) extraite d'une question en langue naturelle pour qu'elle soit pertinente sur les logs issus d'outils différents. Dans cette phase, en apprenant le contexte de la question, nous enrichissons $R_{initiale}$ par les termes les plus significatifs du contexte. Nous appelons, tout au long de cet article, la requête enrichie par ce module $R1_{appr}$.

Le deuxième module s'active pour un deuxième enrichissement de la requête afin d'obtenir une précision plus élevée. Cette phase consiste à pondérer des termes de passages. À cette fin, nous proposons un processus de notation des termes fondé sur une nouvelle fonction de pondération pour en choisir les plus significatifs afin de les intégrer dans la requête. La requête obtenue dans cette phase est appelée $R2_{pond}$.

Dans le troisième module, nous cherchons les passages pertinents dans les logs issues d'un outil différent de ceux utilisés dans la phase d'apprentissage. Les logs de ce corpus ont des structures et un vocabulaire significativement différents de ceux des logs du corpus d'apprentissage.

Dans notre approche, nous utilisons les contextes spécialisés, appelés "monde lexical", des mots qui se trouvent dans la question. Les connaissances terminologiques issues des logs sont également utilisées afin de caractériser et présenter de manière plus pertinente le contexte. Nous développons le concept de monde lexical et l'utilisation des connaissances terminologiques dans les sections suivantes.

4.1. *Monde lexical*

Selon (Léon, 2009), un monde lexical désigne les co-occurrences fréquentes d'une unité lexicale au sein d'une collection de textes. Nous définissons le monde lexical d'un terme comme le contexte dans lequel le terme se trouve. Les mots situés autour du terme ont généralement un lien sémantique et/ou contextuel. Par conséquent, en déterminant le monde lexical d'un terme dans un document, nous pouvons identifier les mots qui ont tendance à apparaître autour de celui-ci dans le document. Nous notons que pour nous, les tendances d'apparitions des termes ne sont pas forcement des tendances fréquentes par rapport aux particularités des logs.

De manière concrète, pour identifier le monde lexical d'un terme dans un document, nous cherchons d'abord un court fragment du document dans lequel le mot se trouve. Ensuite, afin de caractériser et présenter ce fragment, plusieurs solutions sont envisageables.
Nous caractérisons le monde lexical d'un mot, dans un premier temps, par un ensemble de "*mots*" (*appelé aussi sac de mots*) se trouvant autour du terme *et* qui font partie des classes morpho-syntaxiques "Nom", "Verbe" ou "Adjectif". Dans un second temps, le monde lexical d'un mot est présenté par différents syntagmes (Bourigault *et al.,* 2000) qui se trouvent dans le fragment dans lequel le mot se trouve. Nous détaillons cet aspect dans la section suivante.

4.2. *Connaissance terminologique*

Les mondes lexicaux peuvent être caractérisés de manière différente : par des "mots", des "termes" ou des "bigrammes". Selon nos expérimentations, les termes et les mots sont plus représentatifs. Nous créons donc deux types de mondes lexicaux : (1) constitués des mots et (2) constitués des termes (c.-à-d. syntagmes) ainsi que des mots. Afin d'extraire la terminologie des logs, nous utilisons la méthode présentée dans (Saneifar *et al.,* 2009b). Cette méthode adaptée aux spécificités des logs, permet d'extraire des termes selon des patrons syntaxiques dans les fichiers logs. Afin de choisir les termes pertinents et spécifiques au domaine, nous utilisons le protocole de validation et filtrage de terminologie présenté dans (Saneifar *et al.,* 2009a). Nous avons finalement des termes valides et pertinents du domaine pour caractériser des mondes lexicaux.

4.3. *Enrichissement de requêtes par apprentissage de contexte*

Nous détaillons ici le premier module de notre approche. Nous cherchons dans un premier temps à apprendre le contexte de la question et le caractériser par les mots les plus significatifs. Étant donné que ces mots représentent le contexte de la question, il est attendu que les passages correspondant à la question partagent ces mots malgré le fait que les passages soient issus de logs différents. Nous utilisons les mondes lexicaux des termes de la question pour identifier et déterminer le contexte visé par la question.

Nous extrayons d'abord les mots clés de la question par les méthodes classiques de traitement de questions. Il en résulte une liste de mots-clés que nous considérons comme la requête initiale. Pour chaque mot-clé, nous identifions son monde lexical dans le corpus de logs d'apprentissage. Cela signifie que nous cherchons le mot-clé dans le corpus et déterminons le court fragment dans lequel il se situe (c.-à-d. nous déterminons son monde lexical)[2]. Ensuite, nous caractérisons ce fragment (cf. section 4.1) pour finalement obtenir un ensemble de mondes lexicaux.

2. un mot-clé peut correspond à plusieurs mondes lexicaux selon son nombre d'occurrence dans le corpus.

A ce stade, nous cherchons à identifier les mondes lexicaux les plus adaptés à la question. Nous créons donc un système de Recherche d'Information consistant à mesurer la similarité entre les descripteurs de la requête et les mondes lexicaux extraits. Ce processus permettra d'identifier des mondes lexicaux corrélés à la requête initiale.

Nous utilisons un système de RI fondé sur le modèle vectoriel. Ainsi, chaque document (*ici chaque monde lexical*) est représenté comme un vecteur de valeurs où chaque valeur correspond à un terme significatif de documents. Pour calculer ces valeurs, également appelées le poids des termes, les systèmes de RI utilisent une fonction de pondération (indexation) qui attribue un indexe (un poids) aux descripteurs des mondes lexicaux (c.-à-d. les termes et les mots). Il existe plusieurs fonctions de pondération tels que la *représentation binaire*, la *fréquence de terme* (tf) ou *tf-idf* (Salton *et al.,* 1986). Selon les caractéristiques du corpus de logs et nos premières expériences, nous avons choisi la fonction *tf-idf* pour indexer des mondes lexicaux à ce stade. *tf-idf* est une fonction de pondération statistique qui évalue l'importance d'un mot pour un document dans un corpus. Le poids dépend du nombre d'occurrences des mots (tf) dans le document. Elle varie aussi en fonction de la fréquence des mots dans le corpus (idf). Le système de RI utilise également une mesure pour calculer la similarité de chaque monde lexical (présenté par le modèle vectoriel) avec le vecteur de la requête. Nous utilisons et comparons les mesures Cosinus et Jaccard dans nos expérimentations.

Les mondes sont ensuite classés selon leur degré de similarité (score) calculé par le système de RI. A ce stade de l'apprentissage, l'utilisateur intervient en choisissant parmi les m mondes *les mieux classés par le système*, le monde lexical le plus adapté à la question. Ainsi, le monde lexical choisi est considéré comme le *représentant* du *contexte* de la *question*. Nous précisons que d'après nos expérimentations sur les données réelles, le monde lexical pertinent se trouve toujours parmi les quatre mondes lexicaux les mieux classés par le système lorsqu'il existe environ 850 passages dans le corpus de logs (*chaque passage peut être potentiellement un monde lexical représentant le contexte de la question*).

Ensuite, dans le but d'enrichir la requête initiale par les mots les plus représentatifs du contexte, nous cherchons à identifier les termes les plus représentatifs du monde lexical choisi. Nous utilisons l'index des termes (obtenu par tf-idf) pour mesurer leur représentativité en fonction du contexte. Nous choisissons finalement les n termes ayant les scores tf-idf les plus élevés. Nous obtenons la requête enrichie $R1_{appr}$ en intégrant les termes sélectionnés dans la requête initiale $R_{initiale}$.

4.4. *Enrichissement par les termes susceptibles d'être corrélés à la réponse*

Pour améliorer la pertinence de la requête aux différents types de logs, nous envisageons une deuxième phase d'enrichissement de la requête. Dans ce module, nous avons pour entrée la requête enrichie par l'apprentissage $R1_{appr}$ et les logs du corpus d'évaluation dans lequel nous cherchons les passages pertinents (différent de celui utilisé lors de l'apprentissage).

Notre motivation est de trouver les termes dans les logs du corpus d'évaluation qui sont susceptibles d'exister dans le passage adapté et sont donc relatifs à la réponse. À cette fin, nous proposons ici un processus de sélection des termes fondé sur une nouvelle fonction de pondération (scoring). Cette fonction donne un score à chaque terme qui s'appuie sur trois hypothèses :

– les termes les plus corrélés à la réponse se trouvent dans un monde lexical représentant le contexte de la question ;

– la requête doit contenir des mots discriminants (non fréquents dans plusieurs documents (passages)) ;

– la plupart des termes de la requête partagent le passage pertinent.

En raison de la première hypothèse, pour chaque terme de la requête, nous cherchons leurs mondes lexicaux dans les logs du corpus d'évaluation. Nous précisons que le système n'a aucune information sur les logs d'évaluation et les passages pertinents[3]. Nous obtenons finalement un ensemble de mondes lexicaux correspondant aux termes de la requête qui sont extraits dans les logs du corpus d'évaluation.

La fonction de scoring est développée selon les deux dernières hypothèses. Nous cherchons d'abord à favoriser les mots discriminants. Autrement dit, nous cherchons les mots ayant une fréquence très faible dans différents mondes lexicaux. Pour cela, nous utilisons la fonction idf (*inverse document frequency*) en considérant chaque monde extrait dans l'étape précédente comme un document et l'ensemble des mondes lexicaux comme le corpus. De cette manière, nous favorisons les mots qui se trouvent dans un seul ou un très petit nombre de mondes lexicaux.

Ensuite, afin de favoriser les mots susceptibles de faire partie de passages pertinents nous donnons un score à chaque terme des mondes lexicaux en prenant en compte la troisième hypothèse. Pour un terme, ce score que nous appelons lwf (lexical world frequency) dépend du nombre de mots de la requête qui sont associés au monde lexical auquel le terme appartient. Cela présente une forme de df (*document frequency*) où un *documen*t correspond à *l'ensemble des mondes lexicaux* associés à *un mot de la requête*. En effet, par ce score, nous mesurons l'importance du monde lexical dans lequel le terme se trouve. Cette importance est calculée en fonction du nombre de mots de la requête qui sont associés au monde lexical.
La formule de lwf que nous proposons pour le terme i du monde lexical j se calcule ainsi :

$$lwf_{ij} = 1/\log(M/n_j)$$

Ici, M représente le nombre de mots de la requête et n_j représente le nombre de mots de la requête qui sont associés au monde lexical j. Le score final d'un terme que nous appelons TRQ (Term Relatedness to Query) se calcule selon la formule suivante :

$$TRQ = \alpha * (lwf) + (1 - \alpha) * idf$$

3. l'apprentissage de requête se fait sur les logs d'apprentissage qui sont issus d'un autre outil ayant le vocabulaire et des structures significativement différents.

Selon les expérimentations, la valeur de α la plus adaptée est 0.25. Cela signifie que nous donnons plus de poids à la valeur idf des termes et un poids plus modeste, mais qui influence les résultats finaux, à lwf.

Nous expliquons, avec un exemple, le processus de sélection des termes relatifs à la question. Supposons R=$\{mot_a, mot_b, mot_d\}$ comme une requête enrichie par l'apprentissage et $\log_b$ un fichier de logs qui contient 7 segments :
S_1=$\{mot_a\ mot_k\ mot_m\ mot_b\}$ S_2=$\{mot_d\ mot_k\}$ S_3=$\{mot_z\}$
S_4=$\{mot_a\ mot_c\ mot_e\ mot_q\}$ S_5=$\{mot_b\ mot_e\}$ S_6=$\{mot_z\}$
S_7=$\{mot_b\ mot_c\ mot_k\}$.
Si l'on considère que la taille des mondes lexicaux correspond à la taille des segments, en cherchant les mondes lexicaux associés aux termes de R, nous obtenons S_1, S_2, S_4, S_5, S_7 comme l'ensemble de mondes lexicaux. Le tableau ci-dessous montre les mondes lexicaux associés à chaque mot-clé de la question. Nous rappelons qu'étant donné qu'un mot peut être utilisé dans différents fragments du document, il peut être associé à plusieurs mondes lexicaux.

mot_a	S_1, S_4
mot_b	S_1, S_5, S_7
mot_d	S_2

Ici, par exemple, le mot mot_a de la requête R est associé à deux mondes lexicaux S_1 et S_4. La valeur de la mesure idf du mot mot_k, par exemple, est égale à $log(5/3) = 0,22$, car il existe dans trois mondes lexicaux (S_1, S_2 et S_7) parmi les cinq. Ainsi, la valeur de lwf pour le mot mot_k dans le segment S_1 se calcule ainsi : $lwf_{2,1} = 1/\log(3/2) = 5,8$. En effet, il existe deux mots de la requête R (mot_a, mot_b) qui sont associés au monde lexical S_1 (dans lequel le mot mot_k existe). Nous précisons que la valeur de lwf de chaque mot dépend du monde lexical dans lequel il est situé. Nous pouvons donc attribuer des valeurs lwf différentes à un terme en fonction des mondes différents. Par exemple, la valeur lwf du mot mot_k dans le segment S_2 est égale à $lwf_{2,2} = 1/\log(3/1) = 2,1$ car il existe un seul mot de R qui est associé à S_2. Cela signifie que mot_k situé dans le segment S_2 est moins significatif que lorsqu'il est situé dans le segment S_1.

Une fois le score TRQ des termes de tous les mondes lexicaux calculé, nous identifions les k scores les plus élevés et choisissons les termes associés. Pourtant, parmi les termes sélectionnés, il existe des termes qui ont le même score. Afin de distinguer ces termes, nous évaluons leurs tendance d'apparition proche des mots de la requête initiale. Autrement dit, nous cherchons à calculer leur dépendance aux mots de la requête dans le contexte. Pour cela, nous choisissons la mesure de Dice. Cette mesure statistique a un bon comportement pour des tâches de fouille de textes (Roche *et al.,* 2009). Pour un terme T et un mot de requête M, la mesure de Dice se calcule de la manière suivante :

$$Dice(T, M) = \frac{2 * \mid (T, M) \mid}{\mid T \mid + \mid M \mid}$$

Ici, $\mid (T, M) \mid$ correspond au nombre de fois où le terme M et le mot T se trouvent dans la même ligne sur le corpus de logs et $\mid x \mid$ correspond au nombre total d'occurrences de x dans le corpus de logs. Finalement, la valeur de la mesure de Dice permet de distinguer les scores égaux obtenus lors de l'étape précédente. Le score final de ces termes est obtenu par la somme du score de Dice et du score précédent. Nous précisons que nous sélectionnons d'abord les termes en fonction de leur score TRQ (c.-à-d. les termes ayant le score TRQ le plus élevé). Si nous avons les termes ayant le même score TRQ, nous les distinguons en calculant leur valeur de Dice.

Dans la dernière phase, le système ordonne les termes sélectionnés en fonction de leur score TRQ (considérant la valeur de Dice pour les termes ayant le même score TRQ) et recommande à l'utilisateur les k termes les mieux classés pour enrichir la requête. Le système fournit également l'intégration automatique des k premiers termes dans la requête dans le cas d'autonomie de système. La requête enrichie sera ensuite utilisée dans le module d'extraction de passage.

4.5. *Recherche de passages*

Nous détaillons, ici, le processus de recherche de passages pertinents sur le corpus de logs d'évaluation. Dans un premier temps, nous segmentons les logs du corpus d'évaluation. La segmentation est réalisée selon des structures des textes de fichiers logs : les blocs de données, les tableaux etc. Chaque segment est un passage de logs contenant potentiellement la réponse[4]. Ainsi, pour une question donnée, la recherche de passage consiste à rechercher le segment de logs qui contient la réponse.

Ensuite, nous enrichissons la requête initiale pour qu'elle soit adaptée à tous types de logs ayant des vocabulaires différents et ainsi améliorer la précision de recherche de passages.

Finalement, nous créons un système de Recherche d'Information pour trouver le passage pertinent. Le système de Recherche d'Informations dans cette phase, peut être construit en utilisant la fonction *tf-idf* ou la fonction binaire pour indexer les termes. Nous présentons la performance obtenue par chaque fonction et chaque mesure de similarité dans la section 5. Le système de RI attribue un score de pertinence à chaque passage (segment) en fonction de la requête. Ensuite, nous ordonnons les passages en fonction de leur score de pertinence et proposons les passages les mieux classés à l'utilisateur.

Plusieurs des méthodes d'extraction de passage retournent un nombre considérable de passages candidats. Nos expérimentations effectuées sur les données réelles affirment que dans plus de 80% des cas, le passage pertinent se trouve parmi les trois passages les mieux classés par notre approche lorsqu'il existe en moyenne 600 passages potentiels.

---

4. Nous utilisons, dans cet article, les mots "segment" et "passage" pour le même concept.

5. Expérimentations

Nous expérimentons la performance de notre approche sur un corpus de logs issus du monde réel et industriel. Le corpus contient des logs de cinq niveaux de conception de circuits intégrés (CI). Il existe 26 questions qui sont exprimées en langue naturelle et sont conçues par des grandes sociétés de conception de CI. Les logs sont segmentés en fonction de la structure propre des logs (lignes vides, tableaux, bloc de données, etc.). Chaque segment est un passage potentiellement pertinent – contenant la réponse à une question donnée. Nous précisons que pour une question donnée, il n'existe qu'un seul passage pertinent (*segment*) contenant la réponse.
Le corpus d'évaluation contient 625 segments. Chaque segment est constitué d'environ 150 mots. Le corpus d'apprentissage est constitué des logs de même niveaux de conception de CI mais issus d'un outil différent. Cela signifie que le vocabulaire et la structure (*donc les segments*) du corpus d'apprentissage sont différents de celui d'évaluation. Toutes les expérimentations sont effectuées avec une intervention très réduite de l'utilisateur et une autonomie majeure du système.

Les expérimentations sont réalisées dans deux directions principales :

– Évaluation de la performance de recherche de passage en utilisant les requêtes enrichies

– Comparaison de la performance de recherche de passage en utilisant les requêtes enrichies comparativement avec requêtes initiales (non enrichies).

Dans chaque catégorie, le système de RI est implémenté en utilisant différentes fonctions d'indexation et mesures de similarité. Finalement, pour évaluer la performance de notre approche dans ces conditions différentes, nous utilisons la *Moyenne des Réciproques du Rang* (MRR[5]) utilisée dans TREC comme la mesure d'évaluation de performance (Voorhees, 1999). Cette mesure tient compte du rang de la bonne réponse trouvée.

$$MRR = \frac{1}{nb(Question)} \sum_{i=1}^{nb(Question)} \frac{1}{rank(answer)}$$

5.1. *Performance de recherche de passage en utilisant les requêtes enrichies*

Ici, nous mesurons la performance de recherche de passage en utilisant les deux requêtes enrichies (c.-à.d. $R1_{appr}$ et $R2_{pond}$). Nous évaluons d'abord la performance du système en utilisant les requêtes enrichies $R1_{appr}$ qui sont obtenues à la fin de la phase d'apprentissage. Ensuite, nous appliquons le même protocole d'expérimentation, mais en utilisant les requêtes enrichies $R2_{pond}$ qui sont obtenues dans le deuxième module : l'enrichissement par des termes susceptibles d'être corrélés à la réponse.

5. Mean Reciprocal answer Rank

Nous présentons les résultats de la recherche de passage en utilisant les deux types de requêtes enrichies. De cette façon, nous pouvons confirmer si la deuxième phase d'enrichissement améliore encore la pertinence de la requête.

Dans la phase de recherche de passage, nous avons utilisé au sein du système de RI deux fonctions d'indexation différentes (tf-idf & présentation binaire) ainsi que deux différentes mesures de similarité (Cosinus & Jaccard). Le tableau 1a présente

	tf-idf		Binaire	
	Cos	Jac	Cos	Jac
$\%P(1)$	11	12	14	10
$\%P(2)$	8	5	3	4
$\%P(3)$	5	3	2	1
MRR	**0.65**	0.64	0.65	0.55

(a)

	tf-idf		Binaire	
	Cos	Jac	Cos	Jac
$\%P(1)$	20	18	19	13
$\%P(2)$	3	1	0	4
$\%P(3)$	1	3	2	1
MRR	**0.83**	0.78	0.79	0.65

(b)

Tableau 1 – Pourcentage de questions $\%\mathbf{P(n)}$ dont le passage pertinent est classé au rang "$\mathbf{n}$" parmi les passages retrouvés par le système **(a)** performance obtenue en utilisant les requêtes enrichies $\mathbf{R1_{appr}}$, **(b)** la performance obtenue en utilisant les requêtes enrichies $\mathbf{R2_{pond}}$.

les résultats de la performance de recherche de passage en utilisant la requête enrichie $R1_{appr}$. $\%P(n)$ correspond au pourcentage de questions pour lesquelles le passage pertinent est classé au rang n parmi les passages candidats retrouvés par le système. Ici, nous montrons les résultats pour les trois premiers rangs ($n = 1, 2, 3$).

Nous présentons dans le tableau 1b les résultats de la performance de recherche de passages en utilisant la requête enrichie $R2_{pond}$. Nous rappelons que la deuxième phase d'enrichissement suit la première phase (phase d'apprentissage), elle est ainsi considérée comme un re-enrichissement de la requête.

Selon les tableaux, le meilleur MRR obtenu par la requête enrichie dans la phase d'apprentissage est égal à 0.65. Alors qu'en re-enrichissant la requête dans le deuxième module d'enrichissement fondé sur la nouvelle fonction de pondération, nous obtenons une valeur de MRR égale à 0.83. Cela confirme que le ré-enrichissement de requêtes améliore significativement la pertinence des requêtes et donc la performance de recherche de passages.
Selon les résultats, dans la meilleure configuration du système, le passage pertinent est classé dans 76% des cas comme le premier passage parmi les passages retournés. Dans 92% des cas, le passage pertinent est classé parmi les trois premiers passages retournés par le système alors qu'il existe environ 650 passages dans le corpus.
Les résultats obtenus montrent également que la fonction de pondération *tf-idf* et la mesure de similarité "cosinus" dans le système de RI sont plus adaptées que d'autres fonctions ou mesures.

5.2. *Requête enrichie Vs. Requête initiale*

Dans cette section, nous comparons les résultats obtenus en utilisant les requêtes enrichies avec des résultats de recherche de passages en utilisant les requêtes initiales. Cela montre de quelle manière notre approche de double enrichissement de la requête améliore la pertinence des requêtes initiales.

Pour une question donnée, la requête initiale est obtenue en prenant les mots-clés de la question. Dans les expérimentations que nous présentons ci-dessous, nous utilisons les mêmes questions et le même corpus d'évaluation qui sont utilisés dans des expérimentations précédentes. Comme le montre le tableau 2a, en utilisant les requêtes

	tf-idf		Binaire	
	Cos	Jac	Cos	Jac
$\%P(1)$	9	8	11	8
$\%P(2)$	5	4	3	3
$\%P(3)$	4	3	2	2
MRR	0.51	0.50	**0.58**	0.48

(a)

	tf-idf		Binaire	
	Cos	Jac	Cos	Jac
$\%P(1)$	20	18	19	13
$\%P(2)$	3	1	0	4
$\%P(3)$	1	3	2	1
MRR	**0.83**	0.78	0.79	0.65

(b)

Tableau 2 – Pourcentage de questions $\%\mathbf{P}(\mathbf{n})$ dont le passage pertinent est classé au rang "**n**" parmi les passages retrouvés par le système **(a)** performance obtenue en utilisant les requêtes initiales (non enrichies), **(b)** performance obtenue en utilisant les requêtes enrichies $R2_{pond}$.

initiales, nous obtenons une valeur de MRR égale à 0.58 dans les meilleures conditions. Or, tout en enrichissant les requêtes initiales selon l'approche mentionnée dans cet article, MRR est amélioré de façon significative et atteint la valeur 0.83 dans les meilleures conditions. Ceci montre l'intérêt de l'approche développée dans cet article.

6. Conclusions et Perspectives

Nous avons présenté un processus de double enrichissement de requêtes primitives dans le cadre de recherche de passages pertinents dans les fichiers logs. L'hétérogénéité du vocabulaire, de la structure des fichiers logs et le fait que les mots utilisés dans les questions posées en langue naturelle n'existent pas forcement dans les logs rend la tâche difficile. Malgré ces caractéristiques particulières, notre approche permet d'adapter une requête initiale à tous les types de logs quel que soit le vocabulaire. Selon les résultats, nous avons obtenu une valeur MRR égale à 0.83 avec une moindre intervention d'utilisateurs alors que la valeur MRR est égale à 0.58 si l'on utilise les requêtes initiales pour rechercher les passages pertinents.

Nous envisageons d'évaluer notre système avec d'autres modèles de Recherche d'Information. L'amélioration de la fonction de pondération, utilisée dans la deuxième phase d'enrichissement de requêtes, représente un point majeur dans les perspectives à ce travail.

7. Bibliographie

Bourigault D., Jacquemin C., « Construction de ressources terminologiques », *Pierrel J.-M. éd. "Industrie des langues"*, Hermès, p. 215-233, 2000.

Brill E., Lin J., Banko M., Dumais S., Ng A., « Data-Intensive Question Answering », *In Proceedings of the Tenth Text REtrieval Conference (TREC)*, p. 393-400, 2001.

Chalendar G., Dalmas T., Elkateb-Gara F., Ferret O., Grau B., Hurault-Plantet M., Illouz G., Monceaux L., Robba I., Vilnat A., « The Question Answering System QALC at LIMSI, Experiments in Using Web and WordNet », *TREC*, 2002.

Doan-Nguyen H., Kosseim L., « The Problem of Precision on Restricted-Domain Question Answering », *Proceedings the ACL 2004 Workshop on Question Answering in Restricted Domains (ACL-2004)*, ACL- 2004, Barcelona, Spain, July, 2004.

Jing Y., Croft W. B., « An Association Thesaurus for Information Retrieval », *RIAO 94 : Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications)*, p. 146-160, 1994.

Kosseim L., Yousefi J., « Improving the performance of question answering with semantically equivalent answer patterns », *Data Knowl. Eng.*, vol. 66, n° 1, p. 53-67, 2008.

Lamjiri A. K., Dubuc J., Kosseim L., Bergler S., « Indexing Low Frequency Information for Answering Complex Questions », *RIAO 07 : 8th International Conference on Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications)*, Carnegie Mellon University, Pittsburgh, PA, USA, 2007.

Léon S., « Un système modulaire d'acquisition automatique de traductions à partir du Web », *TALN'09 : Traitement Automatique des Langues Naturelles*, 06, 2009.

Lin J., « An exploration of the principles underlying redundancy-based factoid question answering », *ACM Trans. Inf. Syst.*, vol. 25, n° 2, p. 6, 2007.

Mollá D., « Learning of Graph-based Question Answering Rules », *Proc. HLT/NAACL 2006 Workshop on Graph Algorithms for Natural Language Processing*, p. 37-44, 2006.

Ofoghi B., Yearwood J., Ghosh R., « A semantic approach to boost passage retrieval effectiveness for question answering », *ACSC '06 : Proceedings of the 29th Australasian Computer Science Conference*, Australian Computer Society, Inc., Darlinghurst, Australia, Australia, p. 95-101, 2006.

Roche M., Kodratoff Y., « Text and Web Mining Approaches in Order to Build Specialized Ontologies », *Journal of Digital Information*, vol. 10, n° 4, p. 6, 2009.

Salton G., McGill M. J., *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA, 1986.

Saneifar H., Bonniol S., Laurent A., Poncelet P., Roche M., « Mining for Relevant Terms From Log Files », *KDIR'09 : Proceedings of International Conference on Knowledge Discovery and Information Retrieval*, Madeira, Portugal, October, 2009a.

Saneifar H., Bonniol S., Laurent A., Poncelet P., Roche M., « Terminology Extraction from Log Files », *DEXA'09 : Proceedings of 20th International Conference on Database and Expert Systems Applications*, Lecture Notes in Computer Science, Springer, p. 769-776, 2009b.

Voorhees E. M., « The TREC-8 Question Answering Track Report », *In Proceedings of TREC-8*, p. 77-82, 1999.