



QUERI: Un système de question-réponse collaboratif et interactif

Badis Merdaoui, Claude Frasson

► **To cite this version:**

Badis Merdaoui, Claude Frasson. QUERI: Un système de question-réponse collaboratif et interactif. Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie, Oct 2004, Compiègne, France. pp.91-95. edutice-00000700

HAL Id: edutice-00000700

<https://edutice.archives-ouvertes.fr/edutice-00000700>

Submitted on 15 Nov 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

QUERI : Un système de question-réponse collaboratif et interactif

Badis MERDAOUI & Claude FRASSON

Département d'informatique et de recherche opérationnelle
Université de Montréal C.P.6182, Succ. Centre-ville Montréal, Québec Canada H3C 3J7
{merdaoub,frasson}@iro.umontreal.ca

Résumé

Nous présentons dans cet article une nouvelle approche de systèmes de question-réponse. Les systèmes de question-réponse ont montré leur efficacité en ajoutant des informations complémentaires trouvées sur le Web à celles qui sont disponibles dans leurs bases de données. Pour renforcer cette efficacité et améliorer leurs performances nous introduisons des techniques d'agents intelligents, de raisonnement à base de cas, et de modèles d'extraction de réponses. Cette approche permet d'avoir une meilleure interaction avec l'utilisateur.

Mots-clés: aide intelligente, représentation et gestion des connaissances, Raisonnement à base de cas.

Abstract

Systems of question-answer are evolving towards a new generation of search engines. The "user-machine" interaction is going to be changed. Systems of question-answer showed their efficiency by exploiting the great majority of data existing in their data bases with information extracted from the Web. Intelligent agents, Case-Based Reasoning techniques, and pattern extraction are used in the present system which allows a more efficient interaction with the user.

Keywords: cognitive models and agents, formalizing and managing knowledge in organizations, Case-Based Reasoning.

Introduction

Dans toute la panoplie de moteurs de recherche disponible sur le marché, la qualité du résultat présenté à l'utilisateur est cruciale. L'handicap majeur de ces applications est le fait que l'internaute doit initier explicitement toutes les tâches suivantes : consulter les documents présentés, reformuler la question si nécessaire, bien choisir les mots clés (maîtriser le langage requête) [1]. Cette métaphore d'interaction doit évoluer pour permettre aux usagers non spécialisés d'exploiter efficacement la masse d'information disponible sur le Web. Grâce à un système de question-réponse, cet usager peut obtenir une réponse précise à une question posée en langue naturelle.

Un système de question-réponse recherche généralement dans un corpus de documents une réponse précise. Par exemple, si un usager pose la question "Les étoiles se déplacent-elles dans le ciel ?" Le système répondra par une réponse bien spécifique (Oui, Non) et non par une liste de liens hypertextes.

Le processus d'un système de questions-réponses est le suivant:

1. Analyse de la question
2. Indexation de la question (transformer la question en requête)
3. Interroger un moteur de recherche pour chercher le document pertinent
4. Extraction de la réponse

Les tâches 2 et 3 se basent sur des techniques de recherche d'information, tandis que les processus 1 et 4 sont des tâches de traitement automatique de la langue naturelle (TALN) [1].

Actuellement les trois premiers processus ne posent pas de grandes difficultés du point de vue des techniques disponibles en recherche. Cependant, le dernier processus qui s'occupe de l'extraction de la réponse, reste un vrai défi pour le développement de tels systèmes, où les chercheurs posent les questions suivantes:

1. Comment peut-on répondre en temps réel à la question, justifier la réponse et savoir évaluer l'adéquation de la réponse à la question?
2. Comment peut-on déterminer les documents les plus pertinents à la question de l'utilisateur et choisir celui qui contient la réponse?
3. Comment choisir entre plusieurs réponses candidates (qui ont le même degré de similarité) celle qui sera la plus pertinente?

Le premier problème peut être résolu au moyen de techniques d'intelligence artificielle, en particulier "d'agents intelligents" [2 ; 3]. Le deuxième peut être facilité par des approches de raisonnement à base de cas ("Case-Based Reasoning") [4] En ce qui concerne le troisième problème, il peut être résolu par l'interaction directe avec l'utilisateur [5].

Il existe actuellement deux approches pour les systèmes de question-réponse collaboratifs:

1. La première utilise différents agents, chacun pouvant répondre à un type de question. [2 ; 3]
2. La deuxième utilise deux agents de recherche indépendants, l'un destiné à rechercher des réponses sur le web et l'autre sur le corpus local de documents [6]

Dans la présente étude, nous avons adopté la première approche car elle va nous permettre d'utiliser des outils qui vont nous aider à l'extraction de réponse comme WordNet, Cyc, Alembic et autres. La possibilité de choisir la technique convenable à chaque type de question permet d'atteindre des performances proches de la souhaitées tel que l'obtention de la réponse en temps réel. Selon la classification des questions, on affecte un agent à chaque type de question.

Dans la section suivante, nous fournissons une analyse des méthodes qui ont été déjà implémentées, en présentant les difficultés de chacune entre elles. Dans la section 3, nous présentons l'architecture de notre système de question-réponse (*QUERI*) ainsi que notre approche de modélisation basée sur le raisonnement à base de cas et le mode de fonctionnement de l'algorithme que nous avons utilisé pour trouver la réponse demandée par l'utilisateur. Dans la section 4 nous présentons en détail l'implémentation de notre système. Enfin, dans la section 5, nous élaborons des perspectives de développement.

Analyse des méthodes utilisées dans les systèmes de questions-réponses

Dans cette section, nous présentons quelques méthodes qui ont eu un apport à la problématique visée. Nous présentons notamment les systèmes: "PIQUANT d'IBM TREC12 [2]", "LIMSI, QALC TREC12 [6]" et "Modèles d'extraction sous forme d'expressions régulières TREC10 [7]". Toutefois, nous avons d'autres travaux aussi intéressants tels que: "JAVELIN TREC11 [5]", "POWERANSWER TREC11 [8]" et "QUANTUM TREC-X [9].

Dans ce qui suit, nous décrivons en détail chacun de ses systèmes :

PIQUANT d'IBM TREC12 [2]

PIQUANT qui a été présenté dans la conférence TREC12, était une évolution lisse de celui du TREC11 en continuant à utiliser l'architecture multi-agent. PIQUANT utilise différents agents pour répondre à différents types de questions. Cette approche a été essayée pour la première fois dans le TREC11 avec un grand succès. Grâce à cette technique PIQUANT a passé de la 11ème à la 5ème place.

PIQUANT version TREC12 a préservé tous les agents de réponse de la version TREC11 en ajoutant un agent à base de modèle complémentaire, un module de "Résolution de Réponse" et en augmentant l'envergure du contrôleur CYC. CYC est une base de connaissances dites « de sens commun », permettant à l'ordinateur de faire des inférences en apparence simples mais nécessitant en fait de nombreuses connaissances implicites [10 ; 11].

Nous décrivons brièvement ce système. Le traitement commence par l'analyse de la question, qui implique plusieurs étapes comme l'analyse syntaxique, l'identification d'entité-nommée et l'unification de structure de caractéristique.

L'analyse de Question produit un QFRAME qui contient le type de réponse exigé, le type de question, des mots-clés de question et une forme sémantique simple. Dans l'étape suivante le système passe le QFRAME au générateur QPLAN. Un QPLAN est, en principe, un programme général qui dirige le traitement, autrement dit c'est la liste des agents disponibles qui vont être utilisés. Les résultats sont passés au module "Résolution de Réponse". Ce dernier fait le choix de la réponse la plus pertinente suivant le type de la question. La figure ci-dessous montre l'architecture du système PIQUANT.

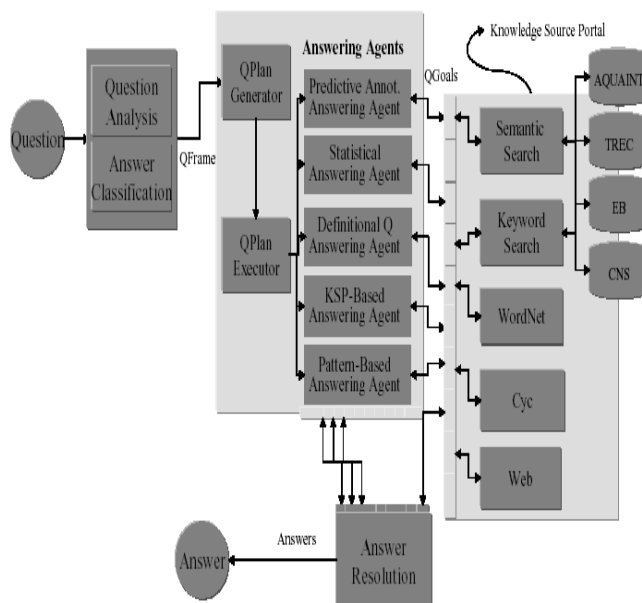


Figure 1 : Architecture du système PIQUANT

PIQUANT utilise une technique qui s'appelle *QA-par-Dossier* pour justifier la réponse. Néanmoins ce dernier reste limité à cause de manque d'interaction avec l'usager, ce qui ne lui permet pas de prendre une décision en cas d'ambiguïté.

LIMSI, QALC TREC12 [6]

Ce système a comme principe l'utilisation de plusieurs sources de données, il utilise en effet deux agents de recherche indépendants, l'un pour Internet et l'autre pour le corpus TREC, ce qui permet au système d'obtenir des scores élevés de certitude pour des réponses données. Le système est constitué des modules suivants: analyse de question, sélection de document, identification d'entité-nommée et extraction de réponse. Vu la grande quantité de documents disponibles sur le Web et aussi leur haute redondance, le système augmente ses chances d'avoir la réponse exacte. Bien que le but du système soit d'obtenir des scores de certitude élevés en utilisant différentes sources de données, le manque d'interaction avec l'usager et l'utilisation de la même technique pour répondre à n'importe quel type de question restent un obstacle pour résoudre le problème d'évaluation de réponse.

Modèles d'extraction sous forme d'expressions régulières TREC10 [7]

Ce système a apporté une nouvelle méthode à utiliser pour l'extraction de la réponse. Au TREC10, le système a utilisé des modèles d'extraction sous forme d'expressions régulières pour classer les réponses possibles. En mettant fin à la méthode classique où le classement des réponses consiste généralement à trouver des entités nommées dans des blocs contenant les mots-clés de la question, puis à regarder les fréquences des réponses.

Malgré ces résultats relativement positifs, le système n'assure pas l'interaction avec l'utilisateur et n'utilise pas l'approche multi-agent pour bien évaluer la réponse choisie.

Notre approche

L'objectif principal de notre article est de répondre à des questions posées en langage naturel à partir d'un corpus de données, où le système de question-réponse devrait pouvoir:

- Lever l'ambiguïté des questions.
- Prendre en compte le contexte général de la question.
- Extraire des réponses.
- Assurer l'interaction avec l'utilisateur, ce qui aide à justifier la réponse et à évaluer l'adéquation de la réponse à la question.
- Reformuler la réponse dans le cas de fusion de documents.
- Répondre en temps réel à la question de l'utilisateur.

Afin d'atteindre l'objectif mentionné précédemment, nous avons eu recours à diverses techniques pour améliorer l'ensemble du processus :

- la recherche d'information et le Traitement Automatique de la Langue Naturelle (TALN)
- le Raisonnement à base de cas "Case-Based Reasoning"
- les Techniques d'intelligence artificielle "multi-agent"

Fonctionnement

Dans cette section nous présentons l'architecture de notre système, ensuite la méthode de raisonnement à base de cas puis nous terminons par montrer le déroulement de notre système.

Architecture

Le système que nous proposons se scinde en quatre modules principaux : l'interface qui assure l'interaction avec l'utilisateur, le module de traitement des questions, la base de données de questions-réponse et le Web.

L'utilisateur communique sa question en langue naturelle via l'interface interactive. La première étape est l'analyse de la question. Celle-ci consiste à :

- Déterminer le type de la question
- Déterminer le contexte général de la question
- Générer une requête à partir de la question

Après avoir transformé la question de l'utilisateur sous forme de requête, une seconde étape consiste à la recherche de la réponse dans la base de données locale. Si le système ne trouve pas de réponses dans la base de données locale, un processus se déclenche afin de chercher la réponse pertinente sur le Web.

La troisième étape est l'extraction de la réponse. Selon le type de question le système peut affecter un agent intelligent qui correspond à l'une des méthodes d'extraction telles que : la recherche sémantique, la recherche statistique, la recherche par mots clés,

WordNet, ... La figure ci-dessous décrit l'architecture de notre système.

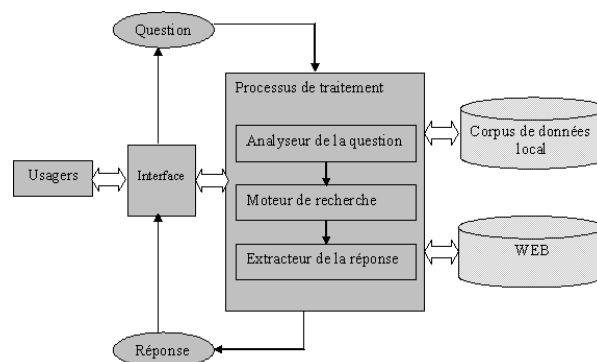


Figure 2 : Architecture du système QUERI

Raisonnement à base de cas

Après avoir décrit l'architecture de notre système, nous définissons la technique adoptée pour la réalisation de notre approche qui est le raisonnement à base de cas. Le raisonnement à base de cas (CBR) est une méthode de résolution de problèmes qui utilise des expériences passées pour résoudre de nouveaux problèmes [LEA 96]. L'ensemble des expériences passées forme une base de cas. Un système de question-réponse est considéré comme un cas CBR car il contient la description d'un problème (la question) et la description d'une solution (la réponse) [4]. La solution est construite en calculant la similarité entre le nouveau cas et les cas qui sont contenus dans la base de cas. Pour cela on utilise une fonction de similarité dont le but est de déterminer la similarité de deux cas dans la base en se basant sur leurs attributs. On peut donc, à partir de la description d'un cas, obtenir une liste ordonnée (en fonction de la valeur donnée par la fonction de similarité) des cas similaires.

La métrique de similarité est une fonction qui prend deux entités en paramètres et qui retourne une valeur reflétant la similarité entre ces deux entités selon un certain but. Dans notre cas, le calcul de la similarité globale entre la question de l'utilisateur et une autre qui existe déjà dans la base de données se base sur le calcul de la similarité de type et la similarité locale [4] :

a- Similarité de type

$$\text{SimType}(\text{user_}q, q_k) = \begin{cases} 1.0 & \text{Si } \text{type } \text{user_}q = \text{type } q_k \\ 0 & \text{Sinon} \end{cases}$$

user_q : Question de l'utilisateur

q_k : Question numéro k, de la base de données

b- Similarité locale

$$\text{SimLoc}(\text{user } a_i, a_{kT}) = \max_{0 \leq J \leq m} \text{SimLoc}_T(\text{user } a_i, a_{kT})$$

user_qi : index (terme) numéro i de la question qi

q_kj : index numéro J de la question qk

m : Est le nombre total des indexs (termes appropriés) de la question qk

$$\text{SimLoc}_j(\text{user_}q_i, q_{k,j}) = \begin{cases} 1.0 & \text{Si } \text{user_}q_i = q_{k,j} \\ 0.9 & \text{Si } ((\text{user_}q_i \text{ Suffixe/Préfixe de } q_{k,j}) \\ & \text{Ou } (q_{k,j} \text{ S/P de } \text{user_}q_i)) \\ 0.6 & \text{Si } ((\text{user_}q_i \text{ Contient } q_{k,j}) \\ & \text{Ou } (q_{k,j} \text{ Contient } \text{user_}q_i)) \\ 0 & \text{Sinon} \end{cases}$$

c- Similarité globale

$$\text{SimGlob}(\text{user_}q, q_k) = \frac{\left[\sum_{(i=1)}^n \text{simLoc}(\text{user_}q_i, q_k) + \text{simType}(\text{user_}q, q_k) \right]}{n+1}$$

n est Le nombre de termes de la question $\text{user_}q$

Déroulement

La métrique choisie pour la partie déjà implémentée est la métrique statistique où la question de l'utilisateur et chaque document du corpus sont convertis selon le modèle vectoriel de la recherche d'information.

Étant donnée une question entrée par l'utilisateur, le programme accomplit les tâches suivantes :

- Récupérer la question de l'utilisateur
- Analyser de la question de l'utilisateur en passant par:
 - L'analyse lexicale (Tokenisation)
 - La détermination du type de la question
 - La correction d'orthographe
 - La normalisation (Ex: prendre juste la racine pour un verbe régulier)
- Extraction des termes appropriés (indexation)
- Calculer la similarité entre la question de l'utilisateur et chaque document du corpus de données toute en calculant:
 - La similarité du type
 - La similarité locale
 - La similarité globale
 - La similarité détachée (En prend en considération le nombre de termes de la question de la base de données)
- Mettre les similarités (globales et détachés) dans deux tableaux différents.
- Trier les tableaux en ordre décroissant avec la méthode mergesort.
- **Si** le tableau des similarités globales contient une et une seule valeur maximale :
 - Si (valeur maximale $\geq 70\%$) alors afficher la réponse adéquate
 - Sinon Si (valeur maximale $\geq 50\%$) alors demander la confirmation de l'utilisateur
 - Sinon Demander à l'utilisateur de mieux formuler sa question
- **Sinon** Si le tableau des similarités détachées contient une et une seule valeur maximale : (Faire comme dans le cas du tableau des similarités globales)
 - Sinon Proposer les trois premières questions du tableau des similarités détachées.

Cette démarche basée sur les approches de : « Case-Based Reasoning » et « Information Retrieval », nous a permis d'obtenir de bons résultats.

Implémentation

La classification utilisée pour déterminer le type de la question est celle de [12]. Voir L'annexe A.

Le langage de programmation que nous avons choisi pour l'implémentation est C#.

L'interaction avec l'utilisateur est un aspect important dans notre système QUERI, principalement pour établir un contexte et pour lever l'ambiguïté des questions. Nous présentons une de nos interfaces qui assure l'interaction avec l'utilisateur

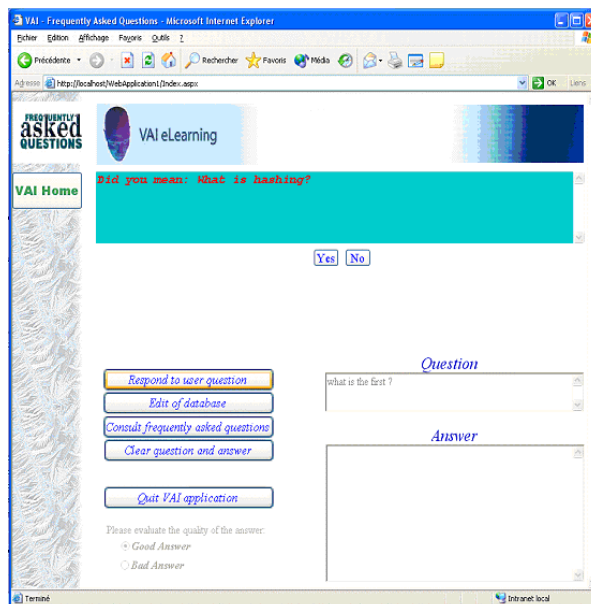


Figure 3 : Interface QUERI

Le système propose une question ou une liste de trois questions en donnant la possibilité à l'utilisateur de faire son choix ou en lui demandant à reformuler sa question.

Conclusion

Nous avons présenté QUERI, un système de question-réponse collaboratif et interactif qui permet de chercher une réponse sur le Web ou dans un corpus de données pour répondre à une question posée en langage naturel. Nous avons combiné des techniques d'intelligence artificielle, de recherche d'information, de TALN et de Raisonnement à Base de Cas (CBR) pour améliorer les performances de notre système en prenant en compte les limites des systèmes précédents, comme le manque d'interaction avec l'utilisateur et la justification de la réponse.

Les systèmes de question-réponse ont montré leur efficacité en exploitant la grande masse de données existante dans leurs bases de données et celle trouvée sur le Web. Parmi les domaines d'applications de ces systèmes nous voyons les systèmes tutoriels, systèmes tutoriels intelligents (eLearning) et les systèmes de réponses automatiques aux questions pour les

organisations et les entreprises. Un système de question-réponse accessible aux clients sur le site Web de l'entreprise permettrait de réduire le volume de courriels.

Comme perspectives de ce projet nous envisageons la prise en considération du profil de l'utilisateur tout en lui garantissant une interaction selon son profil. Ainsi que l'intégration d'un agent intelligent qui s'occupe du rafraîchissement des profils des usagers (profil dynamique).

Références

Actes de Conférences

- [1] T. Poibeau, P. Zweigenbaum et A. Nazarenko Traitement Automatique des Langues pour les systèmes de Question/Réponse Document de travail élaboré dans le cadre de l'action spécifique RIP-WEB, 2003
- [2] J. Chu-Carroll, J. Prager, C. Welty, K. Czuba, D. Ferrucci, IBM T.J. Watson Research Center IBM's PIQUANT in TREC2003 (TREC-12), 2003
- [3] J. Chu-Carroll, J. Prager, C. Welty, K. Czuba, D. Ferrucci, IBM T.J. Watson Research Center A Multi-Strategy and Multi-Source Approach to Question Answering, In Proceedings of the 11 th Text REtrieval Conference (TREC-11), 2002
- [4] C. Gresse von Wangenheim, A. Bortolon, and von Wangenheim "A hybrid approach for the management of FAQ Document in Latin languages". In: iccbr 2001 - international conference of case-based reasoning, 2001, Vancouver. International conference of case-based reasoning 2001.
- [5] E. Nyberg, T. Mitamura, J. Carbonnell, J. Callan, K. Collins-Thompson, K. Czuba, M. Duggan, L. Hiyakumoto, N. Hu, Y. Huang, J. Ko, L.V. Lita, S. Murtagh, V. Pedro, D. Svoboda, Carnegie Mellon University, The JAVELIN Question-Answering System at TREC 2002, In Proceedings of the 11 th Text REtrieval Conference (TREC-11), 2002
- [6] G. de Chalendar, T. Dalmas, F. Elkateb-Gara, O. Ferret, B. Grau, M. Hurault-Plantet, G. Illouz, L. Monceaux, I. Robba, A. Vilnat, LIMSI-CNRS, The Question Answering System QALC at LIMSI, Experiments in Using Web and WordNet, In Proceedings of the 11 th Text REtrieval Conference (TREC-11), 2002
- [7] M.M. Soubbotin, InsightSoft-M, Patterns of Potential Answer Expressions as Clues to the Right Answers, In Proceedings of the 10 th Text REtrieval Conference (TREC-10), 2001
- [8] D. Molovan, S. Harabagiu, R. Girju, P. Morarescu, F. Lacatusu, A. Novischi, A. Badulescu and O. Bolohan, LCC Tools for Question Answering, 2003
- [9] M. Soubbotin, S. Soubbotin, Use of Patterns Detection of Answer Strings: A Systematic Approach, 2003
- [10] Lenat 1990
- [11] Lenat 2001

[12] Graesser et al., 1992

Mémoire ou Thèse

Luc Plamondon, Le système de question-réponse QUANTUM. Mémoire de maîtrise, Université de Montréal, Montréal, Canada, mars 2002