Confronter des sources de connaissances différentes pour obtenir une réponse plus fiable

Gaël de Chalendar, Faïza Elkateb-Gara, Olivier Ferret, Brigitte Grau, Martine Hurault-Plantet, Laura Monceaux, Isabelle Robba, Anne Vilnat

Confronter des sources de connaissances différentes pour obtenir une réponse plus fiable

G. de Chalendar, F. El Kateb, O. Ferret, B. Grau, M. Hurault-Plantet,
L. Monceaux, I. Robba, A. Vilnat

LIMSI – Groupe LIR
BP 133, 91403 Orsay
[nom]@limsi.fr

Résumé – Abstract

La fiabilité des réponses qu'il propose, ou un moyen de l'estimer, est le meilleur atout d'un système de question-réponse. A cette fin, nous avons choisi d'effectuer des recherches dans des ensembles de documents différents et de privilégier des résultats qui sont trouvés dans ces différentes sources. Ainsi, le système QALC travaille à la fois sur une collection finie d'articles de journaux et sur le Web.

A question answering system will be more convincing if it can give a user elements concerning the reliability of it propositions. In order to address this problem, we choose to take the advice of several searches. First we search for answers in a reliable document collection, and second on the Web. When the two sources of knowledge give the system QALC common answers, we are confident with them and boost them at the first places.

Mots Clés – Keywords

Système de question-réponse, recherche d'information, fiabilité des réponses
Question answering system, information retrieval, answer reliability

1 Introduction

La recherche de réponses précises à des questions factuelles portant sur des domaines non restreints constitue un champ de recherche en plein essor. Actuellement, les moteurs de recherche retournent au mieux des extraits, comme le fait Google par exemple[1], et dans ce cas, leur rôle consiste plus à justifier le document proposé qu'à fournir un passage comme

[1] http://www.google.com

réponse, même si celle-ci figure souvent dans ces extraits. L'un des challenges de ce type de recherche consiste à donner une seule réponse, et non une liste plus ou moins longue de propositions, et pour cela, le système doit être suffisamment sûr de ce qu'il a trouvé. La détermination de la fiabilité d'une réponse consiste soit à prouver sa véracité, soit à estimer un degré de confiance. Si la réponse est produite à l'issue d'un raisonnement sur une représentation formelle des connaissances, on peut alors en élaborer une preuve formelle. Ainsi, le système LCC (Moldovan et al., 2002) dérive une chaîne d'inférences à partir d'une représentation logique des connaissances extraites de WordNet[2]. Cette approche nécessite de posséder une base traitant tous les sujets dont relèvent les questions, ce qui ne peut être garanti lorsque l'on fonctionne en monde ouvert. La seconde possibilité, que nous avons choisie, consiste à estimer le degré de fiabilité d'une réponse en lui attribuant un poids qui est fonction des processus et des types de connaissances utilisées. Après avoir constaté que cette méthode endogène de pondération n'était pas suffisante, nous avons opté pour la recherche des réponses dans la collection de référence doublée d'une autre dans une autre source d'informations afin de confronter les résultats des deux recherches. Le principe est de favoriser des réponses trouvées dans les deux sources, par rapport aux réponses, même fortement pondérées, mais trouvées dans une seule collection. Un tel raisonnement s'applique d'autant mieux que les sources de connaissances sont de nature différente, ainsi notre deuxième recherche s'effectue sur le Web, qui, de surcroît, par sa diversité et sa redondance conduit à trouver de nombreuses réponses (Magnini et al., 2002a et 2002b ; Clarke et al., 2001 ; Brill et al., 2001). Après la présentation générale de notre système, QALC, section 2, nous décrivons section 3 la reformulation des questions pour interroger le Web. La section 4 présente ensuite l'extraction des réponses pour une seule source de connaissances, et la section 5 les stratégies pour réaliser le choix final. Les résultats de QALC sont décrits en section 6 avant de rapprocher notre travail de ce qui existe dans le domaine.

2 Le système QALC

Le système QALC (figure 1) participe aux évaluations TREC depuis 4 ans et a été conçu pour rechercher des réponses à des questions factuelles dans une grande base de documents. Le principe est d'extraire un maximum d'informations des questions afin de guider la recherche des réponses (voir (Ferret et al. 2003) pour une description complète). L'analyse des questions vise à déduire des caractéristiques permettant l'extraction de la réponse, et à donner des indices pour reformuler la question afin de produire une requête sur le Web. Cette analyse s'appuie sur les résultats d'un analyseur robuste de l'anglais, IFSP (Aït-Mokthar et Chanod, 1997). Les requêtes construites pour la recherche dans la collection TREC sont formées à l'aide d'opérateurs booléens et envoyées au moteur de recherche MG[3], alors que les requêtes Web essaient d'approcher une formulation exacte de la réponse. En effet, nous supposons que la grande taille du Web permettra de trouver des documents, même si la requête est très précise. Les documents sélectionnés (1500 passages trouvés par MG ou 20 documents provenant du Web) sont alors examinés. Ils sont ré-indexés par les termes de la question et leurs variantes, puis ré-ordonnés suivant le type des termes trouvés dans les documents. Un

[2]Pour les détails sur WordNet, voir la page : http://www.cogsci.princeton.edu/~wn/

[3]Moteur de recherche Managing Gigabytes : http://www.mds.rmit.edu.au/mg/intro/about_mg.html

sous-ensemble de documents est sélectionné parmi les documents TREC, alors que tous les documents Web sont gardés, et ils sont annotés par les types d'entités nommées reconnues. Après pondération des phrases candidates, la réponse est extraite par des traitements différents selon le type attendu, et reçoit un score de confiance. Enfin, les réponses venant du corpus TREC et celles du Web sont comparées, afin d'en choisir une. Le principe appliqué consiste à favoriser une réponse obtenue dans les 5 premières propositions des deux chaînes.

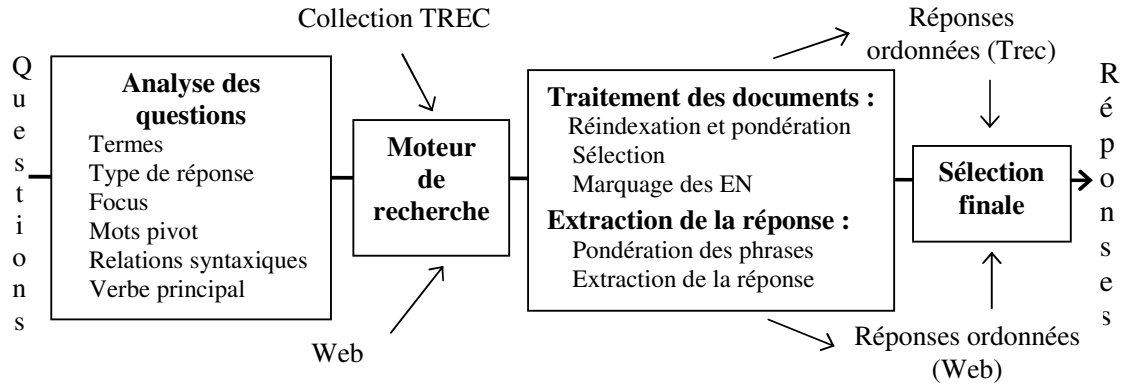

Figure 1 – Le système QALC

3 Reformulation des questions pour la recherche sur le Web

Devant la grande redondance des informations présentes sur le Web, nous avons supposé qu'il était possible de trouver des documents pertinents même avec une requête très spécifique et qu'une requête précise permettrait d'obtenir dans les premières positions les documents susceptibles de contenir la réponse à une question. C'est pourquoi nous avons choisi de reformuler les questions sous une forme affirmative avec aussi peu de variations que possible par rapport à la formulation d'origine. Par exemple, pour la question « *When was Wendy's founded ?* », nous supposons que nous trouverons un document contenant la réponse sous la forme : « *Wendy's was founded on….* » Nous recherchons donc les chaînes exactes fournies par la reformulation, comme dans (Brill et al. 2001) et non pas les différents mots de la requête éventuellement reliés par des opérateurs AND, OR ou NEAR comme dans (Magnini, et al. 2002a) ou (Hermjacob et al. 2002). Ainsi, nous pouvons sélectionner un nombre réduit de documents, 20 dans nos expériences.

La réécriture des questions utilise des schémas de reformulation conçus à partir de l'étude des questions de TREC9 et TREC10. Nous avons d'abord caractérisé les questions en fonction du type de la réponse attendue et du type de la question. Rechercher un nom de personne ou bien un lieu ne mènera pas à la même reformulation, même si les deux questions sont syntaxiquement similaires. "*Who is the governor of Alaska ?*" et "*Where is the Devil's Tower ?*" n'attendent pas des réponses formulées exactement de la même manière : la requête ", *the first governor of Alaska*" fonctionne pour la première question car un nom propre est souvent apposé, tandis que "*The Devil's Tower is located*" est une requête possible pour la seconde car nous recherchons des réponses qui sont la forme affirmative de la question. Nous avons essayé manuellement avec Google ces schémas de reformulation pour trouver les types les plus fréquemment couronnés de succès. Nous avons ainsi examiné environ 50 questions et leurs réponses. Ces tests ont montré la nécessité d'ajouter un critère pour obtenir une

réécriture plus précise : soit un mot introduisant un modifieur dans la forme minimale de la question soit un mot à ajouter à celui de la question (souvent une préposition ou un verbe) pour introduire l'information recherchée dans la forme affirmative. Par exemple, le mot « *when* » introduisant un modifieur est conservé et la présence du mot « *year* » dans une question portant sur une date impose l'ajout de la préposition « *in* » à la forme affirmative.

Un schéma de réécriture est donc construit en fonction des caractéristiques syntaxiques de la question (Ferret et al. 2002) : le focus, le verbe principal, les modifieurs et les relations introduisant des modifieurs du verbe ou de l'objet. La réécriture la plus simple est construite avec tous les mots de la question hormis le pronom interrogatif et l'auxiliaire, comme pour les questions du type « WhatBe ». Par exemple, la question « *When was Lyndon B. Johnson born ?* » se réécrit en : « *Lyndon B. Johnson was born on* », en appliquant le schéma « <focus> <verbe principal> born on ». Google trouve alors en première position, la réponse « *Lyndon B. Johnson was born on August 27, 1908* ». Pour éviter d'être trop restrictifs, nous soumettons les requêtes avec et sans guillemets (recherche de la chaîne exacte ou seulement de l'ensemble des mots) et nous associons à chaque type de question un ou plusieurs schémas, permettant ainsi le relâchement de contraintes par rapport au schéma primitif. Pour évaluer le module de réécriture, nous avons cherché, dans les 20 premiers documents, les patrons des réponses aux 500 questions de TREC11. 372 questions pouvaient être résolues ainsi, soit 74,4% des questions. Parmi celles-ci, 360 permettent de trouver plus d'un document pertinent.

4 Recherche d'une réponse dans les documents

4.1 Sélection des documents

La première phase dans la recherche d'une réponse consiste à restreindre le nombre de documents dans lesquels, compte tenu de son coût, cette recherche est menée. La méthode utilisée est globalement similaire à celle présentée dans (Ferret et al., 2003). Un ensemble de mono et de multi-termes sont extraits de la question au moyen d'un termeur à base de patrons morpho-syntaxiques. Les documents renvoyés par le moteur de recherche sont ensuite indexés par FASTR (Jacquemin, 2001), qui permet de reconnaître les termes de la question ainsi que leurs variantes morphologiques, syntaxiques ou sémantiques. Chaque terme reconnu se voit attribuer un poids en fonction du type de variante qu'il représente, permettant le calcul d'un score global pour chaque document. Finalement, un ensemble restreint de documents est sélectionné lorsque la courbe des scores présente un décrochement significatif ; sinon, les 100 premiers documents sont retenus. Dans cette version de QALC, nous nous sommes attachés à améliorer cette sélection en augmentant la robustesse de la reconnaissance des termes réalisée par FASTR. Celui-ci ayant davantage été conçu pour reconnaître des variantes terminologiques complexes que comme un outil robuste de recherche d'information, il est assez sensible aux erreurs de l'étiqueteur morpho-syntaxique que nous utilisons, le TreeTagger[4]. De ce fait, il lui arrive de passer à côté d'occurrences de termes reconnaissables par un simple appariement. Sur les documents sélectionnés pour les 500 questions de l'évaluation TREC11, nous avons évalué à 71% le rappel de FASTR concernant les mono et

[4] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

les multi-termes sans variation. La référence était constituée par les résultats d'un mécanisme d'appariement de termes s'appuyant sur les résultats du TreeTagger.

	QALC TREC 10	**NIST-100**	**QALC TREC 11**
Nb documents avec une réponse (variation en %)	2041	2479 (+ 21,5)	2313 (+ 13,3)
Nb documents retenus (variation en %)	30992	49900 (+ 61,0)	34568 (+ 11,5)
Rappel (%)	46,0	55,8	52,1
Précision (%)	6,6	5,0	6,7

Tableau 1 : Résultats de la sélection de documents pour les questions QA-TREC10

Pour améliorer la sélection des documents, nous avons combiné les résultats de FASTR et ceux de notre mécanisme d'appariement de termes en éliminant les doublons. L'impact de ce changement est illustré Tableau 1 : les résultats s'appuient sur les jugements réalisés par le NIST sur les réponses des participants à l'évaluation TREC10. NIST-100 correspond à une sélection qui retiendrait les 100 premiers documents renvoyés par le moteur de recherche du NIST. La précision correspond au rapport entre le nombre de documents retenus qui contiennent effectivement une réponse et le nombre de documents retenus par le système considéré. Le rappel est le rapport entre le nombre de documents retenus contenant une réponse et le nombre de documents trouvés par au moins un participant à TREC10 et contenant une réponse. La combinaison des deux algorithmes permet donc d'améliorer le rappel par rapport à notre version précédente tout en maintenant la précision. De plus, le nombre significatif de nouveaux documents pertinents retenus n'entraînent qu'un accroissement linéaire du nombre total de documents à traiter par la suite du système.

4.2 Pondération des phrases

Toutes les phrases des documents sélectionnés sont analysées afin de leur attribuer un poids qui évalue à la fois la possibilité que la phrase contienne la réponse et que QALC puisse la localiser. Les critères de pondération que nous avons choisis découlent des informations de base extraites de la question, à savoir les mots et le type de réponse. Notre objectif est de permettre au module d'extraction de la réponse d'augmenter le poids faible de certaines phrases grâce à des critères de pondération qui leur sont spécifiques. La pondération se fonde sur la présence des mots suivants dans les phrases candidates :

- mots lemmatisés de la question, pondérés[5], et leurs variantes,
- mots exacts de la question et leur proximité mutuelle,
- mots dont le type correspond à celui de l'entité nommée attendue dans la réponse.

Le premier critère (présence des lemmes de la question) permet d'attribuer à chaque phrase un poids de référence. Des poids sont ensuite ajoutés dès qu'un des autres critères est satisfait. Chacun de ces poids ne dépasse pas 10% du poids de référence. Nous obtenons une moyenne de 543 phrases par question. Pour 71% des questions, au moins une phrase contient la bonne réponse et 84% d'entre elles sont classées dans les 30 premières positions.

[5] Le poids est inversement proportionnel à la fréquence des mots calculée dans un large corpus

4.3 Extraction de la réponse

Si le type de réponse attendue est une entité nommée alors QALC sélectionne les mots des phrases qui correspondent à l'entité nommée attendue. Pour évaluer la réponse, il renforce le poids de la phrase par des poids supplémentaires qui dépendent :

- du degré de précision de l'entité nommée (générique ou spécifique),
- de la localisation de la réponse par rapport aux mots de la question dans la phrase réponse, donnée par le barycentre des mots de la question dans la phrase réponse.
- de la redondance de la réponse dans les dix premières phrases.

Quand la réponse attendue n'est pas une entité nommée, nous utilisons des patrons d'extraction. Chaque phrase candidate retenue par le module de sélection des phrases est analysée en utilisant des patrons d'extraction associés au type de la question déterminé lors de l'analyse de la question. Ces patrons représentent un ensemble de contraintes appliquées aux phrases candidates. Ces règles sont constituées de patrons syntaxiques, utilisés pour localiser les réponses possibles dans les phrases, et de relations sémantiques pour valider les réponses. Les patrons syntaxiques repèrent les mots de liaison et simulent les paraphrases possibles de la réponse. Les patrons sémantiques sont établis en utilisant, dans Wordnet, la relation d'hyperonymie et la définition des mots. Les réponses potentielles sont alors pondérées selon les contraintes satisfaites. L'ordre de grandeur du poids dépend de la fiabilité des contraintes (pour plus de détails sur ces 2 sous-sections, voir de Chalendar et al, 2002).

5 Sélection finale de la réponse

Pour l'évaluation TREC11, les systèmes participant devaient fournir une réponse par question et l'ensemble des 500 réponses devait être ordonné en fonction de la confiance du système dans chaque réponse. La métrique d'évaluation est la suivante :

$$\frac{1}{Q}\sum_{i=1}^{Q}\frac{\text{Nb de réponses correctes dans les i premiers rangs}}{\text{i}}$$, avec Q le nombre total de questions.

Ainsi, l'évaluation tient compte non seulement de l'exactitude d'une réponse donnée mais aussi de la confiance que le système a dans ses propres réponses. La stratégie élaborée pour la sélection finale est fondée sur la comparaison des résultats de QALC appliqué à chacune des deux sources de connaissance : la collection TREC et le Web. L'utilisation du Web vise à confirmer des réponses trouvées dans la collection de référence, mais aussi à augmenter le nombre de réponses trouvées par le système. Dans ce dernier cas, notons cependant que parmi les réponses obtenues sur le Web, certaines ne sont pas retrouvées dans un document de la collection TREC, et ne sont donc pas retenues comme réponse finale. Les deux applications du système QALC fournissent pour chaque question un ensemble de réponses qui sont ordonnées selon le score qu'elles ont reçu tout au long du processus d'extraction. Le rôle de la sélection finale est alors d'extraire la réponse de ces deux ensembles. Le tableau 2 donne les ensembles de réponses obtenus pour la question : *Who defeated the Spanish armada ?*

Réponses issues de TREC		**Réponses issues du Web**		**Score final**
0: Queen Elizabeth	score: 1205	0: Elizabeth I	score: 1299	
1: England	score: 1202	1: Elizabeth I	score: 1297	
2 : Francis Drake	**score: 982**	2: Philip II	score: 1282	
3: Spain	score: 872	**3: Francis Drake**	**score: 1252**	**1852**

Tableau 2 : Exemple d'ensembles de réponses

Pour la sélection finale, nous avons élaboré deux algorithmes qui explorent ces ensembles à la recherche de réponses communes. Pour l'instant, seules les cinq premières réponses sont examinées, ce chiffre pouvant être sans difficulté revu à la hausse. Le premier algorithme examine chaque couple ($réponse_i$, $réponse_j$), i étant la position d'une réponse dans l'ensemble provenant de la collection TREC, j, la position d'une réponse dans l'ensemble provenant du Web. Quand les deux réponses d'un couple sont égales, le meilleur des deux scores reçoit un bonus calculé en fonction de i et j : (11 – (i + j)) * 100. La réponse finalement retournée est celle obtenant le meilleur score. L'exemple du tableau 2 montre que la réponse au rang 2 de la collection TREC et celle au rang 3 du Web sont les mêmes. Le bonus reçu est de 600 et la réponse *Francis Drake* est finalement retournée car elle obtient le meilleur score : 1852. Le second algorithme diffère en ce sens qu'il ne remet pas en cause l'ordre des réponses : la première réponse de l'un des deux ensembles est retournée mais avec un score renforcé si cette réponse appartient à l'autre. Dans les deux algorithmes, l'idée sous-jacente est de comparer les résultats provenant de plusieurs sources de connaissance. Dans le cas du premier, est renforcé le score des réponses qui appartiennent aux deux ensembles, ce qui permet à un nombre non négligeable de bonnes réponses d'atteindre le premier rang (voir la section 6). Dans le cas du second, la réponse retournée était déjà au premier rang, mais son score est éventuellement augmenté.

Le tableau 3 contient les résultats des deux algorithmes appliqués à l'ensemble des 500 questions de TREC11. Même si les résultats du premier ne sont pas nettement meilleurs, nous pensons que son approche est plus intéressante et qu'elle pourrait encore être améliorée ; la comparaison des réponses pourrait être étendue à plus de cinq réponses et plutôt que de ne détecter que les cas de stricte égalité, elle pourrait examiner les cas d'inclusion d'une réponse dans une autre. La stratégie pourrait aussi être appliquée à plus de deux versions de QALC.

	Réponses exactes	**Score**
Algorithme 1	165	0.587
Algorithme 2	159	0.574

Tableau 3 : Résultats obtenus par les deux algorithmes pour les 500 questions

6 Résultats

Dans la dernière ligne du tableau 4, nous avons résumé (en gras) les résultats obtenus par QALC lors de l'évaluation TREC11, où nous avons été classé 9èmesur 34 participants. Nous donnons à la fois l'évaluation transmise par le Nist (le Score de Confiance SC1), et l'évaluation obtenue à l'aide des patrons de réponses fournis également par le Nist (SC2). Cette dernière évaluation ne tenant compte ni des réponses non validées par un document de la collection ni des inexactes, ces résultats sont sensiblement meilleurs, mais ils nous servent

de point de comparaison pour les autres tests auxquels nous avons procédé. Nous avons analysé les résultats obtenus par une recherche dans les seuls documents TREC, dans les seuls documents trouvés sur le Web (en n'appliquant donc pas la sélection finale), ou en combinant ces deux sources.

	Bonnes réponses	**Non validées**	**Inexactes**	**SC1**	**SC2**
TREC	128	?	?	?	0.402
Web	122	?	?	?	0.436
TREC+Web	**165**	**20**	**11**	**0.497**	0.587

Tableau 4: Résultats TREC, Web et TREC+Web

On peut noter que la recherche TREC+Web améliore de 46% les résultats obtenus sur les documents TREC seuls. Cette amélioration peut être due à des réponses supplémentaires trouvées dans les documents Web, ou alors à l'algorithme de classement décrit plus haut.

Nous avons d'abord examiné la source des réponses trouvées dans la chaîne TREC+Web, et vérifié si elles étaient ou non obtenues en commun. Sur les 165 bonnes réponses, 106 sont trouvées dans les deux ensembles de documents, 42 uniquement dans les documents TREC, 17 uniquement dans les documents Web (figure 2). Ensuite, nous avons évalué l'influence des réponses trouvées uniquement dans les documents Web. Pour cela, nous avons enlevé les 17 questions correspondant à ces bonnes réponses et évalué les résultats obtenus pour les 483 questions restantes (tableau 5). Le score est encore amélioré de 37% grâce au Web, même si l'évaluation favorise la chaîne TREC par rapport à la chaîne TREC+Web, en supprimant des réponses erronées de la première et des bonnes de la seconde. L'amélioration est donc principalement due au meilleur classement des réponses.

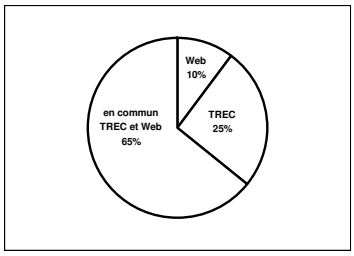

Figure 2 : Origine des bonnes réponses

	Bonnes réponses	**Score**
TREC seul	128	0.414
TREC+Web	148	0.568

Tableau 5: Résultats sur 483 réponses

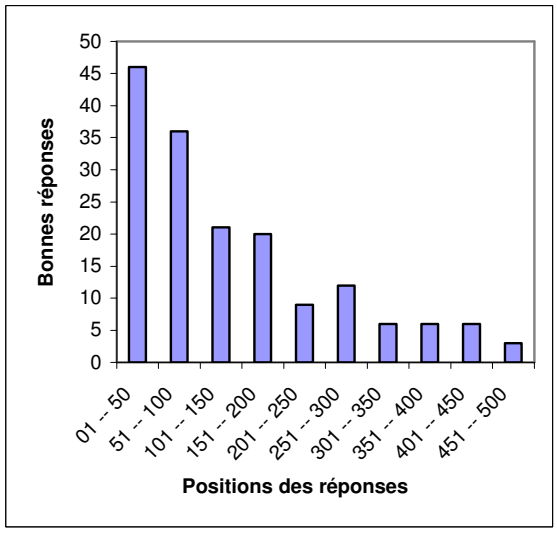

Figure 3: Positions des bonnes réponses

Nous nous sommes ensuite intéressés aux positions des bonnes réponses pour confirmer l'hypothèse qu'elles ont été bien classées. La Figure 3 illustre cette étude. Il est important de noter que dans les 50 premières places se trouvent 46 bonnes réponses. On remarque aussi que la première bonne réponse dont le score n'a pas été modifié par la confirmation dans l'algorithme de classement est en 143ème position (elle vient de la recherche dans les seuls

documents Web) ou en 145ème (venant des documents TREC). Enfin, nous avons fait une évaluation de nos résultats en utilisant le protocole TREC10, c'est-à-dire en donnant pour chaque question non plus une, mais les cinq meilleures réponses trouvées (tableau 6). On note ainsi l'importance de ne pas se limiter à la meilleure réponse pour faire l'interclassement.

	Rang 1 à-5	**%**	**Rang 1**	**%**	**Rang 2 à 5**	**%**
TREC	177	100	128	72	49	28
Web	177	100	122	69	55	31

Tableau 6 : Bonnes réponses entre les rangs 1 et 5

Par ailleurs, (Chu-Carroll *et al*., 2002) ont proposé un score mesurant la compétence des systèmes pour classer leurs réponses et l'ont appliqué aux 15 meilleurs systèmes. La compétence est calculée de la façon suivante :

Compétence = (SC-SCm) / (SCM-SCm), avec : SC=score de confiance TREC, SCm=SC moyen obtenu en ne classant pas les questions, SCM= SC maximum obtenu en classant toutes les bonnes réponses en tête.

Le degré de compétence représente ainsi la part de gain obtenu en classant par rapport au classement idéal. QALC obtient un score de 0.657 et se place en tête des 15 meilleurs systèmes. Avant la sélection finale de la réponse, son score sur les documents TREC seuls est de 0.42, soit une amélioration de 57%.

7 Travaux connexes

Magnini et al. (2002a) utilisent aussi le Web pour valider la réponse. Ils l'interrogent avec une requête composée de mots de la question et de la réponse reliés par des opérateurs booléens et de proximité. Ils ne cherchent pas à obtenir une correspondance exacte de la question. La validité de la réponse est évaluée par rapport au nombre de documents extraits. Cette approche leur a permis d'améliorer la performance de leur système de 28%. Dans TREC11, Magnini et al. (2002b) appliquent la même approche et 40 réponses par question sont validées par le Web. La pondération des réponses est fondée sur le coefficient de validité et la fiabilité du type de la réponse attendue. Clarke et al. (2001) sélectionnent 20 passages de la collection et 40 passages du Web. Ce dernier n'est utilisé que pour augmenter le facteur de redondance des réponses candidates. Cette approche leur a permis d'améliorer leurs résultats de 25 à 30%.

Concernant la réécriture de la question, Brill et al. (2001) gardent les mots de la question dans leur ordre original et déplacent les verbes dans toutes les positions possibles. Ils effectuent, comme dans QALC, une comparaison entre chaînes de caractères. Hermjacob et al. (2002) engendrent des variantes de la question en utilisant des règles de paraphrase syntaxique et sémantique. Ces paraphrases sont utilisées pour former des requêtes booléennes (3 paraphrases par question en moyenne) afin d'interroger le Web.

8 Conclusion

Il est difficile d'estimer la fiabilité d'une réponse quand chaque processus produit successivement des résultats approchés. Une solution consiste à confronter les résultats que

ces mêmes processus obtiennent avec une autre source de connaissances, le Web. En nous fondant sur les propriétés du Web, à savoir le nombre important de documents et la redondance des informations, nous accordons une grande confiance aux réponses communes trouvées dans les cinq premières positions. Cette stratégie nous a permis d'obtenir de meilleurs résultats à TREC que certains systèmes qui trouvent plus de réponses, mais évaluent moins bien leur fiabilité.

Références

Aït-Mokthar S., Chanod J-P., (1997), IFSP, Incremental finite-state parsing. *Proceedings of Applied Natural Language Processing, Washington, DC.*

Brill E., Lin J., Banko M., Dumais S., Ng A., (2001), Data-Intensive Question Answering. *TREC 10 Notebook*, Gaithersburg, USA.

de Chalendar G., Dalmas T. , Elkateb-Gara F. , Ferret O., Grau B., Hurault-Plantet M., Illouz G., Monceaux L., Robba I., Vilnat A., (2002), The Question Answering System QALC at LIMSI, Experiments in Using Web and WordNet, *TREC 11 Notebook*, Gaithersburg, USA.

Chu-Carroll J., Prager J., Welty C., Czuba K., Ferruci D., (2002), A Multi-Strategy and multi-source Approach to Question Answering. *TREC 11 Notebook*, Gaithersburg, USA.

Clarke C.L., Cormack G.V., Lynam T.R., Li C.M., McLearn G.L., (2001), Web Reinforced Question Answering (MultiText Experiments for Trec 2001), *TREC 10 Notebook*, Gaithersburg, USA.

Ferret O., Grau B., Hurault-Plantet M., Illouz G., Jacquemin C., Monceaux L., Robba I., Vilnat A. (à paraître 2003), How a NLP approach benefits question answering. *Knowledge Organization journal, A Special Issue on Evaluation of HLT*. Guest Editor: Widad Mustafa El Hadi. Volume 29 N° 29.3-4

Ferret O., Grau B., Hurault-Plantet M., Illouz G., Monceaux L., Robba I., Vilnat A. (2002), Recherche de la réponse fondée sur la reconnaissance du focus de la question, Actes de *TALN 2002,* Nancy.

Hermjakob U., Echihabi A., Marcu D., (2002), Natural Language Based Reformulation Resource and Web Exploitation for Question Answering, *TREC 11 Notebook*, Gaithersburg.

Jacquemin C., (2001), *Spotting and Discovering Terms through NLP*, Cambridge, MA: MIT Press.

Magnini B., Negri M., Prevete R., Tanev H., (2002a), Is It the Right Answer? Exploiting Web redundancy for Answer Validation, *Proceedings of the 40th ACL*, pp425-432.

Magnini B., Negri M., Prevete R., Tanev H., (2002b), Mining Knowledge from Repeated Co-occurrences: DIOGENE at TREC-2002, *TREC 11 Notebook*, Gaithersburg, USA.

Moldovan D., Harabagiu S., Girju R., Morarescu P., Lacatusu F., Novischi A., Badalescu A., Bolohan O., (2002), LCC Tools for Question Answering, *TREC 11 Notebook*, Gaithersburg.