Quand la réponse se trouve dans un grand corpus

Olivier Ferret, Brigitte Grau, Martine Hurault-Plantet, Gabriel Illouz, Christian Jacquemin

Quand la réponse se trouve dans un grand corpus

Olivier Ferret, Brigitte Grau, Martine Hurault-Plantet, Gabriel Illouz et Christian Jacquemin

LIMSI-CNRS
BP 133
91403 Orsay cedex
[ferret, grau, mhp, illouz, jacquemin]@limsi.fr

RÉSUMÉ. *Traiter des questions factuelles portant sur n'importe quel sujet dont la réponse appartient à un très large corpus entraîne l'application de processus de Traitement Automatique des Langues (TAL) afin de localiser la réponse dans les documents sélectionnés par un moteur de recherche. Le système QALC, présenté dans cet article, a participé à la tâche Question Answering lors des évaluations TREC8 et TREC9. QALC exploite les résultats d'une analyse des documents où les termes de la question traitée, ainsi que leurs variantes, sont étiquetés et pondérés. Ces éléments permettent de pondérer les documents afin de n'en sélectionner qu'une partie pour la suite des traitements. Par ailleurs, ils permettent aussi d'améliorer la sélection des phrases candidates lorsqu'elles sont comparées à une question afin de décider si elles sont susceptibles de contenir la réponse. Cette comparaison exploite également les résultats d'une analyse des questions indiquant le type de la réponse attendue, qui est rapproché du marquage, dans les phrases, des types d'entités nommées reconnues par le système.*

ABSTRACT. *Answering to open-domain factual questions requires to apply Natural Language treatments to retrieved documents in order to be able to locate the answers inside them. We developed a system, QALC, that participated to the Question Answering track of the TREC8 and TREC9 evaluations. QALC exploits an analysis of documents based on the search for multi-words terms and their variations both to select a minimal number of documents to be processed and to give indices when comparing question and sentence representations. This comparison also takes advantage of a question analysis module and a recognition of named entities in the documents.*

MOTS-CLÉS. *Système de question-réponse, extraction de terme, variante terminologique, entité nommée, recherche d'information*

KEYWORDS. *Question answering system, term extraction, terminological variant, named entity, information retrieval*

1. Introduction

L'introduction de la tâche « Question Answering » lors de l'évaluation TREC8 (Text REtrieval Conference), en 1999, est révélatrice du besoin sans cesse croissant de développer des systèmes de recherche d'information plus sophistiqués capables d'apporter l'information attendue par un utilisateur, celle qui répond le mieux à sa requête. Cette pression entraîne le développement de systèmes qui, au delà de la sélection de documents, en extraient les parties pertinentes, que ce soit en proposant la réponse quand il s'agit d'une question d'ordre factuel, ou un résumé si la requête est d'ordre thématique.

La problématique se situe alors à l'intersection de deux domaines, la recherche d'information et le traitement automatique des langues (TAL). La recherche de documents pertinents est enrichie par l'intégration de modules de TAL s'appliquant à large échelle, *i.e.* indépendamment du domaine abordé, et possédant une grande couverture linguistique. Cette intégration permet au-delà la sélection de passages pertinents en exploitant là aussi des traits linguistiques de nature syntaxique ou sémantique.

Dans cet article, nous présentons notre système de question-réponse, QALC, qui a participé aux évaluations TREC. La campagne TREC8 consistait à proposer 5 réponses ordonnées, à retrouver dans un corpus de 1,5 gigaoctets, pour chacune des 200 questions considérées. La campagne suivante, TREC9, s'est accompagnée d'un changement d'échelle, avec 700 questions et 3 gigaoctets de documents. Les documents sont des articles de journaux américains, tels que le *Wall Street Journal*, le *Financial Times*, etc. L'évaluation proposait deux sous-tâches, l'une pour des réponses de 250 caractères maximum et l'autre pour des réponses de 50 caractères.

Le système comporte des modules TAL dédiés au typage des questions, à la reconnaissance d'entités nommées et à l'extraction de termes simples et complexes des questions afin de les retrouver, éventuellement sous la forme d'une variante, dans les documents. Nous insisterons tout particulièrement sur l'apport du traitement de la variation linguistique, à la fois pour réaliser une post-sélection des documents et pour l'appariement entre une question et une réponse possible. En effet, à partir d'un premier ensemble de documents sélectionnés par un moteur de recherche, QALC réduit le nombre de documents sur lesquels vont s'appliquer des modules de TAL afin de limiter le temps de traitement d'une question. Cette deuxième sélection privilégie les documents comportant des multi-termes ou leurs variantes par rapport aux documents comportant les mots de ces termes épars dans le texte. Concernant le deuxième point, la sélection d'une réponse, nous avons considéré que la phrase est l'unité de base à évaluer. Notre système se place en effet dans une perspective un peu différente de l'évaluation TREC, dont l'évolution va vers la production de la seule réponse avec la référence au document qui la contient. Le fait de viser une réponse courte n'empêche pas la production d'un contexte afin que les utilisateurs puissent juger de la validité de la réponse donnée sans devoir retourner au document pour cela. Le niveau de la phrase semble être à cet égard une solution plus

intéressante qu'une fenêtre fixe entourant la réponse, qui ne constitue pas en soi une unité de sens.

Dans la suite de cet article, après avoir défini la problématique et présenté les grandes approches existantes section 2, nous détaillons l'architecture générale de QALC en section 3 afin de positionner les différents modules qui composent le système et qui sont décrits dans les sections 4 à 8. Section 9, nous présenterons les résultats obtenus lors des évaluations TREC, résultats qui donneront lieu à une discussion permettant de mettre en évidence la nécessité d'appliquer des processus de TAL encore plus ambitieux, mais restant réalistes. Enfin nous décrirons plus précisément les différentes approches adoptées pour ce type de système avant de conclure sur les évolutions envisagées. Tous les systèmes de question-réponse de TREC ont globalement la même approche. Aussi nous avons préféré les exposer après avoir présenté les détails de notre solution, de manière à mettre en parallèle les différentes techniques utilisées de façon plus concrète.

2. Problématique

Lorsqu'on se place dans un paradigme consistant à répondre à des questions sans limitation de domaine, on ne peut, à l'heure actuelle, développer une approche purement TAL, comme cela a été réalisé dans les années 70. La problématique des systèmes capables de répondre à des questions n'est de fait pas récente. Dès les premiers travaux en compréhension de la langue, le problème a été étudié, essentiellement dans le cadre de la compréhension d'histoires. Le système le plus représentatif est celui développé par Lehnert, QUALM (Lehnert, 1977 ; Lehnert, 1979). Ce système analysait des petites histoires sur des sujets très spécifiques (voyager en bus, aller au restaurant, etc.), en stockait une représentation conceptuelle et répondait à des questions en consultant cette mémoire et en raisonnant à partir de ses connaissances générales. Lehnert y proposait une typologie des questions en 13 catégories, dont certains systèmes récents se sont fortement inspirés. À chaque catégorie était associé un type de stratégie de recherche de la réponse dans la base de connaissances. Par exemple, une question portant sur l'antécédent causal d'un événement amenait à rechercher des connaissances de type responsabilité causale. Comme le soulignent Zock et Mitkov (Zock *et al.*, 1991), certaines catégories doivent être précisées afin de mieux définir le type de réponse attendue, et donc la stratégie de recherche associée. Par exemple, la catégorie « concept completion » regroupe toutes les questions WH[1], sans distinction du type de concept à trouver. Par la suite, Lehnert a participé à l'élaboration d'un autre système, Boris (Dyer, 1983), qui différait dans le type de connaissances utilisées, introduisant plus de connaissances pragmatiques. Ces systèmes (cf. (Zock *et al.*, 1991) pour une liste assez complète) reposent sur l'utilisation de connaissances

1 Les questions WH sont celles qui contiennent les pronoms interrogatifs : Who, Whom, Where, What, etc.

génériques très élaborées permettant de décrire les situations prototypiques et d'interpréter le comportement des personnages. Lorsque le système répond à des questions, il développe à la fois une recherche parmi les représentations mémorisées (*i.e.* les épisodes) et un raisonnement à partir de connaissances génériques permettant la production d'inférences.

On retrouve ce type d'approche dans un modèle psychologique de réponse à des questions, QUEST (Graesser *et al.*, 1994), testé à l'aide de méthodes expérimentales et qui considère une grande diversité de questions. Ses auteurs définissent 4 composants pour un modèle de question-réponse :

— une catégorisation des questions. Les auteurs proposent une classification en 18 catégories différenciant les questions amenant des réponses courtes en un mot ou un groupe de mots – par exemple les questions WH où on attend la précision d'un concept – et celles qui entraînent des réponses longues couvrant éventuellement plusieurs phrases, telles que les questions « pourquoi » où la réponse explicite une cause ;

— l'identification des sources d'information nécessaires pour répondre. On trouve ici l'utilisation de connaissances sur les épisodes et de connaissances génériques ;

— un mécanisme de convergence permettant de calculer un sous-ensemble des propositions connues, représentant des faits et des événements ;

— la formulation de la réponse en fonction d'aspects pragmatiques, tels que les buts et la culture commune aux participants.

Ce type d'approche ne peut être complètement appliquée lorsqu'il s'agit de réaliser des systèmes automatiques sans restriction sur les domaines traités : la définition et la formalisation des connaissances pragmatiques nécessaires est impossible à réaliser. Néanmoins, une approche purement TAL peut être réalisée pour un domaine d'application limité, tel que cela a été fait dans le système Extrans (Mollá *et al.*, 2000) qui répond à des questions portant sur les commandes Unix. Extrans repose principalement sur une analyse syntaxico-sémantique du manuel Unix permettant la réalisation d'inférences par la mise en œuvre d'un raisonnement logique. Le domaine étant celui des commandes disponibles dans un système informatique, on peut développer une base de connaissances précises pour le représenter, que ce soit pour le lexique, où on peut limiter les ambiguïtés, ou pour les connaissances sémantiques. Signalons que ce système propose aussi un mode de fonctionnement dégradé en cas d'échec de la première approche, mode reposant sur l'exploitation de mots-clefs.

Le fait de s'intéresser à des questions factuelles et encyclopédiques portant sur des événements dont on a rendu compte, et non à des questions visant à connaître les tenants et les aboutissants d'une histoire, a conduit à poser le problème différemment. Dans ce contexte, il s'agit en effet de retrouver une réponse pouvant figurer explicitement au sein d'un ensemble de textes, ensemble le plus vaste possible car sa taille est généralement en relation avec les chances de trouver une

réponse à la question posée. La base de textes joue donc ici le rôle que jouait la base de connaissances dans les premiers travaux sur le sujet. Une approche de type recherche d'information exploitant uniquement des connaissances statistiques sur le corpus conduit à l'élaboration d'un système capable de répondre à moins de la moitié des questions (Morton, 1999). Il est clairement apparu lors des évaluations TREC que les systèmes étaient d'autant plus efficaces qu'ils utilisaient des traitements linguistiques plus élaborés. Pour tous les systèmes de TREC, la sélection de passages contenant la réponse, de 250 caractères maximum dans ce cas précis, est fondée sur leur proximité par rapport à la question. Cette proximité ne repose pas uniquement sur les mots communs. Elle tient également compte du type de la réponse attendue afin de caractériser les parties de texte recherchées, et de la variation linguistique entre la formulation de la question et l'extrait du document. Aussi, les meilleurs systèmes, dès la première évaluation TREC, comportaient un module d'analyse des questions chargé d'identifier le type de la réponse attendue en corrélation avec un module de reconnaissance d'entités nommées. Lors de la deuxième session, la plupart ont introduit l'utilisation de connaissances sémantiques, principalement via l'utilisation de WordNet (Fellbaum, 1998), afin de tenir compte des variations entre la formulation dans les questions et dans les textes, variations dues à la présence de synonymes et d'hyperonymes.

Dans ces systèmes, des méthodes de type recherche d'information sont utilisées pour sélectionner des passages potentiellement intéressants dans un grand corpus. Cette première sélection permet ensuite d'appliquer des traitements issus du TAL pour analyser plus en détail ces passages, traitements envisageables dès lors qu'ils n'obligent pas à se restreindre à un petit nombre de domaines. C'est ainsi que les modules TAL que nous utilisons dans QALC permettent un typage de la réponse attendue lors de l'analyse des questions, la reconnaissance d'entités nommées correspondant aux types de réponse gérés, et enfin, ce qui est spécifique à notre approche, la gestion de variations terminologiques entre les termes de la question et leur équivalent dans les documents.

3. Architecture du système QALC

L'architecture figure 1 est destinée à positionner chacun des modules présenté par la suite, afin d'expliciter à quel moment ils interviennent dans la chaîne de traitement.

L'analyse des questions (cf. section 4.1) assigne une catégorie aux questions lorsque c'est possible. Les catégories correspondent aux types d'entités nommées pouvant répondre à la question. Par exemple à partir de la question « Quel est le $17^{ème}$ président des Etats-Unis ? », l'analyseur prédit une réponse du type « nom de personne ». Cette analyse permet également d'extraire les termes (cf. section 4.2), simples ou composés de plusieurs mots qui seront recherchés, soit tels quels soit

sous forme d'une variante, dans les documents sélectionnés par un moteur de recherche (cf. section 5 pour une comparaison de différents moteurs).

La sélection d'un sous-ensemble de documents (cf. section 6) repose sur le taux de reconnaissance des termes de la question ou de leurs variantes dans les documents sélectionnés. Les variations recherchées sont de nature morphologique, syntaxique et sémantique. Cette reconnaissance est effectuée par *Fastr* (Jacquemin, 1999). Cette sélection revêt toute son importance lorsque le système applique les processus suivants, à savoir la reconnaissance des entités nommées (cf. section 7) et la comparaison entre phrases et questions (cf. section 8), processus pouvant être consommateurs de temps de calcul.

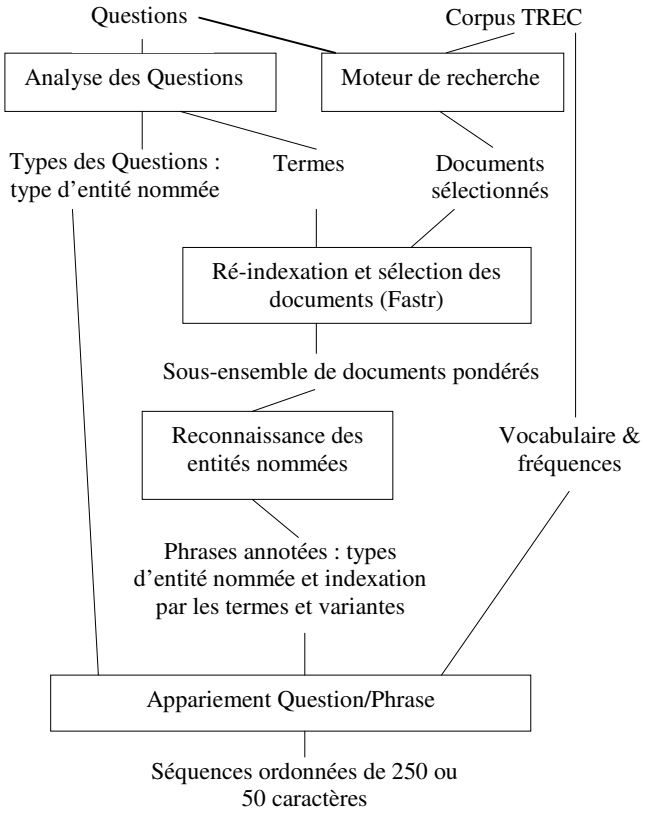


Figure 1. *Architecture du système QALC*

QALC propose des réponses longues, de l'ordre d'une phrase environ, ou des réponses courtes se rapprochant de la seule réponse. Nous nous concentrerons dans le cadre de cet article sur la sélection de réponses longues.

4. L'analyse des questions

L'analyse des questions répond à deux besoins : déterminer le type de la réponse attendue et extraire les termes destinés à ré-indexer les documents sélectionnés en vue de n'en retenir qu'un sous-ensemble et de donner des indices supplémentaires lors de l'appariement final.

4.1. *Détermination du type de réponse*

Connaître le type de la réponse permet de privilégier les phrases qui contiennent un groupe de mots qui lui correspond. Ainsi le typage des questions est-il à mettre en relation avec le module permettant d'étiqueter des expressions, à savoir dans QALC, le module de reconnaissance des entités nommées.

Exemple :

Question : How many people live in the Falklands ? —> type = NUMBER
(Combien de personnes habitent les Falklands ?)

Réponse : Falkland population of <b_numex_TYPE=NUMBER> 2,100 <e_numex> is concentrated…
(La population des Falklands, de 2 100 habitants, est concentrée …)

4.1.1 *Ensemble d'étiquettes*

Les étiquettes utilisées sont présentées en figure 2. Ce sont celles apparaissant dans les feuilles de l'arbre, auxquelles s'ajoutent les étiquettes *nomPropre* et *nombre*. Elles sont similaires aux types définis dans les évaluations MUC (Grishman *et al.*, 1995) pour la tâche d'évaluation des systèmes de reconnaissance d'entités nommées. Le système attribue à chaque question une ou plusieurs étiquettes sur la base de la reconnaissance d'un certain nombre d'indices. Les exemples suivants illustrent cette association « étiquettes : question », les indices permettant leur attribution étant soulignés. QALC détermine ainsi un type de réponse attendue en fonction de types de question.

PERSONNE : Who was the first President of the USA ? *(Qui était le premier président des USA ?)*

ORGANISATION : What laboratory discovered the AIDS virus ? *(Quel laboratoire a découvert le virus du SIDA ?)*

ENDROIT: What is the longest river in Asia ? What is the name of the highest montain in the world ? *(Quelle est la plus longue rivière d'Asie ? Quel est le nom de la plus grande montagne du monde ?)*

VILLE, ENDROIT: Where is Taj Mahal ? (Où est le Taj Mahal ?)

PERIODE : During which period did the dinosaurs vanish ? *(A quelle période les dinosaures disparurent-ils ?)*

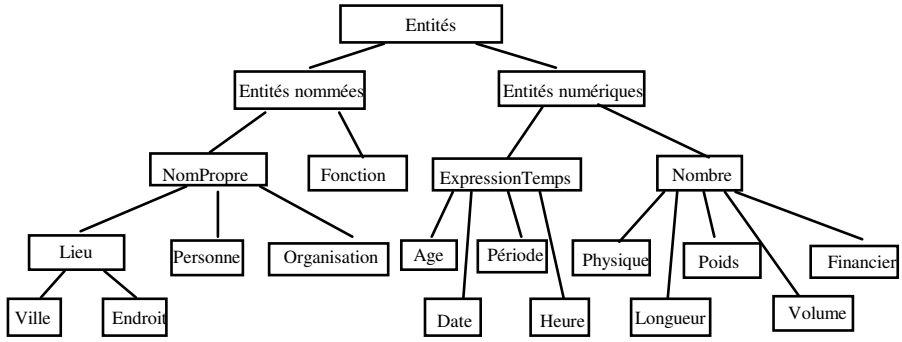


Figure 2. *Hiérarchie de types de réponses et de catégories sémantiques*

4.1.2 *Types de questions*

L'analyse des questions est effectuée par un analyseur dédié. Elle est fondée sur la reconnaissance de règles décrivant différents types syntaxiques de questions. Les indices utilisés dans les règles pour décider de l'attribution d'une étiquette sont :

— d'ordre lexical, avec la détection de mots spécifiques,

— d'ordre syntaxique, avec des critères sur la catégorie syntaxique des mots,

— d'ordre sémantique, avec l'appartenance d'un mot à une catégorie sémantique.

Nous avons identifié sept types de formulation des questions.

TYPE 1 : le type de la réponse ne dépend que du pronom interrogatif, comme c'est le cas avec *who/whom/whose*, *where* et *when*.

Les 3 types suivants sont dédiés à l'analyse des questions *what/which*. Ces règles sont identiques quel que soit le type de la réponse attendue, la détermination du type étant réalisée grâce à la catégorie sémantique du nom de tête du groupe nominal reconnu (*GNsem*). L'ensemble des catégories sémantiques est identique à celui des étiquettes.

TYPE 2 : what/which… be GNsem

TYPE 3 : what/which GNsem

TYPE 4 : what/which is the name of GNsem

Les deux types suivants permettent l'analyse de questions en *How*. Le type 5 conduit à choisir l'étiquette en fonction de la catégorie sémantique de l'adjectif

(*AdjSem*), et le type 6 en fonction de la catégorie du groupe nominal, correspondant ici à un type d'unité permettant de déduire une étiquette d'entité numérique.

TYPE 5 : how AdjSem

TYPE 6 : how much/many GNunité

Enfin le dernier type correspond à la forme impérative « name a film … » et est équivalent au type 4.

La reconnaissance d'un groupe nominal utilise le modèle suivant[2] :

$$(\text{DET})\ (\text{JJ} \mid \text{NPoss} \mid \text{N} \mid \text{V}_{\text{BG}} \mid \text{V}_{\text{BD}})^*\ \text{NSem}$$

en recherchant l'expression la plus longue qui soit conforme à la règle, comme par exemple *Johnny Mathis' high school track coach*. La catégorie sémantique du dernier nom (*NSem*) permet de déterminer l'étiquette attribuée par la règle ; dans l'exemple « *What is the name of Johnny Mathis' high school track coach?* », comme *coach* appartient à la catégorie sémantique *personne*, la règle 4 s'applique et détermine l'étiquette « personne ».

L'analyseur essaie d'appliquer l'une des règles à partir du début de la question, en laissant la possibilité de trouver une préposition en début de phrase dans certains cas. Si aucune règle ne s'applique, les mêmes patrons sont recherchés à l'intérieur des questions, comme dans la question TREC « *The Faroes are part of what northern European country?* ». Dans ce second cas, il y a un risque d'interpréter une proposition relative comme un type de question. Les règles conduisant à attribuer plusieurs étiquettes sont essentiellement celles portant sur le typage des expressions de temps, souvent difficiles à désambiguïser, ainsi que sur le typage de dénomination de lieu (il est souvent impossible de savoir si l'on cherche un nom de ville ou d'endroit).

Le regroupement des noms en catégories sémantiques a été réalisé manuellement, en extrayant les hyponymes d'un type d'étiquette dans WordNet et en filtrant manuellement le résultat. On étiquette ainsi la très grande majorité des questions répondant aux types prévus, soit 60% des 700 questions de TREC9 (seules quelques questions ne sont pas étiquetées, alors qu'elles auraient pu l'être, du fait de l'incomplétude des listes de noms de chaque catégorie).

4.2. *Extraction des termes*

Afin d'acquérir automatiquement des termes à partir des questions, nous utilisons une technique simple de filtrage par des patrons de catégories syntaxiques. Il n'est pas possible de faire un filtrage statistique en sortie de ce module

---

2 DET : déterminant, JJ : adjectif, NPoss : Nom suivi du cas possessif, N : nom, V_{BG} et V_{BD} : verbe au gérondif ou au participe passé, * indiquant un nombre quelconque de l'un ou de plusieurs des éléments entre parenthèses.

d'acquisition car les corpus de questions sont trop petits pour permettre un tel procédé.

Tout d'abord, les questions sont segmentées, étiquetées et lemmatisées par le *TreeTagger* (Schmid, 1999). Des patrons de catégories syntaxiques sont ensuite utilisés pour extraire des termes des corpus étiquetés. Ces patrons ne diffèrent de ceux définis par Justeson et Katz (Justeson *et al.*, 1995) que par le fait que nous ne prenons pas en compte les syntagmes prépositionnels postposés. Les patrons utilisés en acquisition de termes sont donc :

```
( ( ( ( JJ | NN | NP | VBG ) ) ? ( JJ | NN | NP | VBG ) ( NP | NN ) ) ) |
                    ( VBD ) | ( NN ) | ( NP ) | ( CD ) )
```

La chaîne la plus longue est acquise en premier et les sous-chaînes ne peuvent être extraites que si elles ne commencent pas par le même mot que la surchaîne. Par exemple, la séquence $name_{NN}$ of_{IN} the_{DT} US_{NP} $helicopter_{NN}$ $pilot_{NN}$ $shot_{VBD}$ $down_{RP}$ (nom du pilote d'hélicoptère américain abattu*)*, les quatre termes suivants sont acquis : *US helicopter pilot* (pilote d'hélicoptère américain), *helicopter pilot* (pilote d'hélicoptère), *pilot* (pilote) et *shoot* (abattre).

Le mode d'acquisition choisi pour les termes revient à ne prendre en compte que les sous-structures qui correspondent à un attachement de modifieur aux constituants les plus à droite (les plus proches pour la syntaxe anglaise des noms composés). Par exemple, la décomposition de *US helicopter pilot* en *helicopter pilot* et *pilot* équivaut à extraire les sous-constituants de la structure [*US* [*helicopter* [*pilot*]]].

5. Le moteur de recherche

Le premier module de la chaîne de traitement des documents est la sélection, par un moteur de recherche, de documents répondant aux questions. Nous avons testé trois moteurs de recherche différents sur les 200 questions qui avaient été proposées aux participants à la tâche QA de la conférence TREC8. Le premier est *Zprise*, un moteur de recherche de type vectoriel qui a été développé au NIST. Nous l'avons utilisé avec le paramétrage suivant : fonction de pondération bm25idf d'Okapi (Robertson *et al.*, 1999), stemming par l'algorithme de Porter, pas de retour de pertinence. Le second est *Indexal*, un moteur de recherche de type pseudo-booléen, qui nous a été fourni par Bertin Technologies, une société française d'étude et de conseil en technologies qui a développé ce moteur en partenariat avec le Laboratoire d'Informatique d'Avignon (LIA). L'utilisation habituelle d'*Indexal* est la recherche d'information dans une base de données de petite taille avec une indexation des paragraphes de chaque document. Nous avons exploité la capacité d'*Indexal* à réaliser du stemming ainsi que sa notion d'affinité entre mots (de Loupy *et al.*, 1998), une des spécificités de ce moteur. Celle-ci permet de considérer une requête comme un ensemble de mots devant être trouvés à l'intérieur d'une fenêtre de taille donnée. Le dernier moteur de recherche que nous avons testé est le moteur qu'ATT

utilise pour la tâche AdHoc de TREC. Dans ce cas, seuls ses résultats rendus disponibles par les organisateurs de la conférence TREC ont été accessibles, résultats sous la forme de la liste des mille premiers documents pour chaque question, dans l'ordre décroissant de pertinence.

Ces moteurs de recherche renvoient une liste, ordonnée par pertinence décroissante, des documents répondant à une question. L'un des objectifs de ces tests était tout d'abord de déterminer le nombre optimal de documents à retenir à l'issue de la recherche par le moteur. En effet, un trop grand nombre de documents allonge le temps de traitement des questions qui peut alors devenir prohibitif. En revanche, un trop petit nombre de documents diminue les chances d'obtenir la bonne réponse. L'autre objectif était évidemment de déterminer le meilleur moteur, c'est-à-dire celui qui donnait le maximum de documents contenant les réponses.

5.1. *Seuil de sélection des documents*

Nous avons testé le nombre de documents à sélectionner avec le moteur de recherche *Zprise*, en retenant respectivement les 50 premiers documents, puis 100, 200, et enfin les 500 premiers documents. Nous avons évalué les résultats des tests à l'aide de la liste des réponses correctes fournie par les organisateurs de la conférence TREC8. Le tableau 1 présente les résultats de ces tests.

Nombre de documents retenus	**50**	**100**	**200**	**500**
Nombre de questions pour lesquelles au moins un « bon document » a été trouvé	181	184	193	194
Nombre de questions pour lesquelles aucun « bon document » n'a été trouvé	19	16	7	6

Tableau 1. *Comparaison entre les nombres de « bons documents » trouvés pour des seuils différents de documents retenus*

Le tableau 1 montre que l'amélioration des performances du moteur tend à diminuer au-delà du seuil de 200 documents retenus. Dans le traitement des questions de la tâche QA de TREC9, nous avons donc opté pour un seuil de 200 documents, qui nous a semblé offrir le meilleur compromis entre le nombre de « bons documents » obtenus et le temps de traitement d'une question.

5.2. *Performances des moteurs de recherche*

Nous avons comparé les résultats des trois moteurs de recherche pour le seuil des 200 premiers documents sélectionnés. Le tableau 2 donne les performances des moteurs.

Moteur de recherche	**Indexal**	**Zprise**	**ATT**
Nombre de questions pour lesquelles au moins un « bon document » a été trouvé	182	193	194
Nombre de questions pour lesquelles aucun « bon document » n'a été trouvé	18	7	6
Nombre total de « bons documents » trouvés	814	931	1021

Tableau 2. *Performances comparées des moteurs de recherche Indexal, Zprise, et ATT*

Le moteur ATT s'est révélé le plus performant suivant les trois critères que nous avions retenus : le plus grand nombre de questions pour lesquelles tous les bons documents ont été trouvés, le plus petit nombre de questions pour lesquelles aucun « bon document » n'a été trouvé, et le plus grand nombre global de « bons documents » trouvés.

6. Ré-indexation et filtrage de documents

La sélection des documents pertinents repose sur une indexation faisant appel à des techniques de traitement automatique des langues. Les index sont composés de termes simples et de termes composés de plusieurs mots et de liens linguistiques entre les occurrences trouvées dans les documents et les termes d'origine. Les termes de référence sont ceux extraits des questions selon la technique présentée au paragraphe 4.2 ainsi que les mots pleins de la question. L'outil utilisé pour extraire les séquences textuelles qui correspondent aux occurrences de termes et de leurs variantes est *Fastr* (Jacquemin, 1999). Ces index sont utilisés pour le classement des documents par une combinaison pondérée des occurrences de termes et de leurs variantes (cf. 6.2). Ce classement est finalement exploité pour sélectionner les documents *a priori* les plus pertinents dans lesquels sont recherchées les réponses aux questions. Par la réduction du volume textuel qu'elle induit, cette sélection permet de mettre en œuvre pour cette recherche finale des méthodes d'analyse plus élaborées et donc également plus coûteuses en termes de calcul.

6.1. *Indexation par Fastr*

L'indexation automatique des documents est faite par *Fastr*, un analyseur transformationnel surfacique pour la reconnaissance de variantes terminologiques. Les termes sont transformés en règles de grammaire et les mots simples qui les composent sont stockés dans un lexique contenant des liens morphologiques et sémantiques.

La *famille morphologique* d'un mot simple *m* est l'ensemble *M*(*m*) des mots simples de la base CELEX (Celex, 1998) qui ont la même racine que *m*. Par exemple, la famille morphologique du nom *maker* (fabricant) se compose des noms *maker*, *make* (marque) et *remake* (remake), et des verbes *to make* (faire) et *to remake* (refaire).

La famille sémantique d'un mot simple *m* est l'union *S*(*m*) des *synsets* de WordNet 1.6 auxquels ce mot *m* appartient. Un synset est l'ensemble des mots qui partagent un lien de synonymie sur une de leurs entrées sémantiques. Ainsi, la famille sémantique d'un mot *m* est l'ensemble des mots *m'* tels que *m'* est synonyme de *m* pour au moins un de ses sens. Cette notion de famille intègre le fait qu'il n'y a pas de désambiguïsation sémantique des mots dans les textes. La famille sémantique de *maker* obtenue à partir de WordNet 1.6 se compose ainsi de trois noms : *maker*, *manufacturer* (fabricant), *shaper* (façonneur) et la famille sémantique de *car* (voiture) est *car*, *auto*, *automobile*, *machine* et *motorcar* (voiture à moteur).

Les patrons de variations qui reposent sur des familles morphologiques et sémantiques sont générés au moyen de métarègles. Ils sont utilisés pour extraire les termes et les variantes des phrases des documents du corpus TREC.

Le patron suivant (RP sont les particules, PREP les prépositions, ART les articles et V les verbes) extrait l'occurrence *making many automobiles* (fabricant de nombreuses voitures) comme variante de *car maker* (fabricant de voitures) :

```
VM('maker') RP? PREP? (ART (NN|NP)? PREP)? ART?
    (JJ | NN | NP | VBD | VBG)0-3 NS('car')
```

où `VM('maker')` est tout verbe de la famille morphologique du nom *maker* et `NS('car')` tout nom de la famille sémantique de *car*.

En s'appuyant sur les familles morphologiques et sémantiques respectivement extraites de CELEX et WordNet 1.6 et sur le jeu de métarègles pour l'anglais, les occurrences suivantes sont extraites comme des variantes du terme d'origine *car maker* (fabricant de voitures) :

auto maker (faricant d'autos), *auto parts maker* (fabricant de pièces détachées automobiles), *car manufacturer* (fabricant de voitures), *make autos* (fabriquer des voitures) et *making many automobiles* (fabricant beaucoup de voitures)

Quelques variantes incorrectes sont également extraites par cette méthode, comme *make those cuts in auto* (faire ces coupes dans les [...] automobiles) produite par la métarègle précédente.

6.2. *Filtrage des documents*

La sortie de l'indexation par analyse linguistique automatique des documents est une liste d'occurrences de termes composées d'un identificateur de document *d*, d'un identificateur de termes — une paire *t*(*q, i*) composée d'un numéro de question *q* et d'un indice unique *i* —, une séquence textuelle et un identificateur de variation *v* (une métarègle). Par exemple, l'index suivant :

```
LA092690-0038    t(131,1)    making many automobiles
    NtoVSemArg
```

signifie que l'occurrence *making many automobiles* du document *d*=LA092690-0038 est obtenue comme une variante du terme 1 de la question *q*=131 (car maker) au moyen de la variation NtoVSemArg donnée en section 6.1.

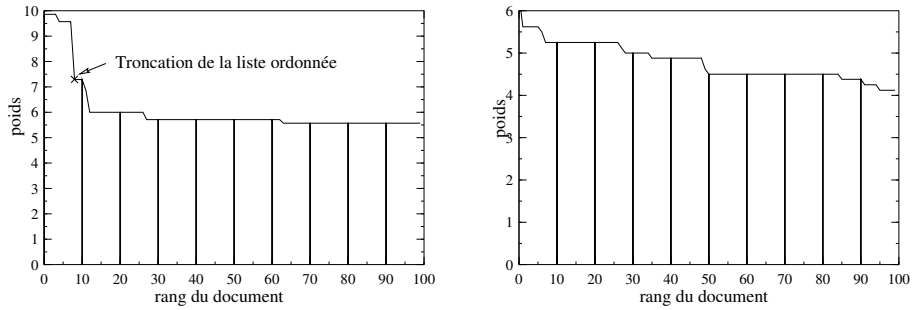
Chaque document *d* sélectionné pour une question *q* par le moteur de recherche utilisé reçoit un poids. La fonction de pondération repose sur une mesure de qualité des différentes familles de variations décrites dans (Jacquemin, 1999). Le poids *w(v)* d'une occcurrence de terme est ainsi d'autant plus important que celle-ci représente une faible variation par rapport au terme originel : il a été empiriquement fixé à 3 en l'absence de variation, à 2 dans le cas des variantes morphologiques et morpho-syntaxiques et à 1 pour les variantes sémantiques et morpho-sémantico-syntaxiques.

Les noms propres représentent des indices importants en recherche d'information dans un corpus tel que celui de TREC composé majoritairement d'articles de journaux. Chaque terme *t*(*q*, *i*) reçoit un poids *P*(*t*(*q*, *i*)) entre 0 et 1 correspondant à sa proportion de noms propres. Par exemple, *president Cleveland's wife* (la femme du président Cleveland) a un poids de 1/3=0,34. Enfin, un dernier facteur de fiabilité est la longueur du terme, il est représenté par la quantité |*t*(*q*, *i*)| dans la formule de pondération des termes ci-dessous et correspond au nombre de mots du terme de référence *t*(*q*, *i*).

Le poids $W_q(d)$ d'un document *d* vis-à-vis d'une question *q*, donné par la formule [1], est obtenu en sommant les pondérations des index extraits du document *d* par l'indexeur (ensemble *I*(*d*) des termes de *q* reconnus dans *d*) et en normalisant la somme résultante par le nombre de termes |*T*(*q*)| de la question *q*.

$$W_q(d) = \sum_{(t(q,i),v)\in I(d)} \frac{w(v)\times(1+2P(t(q,i)))\times|t(q,i)|}{|T(q)|} \qquad [1]$$

Pour chaque question q, les 100 documents les mieux classés sont sélectionnés. On observe principalement deux types de courbes de pondération parmi les documents sélectionnés : les courbes avec un plateau et une chute brutale des valeurs des poids au-delà d'un certain rang (figure 3.a) et les courbes avec des valeurs de poids en décroissance progressive (figure 3.b), les questions correspondant aux courbes de la figure 3 sont issues des données d'entraînement de la conférence TREC8.



(a) Question numéro 87 **(b)** Question numéro 86

Figure 3. *Deux types de courbes de pondération des documents d'une question*

Dans le cas d'une courbe ayant un profil semblable à celui de la figure 3.a, le seuil correspondant à la chute du poids est détecté en analysant simultanément la pente de la courbe (la différence entre le poids d'un document et le poids du document précédent) et la variation de second ordre (la différence entre la pente à une étape et la pente à l'étape précédente). La fonction de décision suivante est utilisée pour calculer le seuil de sélection des documents i_0 correspondant à la fonction de pondération W d'une question q :

$$\text{si } \frac{W_q(d_2)}{W_q(d_1)} < 0.5 \text{ alors } i_0 = 2$$
$$\text{sinon}$$
$$i_0 = \min\left\{ i \in \{3 \ldots 100\} \,.\, \left(\frac{W_q(d_i) - W_q(d_{i-1})}{W_q(d_{i-1}) - W_q(d_{i-2})} \right) > 2 \;\wedge\; \frac{W_q(d_i)}{W_q(d_{i-1})} < 0.8 \right\} \cup \{100\} \quad [2]$$

Cette fonction de décision sélectionne ainsi un nombre restreint de documents lorsqu'un décrochement significatif (pente inférieure à 0,5 dès les premiers documents ou bien variation de second ordre supérieure à 2 et pente inférieure à 0,8

pour les suivants) est observé dans la courbe de pondération des documents ou retient arbitrairement les 100 premiers documents de ce classement.

Par cette méthode, le seuil i_0 est fixé à 8 pour la question 87 (*Who followed Willy Brandt as chancellor of the Federal Republic of Germany?*, figure 3.a) [Qui a remplacé Willy Brandt comme chancelier de la République Fédérale d'Allemagne ?] et à 100 pour la question 86 (*Who won two gold medals in skiing in the Olympic Games in Calgary?*, figure 3.b) [Qui a gagné deux médailles d'or au ski aux Jeux Olympiques de Calgary ?]. Comme indiqué par la figure 3.a, il y a une différence importante de poids entre les documents classés 8 et 9. Le poids du document 8 est égal à 9,57 alors que le poids du document 9 n'est que de 7,29 parce que le terme *Federal Republic* (République Fédérale) n'apparaît que dans le document 8. Ce terme a une forte pondération car il est composé de deux noms propres.

Finalement, le système sélectionne les i_0 documents les mieux classés avec un nombre minimal de documents fixé expérimentalement à 20.

6.3. *Évaluation*

Nous avons réalisé plusieurs tests en vue d'évaluer l'efficacité du filtrage et pour juger ainsi globalement de l'intérêt des termes et des variantes trouvés ainsi que de la pondération que nous avons choisie. Dans un premier temps, nous avons appliqué la chaîne de traitement décrite ci-dessus sur les données de TREC8, une fois avec le processus de filtrage, et une autre sans cette sélection. Sans filtrage, les 200 documents résultat du moteur de recherche étaient donc conservés pour chacune des 200 questions. Le système a obtenu un score de 0,463 dans le premier cas (avec filtrage), et de 0,452 dans le second cas[3] (sans filtrage). Ces résultats montrent que les performances ne diminuent pas, et même augmentent légèrement, quand on traite moins de documents, et que ce sont donc les documents les plus pertinents qui sont gardés.

L'intérêt de procéder à une telle sélection est aussi illustré par les statistiques du tableau 3, statistiques calculées à partir des résultats obtenus pour TREC9. Nous pouvons voir que le processus opère une importance sélection pour 50% des questions (340 questions sont traitées en gardant moins de 100 documents), et que

---

3 Les scores donnés ici correspondent à la mesure d'évaluation utilisée dans le cadre de l'évaluation Question Answering de TREC. Pour un ensemble de *n* questions, le score d'un système S est donné par :

$$score(S) = \frac{1}{n}\sum_{i}^{n}\frac{1}{rang(i)}$$

avec *rang(i)*, le plus haut rang (entre 1 et 5) d'une bonne réponse trouvée par S pour la question *i*., sachant qu'un système fournit pour chaque question un ensemble de 5 réponses ordonnées par ordre décroissant de pertinence. Si le système S n'a pas trouvé de réponse pour la question *i*, son score pour cette question est égal à 0.

cette sélection ne nuit pas à la suite du traitement puisque QALC trouve plus souvent une réponse correcte à ces 340 questions, qui, de plus, est plus souvent mieux placée. Le nombre moyen de documents retenus pour ces 340 questions est 37. Ces résultats indiquent que l'application de processus coûteux en temps, tels que la recherche des entités nommées ou l'appariement entre les questions et les phrases des documents retenus, est possible sans que cela nuise trop au temps de traitement global. Ils permettent aussi d'envisager l'application de traitements encore plus coûteux, tels qu'une analyse syntaxico-sémantique des phrases, sans entraîner pour autant des temps de traitement prohibitifs.

Nombre de documents retenus après filtrage	**100**	**<< 100**
Répartition des questions	342 (50%)	340 (50%)
Nombre de réponses correctes, *i.e.* présence d'au moins une réponse correcte parmi les 5 propositions	175 (51%)	200 (59%)
Nombre de réponses correctes au rang 1 (pourcentage par rapport au nombre de réponses correctes)	88 (50%)	128 (64%)

Tableau 3. *Evaluation du processus de filtrage après ré-indexation des documents*

7. Reconnaissance des entités nommées

Les entités nommées constituent des indices supplémentaires lors de la recherche d'une phrase contenant la réponse et sont véritablement très importantes lorsque l'on veut ne sélectionner que le groupe de mots formant la réponse. Les entités nommées sont marquées par une balise dont le type correspond aux types de réponses présentées en figure 2 ; pour mémoire, nous cherchons à reconnaître les expressions désignant des personnes, des organisations, des lieux et des valeurs. Les types retenus sont définis de manière similaire à ce qui a été fait dans les évaluations MUC (Grishman *et al.*, 1995) et sont reconnus à partir d'une combinaison de :

— consultations de lexiques (pour trouver des traits syntaxiques et sémantiques associés aux mots simples) et de règles exploitant ces traits en complément de traits lexicaux, et

— recherches dans des dictionnaires d'entités nommées.

Les ressources utilisées sont CELEX (Celex, 1998), un lexique de 160 595 mots fléchis auxquels sont associés leur lemme et leur catégorie syntaxique, une liste de 8 070 prénoms (6 763 provenant de l'archive de CLR (Clr, 1998)) et une liste de 211 587 noms de familles (provenant aussi de CLR), une liste de 22 095 entreprises provenant du « Wall Street Research Network » et 649 noms d'organisations

obtenus à partir d'une acquisition lexicale sur Internet (Jacquemin *et al.*, 2000), deux listes dédiées aux noms de lieux, l'une de 7 813 villes et l'autre de 1 144 pays issus de CLR, plus des listes constituées manuellement sur les unités physiques et monétaires.

7.1. *Entités numériques*

Cette catégorie regroupe toutes les expressions de temps et de valeurs, même si elles ne sont pas formulées à l'aide de nombres. La reconnaissance de ces entités s'effectue en trois étapes :

— la reconnaissance des nombres cardinaux et ordinaux, exprimés en lettres ou en chiffres,

— l'application de règles de reconnaissance des expressions complexes formées de nombres et de leurs unités, telles que les distances, les valeurs monétaires, etc.,

— la reconnaissance des expressions de temps à partir des nombres n'ayant pas donné lieu à des expressions complexes,

— enfin, les nombres ne participant pas à l'expression d'une entité spécialisée sont simplement étiquetés *Nombre.*

7.2. *Organisations, personnes et lieux*

Lorsqu'une expression n'appartient pas à l'une des listes utilisées, QALC applique un ensemble de règles dédiées à chaque type d'entité nommée. Pour les organisations, QALC recherche des expressions commençant par un modificateur spécifique, tels que *Democratic, Federal*, etc. ou possédant un nom de tête du type *Academy, Administration, Association*, etc. En ce qui concerne les noms de personne, leur reconnaissance repose sur le fait de trouver au moins deux mots répondant à certaines contraintes. Les règles exploitent des critères typographiques comme la présence d'une première lettre en majuscule pour un mot qui suit un prénom, ou la reconnaissance de titres, tels que *Dr, President, Ayatollah*, etc. avant un mot en majuscule. Un mécanisme d'apprentissage endogène est utilisé afin de reconnaître toutes les occurrences d'un nom repris seul dans un document où il a été précédemment identifié au sein d'une expression du type <*prénom nom*>.

8. Appariement question/phrase

Le module d'appariement question/phrase est chargé de sélectionner pour chaque question soumise au système QALC une liste réduite[4] de N_r phrases extraites des documents conservés à l'issue de la phase de filtrage et susceptibles de contenir

4 N_r est égal à cinq dans le cas de l'évaluation QA.

la réponse à la question posée. Ces phrases sont ordonnées par ordre de pertinence décroissante. Cette sélection s'appuie sur les résultats des modules linguistiques appliqués précédemment :

— à chaque question sont assignés un ensemble de termes ainsi qu'une ou plusieurs catégories caractérisant le type de la réponse attendue ;

— dans chacun des documents sélectionnés pour une question donnée, les entités nommées ainsi que les occurrences des termes de la question ont été repérées et marquées.

8.1. *Principe du module d'appariement question/phrase*

Le module d'appariement considère la phrase en tant qu'unité de réponse de base. Ce choix nous apparaît comme le compromis le plus naturel entre une nécessaire concision de la réponse et la possibilité de conserver un contexte suffisamment large pour que l'utilisateur d'un tel système puisse juger de la validité de la réponse sans avoir à retourner au texte. Même si les évaluations TREC ont montré qu'il est possible d'obtenir d'assez bons résultats sur une telle tâche, elles ont aussi montré l'existence d'un taux d'erreur significatif, en particulier lorsque l'on cherche à diminuer la taille de la réponse. Dans l'optique d'un système opérationnel, *i.e.* utilisé par un large public, il nous paraît indispensable de fournir les moyens de juger rapidement de la validité des réponses fournies.

Le système recherche les réponses dans l'intégralité des documents retenus, et les documents sélectionnés pour une question sont considérés les uns à la suite des autres. Le module d'appariement conserve en permanence la liste des N_r phrases les plus à même de contenir la réponse à la question posée et compare chaque nouvelle phrase aux éléments de cette liste : si l'une de ces comparaisons tranche en faveur de la nouvelle phrase, elle est insérée dans la liste à sa place et la dernière phrase de la liste est éliminée ; sinon, la nouvelle phrase est laissée de côté.

Chaque phrase est transformée en un vecteur regroupant les informations qui permettent d'établir son degré de proximité avec la question. Ce vecteur contient trois types d'éléments, mis en évidence par les traitements linguistiques ayant précédé l'application du module d'appariement :

— les mots simples composant la phrase ;

— les termes de la question ou leurs variantes ayant été reconnus par *Fastr* dans cette phrase ;

— les entités nommées présentes dans la phrase.

Chacun des éléments composant le vecteur est pondéré en fonction de son importance relative vis-à-vis des autres éléments du même type ou de son degré de similarité par rapport à un élément analogue de la question. Un score est calculé pour chaque type d'éléments à partir de ces poids et rend compte de la proximité de la phrase avec la question selon le point de vue représenté par ce type d'éléments.

Ces scores sont ensuite combinés afin de rendre un avis global sur la proximité des deux phrases concernées. Nous avons expérimenté deux mesures, l'une intégrant globalement les différents scores, l'autre opérant cette intégration de façon plus différenciée en hiérarchisant les critères pris en compte (cf. 8.2.4).

8.2. *Ordonnancement des phrases candidates*

8.2.1 *Poids et score des mots simples*

Le vecteur représentant une phrase contient la forme lemmatisée des mots pleins de cette phrase, *i.e.* les adjectifs (y compris les comparatifs et les superlatifs), les verbes, les noms communs et les noms propres, les adverbes, les sigles et abréviations ainsi que les nombres. La lemmatisation et la désambiguïsation morpho-syntaxique sont réalisées par le *TreeTagger* (Schmid, 1999). Les mots de la phrase qui ne sont pas présents dans la question reçoivent un poids nul. Les autres se voient attribuer un poids en rapport avec leur degré de spécificité. Nous avons expérimenté la pondération *tf.idf*, classiquement utilisée en recherche d'information, ainsi que la significativité, utilisée notamment dans (Kozima, 1993) et qui correspond à l'information normalisée du mot par rapport à un corpus.

Le score d'une phrase du point de vue des mots simples est donné par la somme de ces poids.

8.2.2 *Poids et score des termes*

Nous considérons ici les expressions reconnues par *Fastr* dans les phrases des documents, soit les termes complexes ou leurs variantes, et les variantes des termes simples ne relevant pas de la morphologie flexionnelle[5]. Dans la suite nous nommerons terme pour la phrase aussi bien les termes que les variantes trouvées par *Fastr*. Lorsque plusieurs termes se recouvrent (termes inclus les uns dans les autres), nous ne retenons que le terme ayant obtenu le plus fort score du point de vue de *Fastr*. Nous attribuons un poids à chacun des termes conservés et à l'instar des mots simples, le score de la phrase selon cette dimension est donné par la somme de ces poids.

La base du poids des termes est constituée par le score que leur attribue *Fastr*. Nous avons également expérimenté l'introduction d'une modulation de ce poids tenant compte de deux facteurs : d'une part, le fait qu'un terme corresponde ou non à un nom propre et d'autre part, le degré de spécificité d'un terme lorsque celui-ci est un terme simple. Le but est de renforcer les noms propres et de prendre en compte pour les variantes des mots simples les mêmes informations que pour ces derniers. Avec cette modulation, le poids d'un terme est donné par la formule :

---

5 Le cas de la morphologie flexionnelle est couvert par le paragraphe précédent.

$$\text{poids}(T) = \text{score\_fastr}(T) + \left(\frac{\text{score\_fastr}(T) \cdot \text{modulateur\_np}(T) \cdot \text{spécificité}(T)}{\text{force\_modulation}}\right) \quad [3]$$

modulateur_np(T) est plus fort lorsque *T* est un nom propre,

spécificité(T) est égal selon l'option prise (cf. paragraphe 8.2.1) au facteur *tf.idf* de T ou à sa significativité,

force_modulation est une valeur fixe déterminant globalement l'importance que l'on accorde à la modulation du score de *T*.

8.2.3 *Poids et score des entités nommées*

Le score relatif aux entités nommées a pour fonction de déterminer si une entité nommée correspondant au type attendu de la réponse est présente ou non dans la phrase considérée. Dans sa version la plus élémentaire, ce score ne prend que deux valeurs : il est égal au poids de l'entité nommée dont le type est identique à l'un des types attendus de la réponse dans le cas où une telle entité existe dans la phrase ; il est nul dans le cas contraire. Le poids associé aux entités nommées est le même pour toutes les entités.

Comme pour les termes, nous avons aussi testé une méthode de pondération plus élaborée des entités nommées. Cette pondération intègre deux facteurs :

— le premier d'entre eux est la distance entre le type de l'entité considérée et le type de la question dans la hiérarchie présentée au paragraphe 4.1.1. Le module de typage des réponses attendues attribue toujours le type le plus spécifique possible à chaque question. En revanche, le module de reconnaissance des entités nommées n'est pas toujours aussi fin dans le typage des entités qu'il identifie. Afin d'éviter un rappel trop faible, on considère aussi comme une réponse possible une entité nommée dont le type est plus général dans la hiérarchie des types d'entités que le type attendu de la réponse. Néanmoins, le poids d'une entité nommée est d'autant plus faible que son type est éloigné de celui défini pour la réponse ;

— le second facteur pris en compte est la position de l'entité considérée par rapport aux mots de la question présents dans la phrase traitée. Les phrases pouvant être longues, on considère en effet qu'une entité nommée trop éloignée des mots en rapport avec la question posée a peu de chance d'être une réponse à cette question.

Le poids d'une entité nommée *EN* combine ces deux facteurs au travers de la formule suivante :

$$\begin{aligned}\text{poids\_entité}(EN) = {} & \textit{taille\_hiérarchie} - \text{distance}(\textit{type\_réponse}, EN) \\ & + \text{proximité\_réponse}(EN)\end{aligned} \quad [4]$$

avec

hauteur_hiérarchie : nombre de niveaux de la hiérarchie des types d'entités, égal à 3 dans notre cas ;

distance(*type_réponse*, *EN*) : nombre de niveaux séparant le type de *EN* et le type attendu de la réponse ;

proximité_réponse*(EN)* : distance en nombre de mots entre EN et la partie de la phrase regroupant les mots de la question présents dans cette phrase. Dans le cas présent, cette fonction ne prend que deux valeurs : elle est égale à 1 si cette distance ne dépasse par 4 mots ; sinon, sa valeur est nulle.

Avec cette méthode de pondération, le score d'une phrase du point de vue des entités nommées est donné par le poids le plus élevé parmi ceux des différentes entités de la phrase.

8.2.4 *Évaluation d'une phrase*

L'évaluation d'une phrase s'effectue à partir des trois scores définis précédemment. La première méthode que nous avons retenue a été de les combiner directement en réalisant leur somme pondérée (cf. cœfficients α, β, et γ dans [5], empiriquement fixés à α = 1,0, β = 0,5 et γ = 0,5) pour chaque phrase *P*.

$$score(P) = (score\_mots(P) \bullet \alpha) + (score\_termes(P) \bullet \beta) + (score\_entités(P) \bullet \gamma) \quad [5]$$

Nous allons montrer sur l'exemple de la question *What two US biochemists won the Nobel Prize in medicine in 1992?,* qui a été proposée à TREC8, comment chaque phrase est évaluée. La question est d'abord transformée en un vecteur :

two (1,0)	us (1,0)	biochemist (0,9)	nobel (1,0)
prize (0,6)	medicine (0,5)	win (0,3)	1992 (1,0)
<PERSON> (0,5)	16.01 (0,5)	16.04 (0,5)	

où <PERSON> est le type attendu de la réponse, 16.01 est l'identificateur du terme *US biochemist* et 16.04 est l'identificateur du terme *Nobel Prize*.

Le même type de vecteur est construit pour chaque phrase du document FT924-14045, sélectionné pour cette question. Par exemple, la phrase étiquetée par le module des entités nommées : *<NUMBER> Two </NUMBER> US biochemists, <PERSON> Edwin Krebs </PERSON> and <CITY> Edmond </CITY> Fischer, jointly won the <NUMBER> 1992 </NUMBER> Nobel Medicine Prize for work that could advance the search for an anti-cancer drug* donne le vecteur suivant :

two (1,0)	us (1,0)	biochemist (0,9)	nobel (1,0)
prize (0,6)	medicine (0,5)	win (0,3)	1992 (1,0)
edwin (0,0)	krebs (0,0)	edmond (0,0)	fischer (0,0)
work (0,0)	advance (0,0)	search (0,0)	anti-cancer (0,0)
jointly (0,0)	drug (0,0)	<PERSON> (0,5)	<NUMBER> (0,0)
<CITY> (0,0)	16.01 (0,5)	16.04 (0,3)	

où le poids 0,0 est donné aux éléments qui ne font pas partie du vecteur représentant la question. Le terme *US biochemist* est trouvé sans variation et *Nobel Prize* apparaît sous la variante syntaxique *Nobel Medicine Prize*. Finalement, en appliquant [5], on trouve une mesure de similarité de 0,974 entre les deux vecteurs.

Lorsque les scores de deux phrases sont très proches, *i.e.* la différence entre les deux scores est inférieure à un seuil fixé, l'incertitude est jugée suffisamment forte pour faire appel à un second critère. Les phrases pouvant être très longues[6] et les réponses attendues étant en revanche plutôt compactes, nous favorisons les phrases dans lesquelles les mots de la question présents dans la phrase sont les moins dispersés. Nous évaluons cette dispersion par la taille, en nombre de mots, de la partie de la phrase regroupant tous les mots de la question.

En dépit de la présence des cœfficients modulateurs dans [5], la combinaison des scores réalisée par cette première méthode limite les possibilités d'une prise en compte différenciée des dimensions que nous considérons. C'est pourquoi nous avons testé un autre mode de combinaison des scores. Cette méthode différentielle émet un avis initial fondé uniquement sur la somme du score des mots simples et du score des termes. Si cette somme pour une phrase *P1* est significativement plus grande que cette même somme pour une phrase *P2*, *i.e.* la différence entre ces deux sommes est supérieure à un seuil fixé, alors *P1* est classée avant *P2*. Si la différence entre les deux phrases est inférieure au seuil fixé, on classera ces phrases selon leur ordre du point de vue des entités nommées. Compte tenu du faible nombre des valeurs possibles pour le score des entités nommées, les cas d'égalité ne sont pas exceptionnels. Pour les départager, on revient au premier critère mais en baissant la valeur du seuil. Le critère assurant la décision finale en cas de besoin est le même que pour la première méthode : on donne la préférence aux phrases dans lesquelles les mots de la question sont les moins dispersés.

8.2.5 *Pour des réponses plus précises*

Comme nous l'avons déjà souligné, nous avons choisi la phrase en tant qu'unité de base comme un compromis acceptable entre les contraintes d'une utilisation interactive et la concision de la réponse. Lorsque l'on souhaite améliorer cette dernière, comme c'est le cas par exemple pour les évaluations *Question Answering* de TREC, nous nous appuyons sur un ensemble d'heuristiques simples pour réduire la taille des phrases dépassant la limite fixée. Lorsqu'une entité nommée correspond au type attendu de la réponse ou à un type plus général, QALC sélectionne la partie de la phrase entourant cette entité nommée. Dans le cas contraire, ou dans le cas où le type attendu de la réponse n'a pu être déterminé, il extrait une partie de la phrase contiguë à celle contenant les mots de la question. On suppose ainsi que la phrase contenant la réponse possède une structure comparable à ce que serait la forme affirmative de la question posée.

---

6 C'est assez souvent le cas dans les corpus de textes journalistiques sur lesquels nous avons travaillé.

8.2.6 *Tests et discussion*

Nous avons testé le module d'appariement question/phrase dans deux configurations : une configuration dans laquelle la méthode globale de mesure de similarité a été retenue avec les calculs les plus élémentaires pour les poids des éléments représentant chaque phrase, les scores par dimension ainsi que pour la comparaison entre phrases[7] ; et une autre configuration testant la méthode de hiérarchisation des critères, dans laquelle au contraire l'option maximaliste a été choisie pour chacun de ces points en dehors de ce qui a trait aux termes[8]. Ces tests ont d'abord été réalisés sur les questions de l'évaluation TREC8 ainsi que sur celles de l'évaluation TREC9. Globalement, ils ne permettent pas d'affirmer qu'une méthode est plus performante que l'autre. Dans le cas des questions de l'évaluation TREC8, nous avons pu montrer un léger avantage de la 2nde méthode : pour une réponse limitée à 250 caractères, nous sommes passés en effet d'un score de 0,463 à un score de 0,473. Cette différence est néanmoins faible. Par ailleurs, elle ne s'est pas confirmée lorsque nous avons effectué les mêmes tests sur les questions de l'évaluation TREC9.

En dehors de la faible différence intrinsèque en termes de performance entre les deux méthodes, cette absence de confirmation a aussi pour origine une certaine suradaptation de QALC dans sa seconde configuration vis-à-vis des questions de TREC8, celles-ci ayant servi d'étalon pour fixer la valeur de ses paramètres (cf. sections 8.2.2, 8.2.3 et 8.2.4). Cette observation souligne l'intérêt d'un jeu de test suffisamment étendu mais également la nécessité de viser une certaine stabilité des performances sur différents jeux de test plutôt qu'une optimisation forte sur un ensemble de questions réduit.

9. Résultats

Nous avons envoyé à TREC9 deux résultats de test sous la forme de listes de réponses sur 250 caractères (Ferret *et al.*, 2000). Le premier test utilise le moteur de recherche ATT, l'autre *Indexal*. Les résultats sont bien conformes à nos analyses précédentes. En effet, le test utilisant ATT a obtenu un score légèrement supérieur (0,407) à celui obtenu avec le moteur Indexal (0,375). Ce score nous place en 6ème position sur 28 participants. Le tableau 4 récapitule le nombre de réponses trouvées par nos deux tests, par rang de classement.

Nous pouvons voir que le test utilisant le moteur ATT donne davantage de réponses au rang 1 que le test utilisant le moteur *Indexal*. Ce dernier donne en

---

7 Cette configuration est équivalente à ce que nous avions utilisé pour TREC 8 (Ferret *et al.*, 1999).

8 Nous avons ainsi testé l'intérêt d'une stratégie plus complexe de combinaison des différents critères en nous affranchissant des problèmes de rappel accompagnant la reconnaissance des termes complexes et des variantes.

revanche davantage de réponses dans les rangs inférieurs que n'en donne le test utilisant ATT. La différence de score entre les deux tests semble donc résulter à la fois d'une meilleure émergence de la bonne réponse au premier rang et d'un plus grand nombre de réponses trouvées.

Par ailleurs, ces deux tests n'ont qu'un recouvrement partiel : aucun des deux ne trouve la bonne réponse pour 246 mêmes questions (sur environ 310 non trouvées pour chacun). D'autre part, 169 réponses ont été trouvées au même rang pour les deux tests sur environ 370 réponses trouvées, soit un peu moins de la moitié.

Rang de la réponse trouvée	**1**	**2 à 5**	**Total des réponses trouvées**
Test utilisant ATT	216	159	375 / 682
Test utilisant Indexal	187	185	372 / 682

Tableau 4. *Nombre de réponses trouvées par rang de classement pour les deux tests à 250 caractères*

D'autres points importants de la chaîne de traitement se trouvent révélés par les disparités de comportement que nous constatons entre notre système et les autres sur certaines questions. Notre système trouve parfois la bonne réponse là où la majorité des tests ne la trouve pas, et inversement. L'origine de ce phénomène réside notamment dans le choix des termes de la question à retenir pour l'appariement entre question et réponse. Ce choix est particulièrement crucial lorsque la question comporte peu de mots. Pour la question *How far away is the moon?* par exemple, notre module d'extraction des termes a conservé non seulement *moon (NN)*, mais aussi *away (RB)* comme mots pour l'appariement. D'autre part, notre module d'analyse de la question dispose, comme beaucoup d'autres, de *how far* comme locution interrogative permettant de typer la réponse attendue comme une distance. Ces différents choix nous ont permis de trouver la bonne réponse[9].

L'importance relative accordée aux différents termes de la question a également une grande influence. Lorsque la question comporte un nom propre, il est ainsi important de retrouver ce nom propre dans la réponse et il faut alors lui donner un poids fort. C'est le cas par exemple dans la question *Who manufactures the software, « PhotoShop »?* . Le module d'extraction des termes a retenu *software (NN)* , *PhotoShop (NP)*, et *manufacture (VBZ)* comme termes pour l'appariement, mais la méthode globale d'appariement donne à ces différents termes des

9 Seulement 7 tests, sur les 42 présentés à TREC9, ont trouvé la bonne réponse au rang 1. 27 ne l'ont pas trouvée.

importances équivalentes. Cette méthode ne nous a donc pas permis de trouver la bonne réponse[10]. En revanche, lorsque nous avons appliqué la méthode hiérarchisation des critères de manière à donner un poids plus fort aux noms propres, QALC a trouvé la bonne réponse.

L'appariement entre la question et les phrases des documents pertinents repose donc sur un ensemble de critères dont les importances relatives doivent être ajustées. Mais favoriser un critère revient parfois à favoriser un certain type de question. Il faut alors trouver un compromis entre l'optimisation des critères et l'indépendance du système vis à vis du type de question.

10. Travaux connexes

L'architecture des systèmes de question-réponse classiquement utilisée est globalement identique à celle de notre système QALC : détermination du type de la réponse attendue, sélection par un moteur de recherche d'un ensemble restreint de documents pertinents et enfin, recherche dans cet ensemble des réponses possibles.

Nous avons souligné précédemment l'importance de l'étape d'analyse des questions en vue de la détermination du type de la réponse attendue. Comme le fait QALC, la plupart des systèmes déterminent le type de la réponse attendue par la recherche dans la question de patrons pré-définis. Prager *et al.* (2000) utilisent ainsi 400 patrons différents pour identifier environ 50 types de réponse. En revanche, le système développé par IBM (Ittycheriah *et al.*, 2000) se fonde sur un modèle de l'entropie maximum pour la classification des types de réponse. Les auteurs du système FALCON (Harabagiu *et al.*, 2000) ont pour leur part établi une taxonomie des types de réponse d'après les hiérarchies des classes de mots dans WordNet.

Tous les systèmes de question-réponse ayant participé à TREC9 utilisent un moteur de recherche effectuant la sélection d'un sous-ensemble de documents pertinents dans une base d'environ un million de documents. Dans le système QALC, nous gardons le document retrouvé en entier. Cependant plusieurs systèmes ne gardent que le ou les paragraphes jugés les plus pertinents. Le système de Kwok *et al.*[11] (2000), par exemple, utilise le système de recherche d'information PIRCS, développé en interne, qui fait une première sélection d'un ensemble de 300 sous-documents pertinents d'environ 300 à 550 mots. Le système FALCON[12] (Harabagiu

10 22 tests, sur les 42 présentés à TREC9, ont trouvé la bonne réponse au rang 1. Seulement 9 ne l'ont pas trouvé.

11 Ce système, qui s'en tient à des méthodes classiques de recherche d'information tout en utilisant quelques connaissances sémantiques, est arrivé deuxième à la conférence d'évaluation TREC9 sur la tâche Question-Réponse.

12 Ce système est arrivé premier à la conférence d'évaluation TREC9 sur la tâche Question-Réponse.

et al., 2000), comporte également une étape de sélection des paragraphes effectuée par un moteur booléen.

Se focaliser sur les paragraphes potentiellement intéressant permet soit de conserver un plus grand nombre de documents pour la suite du traitement, soit d'appliquer ensuite des analyses syntaxico-sémantiques coûteuses en temps mais performantes. Dans le système de Kwok *et al.* (2000), qui garde un grand nombre de documents, l'appariement est fait sur un ensemble très riche de critères : stemming des mots, synonymes (un dictionnaire de 300 termes manuellement extraits de WordNet), valeur du score du document donné par PIRCS, pondération par la fréquence inverse du mot dans la collection, présence du mot exact quand il est important (certains superlatifs par exemple), proximité des mots dans la phrase, mots composés, présence des mots qui sont en capitales ou entre quottes dans la question. Le système FALCON, qui traite également des paragraphes sélectionnés, est celui qui utilise le plus largement les techniques d'analyse syntaxique et sémantique. Dans ce système, une unification est recherchée entre la représentation sémantique de la question et les représentations sémantiques des paragraphes sélectionnés. Lorsqu'une unification est trouvée, ces représentations sémantiques sont traduites en une forme logique pour inférer une justification de la réponse. Si aucune unification n'est trouvée ou si on ne peut pas justifier la réponse, la formulation de la question est élargie pour une nouvelle étape de sélection des paragraphes. L'élargissement de la question se fait par une fouille sous contrainte de WordNet. C'est le seul système présent à la conférence TREC qui effectue une rétroaction sur la formulation de la question en entrée du système.

Certains systèmes utilisent des techniques d'analyse syntaxique et sémantique sans sélection préalable des paragraphes. Litkowski (Litkowski, 2000) fait une analyse syntaxico-sémantique de l'ensemble des 20 premiers documents retrouvés par le moteur de recherche ATT. Or, la réponse correcte apparaît dans les 20 premiers documents pour seulement 78% des questions, alors qu'elle apparaît pour 92,5% des questions dans les 200 premiers documents ATT. Le peu de documents retenus ne permet pas à ce système d'atteindre un score élevé.

11. Conclusion

Un système de question-réponse doit trouver la réponse à une question précise dans un temps assez bref pour satisfaire un utilisateur en ligne. Cette réponse étant recherchée dans une grande masse de documents, il est tentant d'appliquer des méthodes essentiellement numériques pour la trouver. Néanmoins les expériences montrent que l'ajout de raisonnements fondés sur des connaissances sémantiques et pragmatiques est nécessaire si l'on veut obtenir, à terme, un système réellement efficace. Les futures orientations de la tâche question-réponse de la conférence TREC vont d'ailleurs dans ce sens. À un horizon de 5 ans, les organisateurs prévoient en effet des améliorations à la fois sur la rapidité de la réponse, la

vérification de son exactitude, la possibilité de fusionner plusieurs réponses pour obtenir une réponse complète et enfin, des possibilités de dialogue permettant à l'utilisateur de préciser sa demande. Ces améliorations ne pourront se faire sans une intégration encore plus importante des méthodes relevant du traitement sémantique de la langue.

Les améliorations que nous souhaitons apporter au système QALC que nous avons élaboré concernent donc essentiellement les dimensions sémantiques et pragmatiques. Ainsi, la base de connaissances WordNet, que nous utilisons déjà pour trouver les variantes sémantiques d'un mot, pourra aussi être exploitée pour une classification plus fine des types de réponses. Nous utiliserons aussi une analyse syntaxico-sémantique robuste pour construire les représentations sémantiques de la question et de l'ensemble des phrases candidates afin de sélectionner les réponses à la fois sur les termes de la question et sur les liens syntaxiques ou sémantiques que ces termes entretiennent entre eux. Dans la mesure où les domaines abordés par les questions et les documents ne sont pas circonscrits, WordNet, de par son caractère de base généraliste, constitue pour ce faire la ressource la plus indiquée, même s'il convient de prévoir les mécanismes permettant d'adapter son niveau parfois trop élevé de généralité.

12. Bibliographie

CELEX, http://www.ldc.upenn.edu/readme_files/celex.readme.html, UPenn, Consortium for Lexical Resources, 1998.

CLR, http://crl.nmsu.edu/cgi-bin/Tools/CLR/clrcat#D3, NMSU, Consortium for Lexical Resources, 1998

Dyer M., *In-depth understanding*, MIT Press, Cambridge, MA, 1983.

Fellbaum C., *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998.

Ferret O., Grau B., Hurault-Plantet M., Illouz G., Jacquemin C., Masson N., Lecuyer P., « QALC - The Question-Answering System of LIMSI-CNRS », *Actes de TREC9*, Gaithersburg, MD, 2000, p. 235- 244.

Ferret O., Grau B., Illouz G., Jacquemin C., Masson N., « QALC - the Question-Answering program of the Language and Cognition group at LIMSI-CNRS », *Actes de TREC8*, Gaithersburg, MD, 1999, p. 465-464.

Graesser A.C., McMahen C.L., Johnson B.K., « Question Asking and Answering », dans *Handbook of Psycholinguistics*, M.A. Gernsbcher (Eds), Academic Press, 1994, p. 517-538.

Grishman R., Sundheim B., « Design of the MUC6 evaluation », *Actes de MUC-6*, Morgan Kauffmann Publisher, Columbia, MD, 1995.

Harabagiu S., Pasca M., Maiorano J., « Experiments with Open-Domain Textual Question Answering », *Actes de COLING 2000*, Saarbrucken, Germany, 2000, p. 292-298.

Ittycheriah A., Ratnaparkhi A., Mammone R.J., « IBM's statistical Question Answering System », *Actes de TREC9*, Gaithersburg, MD, 2000, p. 229-234.

Jacquemin C., « Syntagmatic and paradigmatic representations of term variation », *Actes de ACL'99*, 1999, p. 341-348.

Jacquemin C., Bush C., « Fouille du Web pour la collecte d'entités nommées », *Actes de TALN 2000*, Lausanne, 2000, p. 187-196.

Justeson J.S., Katz S.M., « Technical terminology: some linguistic properties and an algorithm for identification in text », *Natural Language Engineering*, vol. 1, 1995, p. 9-27.

Kozima H., Text Segmentation Based on Similarity between Words, *Actes de ACL'93 (student session)*, Columbus, Ohio, 1993, p. 286-288.

Kwok K.L., Grunfeld L., Dinstl N., Chan M., « TREC9 Cross Language, Web and Question-Answering Track experiments using PIRCS », *Actes de TREC9*, Gaithersburg, MD, 2000, p. 419-429.

Lehnert W., « Human and computational question answering », *Cognitive Science*, vol. 1, 1977, p. 47-63.

Lehnert W., *The process of question answering*, Lawrence Erlbaum Associates, 1979.

Litkowski K, « Syntactic Clues and Lexical Resources in Question-Answering », *Actes de TREC9*, Gaithersburg, MD, 2000, p. 157-166.

de Loupy C., Bellot P., El-Bèze M., Marteau P.-F., « Query Expansion and Classification of Retrieved Documents », *Actes de TREC7*, Gaithersburg, MD, 1998, p. 443-450.

Mollá Aliod D., Schwitter R., Hess M., Fournier R., « Extrans, an answer extraction system », *Traitement Automatique des Langues*, vol. 41, n°2, 2000, p. 496-522.

Morton T. S., « Using co-reference in Question Answering », *Actes de TREC8*, Gaithersburg, MD, 1999, p. 685-688.

Prager J., Brown E., Radev D. R., Czuba K., « One Search Engine or two for Question-Answering », *Actes de TREC9*, Gaithersburg, MD, 2000, p 235-240.

Robertson E., Walker S., Beaulieu M., « Okapi at TREC7: automatic ad hoc, filtering, VLC and interactive », *Actes de TREC7*, Gaithersburg, MD, 1999, p. 253-264.

Schmid H., « Improvments in Part-of-Speech Tagging with an Application To German », dans *Natural Language Processing Using Very Large Corpora*, S. Armstrong, K.W. Chuch, P. Isabelle, E. Tzoukermann et D. Yarowski (Eds), Kluwer Academic Publisher, Dordrecht, 1999.

Zock M., Mitkov R., « How to ask a foreigner questions without knowing his language? Proposal for a conceptual interface to communicate thought », *Actes de Natural Language Processing Pacific Rim Symposium*, Singapore, 1991, p. 121-130.