MajecSTIC 2009
Avignon, France, du 16 au 18 novembre 2009

Interrogations de moteurs de recherche par des requêtes formulées en langage naturel

Ludovic Bonnefoy[1], Romain Deveaud[1] et Eric Charton[2]

1 : Centre d'Enseignement et de Recherche en Informatique / Université d'Avignon et des Pays de Vaucluse ludovic.bonnefoy, romain.deveaud {@etd.univ-avignon.fr}
2 : LIA / Université d'Avignon et des Pays de Vaucluse, 339 chemin des Meinajaries, 84911 Avignon eric.charton@univ-avignon.fr

---

Résumé
La difficulté de la tâche d'interprétation de requêtes en langue naturelle réside dans la transformation d'une phrase grammaticale en une requête pertinente pour interroger un système de RI ou d'extraction d'information. Dans cet article, nous présentons un système d'extraction d'information associé à un analyseur de requêtes, dédié à l'analyse et à l'interprétation de questions factuelles formulées en langue naturelle issues de la campagne TREC. Notre système fournit des réponses en exploitant un système composé d'une ressource ontologique et sémantique issue d'un corpus encyclopédique, dont on aura extrait les réponses candidates à l'aide de méthodes statistiques.

Abstract
Nowadays, requesting a search engine with natural language requests is a significant issue in the information retrieval research field, and some of its biggest actors begin to take it seriously. Some prototypes are actually available, but the error rate, inferred by the huge diversity of natural language and the different semantics of words or expressions, is still too large. Sentences naturally contain semantic criteria such as "entities", "concepts" or "categories" which can be combined with standard information retrieval in order to filter the documents with these semantic categories (e.g. "person", "organisation", "date"...). In this article we propose a categorization and keywords extraction system for natural language sentences.

Mots-clés : Sémantique, catégorisation, langage naturel, recherche d'information.

Keywords: Information retrieval, semantic, categorization, natural language.

---

1. Introduction

L'interprétation de requêtes écrites en langage naturel par un système de recherche d'information (RI) ou de dialogue est un enjeu important de l'ingénierie documentaire. Outre le fait que la requête en language naturelle rend possible la formulation d'une requête de recherche selon une forme bien plus précise que la simple saisie de mots clés, elle permet, dans le cadre de la compréhension des languages parlés et des systèmes de dialogue, de déployer des composants de compréhension (SLU) conviviaux pour les utilisateurs.
Dans le cadre des systèmes de Question et Réponses (SQR), l'analyse d'une requête en language naturelle doit permettre de transformer en processus d'extraction d'information, une question factuelle ou complexe. La difficulté de la tâche d'interprétation de requêtes en langue naturelle réside dans la transformation d'une phrase grammaticale en une requête pertinente pour interroger un système de RI ou d'extraction d'information.
Dans cet article, nous présentons le système *NLGbAse QR*. Ce système est un extracteur d'information contenues dans une ontologie, associé à un analyseur de requêtes formulées en langue naturelle. Ce système est testé à l'analyse et à l'interprétation de questions factuelles formulées en langue naturelle issues de la campagne TREC.

La structure de cet article est la suivante :
Nous décrivons dans notre introduction les grands principes de la littérature sur l'analyse de requêtes en langue naturelle, et son déploiement dans le cadre de systèmes de Question Réponses (SQR). Puis nous présentons les principaux systèmes de QR existants, et les méthodes de transformation de requêtes sémantiques et d'interrogation qu'ils mettent en oeuvre. Nous expliquons également dans cette section la relation existant entre ontologies issues de corpus encyclopédiques, et le rôle que ces ontologies peuvent jouer dans des SQR.
Dans la section suivante, nous présentons l'architecture du SQR *NLGbAse.QR*. Ce dernier est divisé en trois parties : analyse des requêtes en langue naturelle, transformation des requêtes en paramètres d'extraction, extracteur d'informations appliqué sur une ontologie. Puis nous détaillons dans la section 4 ces trois modules et les algorithmes qu'ils sous-tendent.
Dans la section 5, nous présentons une évaluation de notre système avec le corpus de question de TREC. Nous concluons par une analyse critique des résultats obtenus, qui sont du niveau de l'état de l'art, et introduisons nos perspectives futures de recherche.

2. Principe de l'analyse de requêtes en langue naturelle et des SQR

Des sociétés telles que Google[1], Powerset[2] (propriété de Microsoft) ou Hakia[3] sont actuellement fortement investies dans le développement de solutions sémantiques à l'interrogation des moteurs de recherche et l'enrichissement communautaire en est un élément central, au moins pour les deux premiers protagonistes. Ils ont en effet recours à de nombreux sites dont les informations sont éditées par des internautes contributeurs, Wikipédia[4] étant le plus célèbre d'entre eux.
C'est également notre cas, puisque le système que nous proposons s'interface avec NLGbAse[5], une base de données classifiées provenant de Wikipédia qui peut être interrogée par le biais de trois moteurs de recherche différents. Le premier d'entre eux met en œuvre un algorithme calculant la *similarité cosinus* entre l'ensemble des mots-clés entrés et les documents issus de Wikipédia ; le deuxième est semblable au premier en tous points, à l'exception que l'on peut affiner la recherche en précisant une catégorie, ainsi seuls les documents classifiés comme appartenant à la catégorie spécifiée seront relevés. Le troisième applique quant à lui un algorithme de compacité [1] permettant de trouver une entité précise appartenant à une catégorie donnée, proche d'un ou plusieurs mots donnés, dans le document Wikipédia se rapportant à une entité nommée donnée, ce qui permet notamment de pouvoir proposer une réponse factuelle à une requête.
Ces outils constituent le système de recherche d'information sur lequel nous appliquons les sorties de notre propre système ; ce dernier peut, à partir d'une phrase en langage naturel - de préférence une question, donner les différents mots-clés et catégories attendus par les moteurs de recherche de NLGbAse, et ainsi obtenir une liste de résultats - et éventuellement des réponses factuelles - pertinents.

3. Encyclopédies, ontologies et SQR

4. Algorithmes déployés

4.1. Analyse morpho-syntaxique et couplage des mots

Pour travailler sur la sémantique, il est indispensable de posséder des outils permettant à la machine de décomposer et d'analyser la structure des phrases. C'est pourquoi nous avons utilisé un analyseur morpho-syntaxique réalisant des couplages de mots selon leur position grammaticale dans la phrase et les liant selon leurs interdépendances [2].

4.2. Catégorisation des phrases

Par définition, la sémantique crée naturellement différentes classes de sens plus ou moins générales, au niveau du mot comme de la phrase. Nous adhérons à l'idée selon laquelle la sémantique glob-

[1] http ://www.google.com
[2] http ://www.powerset.com
[3] http ://www.hakia.com
[4] http ://www.wikipedia.org
[5] http ://www.nlgbase.org

ale d'une phrase est déterminée par un nombre réduit de mots appartenant à une catégorie grammaticale précise, et c'est selon ce point de vue que nous allons présenter la catégorisation de phrases en catégories sémantiques larges que nous avons implémentée.

4.2.1. Catégorisation à base de règles simples

Notre approche pour catégoriser les questions fonctionne avec un ensemble de règles. Cet ensemble est relativement restreint car nous avons pu remarquer qu'une dizaine de règles environ pouvaient couvrir une majorité des cas, et qu'ensuite chaque petit gain se traduisait par la production d'un nombre croissant exponentiellement de nouvelles règles. Les règles que nous avons formulées se basent principalement sur les pronoms interrogatifs des questions. En voici la liste :

- Who, Whom, Whose : *pers* (Person).
- Where, Whence, Wither : si il s'agit de trouver une catégorie pour le deuxième moteur de NLGbAse, et si un nom propre ou un objet est trouvé, la catégorie sera celle du *nom propre* ou de l'*objet* grammatical de la phrase (par exemple : « Where did Patrick Sébastien study ? », il y a peu de chance de trouver dans la fiche de ce lieu une mention de cette personnalité et il est à priori plus judicieux de proposer la fiche de *Patrick Sébastien*, dans laquelle l'utilisateur sera à même de touver l'information). Dans le cas contraire (ou si nous voulons une catégorie pour le troisième moteur), la catégorie *loc* (Location) est attribuée.
- How : nous appliquons la même procédure que précédemment, à l'exception près que si les premières conditions ne sont pas remplies, la catégorie *unk* (Unknown) est attribuée. Pour ce pronom là nous avons ajouté quelques précisions lorsque nous cherchons une catégorie pour le troisème moteur : si le mot suivant directement *how* fait partie de la liste suivante (far, few, great, little, many, much, tall, wide, high, big, old), la catégorie *amount* est attribuée.
- What, Why, Which : le principe est toujours le même, avec *unk* (Unknown) pour valeur par défaut. Nous avons également établi, comme précédemment, une liste de mots pouvant être acceptés comme suivant directement le pronom interrogatif (comme par exemple day : « What day is the Independance Day ? ») et qui vont impliquer automatiquement l'attribution d'une catégorie (*date* dans l'exemple précédent).

Nous allons maintenant détailler les méthodes de catégorisation des noms propres et des noms communs que nous avons mises en place.

4.2.2. Catégorisation par les noms propres utilisant NLGbAse

Comme nous l'avons vu, la majorité des règles ne nous permettent pas de trancher directement, nous devons donc compléter notre analyse par un autre moyen et cela passe notamment par la catégorisation des noms propres. Nous avons remarqué que dans la majorité des cas, si un nom propre est présent dans une question, il en est l'objet ou du moins l'objet est l'une de ses caractéristiques. Prennons par exemple ces deux questions « What is the date of birth of Bruce Dickinson ? », « Who is Batman's team-mate ? » ; nous voyons bien que les informations désirées sont *forcément* en relation avec nos noms propres.

Nous avons donc prit le parti de prendre comme catégorie la catégorie du nom propre se trouvant dans la question - s'il y en a un. Pour cela nous adressons une requête à un script issu de NLGbAse, qui comme nous l'avons vu associe une catégorie à chaque entité de Wikipédia, qui récupère la catégorie de l'entité correspondant à ce nom propre. Si une entité porte exactement le même nom alors la catégorie sera celle de cette entité ; si ce n'est pas le cas mais que des entités ont un nom similaire, alors la catégorie sera celle de la plus pertinente d'entres elles. Enfin si ce n'est pas le cas nous effectuons une recherche par *TF.Idf* en prenant la catégorie qui a le plus fort score, pour cela chaque catégorie se voit attribuée comme score la somme des scores des documents ayant cette catégorie. De ce fait la catégorie qui rassemble le plus de pertinence sera sélectionnée.

Cependant nous sommes conscients qu'il arrive parfois que cette stratégie ne soit pas idéale, comme pour : « What is the name of Batman's car ? ». Il y a des chances que cette méthode donne *pers* (Person) comme catégorie attendue alors que la solution idéale aurait probablement été *prod* (Product). Elle n'est néanmoins pas totalement inappropriée car nous devrions trouver l'information désirée dans la fiche Wikipédia de Batman, néanmoins l'accès à la réponse est moins direct.

Parfois cette méthode ne donne aucun résultat et il y a plusieurs explications possibles à cela. La première est qu'il est possible qu'il n'y ait pas de fiche relative à ce sujet sur Wikipédia, la seconde est que notre système extrait mal le nom propre et soumet donc quelque chose de faux

à NLGbAse. La troisième raison est imputable à l'utilisateur. En effet si l'utilisateur commet une faute dans l'écriture du nom alors NLGbAse ne pourra pas nous donner la bonne réponse.
Pour y remédier nous effectuons une requête avec le nom propre sur Google, ensuite nous regardons si Google suggère une autre orthographe par l'intermédiaire du "Did you mean :". Cependant cette méthode n'est pas infaillible car il est possible que cette même faute d'orthographe soit souvent comise sur Internet.
Enfin si à ce stade le nom n'est toujours pas associé à une étiquette alors nous soumettons la phrase au module de CCG de reconnaissance d'entité nommée [3] basé sur l'architecture SNOW. En effet ce module à de très bonnes performances pour reconnaitre les personnes, organisations et produits.

4.2.3. Catégorisation par les noms communs utilisant WordNet

Toutes les questions ne comportent évidemment pas de noms propres mais généralement des noms communs, c'est pourquoi nous avons du trouver un moyen de traiter ces questions par une approche assez simple. La décomposition morpho-syntaxique nous permet généralement de trouver l'objet de la question, qui possède généralement une forte valeur sémantique ; c'est donc celui-ci que nous allons étudier. En effet, pour la question « What are the generals ? », l'objet de la question est « generals » ; l'enjeu est d'arriver à associer « generals » à l'étiquette *fonc.mil* (fonction militaire).
Pour arriver à cela nous utilisons WordNet[6] et ses hyperonymes ainsi que sa capacité à fournir une classe pour chaque mot : en effet WordNet associe déjà à la totalité des termes une étiquette, par exemple à « general » WordNet associe *noun.person*. L'étiquette que fournit WordNet est bien souvent satisfaisante, cependant son jeu d'étiquette ne correspond pas aux exigences d'Ester auquel notre projet doit se plier.
La première étape fut d'associer « à la main » des étiquettes à des mots qui prendront le dessus sur celles de WordNet ; pour reprendre notre exemple, nous ne voulons pas l'étiquette *pers* (Person) pour « general » mais bien *fonc.mil*. Cependant faire ce travail sur tous les mots demanderait un investissement titanesque, et nous avons pu palier à ce problème en réfléchissant aux mots les plus généraux possibles pour chaque catégorie dont nous avions besoin. Ensuite nous vérifions que les hyponymes de ces mots sur WordNet correspondaient bien à la même catégorie, et si ce n'était pas le cas nous sélectionnions tous les hyponymes pour lesquels c'est le cas et répétions cette opération. Nous sommes donc arrivés à une liste de mots caractérisant parfaitement chaque catégorie - et étant compatibles avec WordNet.
L'algorithme de catégorisation en lui-même consiste en une fonction récursive qui va vérifier si le nom commun ne fait pas partie des mots étiquetés. Si ce n'est pas le cas cette vérification est faite pour son hyperonyme, et ainsi de suite. La récursivité s'arrête si un mot associé à une étiquette est trouvé ou si on arrive sur l'hyperonyme de plus haut niveau. Dans le premier cas le mot de départ se voit associé cette étiquette et donc la catégorie recherchée aussi, dans le deuxième cas c'est l'étiquette associée par WordNet au mot de départ qui prévaut et qui est donc définie comme la catégorie recherchée.
Cette méthode utilisée seule ne peut bien évidemment pas couvrir tous les cas, néanmoins c'est l'association des différentes - mais surtout complémentaires - méthodes de catégorisation qui permet d'obtenir des résultats satisfaisants.

4.3. Extraction des mots-clés

Comme nous l'avons expliqué, les différents moteurs de recherche de NLGbAse n'attendent pas les mêmes entrées, nous allons donc détailler ici les deux types d'extraction de mots-clés - ou mots pertinents.

4.3.1. Extraction destinées aux moteurs de recherche d'information par similarité cosinus

Dans un premier temps, les mots-outils de la phrase sont automatiquement supprimés à l'aide d'un anti-dictionnaire. Nous utilisons ensuite l'analyse morpho-syntaxique de la phrase, et notamment l'arbre constitutif, pour récupérer les mots - qui ne sont pas des mots-outils - qui constituent groupes nominaux ; malgrès son aspect simple, voire simpliste, nous avons pu prouver

[6] http ://wordnet.princeton.edu

empyriquement son efficacité.

4.3.2. Extraction destinées au moteur de recherche d'information par algorithme de compacité (question-réponse)

Nous l'avons déjà précisé plus haut, ce troisième moteur accepte plusieurs entrées différentes, et notamment deux champs de mots-clés. Le premier champ est une entité nommée qui va déterminer dans quel document sera appliqué l'algorithme de compacité, tandis que le deuxième champ consiste en une liste de mots qui représentent l'information cherchée ; par exemple pour la question « When was Albert Einstein born ? », le mot « born » devrait être sélectionné car l'information recherchée - une date de naissance en l'occurence - se trouvera certainement très proche de ce mot.
Dans un premier temps nous devons donc trouver l'entité nommée ; si un nom propre est présent dans la question, il sera directement utilisé comme entité nommée. Dans le cas contraire, nous éxécutons une requête sur NLGbAse avec l'objet de la question afin de récupérer le nom de l'entité nommée la plus pertinente.
Dans un second temps vient l'extraction des mots porteurs de sens ; il s'agit tout d'abord de supprimer tous les mots-outils, les noms propres et le verbe présents dans la phrase. Il s'agit ensuite de récupérer les synonymes des mots restant avec WordNet ; pour reprendre notre exemple précédent, nous ne savons pas si le mot « born » sera effectivement employé dans le document dans lequel l'information sera cherchée, c'est pourquoi nous cherchons des dérivations afin de les rajouter à notre liste et ainsi améliorer nos chances de trouver l'information. Toujours pour notre exemple, cette recherche de synonymes pourrait nous mener au mot « birth », qui serait en effet intéressant à garder dans l'optique de la recherche d'une date de naissance.

5. Expériences et résultats

La vocation de notre algorithme est de transformer une question en langue naturelle en une requête compatible avec un système de RI. Pour mesurer les performances de notre système, nous évaluons dans un premier temps sa capacité à étiqueter une question en vue de l'extraction des informations requise pour construire une requête. Dans un second temps nous construisons une requête d'après les informations extraites et mesurons la pertinence des résultats retournés par l'un des trois moteurs de NLGbAse.

5.1. Corpus de référence et standard de mesure

Nous avons séparé les deux tâches que sont la RI et l'analyse sémantique de la phrase, c'est pourquoi l'évaluation de notre système porte sur la catégorisation et l'extraction de mots-clés. Nous avons utilisé pour ces mesures des corpus de questions déjà existants qui ont été mis au points dans le cadre de campagnes d'évaluation telles que Trec 12.
Le corpus *Question Answering Collections* de Trec est composé de 500 questions factuelles[7], organisées dans un fichier XML comme suit :

```
<top>
<num> Number: 1919
<type> Type: factoid
<desc> Description:
How big is Mars?
</top>
[...]
```

Notre choix s'est porté sur cette année là car ultérieurement le format des fichiers a changé et est devenu à notre sens beaucoup trop spécifique à la tâche de question réponse. En effet les corpus les plus récents proposent des séries de cinq questions en rapport avec une cible donnée. On voit bien là que l'approche est toute autre. En effet pour le moteur de RI classique avec lequel nous travaillons, la meilleure manière d'obtenir un document comportant la réponse est de choisir la fiche correspondant à la cible. L'évaluation de notre application sur ces corpus n'aurait donc eu

[7] En téléchargement sur http ://trec.nist.gov/data/qa/2003_qadata/03QA.tasks/test.set.t12.txt.html

que très peu de sens. De plus nous trouvions qu'une évaluation de ce type est vraiment superficielle et ne reflète pas les résultats que le système fournirait à un utilisateur qui ne préciserait pas la cible de la question.
Ce corpus n'étant pas initiallement prévu pour vérifier l'expérience d'étiquetage de questions tel que prévu dans le cadre de notre algorithme, nous avons procédé à son enrichissement « à la main ».
Cet enrichissement consiste a attribuer à chaque question la catégorie sémantique de la réponse attendue (nous n'avons retenu que les classes racines à savoir *pers*, *org*, *loc*, *date*, *amount* et *unk*), ainsi que les différents mots-clés qui peuvent permettre d'obtenir une réponse. Nous disposons ainsi d'un corpus de questions et de réponses de références complétées par des informations sémantiques compatibles avec notre système[8].
Néanmoins pour des raisons pratiques nous avons créé notre propre formalisme :

```
<question en toutes lettres>#
<mots-clés attendus par les moteurs de RI>#<catégorie de l'entité source>#
<nom de l'entité source>#<mots-clés probablement proches de la réponse>#
<catégorie de l'entité réponse cherchée>
```

Ainsi, dans l'exemple précité, nous complétons la description de la question par plusieurs étiquettes de classes, à savoir que l'entité source est une localisation, et que la réponse cible est une quantité (*amount*, la taille de la planète Mars) :

```
How big is Mars?#Mars#loc#Mars#big#amount#
[...]
```

Nous avons ainsi lancé notre système pour chacune des questions de ce corpus et mesuré les différences entre l'étiquetage « à la main » et les sorties du système. Nous avons pu en déduire des mesures de précision et de rappel complétées par le F-Score[9] que nous vous présentons dans la sous-section 5.2.

5.2. Mesures de la catégorisation sémantique

Les résultats de l'attributions de catégories aux 492 phrases du corpus étiqueté sont présentés dans le tableau 1.
Ces résultats mettent en perspective le fait que des règles simples couplées à une catégorisation par recherche dans une base de données sémantique peuvent être viables. Les résultats de la catégorie *org* peuvent être expliqués par le fait qu'il n'existe pas de règle spécifique à cette catégorie, tous les essais que nous avons fait pour y remédier impactaient fortement les résultats des autres catégories, notamment *pers* et *loc*.

Catégorie	($\bar{p}$)	($\bar{r}$)	($\bar{F}$-s)
Pers	0.81	0.81	**0.81**
Org	0.64	0.61	**0.63**
Loc	0.76	0.77	**0.76**
Date	0.91	0.98	**0.95**
Amount	0.99	0.92	**0.92**
Unk	0.69	0.64	**0.66**
Total	0.80	0.78	**0.79**

TAB. 1 – Précision ($\bar{p}$), Rappel ($\bar{r}$), F-Score ($\bar{F}$-s) obtenus sur le corpus QA de TREC 12

[8] Ce corpus complété est disponible sur www.nlgbase.org
[9] Mesure harmonique combinant la précision et le rappel

5.3. Mesures de l'extraction de mots-clés

Nous avons également mesuré la pertinence des mots-clés extraits par notre système. Comme précédemment nous avons tout d'abord effectué « à la main » le travail d'extraction, puis nous avons comparé ceci avec les résultats du système afin d'obtenir un pourcentage de satisfaction calculé selon la formule suivante :

$$S(C) = \frac{\sum_{i=1}^{N} P(Q_i)}{N}$$

Ils sont présentés dans les tableaux ci-dessous.

Type de mots-clés	(S(C))
Mots-clés extraits pour une recherche par similarité cosinus	**54.03%**
Entités nommées extraites pour une recherche de type question-réponse (compacité)	**57.76%**
Mots-clés extraits pour une recherche de type question-réponse (compacité)	**73.28%**

TAB. 2 – Satisfaction (S(C)) obtenue sur le corpus de test

Ces derniers chiffres peuvent être expliqués par le fait qu'il est assez difficile de se mettre à la place d'un système de RI et d'imaginer quels paramètres seront les plus pertinents pour obtenir un bon résultat, les mots-clés que nous avons extrait « à la main » pour établir notre corpus de test sont donc parfois éloignés de la formulation optimale.

5.4. Commentaires et comparaisons

Ces résultats nous ont permis de comparer notre système avec certains des meilleurs systèmes actuels comme par exemple le système de CCG [4] qui obtient un score de 90% à 95% de bonne classification. Néanmoins celui-ci est basé sur l'architecture d'apprentissage automatique SNOW, contrairement à notre approche basée sur des règles simples et l'utilisation d'hyperonymes. De plus notre méthode est facilement adaptable à d'autres langues et nous obtiendrions alors des résultats à peine plus faible que le système multilingue de Thamar Solorio et al. [5] utilisant des heuristiques complexes et ayant recours aux SVM.

6. Conclusion

Nous avons présenté un système analysant et décomposant des requêtes formulées en langage naturel dans le but d'en extraire leur catégorie sémantique ainsi que les mots porteurs de sens et d'information, destiné à être couplé avec un système de recherche d'information que nous avons présenté.

Bibliographie

1. Laurent Gillard et al. D'une compacité positionnelle à une compacité probabiliste pour un système de questions/réponses. 2007.
2. Daniel Sleator et Davy Temperley. Parsing english with a link grammar. 1993.
3. L. Ratinov et D. Roth. Design challenges and misconceptions in named entity recognition. 2009.
4. Xin Li et Dan Roth. Learning question classifiers. 2002.
5. Thamar Solorio et al. A language independent method for question classification. 2004.