Réordonnancement de réponses par transformation d'arbres pour un système de question-réponse oral interactif

Guillaume Bernard

LIMSI-CNRS
BP 133
91403 ORSAY CEDEX
gbernard@limsi.fr

RÉSUMÉ. *Les techniques traditionnelles de recherche d'information montrent des limites pour extraire certaines réponses précises contenues dans des documents. Cet article présente une méthode de recherche d'informations adaptée au contexte d'un système de question-réponse oral interactif en domaine ouvert. Cette méthode vise à améliorer la sélection des meilleures réponses. Nous proposons une approche consistant à mesurer un coût de transformation entre deux arbres textuels qui rend compte des reformulations possibles entre un texte décrivant l'information recherchée (question) et un passage de document. Nous présentons ensuite une évaluation de la méthode sur le corpus Clef et analysons les résultats mesurés. Nos perspectives présentent des voies d'amélioration et incluent l'exploitation des transformations d'arbres trouvées par notre méthode pour fournir des informations à l'utilisateur sur le déroulement de la recherche.*

ABSTRACT. *Traditionnal Information Retrieval techniques have some limitations when it comes to extracting accurate answers from documents. This paper describes an information retrieval method working in the context of a speech-based, open domain interactive question-answer system. The goal is to improve the selection of the best answers. We present an approach which aims to compute a transformation cost between two textuals trees. This computation uses reformulation paths between a text which describes the information (question) and an extract from a document. We then present a first evaluation of our method on the Clef corpus. We analyze the results and suggest some possible improvements, including the use of the tree transformations to give the user some information on how the answer was extracted.*

MOTS-CLÉS : *Recherche d'information, systèmes à Questions Réponses*

KEYWORDS: *Information Retrieval, question answer systems*

1. Introduction

Notre travail s'inscrit dans le contexte du projet Ritel (Rosset *et al.,* 2006), dont un des objectifs est de réunir les fonctionnalités d'un système de question-réponse et d'un système de dialogue oral. L'utilisateur doit pouvoir poser ses questions à l'oral de façon naturelle et dialoguer avec le système. Du fait de cette interactivité et de la composante orale du système, les requêtes doivent être traitées rapidement pour favoriser l'établissement d'un dialogue naturel. Deux parties du système sont liées à notre travail : le module d'analyse des documents, qui transforme les documents de nos corpus et les questions en arbres, et le module de recherche d'information et de question-réponse.

Nous proposons une méthode pour améliorer la sélection des réponses à une question. Les techniques de recherche d'information habituellement utilisées pour extraire la réponse à une question précise dans des documents montrent certaines limites (Ligozat, 2006), particulièrement lorsque l'information recherchée n'est pas présente telle quelle, par exemple sous forme d'une paraphrase. Pour traiter ce problème, nous proposons une approche basée sur le calcul d'un coût de transformation entre arbres. Notre méthode est conçue pour traiter des questions automatiquement transcrites de l'oral (et pouvant être incomplètes syntaxiquement). Dans le contexte de cet article, on travaille sur des textes écrits.

Le module d'analyse (Villaneau *et al.,* 2007) produit en sortie un ensemble d'arbres dont les nœuds sont typés sémantiquement. Ce système est à base de règles et de lexiques et utilise un moteur d'expressions régulières permettant une analyse très rapide (12 000 mots/s). Les entités sont définies hiérarchiquement. Environ 300 types sont détectés, dont 8 sont des catégories morpho-syntaxiques (mots composés, verbes ...). L'analyse est très rapide, environ 1h30 pour traiter le corpus des évaluations de Clef - ATS et Le Monde 94/95 - soit environ 400 Mo de données). Les arbres produits sont très étendus, c'est à dire plus larges que profonds. La figure 1 présente un exemple d'arbre d'analyse.

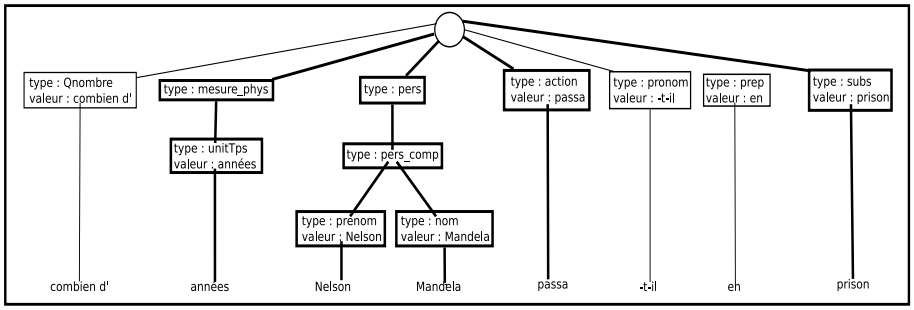

Figure 1. *Exemple d'arbre construit pour «Combien d'années Nelson Mandela passa-t-il en prison ?»*

Le fonctionnement du système RI/QA de Ritel fonctionne en trois étapes : la sélection des documents intéressants, la sélection des passages de texte contenant potentiellement la réponse à la question traitée, et la sélection des réponses candidates. Au début de la recherche, la question est analysée pour déterminer le type de réponse attendu et le système génère un Descripteur De Recherche (DDR) qui permet de guider la recherche de la réponse. Les DDRs représentent les éléments critiques et secondaires de la question ainsi que le type de réponse attendu. Chacune de ces étapes utilise les informations fournies par le DDR. Le premier module calcule un score pour chaque document selon les différentes entités du DDR et retourne une liste de documents. Le deuxième module sélectionne alors les passages des documents pouvant contenir la réponse et retourne une liste de passages ordonnée. Enfin, à partir de cette liste de passages, le troisième module retourne une liste de réponses courtes extraites de ces passages. Les réponses sont ordonnées selon un score. Comme pour le module d'analyse, le système de RI/QR se doit d'être très rapide (en moyenne répondre à une question prend 100 ms).

Pour qu'un extrait de texte soit identifié comme pouvant contenir la réponse, celui-ci doit contenir les éléments critiques du DDR. S'il contient les éléments secondaires, le score global du passage sera plus élevé. Cette méthode est comparable à une approche par sacs de mots typés et transformables. Les scores calculés par le système sont comparables à des scores de compacité (Gillard *et al.,* 2007). Les DDRs ne conservent pas toute l'information contenue dans une question mais seulement les éléments critiques et secondaires (en gras dans la figure 1) et de ce fait certaines informations comme les adjectifs ou les adverbes sont perdues. Cette approche rencontre un certain nombre de problèmes qui proviennent du fait de la non-utilisation de toutes les informations contenues dans le texte. Si dans l'exemple suivant, on essaye d'évaluer les réponses candidates «71» et «27», le système actuel donne «71» comme étant la bonne réponse car elle est proche de «Nelson Mandela», qui est un élément critique, et car «71» est plus fréquent dans les différents documents du corpus. Notre objectif est de proposer une méthode pouvant répondre à ces problèmes.

«Q : Combien d'années Nelson Mandela passa-t-il en prison ?»

«A 71 ans, Nelson Mandela est sorti après 27 ans passé en prison.»

2. Réordonnancement des réponses

Nous avons brièvement présenté le système Ritel, qui sert de contexte à notre travail. Le module d'analyse de Ritel produit en sortie un arbre typé. La figure 2 décrit l'organisation du système RI/QR, incluant notre approche. Notre méthode récupère en entrée une question, la liste de ses réponses potentielles et les passages du texte d'où elles ont été extraites.

Cette méthode se base sur le calcul du coût d'une transformation entre l'arbre de la question et l'arbre du passage contenant la réponse. L'objectif est de calculer la transformation entre l'arbre du texte et l'arbre de la question ayant le plus faible coût. Les extraits des documents pouvant être très longs, nous avons choisi d'extraire la phrase

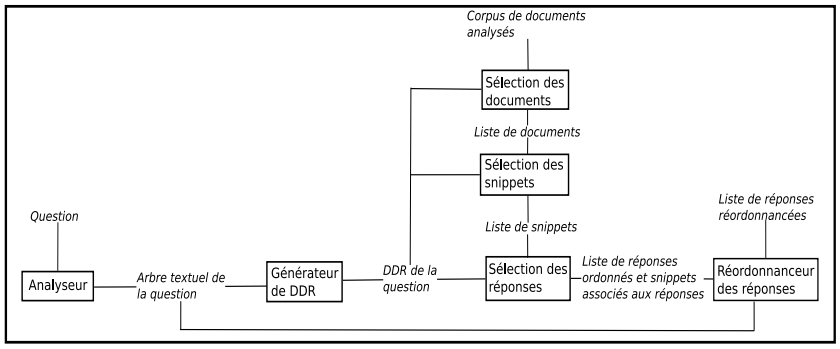

Figure 2. *Contexte d'application du réordonnancement dans le système RI/QR*

du passage contenant la réponse candidate. Des opérateurs de manipulation d'arbre sont utilisés pour transformer un arbre en un autre. Ce coût de transformation est utilisé pour réordonnancer les réponses. Cette méthode est similaire à celle de (Magnini *et al.*, 2006), mais diffère principalement sur les trois points suivants : la méthode de recherche des opérations les moins coûteuses, les opérateurs de transformation et les fonctions de coût associées à ces opérateurs.

2.1. *Approche par transformation d'arbre*

2.1.1. *Algorithme général*

Pour trouver le coût de transformation minimum entre deux arbres, nous utilisons des opérateurs de transformation que nous appliquons sur les différents nœuds de nos arbres. Nous ajoutons aux opérateurs de suppression, d'insertion et de substitution utilisés par Magnini et Kouylekov un opérateur de déplacement. Certains phénomènes linguistiques, tel que le passage d'une phrase de la forme passive à la forme active peuvent être mieux représentés avec le déplacement. Chacun de ces opérateurs est associé à une fonction permettant d'approximer le coût linguistique associé à une manipulation selon le contexte d'application de l'opérateur. Dans la méthode de Magnini et Kouylekov, la recherche de la plus petite distance se fait en utilisant l'algorithme de distance d'édition de Zhang et Shasha (Zhang *et al.*, 1989), conçu pour être appliqué sur des arbres très structurés. Les arbres produits par l'analyseur de Ritel étant peu profonds et très étendus, et l'algorithme de distance d'édition étant conçu pour être utilisé avec trois opérateurs non sensibles au contexte, nous avons décidé d'ajouter l'opérateur de déplacement. Pour ces raisons, l'algorithme de distance d'édition n'est pas adapté. Nous avons décidé d'utiliser l'algorithme de recherche en coût uniforme qui est une méthode d'exploration non heuristique d'un espace de recherche. Dans notre cas, chaque état représente un état de l'arbre que l'on transforme. Les nœuds successeurs sont ceux des arbres résultant de l'application des opérateurs de transformation sur l'arbre de l'état précédent. L'algorithme de recherche en coût uniforme se rapproche d'une recherche en largeur d'abord, mais se différencie par le fait qu'à

chaque itération on développe le nœud dont le coût par rapport à l'état initial est le plus faible. S'il s'agit du nœud d'arrivée (c'est-à-dire l'arbre de la question), l'algorithme a trouvé le but et retourne la solution. Nous procédons aussi à des modifications sur l'arbre de la question. Pour chaque réponse avec le score le plus fort, le passage du texte d'où elle a été extraite est connu. On supprime alors de l'arbre de la question les nœuds correspondant aux pronoms interrogatifs («qui», «comment», etc.) et on ajoute un nœud avec la réponse à évaluer.

2.1.2. *Opérateurs de transformation*

Nous utilisons 4 opérateurs de transformation : l'insertion, la suppression, le déplacement et la substitution. La suppression enlève un nœud de l'arbre du texte si on ne trouve pas de nœuds correspondant dans l'arbre de la question. L'insertion cherche les nœuds de l'arbre de la question non présents dans l'arbre du texte et les ajoute. On effectue une opération de déplacement lorsque l'on détecte des nœuds identiques dans l'arbre du texte et l'arbre de la question. Si le nœud dans l'arbre du texte ne se trouve pas à la même position, on le déplace. On essaye ainsi de se rapprocher de la même structure que celle de l'arbre de laquestion. L'opérateur de substitution permet de convertir un nœud de l'arbre du texte en un nœud de l'arbre de la question. Il s'applique dans deux cas : si les lemmes de deux nœuds sont identiques, ou si l'on identifie une relation de synonymie entre eux. Ce dernier est basé sur la forme lemmatisée des mots. Pour chaque lemme, on a une liste de lemmes synonymes organisés par sens. Pour le moment, la détection de la synonymie se fait en comparant uniquement chacun des synonymes d'un nœud avec ceux d'un autre nœud, sans aucune prise en compte du contexte. Nos synonymes proviennent d'un dictionnaire interne, d'environ 15 000 mots.

2.1.3. *Fonctions de coût*

Dans leur approche, Magnini et Kouylekov utilisent des arbres syntaxiques pour représenter les textes, et de ce fait, leurs fonctions de coût se basent sur la structure des arbres. Nous avons décidé d'avoir une approche basée sur le contexte général de nos arbres. Le but est de modéliser les fonctions de coût en fonction du changement linguistique induit par l'opérateur. On veut par exemple qu'il soit coûteux de supprimer un verbe car cela peut changer radicalement le sens. Au contraire, dans le cadre d'une substitution, si les couples sont par exemple liés par une relation de synonymie, on veut que le coût de l'opération soit faible. La gestion du contexte est également importante. Ainsi, il est par exemple plus coûteux d'insérer un noeud représentant une personne devant un verbe. Dans le cas présent, le contexte d'un arbre correspond aux nœuds immédiatement à gauche et à droite du nœud que l'on traite. La prise en compte du contexte est donc assez restreinte. Par la suite, il sera intéressant d'essayer d'avoir un contexte plus global pour avoir des fonctions de coût qui sont davantage motivées linguistiquement. La structure des fonctions est similaire : une suite de conditionnelles portant sur le type du noeud concerné par l'opérateur, avec en paramètre le contexte gauche et droit du noeud. Il y en une pour chaque opérateur. Un extrait de la fonction de coût de l'opérateur de d'insertion est donné ci-dessous :

```
/*
* ngauche et ndroite correspondent aux noeuds à
* gauche et à droite du noeud sur lequel
* on applique l'opérateur d'insertion.
*/
Insertion(type, valeur, cgauche, cdroite) {
 si le type est une action alors
  si type de ngauche ou ndroite == verbe ou un personnage,
  coût de 5
  sinon coût de 4
 si le type est un personnage, un évènement ou un lieu alors
  si type de ngauche ou ndroite == verbe,
  coût de 5
  sinon coût de 3
}
```

2.1.4. *Exemple de calcul de distance*

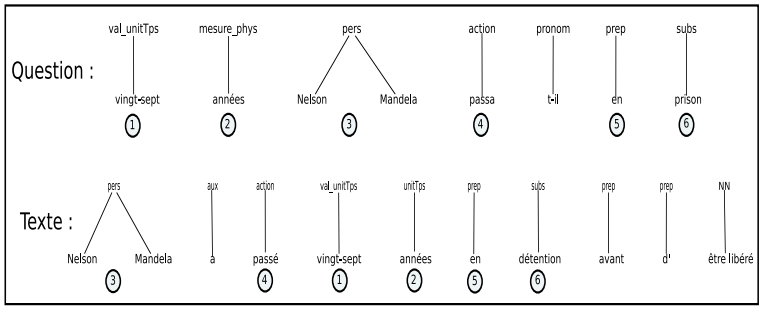

Figure 3. *Points d'ancrage entre les arbres de la question et le texte - Points d'ancrage 1,2,3 et 5, nœuds identiques - Points d'ancrage 5, relation d'équivalence de lemme - Points d'ancrage 6, relation de synonymie*

La figure 3 illustre le fonctionnement général de notre méthode (voir figure 3). Dans un soucis de lisibilité, les arbres sont présentés dans une forme simplifiée (pas de nœud racine et de sous-nœuds). On compare la question *«Combien d'années Nelson Mandela passa-t-il en prison ?»* avec le texte *«Nelson Mandela a passé vingt-sept années en détention avant d'être libéré»*. La réponse évaluée est «vingt-sept», qui est la bonne réponse à la question. Notre méthode supprime le ou les nœuds contenant des pronoms interrogatifs, c'est-à-dire «Combien». Puis le nœud supprimé est remplacé par la réponse évaluée. Dans ce cas, on obtient alors 6 points d'ancrage. Les points 1,2,3 et 5 correspondent à des nœuds identiques, le 4 à une relation d'équivalence de lemme et le 6 à une relation de synonymie. L'algorithme de recherche va alors chercher la suite d'opérations la moins coûteuse pour transformer l'arbre du texte en l'arbre de la question. Le contexte étant pris en compte lors du calcul du coût, l'ordre d'application des opérations est important. Les nœuds n'ayant aucune ancre sont soit supprimés (s'ils sont dans l'arbre du texte), soit insérés (s'ils sont dans l'arbre de la question). Pour les relations de nœuds identiques, l'opérateur de déplacement sera appliqué de manière à récupérer une structure identique entre les deux arbres. Pour les deux autres relations, l'opérateur de substitution sera appliqué.

3. Evaluation

3.1. *Cadre expérimental*

La méthode a été évaluée sur les corpus de questions et de documents des campagne Clef 2004 et Clef 2005 (395 questions). Trois mesures ont été utilisées pour nos tests : l'Accuracy, le MRR et le Rappel. L'Accuracy correspond au pourcentage de questions pour lesquelles la réponse retournée en première position correspond à la réponse recherchée. Le MRR (Mean Reciprocal Rank) est un score permettant de mesurer la position des réponses recherchées. Si pour une question, la bonne réponse est retournée en première position, on ajoute 1 au MRR. Si pour une autre question, la bonne réponse est cette fois en troisième position, on ajoute 1/3, et ainsi de suite. Le Rappel correspond au pourcentage de questions dont la liste des réponses retournées contient la bonne réponse. Lors de l'évaluation, le nombre de réponses maximum pouvant être retournées par Ritel a été fixé à 10. Plus ce nombre est grand, plus le Rappel est potentiellement élevé. Nos questions sont séparés selon le type de la réponse attendue. Cela nous permet de voir pour quelles catégories de questions notre approche est la mieux adaptée. Cette typologie est déterminée automatiquement lors de la génération d'un DDR. Ces résultats sont présentés dans le tableau 1.

3.2. *Analyse des résultats*

		Méthode proposée		Système Ritel		
Type de la réponse	#questions	Acc	MRR	Acc	MRR	Rappel
nombre	42	14.3%	0.27	26.2%	0.39	71.4%
cause	15	20.0%	0.22	20.0%	0.22	40.0%
divers	21	20.0%	0.28	30.0%	0.36	60.0%
organisation	75	11.0%	0.2	31.5%	0.39	54.8%
lieu	56	14.3%	0.28	53.6%	0.59	73.2%
date	50	**28.0%**	**0.38**	24.0%	0.37	66.0%
pers	102	13.9%	0.26	41.6%	0.47	63.4%
inconnu	34	10.7%	0.16	14.3%	0.19	53.6%
total	395	15.2%	0.25	33.2%	0.40	61.0%

Tableau 1. *Résultat de l'évaluation selon la catégorie de la question*

On peut constater que notre méthode dégrade les résultats obtenus par Ritel, avec une baisse de 18%. Ces résultats peuvent être expliqués par l'état actuel des fonctions de coût. Pour le moment, la définition de ces fonctions reste très simple, et ne permet pas de calculer précisément le coût linguistique de chaque opération. De plus, la gestion du contexte est limitée aux nœuds voisins du nœud sur lequel on applique un opérateur, ce qui empêche une gestion globale du contexte. On peut constater une amélioration (pour le type date). Pour certains types, cette baisse est faible (catégories raison et inconnu). Une raison possible est que notre méthode mesure mieux la proximité syntaxique.

4. Conclusions et perspectives

Nous avons présenté une méthode fondée sur un calcul du coût de transformation entre deux arbres issus de l'analyse et représentant la question et un texte pouvant contenir la réponse. Cette méthode a été testée et nous avons analysé nos premiers résultats et évoqué les possibilités d'amélioration. Notre approche donne pour l'instant de mauvais résultats principalement à cause du fonctionnement élémentaire de nos fonctions de coût. Celles-ci ne couvrent pas toutes les modifications de l'information imputables aux transformations. Une voie possible serait de travailler sur des arbres plus profonds et plus structurés apportant d'avantage d'information. Nous comptons aussi améliorer nos résultats actuels en implémentant une meilleure gestion du contexte des phrases qui pour le moment est strictement local. Cette approche serait particulièrement intéressante pour avoir une meilleure gestion de la synonymie : certaines relations de synonymie ne seraient valides que dans un contexte précis. Nous nous intéressons aussi à la gestion des anaphores et en particulier aux pronoms personnels dans les documents (Vicedo *et al.*, 2000). Nous comptons également utiliser cette méthode pour améliorer l'interaction entre l'utilisateur et un système interactif ; les opérations appliquées pour trouver une réponse peuvent permettre de donner des informations sur la conduite de la recherche à l'utilisateur. Enfin, nous voudrions nous inspirer des techniques dites d'implication textuelle (*textual entailment*) pour améliorer nos résultats. Une application du domaine est d'ailleurs présentée dans le challenge RTE du réseau PASCAL (Dagan *et al.*, 2006).

5. Bibliographie

Dagan I., Glickman O., Magnini B., « The PASCAL Recognising Textual Entailment Challenge », *LNAI*, 2006.

Gillard L., Bellot P., El-Bèze M., « D'une compacité positionnelle à une compacité probabiliste pour un système de Questions/Réponses », *Conférence en Recherche d'Information et Applications (CORIA) 2007*, 2007.

Ligozat A.-L., Exploitation et fusion de connaissances locales pour la recherche d'informations précises, PhD thesis, LIMSI - Université Paris XI, 2006.

Magnini B., Kouylekov M., « Tree Edit Distance for Recognizing Textual Entailment : Estimating the Cost of Insertion », *the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy*, 2006.

Rosset S., Galibert O., Illouz G., Max A., « Interaction et recherche d'information : le projet Ritel », *Traitement Automatique des Langues*, 2006.

Vicedo J. L., Ferrandez A., « Importance of Pronominal Anaphora resolution in Question Answering systems », *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL '00)*, p. 555-562, 2000.

Villaneau J., Rosset S., Galibert O., « Semantic Relations for an Oral and Interactive Question-Answering System », *Semantic Representation of Spoken Language (SRSL) 2007*, 2007.

Zhang K., Shasha D., « Simple and fast algorithms for the editing distance between trees and related problems », *SIAM J. COMPUT*, 1989.