Exploitation d'une structure pour les questions enchaînées

Kévin Séjourné
Université de Paris Sud XI, Limsi/CNRS
kevin.sejourne@limsi.fr

Résumé. Nous présentons des travaux réalisés dans le domaine des systèmes de questions réponses (SQR) utilisant des questions enchaînées. La recherche des documents dans un SQR est perturbée par l'absence des éléments utiles à la recherche dans les questions liées, éléments figurant dans les échanges précédents. Les récentes campagnes d'évaluation montrent que ce problème est sous-estimé, et n'a pas fait l'objet de technique dédiée. Afin d'améliorer la recherche des documents dans un SQR nous utilisons une méthode récente d'organisation des informations liées aux interactions entre questions. Celle-ci se base sur l'exploitation d'une structure de données adaptée à la transmission des informations des questions liées jusqu'au moteur d'interrogation. Le moteur d'interrogation doit alors être adapté afin de tirer partie de cette structure de données.

Abstract. We present works realized in the field of the questions answering (QA) using chained questions. The documents search in QA system is disrupted because useful elements are missing for search using bound questions. Recents evaluation campaigns show this problem as underestimated, and this problem wasn't solve by specific techniques. To improve documents search in a QA we use a recent information organization method for bound questions to the interactions between questions. This methode is bases on the operation of a special data structure. This data structure transmit informations from bound questions to the interrogation engine. Then the interrogation engine must be improve to take advantage of this data structure.

Mots-clés : Question réponse enchaînée.

Keywords: chained question answering.

1 Introduction

Dans la foulée des systèmes de réponse à des questions, il a été envisagé de considérer qu'un utilisateur était susceptible de poser plusieurs questions sur une même thématique, des questions qui donc s'enchaînent les unes aux autres. Ainsi, chaque question doit être interprétée en connaissance de l'historique des questions et des réponses précédentes. Il y a eu récemment plusieurs campagnes d'évaluation de systèmes de questions réponses (SQR) où des questions enchaînées étaient proposées. Selon les corpus, les questions enchaînées peuvent faire référence à un contexte global (ou sujet global) préalablement introduit comme ce fut le cas dans la campagne d'évaluation TREC (Zhou *et al.*, 2006). Elles peuvent aussi faire référence aux réponses précédentes ou avoir de multiples références vers d'autres questions. Les questions enchaînées peuvent présenter toutes ces difficultés sans les annoncer explicitement, comme dans la campagne d'évaluation des SQR Clef07 (Penas *et al.*, 2007) ; la première question peut même parfois avoir le rôle d'un introducteur de contexte. Le tableau 1 montre un exemple de groupe de questions enchaînées. On voit sur cet exemple que pour répondre aux questions 2, 3 ou 4, il faut connaître le contexte posé par les questions précédentes. Parfois les SQR sont inter-lingues, c'est-à-dire que la langue des questions est différente de la langue des documents dans lesquels on cherche la réponse, comme c'était le cas pour une des pistes de la campagne Clef07. C'est le corpus de cette campagne que nous utilisons par la suite dans cet article.

Le système Musclef (figure 1) développé au Limsi, et qui a participé aux précédentes campagnes classiques de Questions-Réponses, a globalement une architecture semblable aux SQR classiques. C'est lui qui nous sert de base pour tester ces nouvelles conditions. Le problème que nous nous posons est alors de savoir utiliser aux mieux les informations des dépendances entre questions pour améliorer la recherche des documents, des phrases et ainsi des réponses.

Dans cet article nous présenterons d'abord la structure que nous construisons afin de décrire les interactions entre des questions. Nos résultats vont dépendre des performances de cette étape. Ensuite, nous présenterons une méthode de pondération dynamique des termes des documents dans un moteur de recherche pour la résolution de questions enchaînées.

2 Analyse des questions enchaînées

Nous devons d'abord trouver les dépendances entre les questions d'un même groupe, et pour cela étudier les différents phénomènes linguistiques qui permettent d'inférer leur présence sans

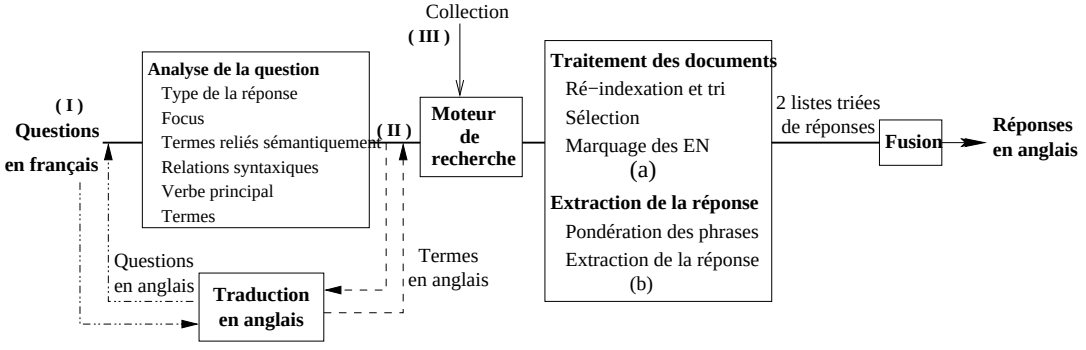

FIG. 1 – Architecture du système Musclef en mode inter-lingue

Exploitation d'une structure pour les questions enchaînées

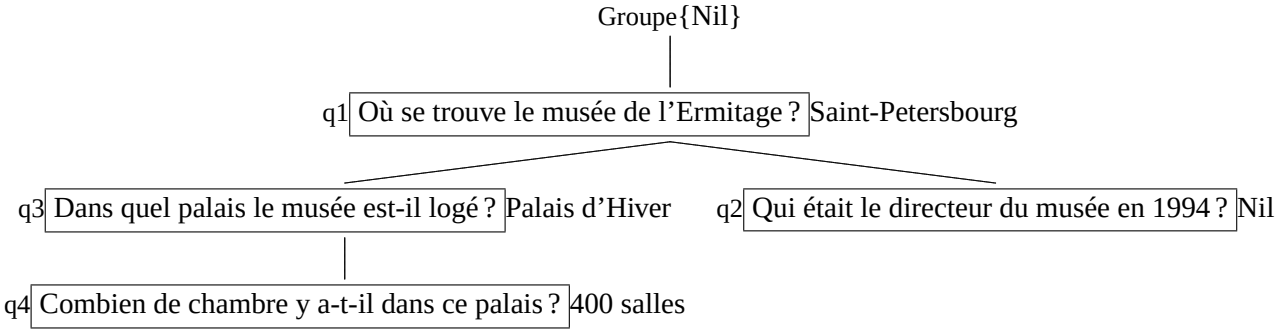

FIG. 2 – L'arbre correspondant au groupe du tableau 1

1	Où se trouve le musée de l'Ermitage ?
2	Qui était le directeur du musée en 1994 ?
3	Dans quel palais le musée est-il logé ?
4	Combien de chambres y a-t-il dans ce palais ?

TAB. 1 – Exemple d'un groupe de questions enchaînées tirées du corpus utilisé pour la campagne d'évaluation CLEF 2007.

trop de bruit (c'est-à-dire de fausses dépendances). Pour améliorer la recherche dans les documents, nous devons représenter les dépendances entre les questions d'un même groupe. En s'inspirant des travaux sur les structures de dialogue (Vilnat, 2005)(van Schooten & op den Akker, 2006), de la nature séquentielle des groupes de questions et du partage des termes des questions déjà résolues du groupe, il a été proposé dans (Séjourné, 2008) d'organiser un groupe de questions en un arbre (figure 2) représentant les liens entre les différentes questions d'un groupe.

À sa racine nous trouvons le contexte commun à toutes les questions dans un nœud nommé *groupe*. Le contexte est composé d'une liste d'éléments faisant éventuellement référence à la réponse. À chaque autre nœud sont indiqués une question et son contexte propre. La structure de l'arbre traduit les dépendances qui sont identifiées.

La structure d'arbre permet de représenter efficacement les groupes où les questions ne reprennent que le contexte issu de la première question. L'ajout des éléments d'informations utiles à la recherche d'information à chaque nœud permet une représentation homogène des groupes où les questions réutilisent des contextes liés les uns aux autres. Les questions qui comme la première, ne réutilisent pas le contexte des précédentes, sont rattachées au nœud *groupe*. Il permet également de recevoir des éléments contraignant l'espace de recherche exprimée hors question à la manière des évaluations de Trec 2006 [1] (Hickl *et al.*, 2006).

Nous présentons maintenant la méthode utilisée pour trouver les dépendances entre les questions d'un même groupe. Nous pouvons formaliser la probabilité d'existence d'une dépendance en un calcul d'argMax sur une collection de traits. Soit α et β deux couples de questions et réponses. Soit Γ l'ensemble des termes que l'utilisateur doit fournir dans ces 2 questions pour que la réponse à β puisse être trouvée. Γ dépend des stratégies du SQR utilisé ainsi que des corpus de documents dans lesquels la réponse est cherchée. La probabilité P à calculer est l'existence

[1] Un contexte était donné explicitement pour chaque groupe de questions.

de l'évènement : β *est une sous-partie de* Γ *strictement plus petite que* Γ. Notons que même si Γ n'est pas optimum (l'utilisateur pourrait fournir plus d'informations), rien n'empêche d'avoir suffisamment d'information pour que la probabilité d'association d'une dépendance soit maximalement correcte. Soit Ψ une collection de traits munis d'une fonction d'évaluation (type de la question, catégorie, ou des combinaisons plus complexes, traits issus de l'analyse de la question comme illustré sur la figure 1) permettant de décrire l'apport et la capacité d'unification de β dans Γ. Alors P est la somme des plus grandes possibilités d'apport et capacité d'unification, soit :

$$\boxed{P_{\alpha\beta} = argMax(\Sigma_{Ti\in\Psi} eval(Ti, \alpha\beta))}$$

C'est une simplification de la méthode présentée dans (Séjourné, 2008). Il est alors plus simple de définir une stratégie utilisant un seuil de probabilité de correction, en dessous duquel nous décidons que la dépendance n'existe pas. Le calcul des dépendances via Ψ est axé sur les informations disponibles dans les SQR classiques, puisqu'il réutilise directement les traits issus de l'analyse de la question.

Nous avons utilisé les mêmes traits pour nourrir l'algorithme générique de construction des dépendances. Nous avons ajouté un trait concernant les répétitions de segments de texte communs à deux questions. Un apprentissage nous a permis de déterminer que la présence de segments communs de plus de 15 caractères qui ne sont en position préfixe ni dans l'une ni dans l'autre, tend à montrer qu'il n'y a pas de dépendance unitaire entre les deux questions.

Quand l'homme politique irlandais Willie O'Dea est-il né ?

Où l'homme politique irlandais Willie O'Dea est-il né ?

Le système effectue donc une recherche du plus long segment commun entre les deux questions, puis il teste sa longueur et celles des préfixes, pour éliminer les cas exposés précédemment. Ce critère est utilisé en complément des autres critères.

Nous avons aussi re-utilisé la même méthode d'évaluation, et le même corpus de question (Clef@QA2007). Des réponses à des questions en français sont cherchées dans des documents en anglais.[2] Soit «Commun» l'ensemble des dépendances communes à l'ensemble des dépendances annotées « à la main » et à l'ensemble de dépendances trouvées par le système. Le rappel est alors calculé en prenant le rapport de « Commun » sur le nombre total de dépendances annotées. La précision est calculée en prenant le rapport de « Commun » sur le nombre total de questions en rang au moins deux d'un groupe. [3] L'ajout de ce trait à ceux utilisés précédemment permet une détection des dépendances unitaires avec une F-mesure d'environ 0.8 pour un rappel de 0.739 et une précision de 0.883. C'est un gain de 11% en terme de F-mesure lié à un gain en précision et en rappel. Nous pouvons alors construire la structure d'arbre présentée ci-dessus en fonction des dépendances ainsi calculées, c'est cette structure qui constituera le contexte dans la suite de ce texte.

3 Moteur de recherche

Pour trouver dans la collection de référence, les documents susceptibles de contenir la réponse à une question posée, nous utilisons des moteurs de recherche à base de réalisation d'une fonction

[2]Ce corpus contient 53 groupes d'au moins 2 questions dont 133 questions en position au-delà de 2, et une majorité de groupe de 4 questions(37). L'annotation «à la main» révèle 96 dépendances.

[3]La F-mesure est calculée par la formule : $(P * rappel * précision)/(rappel + précision)$ Nous avons choisis $P = 2$ pour nos évaluations.

de score. Les documents sont alors ordonnées et les n meilleurs sélectionnés. La pondération consiste à attribuer un poids à chaque terme utilisé pour la recherche, qu'il provienne de la question considérée ou d'un couple question-réponse dont il dépend, en faisant varier leur influence dans le score total en fonction de leur position dans la structure de dépendance. Il n'est pas évident de choisir une pondération qui *a priori* aurait des propriétés correctes pour chaque type de terme, qu'il soit issu de la requête, d'un document, d'une traduction, de l'ajout de synonymes de termes de la question.

Choix de la corrélation des termes. En nous appuyant sur la structure de dépendance calculée précédemment, nous proposons de *réaliser des tests de corrélation des termes d'un niveau à l'autre*. Il s'agit de prendre deux termes de niveaux contigus dans l'arbre et de regarder s'ils sont présents simultanément dans un document. Le principe est ensuite généralisé aux arbres ayant un nombre quelconque de niveaux. Pour chaque terme d'un niveau, il faut regarder s'il existe au moins un terme de chaque niveau dont il dépend avec lequel il est présent dans le document dont il faut calculer le score. Des tests incrémentaux par niveau de la présence simultanée des termes servent alors de pondération implicite et dynamique.

3.1 Utilisation pour la recherche des documents.

Formes possibles de la généralisation. Cette généralisation peut prendre plusieurs formes. Il est possible de choisir que tous les termes des niveaux précédents soient présents, mais comme les stratégies de sélection et extension de termes ajoutent de nombreux mots clefs de sens voisins dans la requête, il est peu probable d'obtenir un effet satisfaisant. Il est possible de choisir que seule contribuera au score du document, soit la plus grande corrélation de termes soit chaque sous corrélation de termes. Il est possible d'éliminer arbitrairement les documents ne présentant aucun terme d'un rang donné, mais l'impact des termes de rang inférieur est ignoré et la résistance au glissement de sujet est inférieure. Il est possible d'oublier le contexte de rang supérieur à un rang où aucun document ne possède au moins un terme de ce rang etc ...

Score à base de somme de corrélation La corrélation des termes rang à rang avec une généralisation et une contribution au score pour chaque sous corrélation de termes possède d'autres avantages. [4].

1) *Tailles des groupes* : Un terme n'est effectivement pris en compte que s'il existe au moins un document contenant au moins un exemplaire de terme pour chaque rang du contexte. Jamais un terme de rang n ne peut prendre plus d'importance relative que la totalité des termes de rang $n-1$. La taille des groupes de termes pour chaque rang du contexte à un impact moins important que dans les stratégies de pondérations par rang du contexte.

2) *Divergence de score* : Si la généralisation aboutit, alors cette méthode résout les problèmes liés à la pondération. La pondération est exprimée en fonction des termes. La divergence est alors contrôlée par la présence corrélée des termes dans les documents. Le terme d'une question ne sera jamais écrasé par un gros coefficient, car ou bien les termes devront être présents simultanément ou bien ils ne comptent pas. La présence corrélée est en elle-même une garantie de pondération qui respecte le critère de divergence.

[4] Nous faisons l'hypothèse que la stratégie de sélection des termes dispose d'un maximum(soit m ce maximum) dans le nombre de termes par rang sélectionné. Pour les calculs de convergence nous faisons l'hypothèse que m est aussi une valeur cible pour le nombre de termes à sélectionner dans la stratégie de sélection de termes. m n'est utilisé qu'à la section suivante.

3.2 Construction du scoring par cooccurence.

La métrique du TfxIdf peut être déclinée en différentes variantes. Nous pouvons trouver les plus utilisées dans (Manning *et al.*, 2008). Nous noterons :
– $\#Term(t, D)$ = Nombre d'occurrences du terme «t» dans le document D. ($\#Term$)
– $\#Docs(t)$ = Nombre de documents présentant au moins une occurrence du terme «t» dans une collection donnée. ($\#Docs$)
– N = Nombre total de document dans la collection.
Notre méthode consiste à modifier la manière dont le score est calculé de manière à tenir compte de la présence simultanée des termes de la question et du contexte dans les documents. Nous faisons l'hypothèse qu'un terme d'un contexte utilisé sans aucun terme de la question a moins de valeur qu'un terme trouvé de la question sans son contexte.

Une variante du TFxIDF. Le Tf[5] est construit sur la base de la fréquence des termes dans un document. l'Idf[6] est construit sur la base du nombre de documents contenant un terme par rapport au nombre total de documents. Le score est construit de cette manière $Score(Q, D) = \Sigma_{t_i \in Q} Tf * Idf$. Souvent une méthode de normalisation est ajoutée pour remédier aux disparités de longueur des documents et de dispersion des termes (Salton & Buckley, 1988). Comme nous ne cherchons pas seulement un terme x, mais des corrélations de termes, nous devons calculer une valeur fondée sur le nombre de documents contenant un terme de la question et des termes du contexte par rapport au nombre total de documents. Pour un même document, il faut tenir compte des risques d'absences et de mauvais choix des termes. Ces risques sont importants pour les termes du contexte dont l'erreur réelle dépend aussi de la détection des dépendances entre les questions. Il nous faut donc étendre le TfxIdf, pour tenir compte des niveaux du contexte.

La «partie» Tf est augmentée avec les cooccurrences éventuelles des termes dans le document tout en tenant compte des erreurs faites à la détermination des termes. La «partie» Idf est réduite pour tenir compte de la quantité de documents qui présente ces mêmes cooccurrences. Soit t_{ij} le terme de rang du contexte i qui est le j-ieme de son niveau. Si $i = 0$ alors il s'agit d'un terme de la question. Soit $nombreDeRangs$ le nombre de rang du contexte.

Construisons un indicateur de la fréquence des termes de la question et du contexte dans un document, le Tf'. Nous accordons de l'importance à un terme du contexte de rang n uniquement si un terme du contexte du rang $n-1$ est présent dans le document. Cela se fait selon l'algorithme suivant :

$\boxed{freq(t, D) = 1/\#Term \mid \#Term > 0}$ et $\boxed{freq(t, D) = 0 \mid \#Term = 0}$.

Construisons l'indicateur de fréquence des dépendances comme un système de fréquence des termes d'un rangs pondéré par les fréquences des termes précédents.

$$\boxed{\begin{array}{rl} Tf'(D) \ = & \sqrt{\Sigma_1^i \Pi_1^i \Pi_1^j (freq(t_{i,j}, D) \ + \ 1) - nombreDeRangs} \\ & \| i \ \in rangs\ du\ contexte\ ,\ j \ \in terme\ du\ rang(i) \end{array}}$$

C'est la somme des produits des fréquences d'un rang par le produit des fréquences des sous rangs, donc une corrélation niveau à niveau.

Nous commençons par calculer l'impact pour les termes de rang 1, nous réalisons un produit des fréquences (au sens défini ci-dessus) pour obtenir un impact global pour le rang. Par rapport

[5] $Tf(t_i, D) = 1/(\#Term(t_i, D))$
[6] $Idf(t_i) = log(N) \ - \ log(1 + \#Docs(t_i))$

au Tf traditionnel, chaque rang est traité comme s'il s'agissait d'un terme unique, mais chaque rang est pondéré non pas par une valeur fixe, mais par le produit des fréquences de tous les sous-rangs précédents. *Il en résulte que moins les termes des premiers rangs sont présents, moins l'impact des termes des rangs les plus anciens est important.* Notons que si un terme de rang n est absent, alors il représente un élément neutre pour l'opération de multiplication Π. Si tous les termes de rang n sont absents, leur impact est exactement compensé par la soustraction finale par le nombre de rangs. Par exemple pour un contexte de profondeur 3 avec $m = 2$ nous obtenons le développement suivant :

Soit $t_{p,q}$ le q-ième	terme du p-ième rang et $freq(x, D) + 1 = f(x)$ alors
$Tf'(D)^2 \ + \ 3 \ =$	$f(t_{1,1}) * f(t_{1,2})$
	$+f(t_{1,1}) * f(t_{1,2}) \ * \ f(t_{2,1}) * f(t_{2,2})$
	$+f(t_{1,1}) * f(t_{1,2}) \ * \ f(t_{2,1}) * f(t_{2,2}) \ * \ f(t_{3,1}) * f(t_{3,2})$

Il est alors évident que les $i - 1 | i \in rangs\ du\ contexte$ premiers termes du produit des rangs agissent comme une pondération définie dynamiquement.

Construisons un indicateur de la fréquence des documents possédant des termes corrélés, l'Idf'[7]. Soit $\odot$ l'opérateur binaire commutatif de corrélation de présence de deux termes dans un document. $\#docs(t_{i,j} \odot t_{x,y})$ désigne donc le nombre de documents dans un corpus qui contiennent à la fois le y-ième terme du rang x du contexte et le j-ième terme du rang i du contexte. Un terme d'un rang du contexte ne peut être utilisé que si au moins un terme de chaque rang inférieur peut aussi être utilisé pour déterminer l'importance d'un nombre de documents. Dans le cas où tous les termes sont corrects et effectivement présents dans tous les documents contenant la bonne réponse la quantité $\#docs(t_i)$ peut donc être substituée par $\#docs(t_i \odot t_{1,x} \odot t_{2,y} \odot ... \odot t_{n,z})$ où les valeurs x y ... z varient dans les limites possibles du rang du contexte concerné. Notons que les t_i de la requête sont intégrés aux calculs séparément les uns des autres. Nous pouvons réduire nos contraintes en relâchant des termes du contexte de manière à autoriser des corrélations de présences de termes moins fortes. Plus la mesure est faible plus il existe un grand nombre de documents possédant ces termes corrélés. Par récursion nous pouvons obtenir la méthode de calcul suivante :

$Idf'(t_i) \ =$	$1 + log(N) - log(1+$
	$\#docs(t_i)$
	$+ \Sigma_1^x(\#docs(t_i \ \odot \ t_{1,x}) \mid x \in t_1 \ )$
	$+ \Sigma_1^x \Sigma_1^y(\ \#docs(t_i \ \odot \ t_{1,x} \ \odot \ t_{2,y}) \mid x \in t_1 \ , \ y \in t_2)$
	$+ ...$
	$+ \Sigma_1^x ... \ \Sigma_1^z(\ \#docs(t_i \ \odot \ t_{1,x} \ ... \ t_{n,z} \ ) \mid x \in t_1 \ , \ ... \ , \ z \in t_n$
	$, n \ = \ nombreDeRangs \ - \ 1))$

Pour un terme unique sans aucune dépendance nous retrouvons bien la formule de base[8] de calcul de l'Idf ($1 + log(N) - log(1 + docs(t_i))$). Imaginons maintenant que nous disposons d'un rang supplémentaire de dépendance. Le rang est ajouté à la partie précédente du calcul en faisant attention à la présence simultanée avec les termes de rangs inférieurs. Pour la présence simultanée, le système utilise l'opérateur de corrélation de présence. Chaque terme du rang est ajouté 1 à 1 en vérifiant la présence des termes de rangs inférieurs, la formule visualise bien cela

[7] Rappelons que $log(N/(1 + \#Docs(y_i))) \ = \ log(N) \ - \ log((1 + \#Docs(t_i)))$

[8] $log(N/(1 + \#Docs(y_i))) \ = \ log(N) \ - \ log((1 + \#Docs(t_i)))$

sous la forme $\Sigma_1^x | x \in t_1$. L'addition (Σ) et la corrélation de présence($\odot$) étant commutatives, la généralisation pour des dépendances avec plus de rangs ne pose pas de problèmes.

Variante du Score A notre variante du TfxIdf nous associons alors une variante de la méthode de calcul du score d'un document. Par généralisation, c'est une extension des méthodes de scores par fréquences[9].
Le score d'un document est alors défini par : $\boxed{score(Q, D) = \Sigma_{t_i \in Q} Tf'(D) * Idf'(t_i)}$

3.3 Evaluer la modification.

Voyons maintenant la méthodologie retenue pour évaluer l'impact sur les performances de la recherche dans les documents.

Déterminer la présence de la réponse dans un document. Dans un premier temps, une liste des réponses courtes attendue est réalisée pour chaque question semi-automatiquement. De ces réponses courtes, nous en déduisons des ensembles de patrons figés qui permettent de les identifier dans des documents. Nous calculons alors l'ensemble des documents contenant ces patrons. Nous bouclons alors sur deux opérations jusqu'à ce que le premier choix soit systématiquement réalisé. Soit, il y a suffisamment peu de documents, nous vérifions « à la main » pour chaque document que le patron figé qui est trouvé correspond bien à la réponse. Sinon nous sélectionnons alors un échantillon de documents que nous analysons à la main. Ces documents permettent de déterminer un ensemble de patrons secondaires « le contexte » qui doivent être présents dans le document pour que le patron réponse identifie vraiment la réponse. Et nous recalculons l'ensemble des documents contenant les patrons avec « le contexte ». Nous obtenons alors 2 ensembles, un ensemble de documents contenant les réponses, un ensemble de patrons de réponses sur une logique de type «et/ou» permettant d'obtenir les documents contenant les réponses. *In fine*, nous avons adapté le programme de sélection des documents réponses pour qu'il puisse évaluer les résultats retournés par les différentes versions des tests sur la recherche de document.

Caractéristiques de l'évaluation sur corpus. Notre évaluation a porté sur les 200 questions du corpus QA@Clef2007 en français avec réponse attendue à partir du corpus anglais de la Wikipédia de novembre 2006 et de l'année 1994 des journaux LA et GH. Nos patrons de bonnes réponses nous permettent de découvrir un maximum de 116 bonnes réponses et nous savons qu'il existe au moins 3 questions pour lesquelles aucune réponse ne se trouve dans les documents.

Les résultats bruts de nos évaluations sont récapitulés dans le tableau 3.3. Qalc récupère 100 documents qui sont transmis au module de sélection des phrases. La récupération de $n = 100$ ne se réalise vraiment que si la posting-list fait au moins n documents. Notre méthode de recherche de documents en une seule interrogation ne cherche pas à obtenir des documents supplémentaires en formulant une requête alternative. Une des raisons est que certaines questions n'ont pas de réponse dans les documents.

Le MRR(Ok) est calculé en ne tenant compte que des questions pour lesquelles au moins une

[9] Nous avons proposé ici les versions, «*racine carrée*» et «*quantité d'information*» des Tf' et Idf'. La raison en est que nous voulions obtenir un modèle proche de celui de Lucene pour les tests. La version «*quantité d'information*» du Tf' peut s'obtenir simplement en remplaçant la fonction «*racine carrée*» par une fonction du «$log + 1$». $Tf(D) = log(1 + 1/\#Term) = log(freq(D) + 1)$ or par construction nous avions choisi une étude à base de $\sqrt{(freq(D) + 1)}$.

Stratégie	Bonnes réponses	MRR(Ok)	Moyenne(Ok)	MRR(All)	Moyenne(All)
A	59	0.20	19.15	0.06	76.15
B	88	0.24	16.17	0.10	63.12
C	88	0.32	16.78	0.14	63.38
D	106	0.19	76.54	0.10	510.5
E	106	0.15	193.7	0.08	572.7
F	92	0.31	18.57	0.14	62.54

TAB. 2 – Caractéristiques des bonnes réponses pour différentes stratégies d'attribution de scores.

réponse a été trouvé : c'est la moyenne des inverses des rangs des questions pour lesquelles un document-réponse a été trouvé dans les n premiers documents. De même pour la Moyenne(Ok) qui est une moyenne de rang de document-réponse. Les MRR(All) et Moyenne(All) sont les approximations avec autant de décimales significatives que le MRR et la Moyenne traditionnelles. Contrairement au MRR(Ok) si une réponse n'est pas dans les n premiers documents nous comptons simplement zéro. C'est ou bien la somme inverse des rangs des questions ou bien zéro, divisé par le nombre total de questions. De manière similaire, la Moyenne(All) est calculée en comptant $n + 1$ s'il n'y a pas de document-réponse dans les n premiers documents. Les calculs des All sont réalisés sur une base de 200 questions, mais ce qui est vraiment intéressant, c'est l'apport relatif des différentes méthodes. Il est facile de recalculer à partir des OK n'importe quel MRR ou Moyenne.

Méthode A, le hors contexte Les questions sont traitées de manière traditionnelle sans prise en compte du contexte. Elles sont envoyées dans la méthode classique de Lucene. Nous constatons que par rapport à la plus mauvaise méthode en contexte basique, 29 nouvelles bonnes réponses sont trouvées, soit un gain de 49.15% en introduisant des termes du contexte.

Méthode B,D et C,E Les méthodes D et E ont été réalisés avec $n = 1000$ alors que les méthodes B et C ont été réalisé avec $n = 100$ Les méthodes B et D ont été réalisés avec la méthode par défaut de Lucene où l'origine des termes est oubliée... Les méthodes C et E ont été réalisés avec notre nouvelle méthode d'attribution des scores aux documents.

Méthode F, la fusion. Nous avons remarqué que les méthodes B et C ont des moyennes d'OK très inférieures à 50 ; or nous sélectionnons plus de 100 documents. Il est donc sans risque de soit réduire le nombre de documents soit prendre les 50 premiers des 2 méthodes. Ici nous avons pris les 50 premiers documents de B et C, puis retirés les doublons. Il aurait été possible d'aller chercher plus de 50 documents une fois les doublons retirés. [10]

L'explication principale vient de la nature du corpus. Comme les SQR modifiés en vue de faire de l'interaction ne sont pas encore vraiment déployés, la majorité des questions sont indépendantes. Par la nature même des liens entre les questions, il est difficile de créer des classes pour séparer les différents types de questions : Il apparait très clairement que l'ajout du contexte est un plus non négligeable. Il est moins évident que tenir compte de l'affordance du contexte

[10] Il aurait aussi été possible de faire un mélange alternatif tenant compte des rangs des questions (En rang 1 et 2 nous prendrions les questions en rang 1 de chaque méthode, etc...), cela permettrait d'augmenter le Mrr. En effet, les bonnes réponses sont classées en moyenne au rang 16-17 par les deux méthodes, cela remonterait leur rang moyen de 50+17 à 17+17. Les tests de fusion exacts n'auraient pas forcément été plus intéressants, car ce sont déjà nos meilleurs résultats. Nous observons le même Mrr(All) que pour le système C car ce sont les réponses du système C qui ont été mises en première place.

soit un plus. Si nous réduisons notre étude à l'ensemble des groupes de questions disposant d'une structure de dépendance non triviale, les résultats sont significativement meilleurs, mais en contrepartie la significativité des résultats est bien plus faible.

4 Conclusion

En exploitant les informations des dépendances entre questions, nous avons construit un modèle dynamique de la pondération des termes et documents basé sur la corrélation de présence de deux termes dans un document. Nous n'avons pas pu établir de boucle d'optimisation : tests, analyse, modification. Nous ne disposons pas de suffisamment de corpus pour cela, la difficulté imposée par le domaine ouvert empêche notamment des analyses trop fines des questions et des corpus de documents. Nous ne voulions pas risquer la critique du surapprentissage, seule la tactique de la fusion des deux sources de résultats a été réalisée puisque les analyses montrent qu'elle est statiquement fondée. Cette astuce déduite de la répartition des résultats a permis d'améliorer les résultats par rapport à celles existantes de la tâche de récupération des documents dans un SQR avec des questions enchaînées.

Un nouvel objectif serait d'optimiser à partir de nouveaux corpus, que ce soit par une meilleure organisation des calculs, une meilleure propagation des conséquences de l'existence d'un modèle d'enchaînement de questions. Nous n'avons pas tenu compte de l'impact de l'indexation des documents spécifiquement pour les dépendances. Il serait intéressant de tester s'il est possible de construire l'index différemment, de nettoyer les documents différemment afin de tenir compte dès l'indexation du type de calcul que nous allons réaliser. Nous devons aussi approfondir les avantages de la fusion des 2 stratégies de recherche que nous avons testés.

Références

HICKL A., WILLIAMS J., BENSLEY J., ROBERTS K., SHI Y. & RINK B. (2006). Question answering with lcc's chaucer at trec 2006. *15th Text REtrieval Conference, Gaithersburg*, p.1̃.

MANNING C. D., RAGHAVAN P. & SCHÜTZE H. (2008). *Introduction to Information Retrieval*.

PENAS A., FORNER P. & GIAMPICCOLO D. (2007). Guidelines for participants in qa at clef 2007. *CELCT, Trento(IT) and UNED, Madrid*, p.1̃.

SALTON G. & BUCKLEY C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, **24**(5), 513–523.

SÉJOURNÉ K. (2008). Une structure pour les questions enchainées. *RECITAL, Avignon, 9-13 juin*.

VAN SCHOOTEN B. & OP DEN AKKER R. (2006). Follow-up utterances in qa dialogue. *TALN-05*, **1**(46(3)).

VILNAT A. (2005). *Habilitation à diriger les recherches : Dialogue et analyse de phrases*. PhD thesis, University de Paris-Sud XI LIMSI/CNRS. 2009 :http :// www.limsi.fr /Individu/anne/HDR/MemoireHDR.pdf.

ZHOU Y., YUAN X., CAO J., HUANG X. & WU L. (2006). Fduqa on trec2006 qa track. *15th Text REtrieval Conference, Gaithersburg*, p. 1026–1033.