Répondre à des questions à réponses multiples : premières expérimentations

Mathieu-Henri Falco — Véronique Moriceau — Anne Vilnat

LIMSI-CNRS
Université Paris-Sud
91403 Orsay, France
prenom.nom@limsi.fr

RÉSUMÉ. *Une des difficultés majeures des systèmes de question-réponse concerne l'extraction des bonnes réponses depuis les documents sélectionnés par un moteur de recherche. En effet, il est souvent difficile de procéder à un recoupement des candidats réponses, notamment dans le cas des questions qui attendent plusieurs réponses. Afin de nous focaliser sur les conditions d'extraction et de recoupement de réponses multiples, nous avons construit un corpus en « conditions idéales » pour une dizaine de questions à réponses multiples. Nous mesurons ensuite l'apport de plusieurs techniques d'extraction et de validation des réponses et comparons les résultats obtenus sur ce corpus par deux systèmes de question-réponse : celles du système de question-réponse FIDJI (orienté vers les campagnes d'évaluation) et celles de Citron spécialisé dans l'extraction et le recoupement de réponses multiples.*

ABSTRACT. *Question-answering systems have multiples problems dealing with answer extraction from retrieved documents, especially when cross-checking is needed for questions expecting multiple answers. We focused on answer extraction and answer crossing by building an ideal corpus for a dozen of multiple answer questions. We performed a fine-grained evaluation on each strategy for answer extraction and validation. We compare the results of two question-answering systems : FIDJI which has successfully participated to several evaluation campaigns and Citron which is designed for multiple answer extraction and cross-checking.*

MOTS-CLÉS : *question-réponse, questions à réponses multiples, question liste*

KEYWORDS: *question-answering, multiple answer questions, list question*

1. Introduction

Les systèmes de question-réponse (*SQR*) ont pour but de fournir une réponse précise à une question formulée en langue naturelle ; nous nous intéressons ici aux SQR recherchant cette réponse dans une collection de documents. Cette recherche de la réponse comporte plusieurs étapes, dont notamment l'analyse de la question pour en extraire des mots-clefs qui serviront ensuite à la recherche de documents pertinents. Ces deux étapes génèrent inévitablement du bruit et les SQR essaient souvent d'extraire un candidat à la réponse (*candidat-réponse*) depuis un document ne contenant pas de bonne réponse.

Nous nous intéressons ici aux questions à réponses multiples (questions-ARM) c'est-à-dire des questions qui attendent plusieurs réponses. Nous cherchons dans un premier temps à améliorer nos techniques d'extraction de bonnes réponses pour ce type de questions et plaçons ainsi nos conditions de travail dans un cadre idéal : l'analyse de la question a été effectuée manuellement et les documents ont été filtrés pour contenir de façon certaine au moins une bonne réponse chacun. Dans un second temps, nous nous concentrons sur le recoupement des bonnes réponses extraites d'un ou plusieurs documents afin de constituer l'ensemble de bonnes réponses attendues.

Dans cet article, nous présentons d'abord un état de l'art sur comment les questions-ARM sont traitées en question-réponse et comment les réponses multiples sont structurées dans les documents. Puis nous définissons notre cadre de conditions idéales dans la section 3. Dans la section 4, nous présentons notre approche pour l'extraction de réponses à des questions-ARM. Enfin, nous discutons les résultats de l'extraction des réponses dans la section 5 et le recoupement des réponses dans la section 6.

2. Contexte et état de l'art

Nous définissons une question-ARM comme une question attendant plusieurs réponses correctes. Une question-ARM peut prendre la forme d'une question de type liste (*question-liste*) avec une formulation au pluriel (*Quelles sont les sept merveilles du monde ?*) mais peut également prendre la forme d'une question factuelle au singulier comme dans l'exemple suivant :

question : Quand la France a-t-elle perdu son triple A ?
passage 1 : *Le 13 janvier 2012, lors de sa revue des pays de la zone euro, S&P avait privé la France de son triple A historique.*
passage 2 : *Moody's retire son triple AAA à la France (le 19.11.2012, 22h18)*

Il existe 2 réponses correctes car 2 agences différentes ont dégradé la France en 2012 à 2 dates différentes.

Figure 1. *Exemple de question-ARM*

Il existe donc des questions-ARM pour lesquelles une seule réponse correcte pourrait suffire (cf. Figure 2) alors que pour d'autres, une liste exhaustive de réponses est attendue (par exemple, pour *Quelles sont les sept merveilles du monde ?*). Ces deux types de questions sont des questions à réponses multiples au sens où il existe plusieurs réponses correctes possibles.

Les éléments composant la liste de réponses peuvent être déjà sous la forme d'une liste dans un document mais ils peuvent aussi être répartis dans un document ou même dans plusieurs documents. Nous avons choisi de nous intéresser à l'extraction de réponses multiples sur le Web car cela nous permet de travailler en domaine ouvert et, étant donné le nombre important de documents, le travail de recoupement des réponses multi-documents s'avère indispensable. Nous utiliserons le terme *candidat-réponse* pour désigner un élément extrait d'un document et étant possiblement une des réponses multiples d'une question-ARM.

Nous présentons ici comment sont évaluées les questions-ARM durant les campagnes d'évaluation puis comment elles sont traitées par les SQR durant ces campagnes. Enfin, nous présentons comment les réponses multiples peuvent être structurées dans les documents.

2.1. *Les questions à réponses multiples dans les campagnes d'évaluation*

Les SQR peuvent se placer dans un cadre applicatif vis-à-vis d'un utilisateur ou dans un cadre évaluatif lors de campagnes d'évaluations. Nous présentons ici les SQR dans le contexte des campagnes d'évaluation. Un nombre de questions significatif et de différents types (factuelles, complexes, booléennes, définitions, liste et nil (question n'ayant pas de réponse dans les documents à disposition)) sont proposées aux SQR qui peuvent fournir, pour chaque question, plusieurs réponses classées (généralement de trois à cinq). Les réponses sont ensuite évaluées, souvent manuellement car il est très difficile de garantir l'unicité d'une bonne réponse dans une grande collection de documents.

Les questions-ARM telles que nous les définissons ne sont pas définies dans les campagnes d'évaluation : ainsi, si un SQR répond deux réponses correctes pour une même question, la métrique MRR (Mean Reciprocal Rank) utilisée dans les campagnes ne prendra en compte que le rang de la première réponse correcte.

La métrique d'évaluation est différente pour les campagnes proposant des questions-listes, selon que ces campagnes imposent un nombre de réponses explicite dans la question (*Quelles sont les 4 localisations possibles des neuroblastomes ?*) ou non (*Quels sont les secteurs qui recrutent ?*). Pour le premier type de question, la métrique utilisée est la précision moyenne (nombre de réponses correctes/nombre de réponses attendues) ; pour le second, c'est la F-mesure (en considérant l'ensemble des réponses jugées correctes par les évaluateurs). Dans les deux cas, les campagnes d'évaluation imposent que les réponses proviennent d'un seul extrait de document limité en nombre de caractères.

2.2. *Le traitement des questions à réponses multiples par les SQR*

Les SQR ayant participé à des campagnes comportant des questions-listes ont eu deux types d'approches : soit adapter leur traitement de questions factuelles aux listes, soit développer un traitement spécifique. L'adaptation à des questions-listes pouvait être de répondre un top-N des réponses trouvées : N étant fixe (5 pour (Chu-carroll *et al.*, 2004) et 20 pour (Wu *et al.*, 2003)), N pouvant dépendre d'un seuil déterminé par le SQR selon son système d'ordonnancement (Kaisser *et al.*, 2004) (Schlaefer *et al.*, 2007), ou N étant le nombre explicite d'éléments attendus lorsque ce dernier est mentionné dans la question-liste (Harabagiu *et al.*, 2001). Une autre adaptation est de reformuler la question-liste en question factuelle afin de déterminer le nombre d'éléments attendus, par exemple à propos de *the Gaza Strip* : *What were the settlements that were evacuated ?* en *How many were the settlements that were evacuated ?* (Bos *et al.*, 2007). Si aucun nombre n'est trouvé dans la collection, alors le système utilise soit le nombre explicite d'éléments attendus s'il est mentionné dans la question, soit un nombre fixé par défaut (Bos *et al.*, 2007).

Les SQR ayant développé un traitement spécifique pour les listes ont notamment utilisé la détection de doublons pour éviter la redondance de candidats-réponses, par exemple à l'aide d'une métrique de recouvrement de candidats-réponse (Monz Christof, 2001), de distance d'édition (Levenshtein sur les candidats-réponse), d'une mesure de similarité (cosinus sur la représentation syntaxique) (Schlaefer *et al.*, 2007). Certains SQR utilisent en plus la réconciliation de références à l'aide de ressources extérieures comme le Web, Wordnet, Wikipédia, etc. (Schlaefer *et al.*, 2007) (Dan I. Moldovan and *et al.*, 2007). À travers l'expansion de requête, la co-occurrence des candidats-réponses (au niveau de la phrase ou du document) est également très fréquemment utilisée comme critère de validation (Razmara *et al.*, 2008) (Wang *et al.*, 2008) (Figueroa *et al.*, 2008).

2.3. *Les listes dans les documents textuels*

Nous nous intéressons ici aux éléments structuraux, c'est-à-dire aux objets regroupant plusieurs éléments de façon structurée dans des documents textuels ou HTML, et plus particulièrement les listes et les tableaux. En effet, ces objets structurants sont suspectibles de contenir plusieurs candidats-réponses à des questions-listes en particulier et à des questions-ARM en général.

Les listes ont été beaucoup étudiées du point de vue discursif et les travaux de (Péry-Woodley, 2000), (Luc, 2001), (Bras *et al.*, 2008), (Laignelet, 2009), (Ho-Dac *et al.*, 2010) ont notamment abouti à la définition du terme *structure énumérative* pour désigner l'objet composé d'une amorce (phrase introductrice), d'une énumération composée d'items (entité co-énumérée caractérisée par diverses marques typographiques, dispositionnelles, lexico-syntaxiques) et éventuellement d'une conclusion. Les structures énumératives peuvent exister à plusieurs niveaux : document, paragraphe, phrase. Nous reprenons ce concept de structure énumérative et posons les

définitions suivantes :

– *énumération verticale* : l'amorce est délimitée par un " :" et l'amorce ainsi que chacun des items sont séparés par retour-chariot ;

– *énumération horizontale* : l'amorce est délimitée par un " :" et les items sont séparés par un symbole de ponctuation comme un point-virgule ou une virgule ;

– *énumération intra-phrastique* : c'est une énumération qui ne comporte pas d'amorce délimitée par " :" (l'amorce et le premier élément sont dans la même phrase) et ne dépasse pas le cadre de la phrase (par opposition aux *énumérations intra-paragraphiques*).

Quant aux structures de tableaux, elles ont surtout été étudiées dans le cadre du traitement de documents HTML dans le but notamment de typer les cases, soit à des fins de visualisations ergonomiques, soit pour de l'extraction d'information. Deux types d'approches dominent : à bases de règles (Gatterbauer *et al.*, 2007), (Tajima *et al.*, 2008) et par apprentissage automatique sur un corpus annoté manuellement (Wang *et al.*, 2002).

3. Corpus en conditions idéales

Afin de mieux nous concentrer sur la tâche de l'extraction de bonnes réponses ainsi que celle de leur recoupement, nous avons mis en place un corpus de conditions idéales aussi bien pour les questions-ARM que pour les documents Web utilisés.

3.1. *Pourquoi un corpus en conditions idéales ?*

Lors de notre étude des questions-ARM en corpus (Falco *et al.*, 2012), nous avons constaté que pour la plupart des questions, les réponses étaient présentes sur le Web sous forme de structures énumératives ou de tableaux et qu'elles étaient aussi largement réparties dans plusieurs documents. Or les données des campagnes d'évaluation des SQR (questions et collections de documents) pour le français proposant des questions-listes (EQueR (Ayache *et al.*, 2006) et Quaero (Quintard *et al.*, 2010)) ne sont pas représentatives de ces phénomènes. Nous avons donc choisi de constituer un corpus issu du Web que nous appelons "corpus en conditions idéales" car il a été construit de manière à s'assurer de la présence des phénomènes à traiter dans les documents.

Ces conditions idéales nous permettent donc de disposer de façon certaine de documents issus du Web contenant des structures énumératives et imposant une phase de recoupement des candidats-réponses. De plus, les questions-ARM sélectionnées présentent des problématiques significatives découvertes dans notre étude de corpus. Enfin, ces conditions idéales comportant peu de questions nous permettent de réaliser une évaluation manuelle plus rapidement et avec une certitude sur le rappel (le nombre de réponses correctes étant arrêté).

Comme nous l'expliquons par la suite, ces conditions nous permettent également de résoudre plusieurs difficultés inhérentes aux SQR pour se concentrer uniquement sur l'extraction de réponses multiples. Ainsi, une erreur lors du prétraitement des documents ou de l'analyse de la question ne pourra se répercuter jusqu'à l'extraction d'un candidat-réponse ; chaque document contiendra au moins un candidat-réponse correct et enfin, la mauvaise résolution d'un pronom anaphorique ne conduira pas à l'extraction d'un candidat-réponse incorrect.

3.2. *Conditions idéales pour les questions*

Nous avons choisi quatorze questions-ARM issues de notre corpus d'étude des questions-ARM (Falco, 2012) : des questions-ARM de type temporel (les plus fréquentes dans notre corpus) et des questions-listes volontairement explicites (marque de pluriel dans la question et nombre de réponses attendues pour certaines) :

– questions générées ex-nihilo : *Quand s'est déroulée la Commune de Paris ?*, *Quand la deuxième guerre mondiale s'est-elle terminée ?*, *Quand est sorti l'Ibook ?*, *Quand se déroule la fête de la bière ?*, *Quand la France a-t-elle perdu son triple A ?*, *Quels sont les fruits à consommer en automne ?* ;

– questions provenant de campagnes d'évaluation (Quaero 2008 et EQueR) : *Quels pays étaient candidats à l'organisation de la coupe du monde 2018 ?*, *Dans quels clubs a joué Nicolas Anelka ?*, *Quand le PSG a-t-il gagné la coupe de France ?* ;

– questions générées à partir de documents contenant les réponses : *Quels sont les noms des sept nains ?*, *Quelles sont les sept merveilles du monde ?* ;

– questions générées à partir de documents du corpus Annodis[1], contenant des structures énumératives : *Quelles sont les architectures possibles d'un système de télécommunications ?*, *Quels polluants ont été dispersés dans l'atmosphère lors de l'effondrement du World Trade Center le 11 septembre 2001 ?*, *Quelles sont les distributions Linux ?*.

Certaines questions provenant des campagnes d'évaluation ont été modifiées partiellement, par exemple en modifiant la date mentionnée afin de disposer de documents plus récents.

L'analyse de chacune des questions a ensuite été réalisée manuellement de façon à en extraire idéalement toutes les informations nécessaires : le type de la question (liste, factuelle), le type de la réponse attendue, le focus (l'élément de la question qui porte l'information), le verbe principal et des mots-clefs. Ainsi, pour la question *Dans quels clubs joue Nicolas Anelka ?*, le type de la réponse attendue est un *club*, le focus est *Nicolas Anelka* et le verbe principal *jouer*.

1. http ://redac.univ-tlse2.fr/corpus/annodis

3.3. *Conditions idéales pour les documents*

Les documents contenant les réponses correctes ont été récupérés depuis Internet au format HTML à partir de plusieurs moteurs de recherche (Bing, Exalead et Google) puis une validation manuelle de la présence des réponses a été effectuée (pour cette raison, le nombre de documents à récupérer a été volontairement restreint à une dizaine maximum par question).

Chaque document a ensuite été découpé en plusieurs passages de quelques phrases contenant chacun au moins une réponse correcte.

Les coréférences ont également été résolues manuellement en remplaçant tous les référents anaphoriques par les référés. La date de publication des documents a été également été manuellement extraite. Enfin, un recensement des réponses correctes dans chacun des passages a été effectué, un passage pouvant contenir plusieurs réponses correctes.

4. Extraction de réponses multiples avec Citron

Nous avons développé le programme Citron qui est spécialisé dans l'extraction et le recoupement de réponses multiples. Il est codé en Java et a pour objectif de pouvoir être utilisable par tout SQR à des fins de validation de réponses. Citron procède par étapes centrées sur le type attendu de la réponse et travaille au niveau de la phrase. Les étapes s'appliquent séquentiellement puis recoupent les informations recueillies.

Citron commence par analyser syntaxiquement tous les passages sélectionnés pour une question afin de disposer de la totalité des dépendances syntaxiques puis il extrait des candidats-réponses du type attendu depuis ces dépendances à l'aide de patrons et réalise de la résolution temporelle si besoin. Puis, il recherche des candidats-réponses indirectement depuis ces dépendances par similarité contextuelle sur le focus et le verbe principal de la question. Ensuite, il recherche des candidats-réponses du bon type dans les structures énumératives et les tableaux. Enfin, il utilise la Wikipédia pour désambiguïser et valider si nécessaire le type des candidats recueillis.

Dans les sous-sections suivantes, nous illustrons les différentes stratégies avec la question-ARM *Dans quels clubs a joué Nicolas Anelka ?*.

4.1. *Règles de réécriture syntaxique*

Citron utilise l'analyseur syntaxique XIP (Aït-Mokhtar *et al.*, 2002) qui produit une analyse en dépendances et détecte les entités nommées. XIP permet la création de lexique et de règles de réécriture : Citron utilise les règles du SQR FIDJI (Moriceau *et al.*, 2010) qui permettent entre autres d'extraire des relations de définition, auxquelles s'ajoutent celles de détection de certaines structures énumératives (par exemple, pour détecter les structures énumératives commençant par la préposition "parmi"). Pour le

moment, nous n'avons repéré syntaxiquement que les structures énumératives horizontales et intra-phrastiques. Les énumérations verticales ont été formatées en énumérations horizontales (remplacement des retours-chariots par un espace) sinon elles sont trop difficilement analysables syntaxiquement.

4.2. *Résolution temporelle*

Citron utilise également des règles syntaxiques pour repérer des désignations d'expressions temporelles. Nous nous sommes limités pour le moment aux expressions temporelles sans prendre en compte les désignations d'événements. Quand une expression temporelle est décrite de façon relative à une autre, Citron effectue un calcul de normalisation. Par exemple, dans la phrase *Standard & Poor's a dégradé* ***vendredi*** *d'un cran la note de la dette française, de AAA à AA+*, la date de publication de l'article (le samedi 14/01/2012) est nécessaire pour normaliser "vendredi" en "vendredi 13 janvier 2012".

4.3. *Recherche de candidats-réponses du type attendu*

Citron recherche dans un premier temps des candidats-réponses identifiés syntaxiquement comme étant du type attendu. Les dépendances recherchées sont : *DEFINITION(type attendu,candidat-réponse)*, *ATTRIBUT-DE* pour les compléments du nom, *ATTRIBUT-NN* pour les modifieurs du nom et *TYPE(candidat-réponse)*, où TYPE désigne un type d'entité nommée reconnu par XIP. La relation DEFINITION est créée par XIP lors de l'analyse en dépendance grâce aux règles de réécriture. Dans notre exemple, le passage *Anelka rejoint le club anglais de Manchester City* valide le type de la réponse (ici "club") grâce à la relation ATTRIBUT-DE entre *club* et *Manchester City*.

4.4. *Utilisation de la similarité contextuelle*

Une première étape commence par rechercher dans la phrase la présence d'une dépendance syntaxique entre le verbe principal, le focus et un candidat-réponse. Pour notre exemple, on recherche les relations entre le verbe *jouer*, le focus *Anelka* et un candidat-réponse : le candidat *Chelsea* est donc extrait de la phrase *Agé de 33 ans, Anelka joue depuis la saison 2007-2008 à* ***Chelsea*** *pour lequel il a inscrit 59 buts* car on trouve les relations *SUJET(jouer, Anelka)* et *VMOD(jouer, Chelsea)*.

Dans la deuxime étape, si un candidat-réponse dont le type a été validé précédemment a été détecté par XIP comme une entité nommée correspondant au type attendu alors toutes les entités nommées de ce type dans les documents deviennent des candidats-réponses. Dans notre exemple, l'analyse syntaxique donne les relations *DEFINITION(club, Arsenal)* et *ORGANISATION(Arsenal)* : toutes les entités nommées

du type ORGANISATION deviennent donc des candidats-réponses dont il faudra valider le type “club” : pour notre exemple, on obtient les entités *Chelsea FC*, *Arsenal*, *Real Madrid*.

4.5. *Utilisation de la Wikipédia pour la validation du type des réponses*

Nous utilisons Wikipédia pour valider le type des candidats-réponses quand la validation n’a pas été pas possible lors des étapes précédentes. Nous reprenons l’approche de (Grappy, 2011) qui valide le type d’un candidat-réponse dans la page Wikipédia associée, à la différence que nous faisons l’hypothèse que cette validation peut se faire à l’aide des seuls paragraphes d’introduction de la page : en effet, l’introduction contient très souvent une définition de l’entité qui permet de valider son type (par exemple, la première phrase d’introduction de la page sur “Manchester City” indique que *Manchester City Football Club est un club de football basé à Manchester*). Ainsi, nous n’analysons que les quelques phrases d’introduction plutôt que l’article entier. Pour chaque candidat-réponse à valider, si l’article Wikipédia correspondant existe, l’introduction est analysée syntaxiquement pour vérifier si le type est bien conforme au type de la réponse attendue. Nous utilisons les mêmes patrons syntaxiques que ceux présentés précédemment pour la recherche des candidats-réponses.

Si une page d’homonymie existe pour un candidat-réponse, chaque définition est analysée et les termes de la question sont également recherchés afin de valider le type : par exemple, le candidat-réponse "Chelsea" est défini dans Wikipédia comme un prénom (*Chelsea est un prénom féminin*), un lieu (*Chelsea est un nom de lieu*) ou un club de football (*le Chelsea Football Club, un club de football anglais)*. Ici, c’est la définition contenant le mot “club” qui sera retenue.

Un dump de la Wikipédia a été prétraité de manière à obtenir un index des termes possédant une page d’homonymie.

4.6. *Détection des structures énumératives*

L’algorithme de détection de structures énumératives se concentre sur les tableaux, les énumérations horizontales et les énumérations intra-phrastiques.

Le contenu des tableaux est extrait de façon à relier les cases entête aux cases données selon une syntaxe prédéfinie, facilitant ainsi le repérage et l’extraction syntaxique des informations (Falco *et al.*, 2012). Par exemple, le contenu du tableau de la figure 2 est extrait et formaté pour obtenir une relation syntaxique de type définition entre “club” et “Paris-SG”. En effet, après notre formatage, on obtient (case entête en gras, case donnée en italique) : **Saison** : *95-96* / **Club** : *Paris-SG* / **pays** : *FRA* .

L’extraction du contenu des énumérations horizontales et des énumérations intra-phrastiques s’appuie sur des patrons typographiques et la détection de l’amorce si elle est précédée d’un symbole “ :”. Les items d’une structure énumérative partagent un

Saison	**Club**	**Pays**	(...)
95-96	Paris-SG	FRA	(...)
96-97(fév)	Paris-SG	FRA	(...)
96-97	Arsenal	ANG	(...)
(...)	(...)	(...)	(...)

Figure 2. *Extrait d'un tableau avant son formatage*

point commun identitaire appelé *enumeraTheme* (Ho-Dac *et al.*, 2010) généralement présent dans l'amorce et qui type chacun de ces items. Par exemple, l'amorce "*L'effondrement des tours a dispersé dans l'atmosphère de Manhattan de nombreux polluants dangereux : de la dioxine, du plomb (...)*" permet avec XIP de créer les relations de typage POLLUANT(dioxine) et POLLUANT(plomb).

5. Expériences et résultats

Nous étudions à présent l'extraction de candidats-réponses sur notre corpus en conditions idéales avec le programme *Citron*. Une évaluation granulée est proposée de façon à quantifier l'apport de chacune des stratégies de Citron (toutes cumulables) et également l'apport de la résolution des coréférences. Les performances de Citron sont comparées à une baseline et au SQR FIDJI (Moriceau *et al.*, 2010) capable de traiter les questions-listes dans un cadre de campagne d'évaluation classique.

Nous avons défini une baseline qui extrait les candidats-réponses selon deux critères :

– les noms en relation avec le focus et le verbe principal de la question ;

– les candidats-réponses dont le type peut être vérifié syntaxiquement dans la collection de documents.

Le but est de comparer les résultats de Citron avec ceux de la baseline d'un point de vue général mais aussi de mesurer l'apport de chacune des stratégies adoptées par Citron.

5.1. *Performances globales de Citron*

Le tableau 1 montre les résultats (précision moyenne, rappel et F-mesure) obtenus par la baseline et Citron. Ces résultats sont calculés à partir du nombre de réponses correctes identifiées manuellement dans notre corpus, chaque candidat-réponse ne pouvant être proposé qu'une seule fois (recoupement sur la forme de surface). Les résultats montrent une F-mesure supérieure à la baseline, notamment grâce à de meilleures performances pour les questions dont les réponses se trouvent dans des structures énumératives (7 à 14) alors que les performances sont plus similaires pour les questions

Système	baseline			Citron		
Métrique / Question	P	R	F	P	R	F
(1) Quand s'est déroulée la Commune de Paris ?	0,92	0,80	**0,86**	0,72	0,87	0,79
(2) Quand la 2ième guerre mondiale s'est... ?	0,85	0,79	0,81	0,86	0,86	**0,86**
(3) Quand est sorti l'Ibook ?	0,78	0,78	**0,78**	0,47	0,78	0,58
(4) Quand se déroule la fête de la bière ?	1,00	1,00	**1,00**	0,86	1,00	0,92
(5) Quand la France a-t-elle perdu son triple A ?	0,71	0,38	**0,50**	0,63	0,38	0,48
(6) Quand le PSG a-t-il gagné la coupe de... ?	0,71	0,19	0,29	0,52	0,41	**0,46**
(7) Quels pays étaient candidats à... ?	1,00	0,93	**0,97**	1,00	0,93	**0,97**
(8) Quels sont les fruits à consommer... ?	0,00	0,00	0,00	0,91	0,61	**0,73**
(9) Dans quels clubs a joué Nicolas Anelka ?	0,33	0,18	0,24	0,80	0,55	**0,65**
(10) Quels sont les noms des sept nains ?	0,00	0,00	0,00	1,00	1,00	**1,00**
(11) Quelles sont les sept merveilles du monde ?	0,00	0,00	0,00	0,75	0,86	**0,80**
(12) Quelles sont les architectures possibles... ?	0,50	0,50	**0,50**	0,50	0,50	**0,50**
(13) Quels polluants ont été dispersés dans... ?	0,00	0,00	0,00	0,83	0,63	**0,71**
(14) Quelles sont les distributions Linux ?	0,50	0,10	0,17	0,78	0,70	**0,74**
Moyennes	0,52	0,4	0,44	0,76	0,72	**0,73**

Tableau 1. *Résultats de la baseline et de Citron.*

temporelles (1 à 6). Nous avons également mesuré la précision locale de l'extraction de réponse, à savoir le nombre moyen de réponses correctes extraites par passage par rapport aux nombres de bonnes réponses correctes contenues dans le passage. En moyenne, la baseline extrait 0,33 réponse par question contre 0,44 pour Citron.

5.2. *Apports de chacune des stratégies de Citron*

Nous avons également mesuré l'apport individuel de chacun des paramètres de Citron (voir tableau 2) : d'abord en n'utilisant qu'une seule des stratégies à la fois, puis en les combinant. Nous avons calculé la F-mesure, ainsi que le nombre de réponses correctes différentes et le nombre de réponses fournies différentes pour l'ensemble de questions Les résultats montrent que certaines stratégies sont inefficaces lorsqu'elles sont utilisées seules. Les résultats obtenus sur les passages où les coréférences ont été résolues s'expliquent par le fait que seules deux questions sont concernées (2 passages pour la question (5) et 42 pour la question (9)). La recherche de candidats-réponses par similarité contextuelle obtient des résultats similaires à la baseline. La première partie de notre approche par similarité utilise en effet la même approche que la baseline mais nous pensions que la deuxième étape de notre approche apporterait un gain. Ce n'est pas le cas, sans doute car dans notre corpus, le nombre de passages par question est restreint et ne permet pas de valider cette stratégie. Quant à l'utilisation de la Wikipédia, elle permet bien de filtrer des candidats-réponses incorrects mais en élimine un

Stratégie utilisée individuellement	P	R	F	Nombre réponses correctes différentes	Nombres réponses fournies différentes
Validation par Wikipédia	0,53	0,40	0,43	66	82
Similarité contextuelle	0,52	0,40	0,44	67	88
Coréférence résolue	0,52	0,40	0,44	67	88
Baseline	**0,52**	**0,40**	**0,44**	67	88
Règles syntaxiques	0,54	0,49	0,50 (+14%)	112	155
Structures énumératives	0,71	0,61	0,62 (+41%)	91	115
Toutes les stratégies ensemble sauf	P	R	F	Nombre réponses correctes différentes	Nombre réponses fournies différentes
Structures énumératives	0,55	0,49	0,50 (-32%)	112	151
Règles syntaxiques	0,73	0,59	0,63 (-14%)	90	109
Validation par Wikipédia	0,76	0,71	0,72	136	178
Similarité contextuelle	0,75	0,72	0,72	138	184
Coréférence résolue	0,76	0,72	0,73	138	180
Toutes ensemble	**0,76**	**0,72**	**0,73**	138	180

Tableau 2. *Résultats détaillés de Citron.*

correct, tombant ainsi légèrement en-dessous la baseline. Les résultats montrent cependant l'importance de traiter les structures énumératives ainsi que l'apport de nos règles de réécriture syntaxique : avec ces deux stratégies seulement, on obtient une F-mesure de 0,72, soit un résultat quasiment identique à l'utilisation des cinq stratégies en même temps.

5.3. *Comparaison avec FIDJI*

Le tableau 3 montre les résultats obtenus par Citron et FIDJI. Nous souhaitions confronter Citron à un SQR qui obtient globalement de bons résultats aux campagnes d'évaluation.

Citron obtient pour quasiment toutes les questions une meilleure F-mesure. Pour chaque question, FIDJI a réalisé une bonne analyse des questions et trouve les bons documents mais il ne parvient pas à en extraire les réponses correctes. Citron extrait également plus de réponses correctes différentes que FIDJI : 138 sur 180 réponses différentes contre 63 sur 81, FIDJI se situant sous la baseline. Cela s'explique notamment par le fait que de nombreuses réponses se trouvent dans des structures énumératives : FIDJI est capable de les repérer dans les passages mais ne réussit pas à en extraire les réponses correctes alors que Citron possède une stratégie dédiée à ces structures.

Système	FIDJI			Citron		
Métrique / Question	P	R	F	P	R	F
(1) Quand s'est déroulée la Commune de Paris ?	0,92	0,80	**0,86**	0,72	0,87	0,79
(2) Quand la 2ième guerre mondiale s'est... ?	0,70	0,50	0,58	0,86	0,86	**0,86**
(3) Quand est sorti l'Ibook ?	0,00	0,00	0,00	0,47	0,78	**0,58**
(4) Quand se déroule la fête de la bière ?	1,00	0,33	0,50	0,86	1,00	**0,92**
(5) Quand la France a-t-elle perdu son triple A ?	0,50	0,08	0,13	0,63	0,38	**0,48**
(6) Quand le PSG a-t-il gagné la coupe de... ?	0,75	0,33	**0,46**	0,52	0,41	**0,46**
(7) Quels pays étaient candidats à... ?	1,00	0,60	0,75	1,00	0,93	**0,97**
(8) Quels sont les fruits à consommer... ?	0,79	0,29	0,43	0,91	0,61	**0,73**
(9) Dans quels clubs a joué Nicolas Anelka ?	0,71	0,23	0,34	0,80	0,55	**0,65**
(10) Quels sont les noms des sept nains ?	0,00	0,00	0,00	1,00	1,00	**1,00**
(11) Quelles sont les sept merveilles du monde ?	1,00	0,14	0,25	0,75	0,86	**0,80**
(12) Quelles sont les architectures possibles... ?	0,67	0,50	**0,57**	0,50	0,50	0,50
(13) Quels polluants ont été dispersés dans... ?	0,00	0,00	0,00	0,83	0,63	**0,71**
(14) Quelles sont les distributions Linux ?	0,00	0,00	0,00	0,78	0,70	**0,74**
Moyennes	0,57	0,27	0,35	0,76	0,72	**0,73**

Figure 3. *Résultats de FIDJI et de Citron.*

6. Recoupement de réponses avec Citron

Comme nous l'avons vu dans la section 2, les SQR traitant des questions-listes lors des campagnes d'évaluation effectuent souvent une détection des doublons dans leur ensemble de réponses. Nous posons ici qu'un doublon existe lorsque deux candidats-réponses font référence à une mme entité (réconciliation de référence). Dans le cadre des questions-ARM, et plus encore dans un cadre utilisateur, cette détection est donc fondamentale. Nous l'effectuons à deux niveaux que nous désignons par recoupement de surface et partitionnement temporel.

6.1. *Recoupement de surface*

Nous appliquons plusieurs techniques sur la forme de surface afin d'effectuer, dans un premier temps, des groupes de plusieurs candidats-réponses. Pour cela, nous utilisons entre chaque couple de candidats-réponses :

– la distance de Jaro-Winkler avec 0,85 pour seuil (par exemple, pour regrouper *Real de Madrid* et *Real Madrid*) : nous l'avons modifiée en ajoutant une pénalité pour les insertions de plus de 3 lettres (*fraise* et *fra**mbo**ise*) ainsi que pour l'ajout de lettres en début de mot (*airelle* et ***mir**abelle*),

– la plus longue chaine commune (*FC Trappes* pour la comparaison de *FC Trappes* et *FC Trappes-St Quentin*),

– une détection d'acronymes (*PSG* et *Paris-SG* pour *Paris Saint-Germain*).

6.2. *Partitionnement temporel*

Lorsque Citron a extrait plusieurs candidats-réponses de type temporel, il les normalise de façon à pouvoir, dans un premier temps détecter les dates, et dans un deuxième temps les partitionner. La normalisation nous permet de résoudre par exemple les dates relatives en dates absolues grâce à la date du document. Cette opération permet ensuite de regrouper plusieurs candidats-réponses désignant une même date. Enfin, un partitionnement reposant sur la distance moyenne entre chaque année d'un candidat-réponse est effectué de manière à faire émerger des segmentations parmi les groupes de candidats-réponses. Par exemple, pour la question (5), nous obtenons le partitionnement suivant :

– 2012-01-13 : *vendredi, vendredi soir*

– 2011-07 : *en juillet*

– 2012-11-19 : *lundi soir, le 19.11.2012, lundi, Novembre 19*

6.3. *Évaluation du regroupement des réponses*

Nous avons évalué manuellement le regroupement de surface et le partitionnement temporel sur les résultats de Citron. Le recoupement de surface a produit un mauvais regroupement de réponses pour treize corrects (92,3 %), l'erreur étant due à notre mesure de Jaro-Winkler modifiée qui est trop laxiste. Le recoupement de surface a été notamment très utile pour les questions (8) (par exemple, recoupement de *pêche de vigne* et *pêche*) et (9) (par exemple, recoupement de *Liverpool FC* et *Liverpool* ou de *Fenerbahce* et *Fenerbahçe*).

Le partitionnement temporel a réalisé 15 partitions pour les six questions temporelles. Il s'est avéré bon pour toutes les questions et particulièrement pertinent pour les questions (1) et (5), par exemple pour la question (1) :

– une période 1789-1795 contenant 8 candidats-réponses compris entre ses deux dates ;

– une période 1871 contenant 5 candidats-réponses désignant cette année-là.

Le partitionnement des candidats-réponses temporels est pour le moment réalisé en fin de programme mais nous réfléchissons à l'implémenter plus en amont dans la phase de validation. Il nous permettrait alors notamment de détecter un éventuel critère variant en étudiant les similarités dans chacune des partitions. Par exemple, le critère variant pour la question (5) serait alors l'agence de notation ayant dégradé la France :

– 2012-01-13 : *Standard and Poor's* ;

– 2011-07 : *Egan-Jones* ;

– 2012-11-19 : *Moody's*.

7. Conclusion

Nous nous sommes intéressés à la tâche d'extraction de réponses à des questions à réponses multiples. En nous mettant dans des conditions idéales, nous avons pu tester les performances du programme de validation de réponse Citron : nous avons constaté qu'une utilisation combinée de tous ses paramètres apportait de meilleurs résultats pour l'extraction de réponses multiples, notamment grâce aux structures énumératives. Ces dernières nécessiteront un prétraitement important, mais très probablement rentable, pour être analysées efficacement en conditions réelles. Citron obtient également de meilleurs résultats que le SQR FIDJI, qui obtient de bons résultats lors des campagnes d'évaluation classiques. Ceci nous laisse penser que Citron pourrait obtenir d'aussi bons résultats que FIDJI sur des données plus larges.

À très court terme, nous allons donc d'évaluer Citron sur un plus grand nombre de questions et sur une collection de documents rééls. Nous envisageons également une évaluation de Citron dans un cadre utilisateur pour évaluer le système selon plusieurs points : la qualité des réponses proposées et la satisfaction des utilisateurs vis-à-vis de la façon dont lui sont présentées les réponses.

8. Bibliographie

Ayache C., Grau B., Vilnat A., « EQueR : the French evaluation campaign of question-answering systems », *Proceedings of The Fifth International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy*, 2006.

Aït-Mokhtar S., Chanod J.-P., Roux C., « Robustness beyond shallowness : incremental deep parsing », *Nat. Lang. Eng.*, 2002.

Bos J., Guzzetti E., Curran J. R., « The Pronto QA System at TREC 2007 : Harvesting Hyponyms, Using Nominalisation Patterns, and Computing Answer Cardinality », *TREC-16*, 2007.

Bras M., Prévot L., Vergez-Couret M., « Quelle(s) relation(s) de discours pour les structures énumératives ? », CMLF (Congrès mondial de linguistique française), 2008.

Chu-carroll J., Czuba K., Prager J., Blair-goldensohn S., « IBM's PIQUANT II in TREC2004 », *TREC-13*, 2004.

Dan I. Moldovan and C. C., Bowden M., « Lymba's PowerAnswer 4 in TREC 2007 », *TREC-16*, 2007.

Falco M.-H., « Typologie des questions à réponses multiples pour un système de question-réponse (Typology of Multiple Answer Questions for a Question-answering System) [in French] », *Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 3 : RECITAL*, Grenoble, France, 2012.

Falco M.-H., Moriceau V., Vilnat A., « Kitten : a tool for normalizing HTML and extracting its textual content », *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, 2012.

Figueroa A., Neumann G., « Finding Distinct Answers in Web Snippets », *In the 4th International Conference on Web Information Systems and Technologies*, INSTICC Press, p. 26-33, 5, 2008.

Gatterbauer W., Bohunsky P., Herzog M., Krüpl B., Pollak B., « Towards domain-independent information extraction from web tables », *Proceedings of the 16th international conference on World Wide Web*, WWW '07, ACM, p. 71-80, 2007.

Grappy A., Validation de réponse dans un système de question-réponse, PhD thesis, 2011.

Harabagiu S., Moldovan D., Pasca M., Moldovan D., Surdeanu M., Gîrju R., Mihalcea R., Lacatusu F., Morarescu P., Bunescu R., Rus V., « Answering complex, list and context questions with LCC's QUestion-Answering Server », *TREC-10*, 2001.

Ho-Dac L.-M., Péry-Woodley M.-P., Tanguy L., « Anatomie des structures énumératives », 19-23 juillet 2010, 2010.

Kaisser M., Becker T., « Question Answering by Searching Large Corpora With Linguistic Methods », *TREC-13*, 2004.

Laignelet M., Analyse discursive pour le repérage automatique de segments obsolescents dans les documents encyclopédiques, PhD thesis, 2009.

Luc C., « Une typologie des énumérations basée sur les structures rhétoriques et architecturales du texte », *TALN*, 2001.

Monz Christof M. d. R., « Tequesta : The University of Amsterdam's Textual Question Answering System », *TREC-10*, 2001.

Moriceau V., Tannier X., « FIDJI : Using Syntax for Validating Answers in Multiple Documents », *Information Retrieval, Special Issue on Focused Information Retrieval*, 2010.

Péry-Woodley M.-P., « Une pragmatique à fleur de texte : approche en corpus de l'organisation textuelle », 2000. HDR.

Quintard L., Galibert O., Adda G., Grau B., Laurent D., Moriceau V., Rosset S., Tannier X., Vilnat A., « Question Answering on web data : the QA evaluation in Quæro », *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta*, 2010.

Razmara M., Kosseim L., « Answering List Questions using Co-occurrence and Clustering. », *LREC*, European Language Resources Association, 2008.

Schlaefer N., Ko J., Betteridge J., Sautter G., Pathak M., Nyberg E., « SEMANTIC EXTENSIONS OF THE EPHYRA QA SYSTEM FOR TREC 2007 », *TREC-16*, 2007.

Tajima K., Ohnishi K., « Browsing large HTML tables on small screens », *UIST*, p. 259-268, 2008.

Wang R. C., Schlaefer N., Cohen W. W., Nyberg E., « Automatic Set Expansion for List Question Answering », *EMNLP*, 2008.

Wang Y., Hu J., « A machine learning based approach for table detection on the web », *Proceedings of the 11th international conference on World Wide Web*, ACM, p. 242-250, 2002.

Wu M., Zheng X., Duan M., Liu T., Strzalkowski T., « Questioning Answering By Pattern Matching, Web-Proofing, Semantic Form Proofing », *TREC-12*, p. 578-585, 2003.