Répondre à des questions à réponses multiples sur le Web

Mathieu-Henri Falco

UNIVERSITÉ PARIS-SUD

École Doctorale : ED 427

École doctorale d'Informatique de Paris-Sud (EDIPS)

Laboratoire : LIMSI-CNRS

Laboratoire d'Informatique Mécanique et Sciences de l'Ingénieur (UPR 3251)

Discipline : INFORMATIQUE

THÈSE DE DOCTORAT

Soutenue le 22 mai 2014 par

Mathieu-Henri Falco

Répondre à des questions à réponses multiples sur le Web

Composition du jury

Co-encadrante :	M^me^ Véronique Moriceau	**Maitre de conférence** (LIMSI-CNRS - Université Paris-Sud)
Directrice de thèse :	M^me^ Anne Vilnat	**Professeure** (LIMSI-CNRS - Université Paris-Sud)
Rapporteure :	M^me^ Marie-Francine Moens	**Professeure** (Katholieke Universiteit Leuven)
Rapporteur :	M. Patrice Bellot	**Professeur** (LSIS - Aix-Marseille Université)
Examinatrice (présidente) :	M^me^ Sophie Rosset	**Directrice de recherche** (CNRS)
Examinateur :	M. Thierry Baccino	**Professeur** (LUTIN - Université de Paris 8)

Résumé

Les systèmes de question-réponse renvoient une réponse précise à une question formulée en langue naturelle. Les systèmes de question-réponse actuels, ainsi que les campagnes d'évaluation les évaluant, font en général l'hypothèse qu'une seule réponse est attendue pour une question. Or nous avons constaté que, souvent, ce n'était pas le cas, surtout quand on cherche les réponses sur le Web et non dans une collection finie de documents.

Nous nous sommes donc intéressés au traitement des questions attendant plusieurs réponses à travers un système de question-réponse sur le Web en français. Pour cela, nous avons développé le système Citron capable d'extraire des réponses multiples différentes à des questions factuelles en domaine ouvert, ainsi que de repérer et d'extraire le critère variant (date, lieu) source de la multiplicité des réponses. Nous avons montré grâce à notre étude de différents corpus que les réponses à de telles questions se trouvaient souvent dans des tableaux ou des listes mais que ces structures sont difficilement analysables automatiquement sans prétraitement. C'est pourquoi, nous avons également développé l'outil Kitten qui permet d'extraire le contenu des documents HTML sous forme de texte et aussi de repérer, analyser et formater ces structures.

Enfin, nous avons réalisé deux expériences avec des utilisateurs. La première expérience évaluait Citron et les êtres humains sur la tâche d'extraction de réponse multiples : les résultats ont montré que Citron était plus rapide que les êtres humains et que l'écart entre la qualité des réponses de Citron et celle des utilisateurs était raisonnable. La seconde expérience a évalué la satisfaction des utilisateurs concernant la présentation de réponses multiples : les résultats ont montré que les utilisateurs préféraient la présentation de Citron agrégeant les réponses et y ajoutant un critère variant (lorsqu'il existe) par rapport à la présentation utilisée lors des campagnes d'évaluation.

Abstract

Question answering systems find and extract a precise answer to a question in natural language. Both current question-answering systems and evaluation campaigns often assume that only one single answer is expected for a question. Our corpus studies show that this is rarely the case, specially when answers are extracted from the Web instead of a frozen collection of documents.

We therefore focus on questions expecting multiple correct answers from the Web by developping the question-answering system Citron. Citron is dedicated to extracting multiple answers in open domain and identifying the shifting criteria (date, location) which is often the reason of this answer multiplicity. Our corpus studies show that the answers of this kind of questions are often located in structures such as tables and lists which cannot be analysed without a suitable preprocessing. Consequently we developed

the Kitten software which aims at extracting text information from HTML documents and also both identifying and formatting these structures.

We finally evaluate Citron through two experiments involving users. The first experiment evaluates both Citron and human beings on a multiple answer extraction task : results show that Citron was faster than humans and that the quality difference between answers extracted by Citron and humans was reasonable. The second experiment evaluates user satisfaction regarding the presentation of multiple answers : results show that users have a preference for Citron presentation aggregating answers and adding the shifting criteria (if it exists) over the presentation used by evaluation campaigns.

Quand la vie vous envoie des citrons, faites-en de la limonade **Proverbe**

L'équilibre acido-basique est une grande fonction du corps humain qui vise à réguler le pH de celui-ci. (—) Le citron, bien qu'acide au goût, a paradoxalement un effet basifiant/alcalinisant sur l'organisme **Article *Équilibre_acido-basique* de Wikipédia (version du 10 juin 2014)**

- (le doctorant) *Ce serait génial si Citron pouvait être accessible depuis l'extérieur dès demain.*
- (le service informatique) *Si c'est pour Citron, c'est normal que ça soit pressé.*

Remerciements

Merci aux co-encadrantes, rapporteurs et examinateurs d'avoir accepté respectivement de co-encadrer, rapporter et examiner cette thèse. Merci pour votre sincérité : ce fut un plaisir de défendre ma thèse devant vous.

Un énorme merci supplémentaire à Véronique pour sa lucidité et maturité d'encadrement.

Merci au laboratoire riche en chaleur humaine, sentiments authentiques et performances sportives : les ressentir quotidiennement était un réel plaisir intériorisé.

Table des matières

Table des figures

Liste des tableaux

INTRODUCTION

Les systèmes de question-réponse se différencient des moteurs de recherche par des données d'entrée différentes et la finalité de leur objectif. En effet, là où les moteurs de recherche renvoient des liens vers des documents pertinents en regard d'une requête composée de mots-clés, les systèmes de question-réponse renvoient une réponse précise à une question formulée en langue naturelle. La première différence provient donc des données d'entrées puisqu'il va s'agir, pour les systèmes de question-réponse, d'une question ciblant (quasiment) sans ambiguïté une information précise alors que, pour les moteurs de recherche, une juxtaposition de mots-clés peut recouvrir plusieurs champs sémantiques. La seconde différence provient de la tâche supplémentaire d'extraction d'information pour non seulement trouver les documents contenant la réponse correcte mais également l'extraire, d'autant plus qu'il peut exister plusieurs réponses correctes à une même question.

Regardons par exemple les problèmes qui apparaissent lorsqu'on pose la question *Quels sont les trois pays frontaliers de la Bosnie-Herzégovine ?* à un moteur de recherche, en l'occurrence Google. La figure 0.1 illustre les trois problèmes suivants :

1. les documents contenant l'intitulé exact de la question (mais pas les réponses) sont renvoyés ;
2. des mots vides (*sont*, *qui*) sont utilisés pour la recherche qui renvoie alors des documents peu pertinents ;
3. des termes discriminants de la question (*bordent*, *trois pays*) permettent de renvoyer des documents thématiquement pertinents mais ne contenant pas la réponse.

Les trois réponses attendues au moment de la saisie de la requête (2011) sont la Croatie, la Serbie et le Monténégro. En utilisant le moteur de recherche avec une requête composée du focus de la question (ce sur quoi porte la question : *Bosnie-Herzégovine*), du type de la réponse (*pays*) et d'un attribut qualifiant la réponse (*frontalier*), on réduit le bruit. La figure 0.2 montre que Google renvoie alors deux des réponses attendues (Monténégro et Croatie) avec des extraits permettant d'expliquer ces réponses mais qui se trouvent dans des documents différents (documents 1 et 4). Ceci impose donc à l'utilisateur d'ouvrir les documents renvoyés et de les parcourir pour repérer les réponses à sa question, ce qui évidemment peut prendre plus ou moins de temps.

Les systèmes de question-réponse, grâce à une analyse de la question et à une extraction de la (les) réponse(s) précise(s), permettent de corriger ces inconvénients. Traditionnellement, ils extraient les réponses de documents au format texte (articles de journaux, encyclopédiques, etc.) et seuls quelques systèmes utilisent des collections de documents

FIGURE 0.1 : Requête Google : *Quels sont les trois pays qui bordent la Bosnie-Herzégovine ?* (3 octobre 2011).

issus du Web pourtant disponibles en grande quantité et sur de nombreuses thématiques. Pouvoir traiter des documents issus du Web permettrait donc non seulement de couvrir énormément de thèmes mais également de disposer de documents récents. De plus, les documents issus du Web utilisent des éléments structuraux comme les listes ou les tableaux susceptibles de contenir des réponses aux questions, or ces éléments sont très difficilement exploitables une fois les balises de mise en page supprimées.

La tâche d'extraction d'information d'une réponse correcte étant déjà extrêmement difficile, les systèmes de question-réponse sont principalement développés pour se concentrer sur l'extraction d'une seule et unique réponse correcte (souvent la plus fréquente), délaissant ainsi les questions attendant plusieurs réponses comme celle de l'exemple précédent. Or, il existe plusieurs types de questions pouvant attendre plusieurs réponses correctes. C'est notamment le cas des questions temporelles dont on ne peut savoir *a priori* si plusieurs réponses correctes existent (*Quand le PSG a-t-il remporté la coupe de France ?*

FIGURE 0.2 : Requête Google : *+pays +frontalier +Bosnie-Herzégovine* (3 octobre 2011)/

En 1982, 1983, 1993, 1995, 1998, 2004, 2006 et 2010) et c'est également le cas de questions dont la validité des réponses évoluent en fonction du moment où elles sont posées (*Qui a gagné le tour de France 2010 ? le 6 février 2012, Alberto Condador a été déclassé au profit d'Andy Schleck)*. De plus, certaines de ces réponses peuvent faire référence à une même entité comme *Olympique de Marseille* et *OM* : il est donc nécessaire de pouvoir agréger les réponses, c'est-à-dire pouvoir regrouper les références à une même entité.

Les systèmes de question-réponse sont évalués durant des campagnes d'évaluation et cet aspect de réponses multiples à une question y est difficilement évalué. En effet, dans le cas des questions attendant plusieurs réponses (une liste de réponses par exemple), il est souvent imposé que toutes les réponses proviennent d'un même document, voire même parfois d'un même paragraphe ou d'une même phrase, ce qui empêche de fournir des réponses correctes différentes trouvées dans des documents différents (comme dans la figure 0.2). De plus l'évaluation durant ces campagnes ne mesure que le statut correct ou incorrect de la réponse mais ne s'intéresse pas à la satisfaction des utilisateurs vis-à-vis de la réponse proposée.

MOTIVATIONS ET OBJECTIFS

À la suite de ces observations, nous avons choisi de nous intéresser au traitement des questions attendant plusieurs réponses à travers un système de question-réponse sur le Web en français. Un certain nombre de constats nous ont guidé dans ce choix :

- les documents issus du Web contiennent beaucoup d'éléments structuraux (tableaux, listes) susceptibles de contenir des réponses à des questions attendant des réponses multiples ;
- un utilisateur qui cherche une réponse à une question sur le Web par l'intermédiaire d'un moteur de recherche peut y passer beaucoup de temps surtout si les réponses sont multiples et réparties dans plusieurs documents, et ceci sans garantie d'être finalement satisfait ;
- les systèmes de question-réponse sont souvent contraints par les campagnes d'évaluation en terme de format des réponses et donc peu adaptés au traitement des questions à réponses multiples ;
- les campagnes d'évaluation ne s'intéressent que très peu à l'évaluation des passages justificatifs des réponses et à la compréhension plus ou moins aisée des réponses par des utilisateurs.

Face à ces constats, nos objectifs sont les suivants :

- développer un système capable de repérer, analyser voire transformer les documents HTML, en particulier les éléments structuraux, pour pouvoir en extraire de l'information plus facilement ;
- développer un système d'extraction de réponses multiples qui soit rapide ;
- proposer un cadre plus large que celui des campagnes d'évaluation pour le format et la présentation des réponses : par exemple, en agrégeant les réponses, en les présentant de façon compréhensible ;
- proposer un cadre d'évaluation différent des campagnes d'évaluation : les questions auxquelles nous nous intéressons plus particulièrement sont : qu'est-ce qu'une réponse correcte pour un utilisateur ? quelle forme les réponses doivent-elles avoir pour satisfaire un utilisateur et faciliter sa compréhension ?.

ORGANISATION DU MANUSCRIT

La suite de ce manuscrit montre comment nous avons tenté d'atteindre ces objectifs. Il est composé de cinq chapitres et d'une conclusion générale.

Chapitre 1 : État de l'art

Nous y présentons un état de l'art des systèmes de question-réponse ayant traité des questions de type liste ainsi que les campagnes d'évaluation pour ces systèmes. Nous nous intéressons également aux systèmes de question-réponse destinés à de vrais utilisateurs, notamment du point de vue de la présentation des réponses.

Chapitre 2 : Observations en corpus

Dans ce chapitre, nous détaillons les observations réalisées durant nos études des corpus de questions de type liste et des documents ayant servi durant les campagnes d'évaluations pour le français. Nous y présentons également la constitution du corpus *Frites* qui se compose de questions attendant des réponses multiples et de documents issus du Web contenant les réponses.

Chapitre 3 : Outils et hypothèses de travail

Le chapitre 3 présente les différents outils que nous avons développés, adaptés ou simplement utilisés au sein de notre système de question-réponse *Citron*. Il s'agit notamment :

- de *Kitten* qui permet l'extraction de texte depuis des documents HTML avec structuration des éléments structuraux (tableau, liste) et que nous avons développé ;
- du moteur de recherche *Lucene* que nous avons paramétré ;
- de l'analyseur syntaxique et l'étiqueteur d'entités nommées *XIP* dont nous avons complété les règles ;
- de Wikipédia que nous utilisons durant la validation de réponses.

Nous y présentons également le cadre de développement de Citron.

Chapitre 4 : Citron, un système d'extraction de réponses multiples sur le Web

Ce chapitre est consacré à la présentation du fonctionnement de Citron qui permet d'extraire des réponses multiples depuis des documents HTML prétraités par Kitten. Nous y détaillons notamment l'exploitation des éléments structuraux (liste, tableau) ainsi que la validation et l'agrégation des réponses multiples.

Chapitre 5 : Évaluations

Enfin, nous présentons dans le chapitre 5 les évaluations de Citron : une première en « conditions idéales » afin de réaliser une étude des performances des différents modules

de Citron, puis une seconde en conditions réelles. Enfin, une première expérience utilisateur a été réalisée afin de comparer les performances de Citron en terme de rapidité et de qualité de réponses à des performances humaines. Une seconde expérience utilisateur permet d'analyser ensuite la satisfaction des utilisateurs quant à la présentation des réponses.

1

ÉTAT DE L'ART

Sommaire

Ce chapitre a pour but de présenter dans un premier temps les différents objets pouvant contenir des réponses à des questions de type liste et de définir notre terminologie. Ainsi, nous verrons du point de vue général comment les listes se présentent et comment elles ont été traitées jusqu'à présent. Dans un second temps, nous verrons comment la notion de questions de type liste a été traitée dans un cadre applicatif en accentuant surtout sur les systèmes de question-réponse, systèmes que nous présenterons succinctement ; le but n'étant pas de présenter ces systèmes de façon exhaustive mais de montrer la variété de ce qui a été demandé dans les campagnes d'évaluations et des réponses apportées. Ayant choisi de travailler sur le Web, nous considérons dans ce chapitre deux types de documents : les documents textes ainsi que les documents HTML interprétés, c'est-à-dire affichés dans un navigateur Internet. Ces documents interprétés peuvent notamment être convertis en texte par des programmes spécifiques comme nous le verrons plus loin mais en attendant, les discussions ici se basent sur le document interprété dans un navigateur et perçu par un être humain ; sont donc laissés de côté pour le moment les images, figures, animations Flash, etc.

1.1 LES ÉLÉMENTS STRUCTURAUX DANS LES DOCUMENTS HTML ET TEXTUELS

Nous regroupons sous le terme d'*éléments structuraux* les objets porteurs d'une structure permettant la présentation de plusieurs informations dans un document. Intuitivement, les éléments structuraux que l'on peut trouver dans des documents textes ou HTML susceptibles de contenir une réponse à une question de type liste sont les énumérations (dont les listes) et les tableaux. Nous posons cependant une restriction à savoir que nous nous sommes intéressés aux énumérations et tableaux dans un cadre discursif « local ». En effet, nous avons laissé de côté les structures plus globales comme les tomes et les chapitres. Nous resterons au niveau de la phrase pour les tables des matières et prendrons en compte les sections d'un texte mais sans réaliser de découpage discursif.

1.1.1 *Définitions et règles typographiques*

Pour mieux détecter les éléments structuraux nous intéressant, il est utile de connaître les règles typographiques qui les régissent. Nous présentons ici les règles typographiques conventionnelles pour chacune des deux structures : les énumérations et les tableaux.

1.1.1.1 *Les énumérations*

L'énumération se définit comme étant l'action d'énumérer, à savoir :
- *Énoncer successivement les parties d'un ensemble, les donner en détail ; dénombrer* (Larousse) ;
- *Énoncer un à un les éléments d'un ensemble* (TLFi [1]) ;
- *Énoncer une à une les parties d'un tout* (Wiktionnaire [2]).

Il existe plusieurs termes pour désigner l'objet support de cette énonciation selon son contexte d'application : catalogue, liste, recensement, etc. Dans les documents HTML, nous rencontrons celui de *liste*. Une liste est définie comme une :
- *suite de mots, de nombres, de noms de personnes, de choses le plus souvent inscrits l'un au-dessous de l'autre. Longue énumération* (Larousse) ;
- *suite continue, hiérarchisée ou non, de noms (de personnes ou d'objets) ou de signes généralement présentés en colonne* (TLFi) ;
- *suite de noms de personnes ou de choses, rangées ou non par ordre alphabétique* (Wiktionnaire).

Les termes *énumération* et *liste* ayant des définitions relativement similaires, nous avons choisi d'utiliser le terme **énumération** afin de désigner l'objet énumérant une séquence d'éléments. Les règles communes aux différents guides de disposition typogra-

1. TLFi : http ://atilf.atilf.fr/
2. Wiktionnaire : http ://fr.wiktionary.org

phique en français se retrouvent dans la définition de l'énumération donnée par la société Synapse Développement[3] (société spécialisée dans le Traitement Automatique des Langues et proposant notamment des programmes de correction orthographique, grammaticale et typographique du français) :

- les énumérations commencent, en principe, par un deux-points ;
- une énumération peut être en ligne ou en colonne ;
- habituellement, chaque élément de l'énumération (signalé ou non par un tiret, introduit ou non par un chiffre ou une lettre), doit être séparé par un point-virgule, le dernier se terminant par un point ;
- il faut noter que, malgré les retours à la ligne, les initiales ne sont pas en majuscules ;
- si un élément de l'énumération se subdivise à son tour, chaque sous-élément se termine par une virgule sauf le dernier qui reprend le point-virgule ;
- si la phrase se poursuit après l'énumération, le dernier élément de celle-ci s'achèvera sur une virgule et non sur un point-virgule.

Avec ces définitions typographiques, nous voyons qu'il faut se préparer à rencontrer des énumérations pouvant être récursives suite à la subdivision d'un élément de l'énumération ; cette utilisation récursive est très fréquente dans les tables des matières. Nous voyons également que les symboles typographiques, et tout particulièrement les puces, seront un indice très précieux pour distinguer la signification d'un retour à la ligne, par exemple pour distinguer un retour à la ligne séparant deux éléments d'une énumération d'un retour à la ligne marquant la séparation entre deux paragraphes. Ces règles ne sont toutefois toutes suivies à la lettre que dans des domaines encadrés comme les publications de texte de lois, de séances à l'Assemblée Nationale, de rendu de verdict juridique, de rapports médicaux ou d'articles encyclopédiques et de journaux (avec des entorses toutefois selon les journaux ou les publications scientifiques). Plus qu'un manque de soin du rédacteur, il s'agit surtout de règles peu connues et, lorsqu'il est question de documents HTML, toutes les combinaisons de ponctuation se retrouvent notamment possibles pour délimiter un élément : absence de point-virgule, virgule, point, émoticône.

Les énumérations ont été l'objet de nombreuses études d'un point de vue linguistique et notamment discursif afin de mieux cerner la structure d'un document. Les travaux de [Péry-Woodley, 2000], [Luc, 2001], [Ho-Dac, 2007], [Bras *et al.*, 2008], [Laignelet, 2009], [Ho-Dac *et al.*, 2010] ont beaucoup traité de cette question et ont ainsi défini la **structure énumérative** (*SE* dorénavant) comme étant composée d'une **amorce** (phrase introductrice) et d'une énumération composée d'**items** (entité co-énumérée caractérisée par diverses marques typographiques, dispositionnelles, lexico-syntaxiques). Leur définition d'une énumération est la suivante : *ensemble d'items pouvant entretenir entre eux des relations diverses*.

3. http://www.synapse-fr.com/manuels/ENUMERA.htm

La SE existe aussi bien au niveau du document (voir figure 1.1 au niveau d'une section et figure 1.2 au niveau du document entier [Ho-Dac *et al.*, 2010]) qu'à un niveau plus « local » comme dans l'exemple suivant [Ho-Dac *et al.*, 2004] où le titre est l'amorce de la SE principale et où les deux items sont eux-mêmes des SE :

[GD] 2.1.2 LES ACTIVITES A COORDONNER : (*amorce SE1*)
- En temps différé : (*item SE1 et amorce SE2*)
  - La constitution d'une base de données sur les déplacements dans l'agglomération toulousaine (*item SE2*)
  - L'évaluation des politiques d'exploitation (*item SE2*)
  - Les études et la recherche opérationnelle (*item SE2*)
- En temps réel : (*item SE1 et amorce SE3*)
  - Les activités de coordination notamment en temps de "crise" (*item SE3*)
  - L'information des usagers (*item SE3*)
  - Le recueil en temps réel des données nécessaires à la coordination et à l'information (*item SE3*)

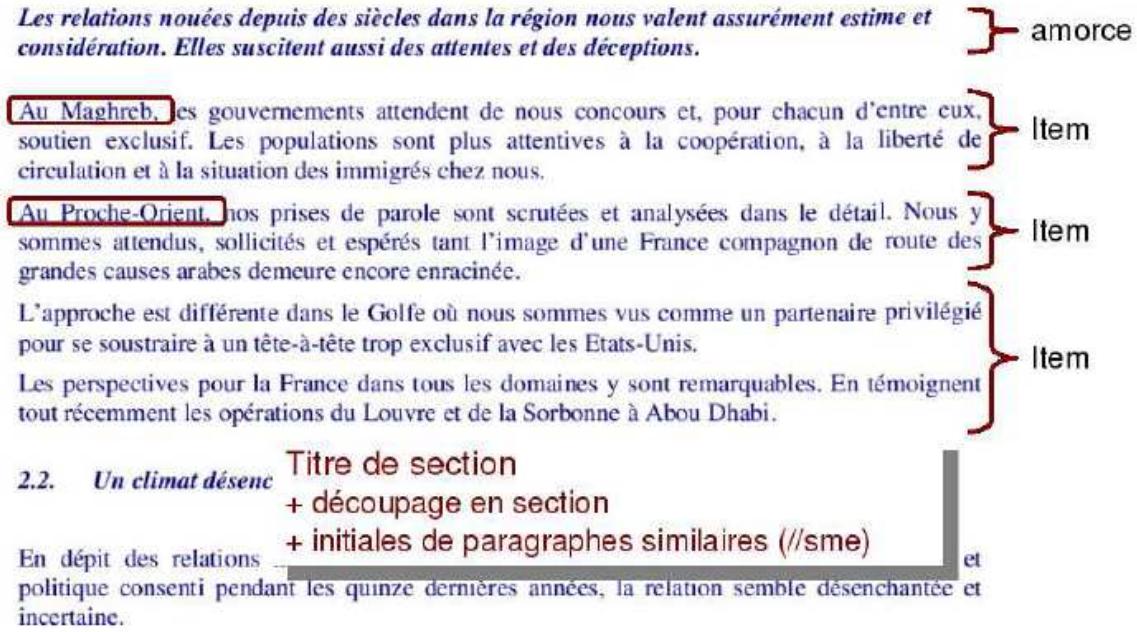

FIGURE 1.1 : Exemple de SE au niveau d'une section.

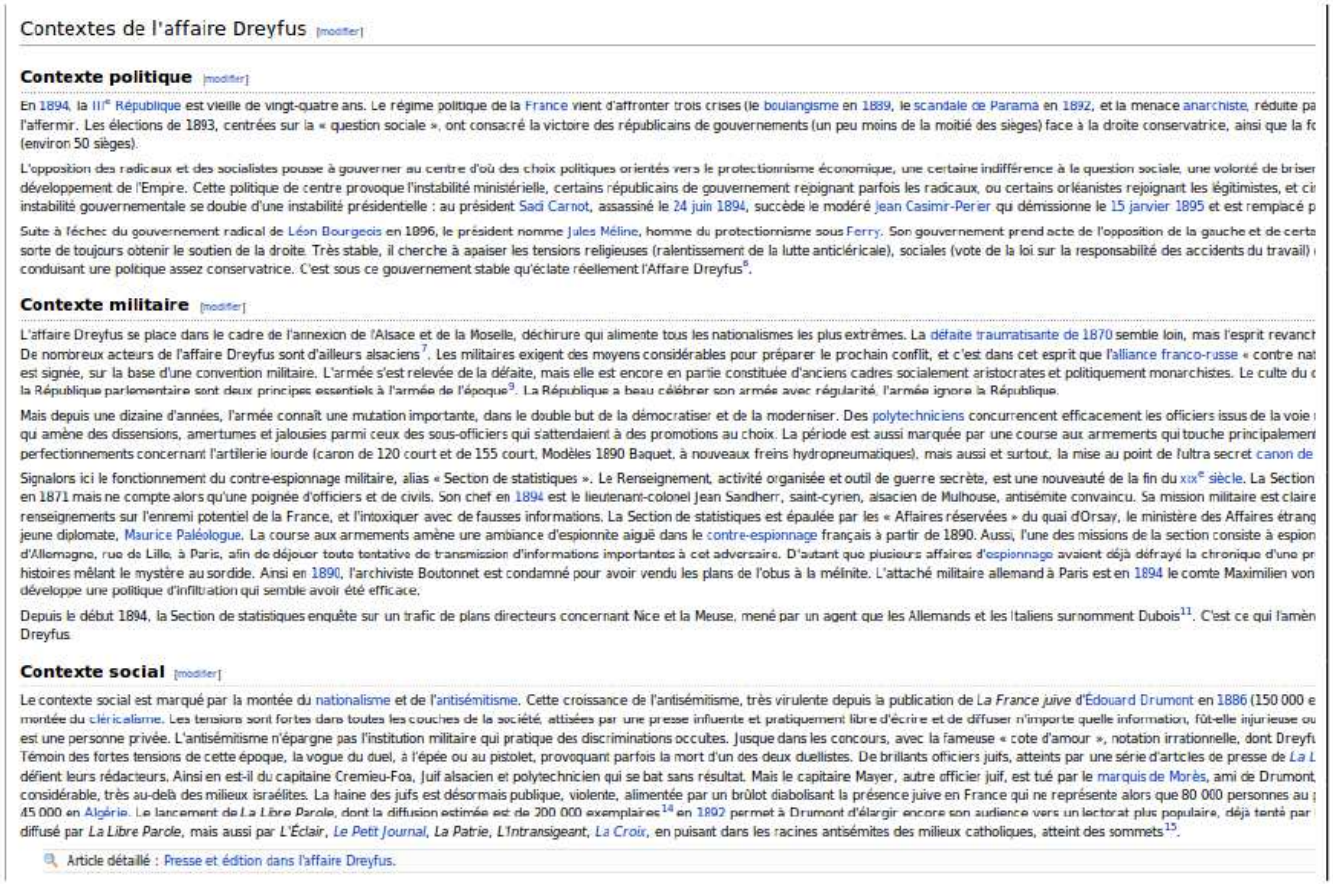

FIGURE 1.2 : Exemple de SE au niveau du document.

[Gala, 2003] utilise également les termes *ouvert* et *fermé* pour désigner la complétude de la séquence d'items : une énumération se terminant par *etc.* ou par des points de suspension sera par exemple considérée comme ouverte.

Nous reprenons donc ce concept de SE et, les dispositions typographiques pouvant varier, nous posons les définitions suivantes (conformes à la terminologie de [Kamel et Rothenburger, 2011]) :

- **énumération verticale** : l'amorce de la SE est délimitée par un " :" et l'amorce ainsi que chacun des items de l'énumération sont séparés par un retour à la ligne (voir exemple précédent) ;
- **énumération horizontale** : deux cas sont possibles :
  - l'amorce de la SE est délimitée par un " :" et les items de l'énumération sont séparés par un symbole de ponctuation comme un point-virgule ou une virgule (par exemple : *On trouve quatre nucléotides différents dans l'ADN : A, G, C, T*) ;
  - la SE ne comporte pas d'amorce délimitée par " :" et ne dépasse pas le cadre de la phrase : nous désignons ce type d'énumération comme des **énumérations intra-phrastiques**. Deux possibilités alors :
    - les items sont énumérés en étant reliés par une marque de coordination ou un adverbe (par exemple, par l'adverbe *respectivement*). Cette marque peut se retrouver entre chacun des items (*La réunion aura lieu soit en mars, soit en avril, soit en mai*) ou seulement entre les deux derniers items (*La réunion aura lieu en mars, en avril ou en mai*) ;
    - les items énumérés ne sont pas reliés entre eux mais un patron lexicographique permet de déduire ce sur quoi porte l'énumération, par exemple une mise entre parenthèses : *Le problème essentiel des corticoïdes réside dans leurs effets secondaires (aspect cushingoïde, myopathie, complications psychiatriques, ostéoporose, syndrome de sevrage).*

Pour notre terminologie, nous avons également choisi de garder le terme **item** utilisé traditionnellement pour désigner les éléments d'une énumération, cet élément pouvant tout aussi bien être une phrase ou un syntagme. Nous utiliserons le terme **puce** pour désigner le symbole précédant l'item d'une énumération verticale. Une puce peut être typographiquement un nombre, un astérisque, un tiret ou même une image, notamment pour les documents HTML.

Nous n'avons cependant pas cherché à appliquer les approches de formalisation discursive pour deux raisons : d'abord parce que ces approches dépendent forcément d'un formalisme discursif (RST, SDRT) mais aussi parce que les outils de formalisation discursive sont généralement peu adaptés aux documents HTML qui sont souvent mal structurés.

1.1.1.2 *Les tableaux*

Un tableau est une structure en deux dimensions dont les dictionnaires donnent les définitions suivantes :

- *composition, encadrée ou non, comportant des chiffres et/ou des textes et divisée en colonnes* (Larousse) ;
- *support plan vertical destiné à recevoir des informations, des renseignements, des inscriptions* (TLFi) ;
- *liste didactique et méthodique rédigée, pour être vue d'un coup d'œil* (Wiktionnaire).

Il s'agit donc d'un objet destiné à faciliter la prise d'information par un être humain et son placement va donc forcément se dissocier visuellement des paragraphes le précédant et le suivant. Là où la liste pouvait être intégrée de façon continue entre deux paragraphes, le tableau marquera clairement une rupture.

Il n'existe pas, à notre connaissance, de règles définissant formellement un tableau en typographie. Nous considérerons ici le côté informatique du tableau au sens d'une grille définie par des colonnes et des lignes. L'interprétation des **cases de données** se fera sur la base de **cases en-têtes** qui seront porteuses de la définition du sens des cases contenues dans cette ligne ou cette colonne.

1.1.2 *Travaux applicatifs*

Les travaux applicatifs sur les énumérations et les tableaux ont également donné lieu à des études de corpus mais un traitement plus automatique a été introduit, avec notamment une évaluation pour certains des programmes présentés.

1.1.2.1 *Les énumérations*

Plusieurs travaux applicatifs ont été consacrés aux SE contenant des énumérations verticales dans le cadre du peuplement d'ontologies [Laignelet *et al.*, 2011] et d'enrichissement de base d'entités nommées [Hearst, 1992] [Jacquemin et Bush, 2000]. En effet, les énumérations verticales présentent plusieurs items ayant un point commun les reliant entre eux. Ce point commun est souvent annoncé dans l'amorce et en fait donc une source riche d'entités en relation hyperonymique. [Laignelet *et al.*, 2011] utilise les énumérations verticales pour construire une ontologie d'objets susceptibles d'intégrer des bases de données géographiques et cartographiques à partir d'un corpus composé de documents très structurés et peu bruités, décrivant ces objets du domaine géographique. Dans un premier temps, l'approche était symbolique pour identifier les relations sémantiques au sein des SE et les résultats étaient encourageants (moins de 20 % d'erreur pour la méronymie et l'holonymie) [Kamel et Rothenburger, 2011], [Laignelet *et al.*, 2011]. Dans un second temps, une approche par apprentissage automatique a été proposée avec deux systèmes de classification [Fauconnier *et al.*, 2013] : traits lexico-syntaxiques et typo-dispositionnels

pour le premier système, trigrammes de tokens étiquetés en POS pour le second. Le but était de classer les relations sémantiques entre l'amorce et l'item à l'intérieur des SE et l'approche par traits a obtenu une exactitude de 61 % et celle par trigrammes de 59, 80 %.

Les structures énumératives ont beaucoup été étudiées du point de vue de leur analyse syntaxique. Elles sont en effet par nature très difficiles à analyser syntaxiquement car il faut pouvoir tenir compte de la verticalité, des marques de ponctuation entre les items, d'une typographie assez libre (par exemple, pour le choix des puces) et créer des liens syntaxiques entre l'amorce, les items et la conclusion. [Aït-Mokhtar *et al.*, 2003] a étudié les énumérations verticales d'un point de vue syntaxique afin de pouvoir correctement identifier ses éléments et de réaliser ensuite leur extraction. Le corpus utilisé pour leur étude était conséquent (81 513 mots pour 410 énumérations verticales) et composé de documents en anglais provenant de manuels de maintenance et descriptions de protocoles. Il s'agissait donc de documents ciblés thématiquement et structurellement, et prétraités sur la base des critères suivants :

- rejet des énumérations suivies d'une conclusion ;
- rejet des énumérations sans amorce ;
- rejet des énumérations imbriquées ;
- les items doivent être de la même catégorie syntaxique ;
- une introduction incomplète (syntagme ou phrase) doit être suivie d'items du type du syntagme manquant à l'introduction.

La typologie ainsi obtenue définit deux types d'énumérations verticales : celles comportant une dépendance syntaxique entre l'amorce et les items, et celles n'en comportant pas. Par exemple :

- cas où chaque item doit être rattaché individuellement à l'amorce pour obtenir des relations de dépendance correctes :

  - This section will help you **to** :
    - **discover** the wide range of work available
    - **find** out the specifics of a particular job
    - **detect** work-related trends

- cas où il n'y a pas de dépendance syntaxique entre l'amorce et les différents items :

  - Introduction :
    - slide the SOP into the telescope body
    - press the SOP up into the rail and install the screws

L'évaluation préliminaire a porté sur l'étiquetage d'un corpus HTML comportant 36 énumérations verticales, toutes étiquetées par une balise HTML explicite. Les 36 énumérations verticales ont été correctement détectées par l'analyseur syntaxique XIP (Xerox Incromental Parser) [Aït-Mokhtar *et al.*, 2002] et sur ces 36 énumérations, 25 ont été correc-

tement segmentées au niveau de l'amorce [Banik *et al.*, 2002].

[Gala, 2003] a également étudié le problème de l'analyse syntaxique posé par les énumérations verticales et intra-phrastiques, l'énumération intra-phrastique étant définie ici comme ayant l'avant-dernier et le dernier items de l'énumération reliés par une conjonction. Un corpus de documents en français issus du Web a d'abord été constitué avec Altavista. Le but était d'intégrer dans l'analyseur syntaxique XIP les phénomènes observés dans le corpus. Ce corpus couvrait plusieurs domaines (juridique, économique, journalistique, technique et scientifique) et contenait 5 205 phrases. 217 énumérations verticales ont été récupérées. Le patron suivant a été défini pour identifier une amorce qui est composée :

- d'un noyau qui contient :
  - un classificateur : il agit en tant qu'hyperonyme de l'ensemble d'items de l'énumération verticale,
  - des modifieurs : ils apportent des précisions, restreignent le champ sémantique du classificateur ;
- d'annexes : elles introduisent ou apportent un complément d'information au noyau ou aux items.

Par exemple [Gala, 2003] :

<annexes>En procédant de la sorte, nous avons pu mettre en évidence</annexes> <classificateur>les ingrédients</classificateur> <modifieurs>nécessaires à la production d'un catalyseur pour la réduction de l'oxygène en PEFCs</modifieurs> :
- un métal de transition,
- une source d'azote,
- une source de carbone,
- un traitement thermique.

Les énumérations intra-phrastiques ont aussi été étudiées mais aucune n'a été rencontrée en début de phrase dans le corpus et seuls deux traits ont été identifiés pour les caractériser :

- présence ou non d'une conjonction de coordination entre les deux derniers items ;
- énumération intra-phrastique fermée ou ouverte.

Des règles ont été ajoutées à XIP puis une évaluation de ce chunking a donné un F-score de 99, 3 % pour la détection des amorces et de 90, 9 % pour les items (les items longs sur plusieurs lignes ont posé problème).

[Bouraoui et Vigouroux, 2003] ont également réalisé une étude sur un corpus issu du Web composé d'articles scientifiques, textes procéduraux, présentations de thèmes de

recherche de laboratoires et de pages d'accueil (en anglais et français). Seules les SE avec une composante visuelle verticale ont été étudiées. Ils ont observé :

- à propos des marqueurs dispositionnels : les balises HTML pour les listes et les tableaux ne sont pas suffisantes pour repérer une SE car une majorité des SE ne sont pas mises en forme de façon standard, que ce soit au niveau des balises utilisées pour les formater ou des normes typographiques ;
- à propos des marqueurs lexico-syntaxiques inspirés de [Jacquemin et Bush, 2000] comme les introducteurs (*voici*, *ci-dessous*), les classifieurs (*étape*), les annonceurs (*liste*) ou les marqueurs de relations (*parties*, *composants*) : il y a peu d'occurrences de ces marqueurs dans l'amorce et les items. Ils émettent donc l'hypothèse que le concepteur de la page Web compte sur la composante visuelle de la SE pour que le lecteur interprète correctement l'intention d'utiliser une SE.

1.1.2.2 *Les tableaux*

Il existe deux utilisations très différentes de la structure tableau en HTML : l'une, même si fortement déconseillée par le *W3C*, pour le formatage du rendu général d'une partie de la page (ou de la page entière) à l'aide de tableaux imbriqués et l'autre pour présenter des données de façon structurée. Nous appellerons *tableau de formatage* les premiers et *tableau de données* les seconds. Une étude à très grande échelle (14 milliards de tableaux HTML) a été réalisée par [Cafarella *et al.*, 2008b] qui estime à seulement 1,1 % le nombre de tableaux de données ; la détection et la distinction entre ces deux types de tableau sont, par conséquent, fondamentales. Comme le montre la figure 1.3 tirée de [Wang et Hu, 2002], un tableau de formatage utilise la structure d'un tableau pour disposer des éléments sur la page. En revanche, un tableau de données utilise la structure sémantique d'un tableau reliant des cases de données à des cases en-têtes pour présenter une donnée et apporter en même temps un sens à cette donnée. Bien qu'extrêmement rare, il est toutefois possible que des tableaux de données ne comportent pas de cases en-têtes comme le montre le tableau de données *MARKET WATCH* à droite sur la figure 1.3 : les cases en-têtes sont alors inférées par l'être humain lisant le tableau. Ce cas très particulier est naturellement extrêmement difficile à traiter, ne serait-ce même que pour les êtres humains s'ils ne disposent pas de connaissances nécessaires sur la thématique abordée par le tableau.

Le traitement automatique des tableaux HTML a fait l'objet de travaux avec des objectifs différents mais avec quasiment toujours une même visée : réussir à typer les cases d'un tableau, et notamment le type des cases en-têtes. Parfois un objectif supplémentaire existe en amont de ce typage de cases : repérer s'il s'agit bien d'un tableau de données. Deux objectifs prédominent : optimiser la visualisation ergonomique [Tajima et Ohnishi, 2008] et l'extraction d'information [Gatterbauer *et al.*, 2007] [Li, 2011]. Il est toutefois à noter que la plupart des travaux en extraction d'information visent à vérifier/découvrir

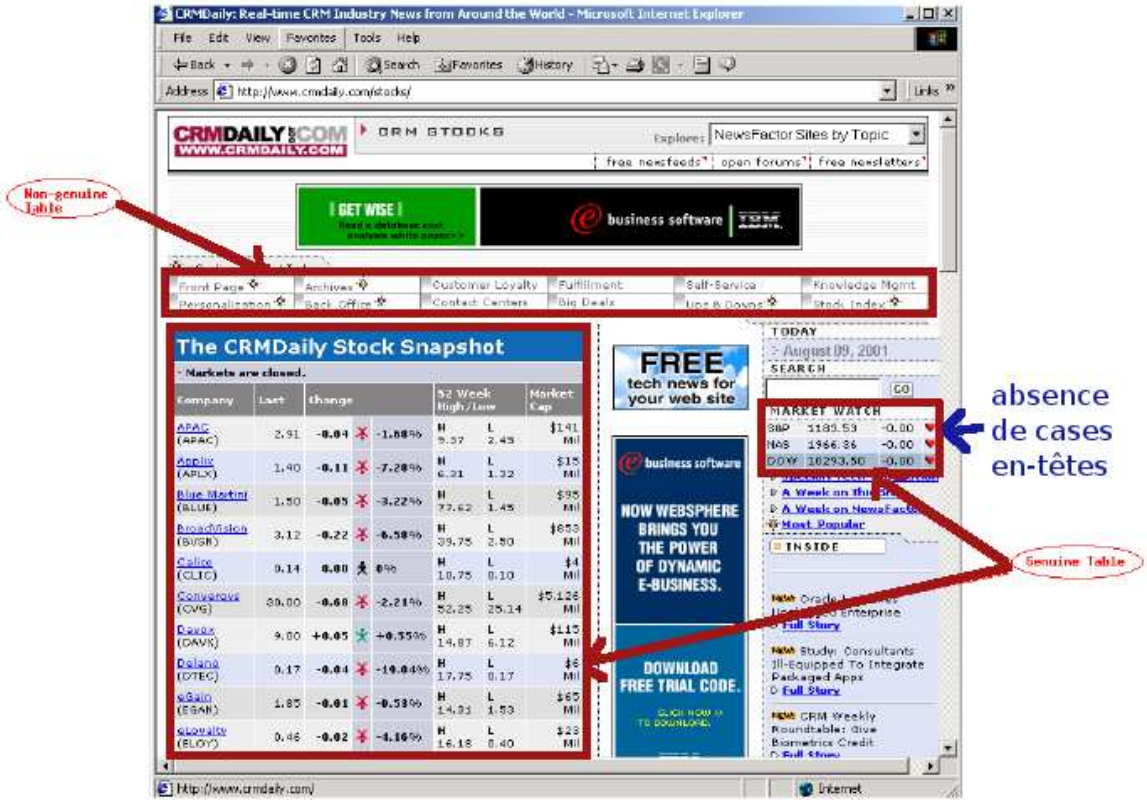

FIGURE 1.3 : Tableau de données (*genuine table*) et tableau de formatage (*non-genuine table*).

des relations entre entités (notamment dans le cadre de la tâche TREC-Entity [Balog *et al.*, 2009] [Balog *et al.*, 2010] [Balog *et al.*, 2011]) et la recherche de paires d'entités sous la forme *concept-instance* [Cafarella *et al.*, 2008a] [Kopliku *et al.*, 2011] [Abbes *et al.*, 2013].

Pour l'extraction d'information dans des tableaux HTML, une large palette d'approches a été utilisée avec notamment l'utilisation de règles [Gatterbauer *et al.*, 2007] [Tajima et Ohnishi, 2008], l'utilisation d'un modèle probabiliste [Li, 2011], la manipulation d'arbre à partir du code HTML [Zhai et Liu, 2005] et l'apprentissage automatique sur un corpus annoté manuellement (arbre de décision pour [Wang et Hu, 2002], CRF (Conditional Random Fields) pour [Pinto *et al.*, 2003]). Toutes ces approches produisent de très bons résultats dans leur tâche mais plusieurs nécessitent un traitement en amont afin de constituer notamment cette base de paires d'entités concept-instance.

1.1.3 *Synthèse*

Des études de corpus ont permis de définir ce qu'est une structure énumérative. Cependant, peu d'implémentations ont été réalisées pour les extraire automatiquement et

elles imposent souvent de nombreuses restrictions (thème, structure des documents) que nous ne pourrons pas utiliser vu notre approche en domaine ouvert sur le Web. Nous devrons donc en mettre en place. Quant aux travaux sur les tableaux HTML, ils montrent de bons résultats mais au prix d'efforts computationnels assez importants pour certains. Nous doutons de la possibilité de pouvoir utiliser ces derniers du point de vue du temps de traitement dans un cadre applicatif question-réponse et nous avons donc seulement gardé les approches existantes présentant une possibilité de prétraitement rapide. Nous détaillerons ce point au chapitre 4.

1.2 LES QUESTIONS-LISTES

Nous utiliserons à partir de maintenant le terme *question-liste* pour désigner les questions de type liste. Dans les campagnes d'évaluation en question-réponse, il s'agit de questions portant toujours un indicateur de leur nature, cet indicateur étant exclusivement le pluriel. Les questions-listes attendent plusieurs réponses, le nombre de réponses attendues n'étant pas forcément déterminé. Ainsi, les questions suivantes sont des questions-listes :

Question-liste : *Quels sont les 7 astres du système solaire visibles à l'oeil nu ?*
Question-liste : *Quels pays étaient candidats à l'organisation de la coupe du monde 2006 ?*

Nous allons voir comment ces questions ont été traitées en question-réponse et en recherche d'information.

1.2.1 *Définition en question-réponse*

Le domaine question-réponse est naturellement celui qui nous intéresse le plus. Bien que tous applicatifs au sens propre du terme, les systèmes de question-réponse se divisent toutefois en deux catégories : les systèmes destinés à un cadre évaluatif et ceux destinés à un cadre utilisateur, les seconds pouvant également tout à fait participer à des cadres évaluatifs sous réserve de modifications que nous allons voir. Après un rappel succinct de ce que sont les systèmes de question-réponse, nous présenterons comment ils sont évalués durant les campagnes d'évaluation, puis les méthodes de fonctionnement de ces systèmes participant à ces campagnes et enfin nous parlerons des applications utilisateurs.

1.2.1.1 *Présentation des systèmes de question-réponse*

Les **SQR** (systèmes de question-réponse) ont pour but de fournir une réponse précise à une question formulée en langue naturelle par un utilisateur. Les réponses peuvent être recherchées dans des bases de données et/ou des collections de documents. Un SQR peut simplement consister en une traduction de la question formulée en langue naturelle en

une requête d'interrogation de base de données, le procédé étant similaire pour interroger une ontologie. Nous nous intéressons ici uniquement à ceux interrogeant un corpus de documents, ce qui n'exclut pas pour autant l'utilisation d'une base de données ou d'une ontologie durant le processus, notamment à des fins de validation de réponses.

Les SQR cherchant une réponse dans une collection de documents combinent plusieurs domaines dont notamment la recherche d'information et le traitement automatique des langues à travers l'extraction d'information. En effet, là où des moteurs de recherche renvoient des références de documents (avec éventuellement un extrait de ces documents) suite à une requête saisie sous forme de mots-clés, les SQR travaillent à partir d'une question en langue naturelle dont tous les mots ne sont pas forcément pertinents pour la recherche d'information. Après une analyse de la question propre à chaque système, ils sélectionnent un ensemble de documents de la collection puis extraient des candidats pour la réponse recherchée depuis ces documents. Les SQR extraient plusieurs candidats à la réponse, que nous désignerons désormais sous le terme de **candidats-réponses**, et les ordonnent selon des critères qui leur sont propres. Enfin, une étape de validation de réponse peut également intervenir, aussi bien durant l'ordonnancement qu'après. L'ultime étape est bien évidemment de présenter la ou les réponses à l'utilisateur.

Pour résumer, on distingue donc plusieurs parties communes à la quasi-totalité des SQR :

- l'analyse de la question afin de :
  - transformer la question en langue naturelle en une requête formalisée,
  - déterminer le type de la question et le type de la réponse attendue ;
- la recherche de documents à l'aide de cette requête formalisée ;
- l'extraction des candidats-réponses ;
- l'ordonnancement, l'agrégation et éventuellement la validation des candidats-réponses ;
- la présentation de la ou des réponses.

Les SQR existant utilisent des approches très variées qui peuvent s'appliquer sur la totalité du système ou seulement certaines parties, comme une utilisation de méthode d'apprentissage automatique uniquement pour l'extraction des candidats-réponses. Par exemple, certains SQR utilisent une représentation logique de la question et des documents [Dan I. Moldovan et Bowden, 2007] ou discursive [Bos *et al.*, 2007], ce qui les fait procéder à une étape de formalisation des documents mais pas forcément de toute la collection : après une requête par un moteur de recherche, seuls les documents les plus pertinents sont alors formalisés. Il est à noter toutefois que ce genre d'approche amène irrémédiablement un temps de traitement plus long.

L'analyse syntaxique est également très utilisée par les SQR, notamment pour l'analyse de la question et l'extraction de la réponse. Par exemple, [Katz et Lin, 2003] et [Moriceau et Tannier, 2010] utilisent les dépendances syntaxiques produites par un analyseur pour rechercher dans les documents des phrases syntaxiquement proches de la question

et procéder à une fusion d'informations provenant de plusieurs documents. Parmi les approches du même genre, on trouve aussi celles de [Cui *et al.*, 2005], [Shen et Klakow, 2006] ou [Bouma *et al.*, 2005]. Les réponses peuvent également être extraites à l'aide de méthodes par apprentissage automatique : CRF sur la distance d'arbres syntaxiques [Yao *et al.*, 2013], Perceptron et SVM sur des traits syntaxiques et sémantiques pour ordonnancer les réponses [Surdeanu *et al.*, 2011].

L'étape d'ordonnancement et de validation des candidats-réponses utilisent très fréquemment des heuristiques de distance entre les mots-clés issus de la question et les candidats-réponses [Fangtao *et al.*, 2008] ; un apprentissage automatique basé sur des critères comme la distance entre les mots et le type de la réponse peut également être utilisé [Grappy, 2011].

Enfin, plusieurs systèmes utilisent également une étape d'agrégation des réponses, ne serait-ce que sur la forme de surface, afin d'éviter les doublons [Harabagiu *et al.*, 2001], [Yang *et al.*, 2003], [Chen *et al.*, 2004] , [Fan *et al.*, 2005], [Katz *et al.*, 2006], [Schlaefer *et al.*, 2007].

1.2.1.2 *Les campagnes d'évaluation en question-réponse*

PRÉSENTATION GÉNÉRALE. L'évaluation des SQR peut se faire par la mesure de la satisfaction des utilisateurs (point de vue applicatif et qualitatif) ou par l'intermédiaire d'une mesure telle que celles utilisées dans les campagnes d'évaluation (point de vue quantitatif). Les campagnes d'évaluation des SQR ont pour but de jauger les performances des systèmes et proposent pour cela un nombre de questions significatif pour les catégories les plus fréquentes [Voorhees, 2001] : *factuelles* (*Quel est le numéro de département du Doubs ?*), *booléennes* (*Le roquefort est-il au lait cru ?*), *définition* (*Qu'est-ce qu'un coupe-choux ?*), *complexes* (question principalement en Comment ? Pourquoi ? : *Comment construire une serre de jardin ?*, *Pourquoi les bombes de mousse à raser polluent ?*), *liste* (*Quels sont les pays frontaliers de la Belgique ?*) et *nil* (pas réponse dans la collection : *Qui est la Reine de France ?*). La figure 1.4 tirée de [Bernard, 2011] présente une synthèse des différentes campagnes d'évaluation des SQR avec le type des questions proposées ainsi que les thématiques des documents de la collection imposée. Les principales campagnes sont les suivantes (du fait de la variation des tâches proposées lors des éditions d'une campagne, les indications sur le nombre d'éditions, de langues proposées et de participants ne sont qu'une tendance) :

- EQueR : campagne unique sur le français avec 7 participants [Ayache *et al.*, 2006] ;
- Quaero : campagne sur le français et l'anglais avec 4 éditions et 5 participants [Quintard, 2010], [Quintard *et al.*, 2010] ;
- NTCIR : campagne régulière sur le japonais, le chinois et l'anglais avec 5 éditions et une quinzaine de participants [Fukumoto *et al.*, 2003], [Fukumoto *et al.*, 2004], [Kato *et al.*, 2004], [Kato *et al.*, 2005], [Sasaki *et al.*, 2005], [Sasaki *et al.*, 2007], [Fukumoto *et al.*, 2007], [Mitamura *et al.*, 2008] ;

– QAst : campagne régulière sur le français, l'anglais et l'espagnol avec 3 éditions utilisant un corpus de transcriptions de paroles, 4 participants [Turmo *et al.*, 2007], [Turmo *et al.*, 2008], [Turmo *et al.*, 2010] ;
– CLEF : campagne régulière sur plusieurs langues variant selon les éditions (7) : l'anglais, le français, l'allemand , l'espagnol... [Magnini *et al.*, 2004], [Magnini *et al.*, 2005], [Vallin *et al.*, 2006], [Magnini *et al.*, 2007], [Giampiccolo *et al.*, 2008], [Forner *et al.*, 2009], [Peñas *et al.*, 2010] ;
– TREC : campagne régulière sur l'anglais avec 11 éditions et une trentaine de participants [Voorhees, 1999] [Voorhees, 2000] , [Voorhees, 2001], [Voorhees, 2002], [Voorhees, 2003], [Voorhees, 2004], [Voorhees et Dang, 2005], [Dang *et al.*, 2006], [Dang *et al.*, 2007].

	TREC								QA@Clef Main Track							QAst			NTCIR					EQueR	Quaero		
	1999	2000	2001	2002	2003	2004	2005	2006	2003	2004	2005	2006	2007	2008	2009	2007	2008	2009	2004	2005	2007	2008	2009	2010	2008	2009	2010
Factuelles	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Définitions simpl.		●	●							●	●	●	●	●							●			●			
Définitions					●												●	●				●	●		●	●	●
Pourquoi																						●			●	●	●
Comment											●										●				●	●	●
Oui/non																									●	●	●
Listes ouvertes					●	●	●	●	●			●	●	●					●		●	●	●				
Listes fermées			◇	◇								●	●	●					●		●	●		●	●	●	●
Journaux	●	●	●	●	●	●	●	●	●	●	●	●	●	●					●	●	●	●	●	●			
Parole																●	●	●									
Politique																								●			
Médical																								◇			
Juridique														●													
Wikipédia												●	●														
Blogs								●																			
Web en général																									●	●	●

◇ : tâche à part

FIGURE 1.4 : Synthèse des types de question et de documents des campagnes d'évaluation de SQR.

Pour résumer et continuer l'étude de [Gillard *et al.*, 2006] au-delà de 2006, nous pouvons voir les tendances suivantes émerger (en ne considéant que TREC, CLEF, NTCIR, EQueR et Quaero). tout d'abord concernant les questions factuelles :

- **taille de la réponse :** les campagnes d'après 2001 se concentrent sur la réponse exacte après des premières années où la réponse était limitée en taille. Pour ResbpliQA [Peñas *et al.*, 2010] (CLEF) par exemple, un élargissement de la réponse à la balise paragraphe (dans des documents balisés) a été autorisé suite à l'intégration de questions complexes (raison, but, procédure) ;
- **catégorie syntaxique de la réponse :** la réponse exacte était limitée à un groupe nominal pour les premières campagnes NTCIR puis à des types d'entités nommées (date, expression numérique). Suite à l'introduction de questions non factuelles (biographie, complexe), cette limitation a été levée. ;
- **génération de la réponse :** la réponse exacte devait être absolument extraite depuis un document pour la quasi-totalité des campagnes : seules quelques éditions ont autorisé une génération de la réponse. La génération pour NTCIR 2002 était totale puisqu'elle autorisait les systèmes à utiliser des ressources extérieures à la collection et à proposer des réponses ne se situant pas dans les documents, tant que ce dernier justifiait leur réponse (cas de paraphrase). La génération n'était que partielle pour CLEF 2007 et 2008 puisqu'elle portait sur la correction orthographique et grammaticale de réponse extraite, ces campagnes portant notamment sur des langues à flexion nominale (allemand, néerlandais).
- **nombre de réponses par question :** il était possible de répondre plusieurs réponses pour les campagnes TREC de 1999 à 2001, les campagnes NTCIR et CLEF 2006. Mais les campagnes se sont rapidement concentrées sur la tâche de ne fournir qu'une réponse par question ;
- **justification de la réponse :** à part pour Quaero et CLEF (2006 et 2008), la justification d'une réponse était un document complet ; chaque réponse devait être fournie avec l'identifiant du document dont elle provenait, ce qui est alors très proche de la recherche d'information au sens où l'assesseur est obligé de parcourir tout le document pour juger la validité de la réponse. Quaero imposait un passage contenant et justifiant la réponse de 250 caractères pour les questions factuelles, tandis que CLEF 2006 limitait à 500 caractères un passage unique et CLEF 2008 à 750 caractères au total pour une compilation possible de passages d'un même document ;
- **statut de la réponse :** pour toutes les campagnes, aux deux statuts *correct* et *incorrect* des premières éditions se sont toujours rajoutés le statut *inexact* (la réponse manque d'information ou au contraire possède des informations inutiles) et *non supporté* (la réponse est correcte mais le document dont elle est tirée n'en apporte pas la preuve). TREC 2006 et 2007 ont ajouté les statuts *correct localement* et *correct globalement* où seulement la réponse provenant du document le plus récent était considérée comme correcte globalement. Du fait de l'obligation de fournir un pas-

sage justificatif, Quaero a ajouté les statuts *supporté* (la réponse n'est pas correcte mais le passage est pertinent) et fait une distinction entre *correct complet* (le passage justifie la réponse) et *correct* (le document contenant le passage justifie la réponse).

Concernant les réponses aux questions-listes :
- **taille de la réponse :** une réponse à une question-liste n'est quasiment jamais limitée en taille. Seul Quaero a imposé que cette séquence ne dépasse pas 250 caractères ;
- **liste ouverte ou fermée :** le nombre de réponse attendues peut être spécifié ou non dans la question. De plus, si ce nombre est spécifié, cela ne correspond pas forcément au nombre de réponses correctes dans la collection. Certaines campagnes limitaient le nombre maximum de réponses à fournir comme EQueR avec 20 réponses (même s'il pouvait en exister plus) ;
- **nombre de réponses par question :** seule la campagne Quaero a permis de répondre plusieurs réponses (trois), au sens où une réponse à une question-liste se compose de plusieurs éléments ;
- **justification de la réponse :** seule la campagne Quaero a limité la taille du passage justificatif mais à une très grande valeur, 8000 caractères ;
- **statut de la réponse :** chaque élément de réponse composant la réponse à la question-liste possède un statut ;
- **restriction à l'échelle du document :** même contrainte que pour les questions factuelles, toutes les réponses doivent provenir d'un même document.

Lorsque les SQR peuvent fournir plusieurs réponses pour une chaque question factuelle non-liste (généralement de trois à cinq), ils sont alors le plus souvent évalués grâce à la mesure du *MRR* (Mean Reciprocal Rank) qui favorise ainsi les SQR fournissant une réponse correcte dans les premiers rangs. En effet, le MRR se calcule en additionnant l'inverse du rang de la réponse la mieux classée : une réponse correcte au premier rang permet de gagner 1 point, une au deuxième rang permet de gagner $\frac{1}{2}$ point, etc. Le total est ensuite divisé par le nombre de questions de façon à obtenir le MRR moyen du SQR.

Il est très difficile de garantir qu'une seule et unique réponse correcte puisse être obtenue à partir de la collection de documents disponible pour l'évaluation, ce qui serait peu intéressant d'ailleurs. Une évaluation humaine des réponses proposées par les SQR doit souvent avoir lieu pour juger la réponse ainsi que le passage justificatif l'accompagnant (si exigé par la campagne).

LES QUESTIONS-LISTES. Elles ont été abordées dans plusieurs campagnes (figure 1.4). Pour indiquer que la question était de type liste et n'attendait donc pas une réponse unique, une marque de pluriel était toujours présente ; le nombre de réponses attendues

n'était toutefois pas toujours mentionné dans la question. Si le nombre est imposé, la question-liste est considérée comme fermée, ouverte sinon. Il faut noter qu'une question-liste fermée restreint le nombre de réponses mais cette borne n'est pas forcément celle de toutes les réponses possibles : c'est le cas par exemple de la question *Citer 3 aéroports français*. Le nombre de réponses attendues était mentionné pour les campagnes EQueR [Ayache, 2005], Quaero 2008, 2009 [Quintard *et al.*, 2010], TREC 2001, 2002 comme dans l'exemple : *Quelles sont les 4 localisations possibles des neuroblastomes ?* (EQueR). Il ne l'était pas pour les campagnes Quaero 2010, TREC 2003 à 2007 comme dans l'exemple *Quels sont les secteurs qui recrutent ?* (Quaero 2010). Toutes les éditions et guidelines des campagnes CLEF [4] et TREC [5] sont en ligne.

Deux mesures sont principalement utilisées pour évaluer les réponses aux questions-listes :

- la précision moyenne : nombre de réponses correctes divisé par le nombre de réponses attendues (cas des questions-listes fermées) ;
- la F-mesure : $F = \frac{2.\text{Précision.Rappel}}{\text{Précision+Rappel}}$

  La précision est le nombre de réponses correctes extraites divisé par le nombre de réponses fournies et le rappel est le nombre de réponses correctes extraites divisé par le nombre de réponses correctes existant dans les documents en considérant l'ensemble des réponses jugées correctes par les assesseurs (cas des questions-listes ouvertes).

Pour conclure sur le format des campagnes d'évaluation, notons que les assesseurs jugent de la validité d'une réponse sur la base de 1 à 5 triplets question/réponse/document (ou passage), ce qui oblige donc les SQR à faire un choix d'au plus N réponses par question s'ils en trouvent plusieurs qu'ils jugent correctes. Une réponse issue d'un recoupement d'informations entre plusieurs documents est aussi difficile à justifier dans le cadre d'une campagne d'évaluation, voire impossible si un seul triple question/réponse/document est permis. On constate en effet qu'un recoupement de deux extraits de documents permettant à eux deux de justifier la réponse se retrouvera pénalisé au niveau de l'évaluation par un assesseur du fait même de cette limitation à un seul extrait. Par exemple [Moriceau et Tannier, 2010] :

4. http://nlp.uned.es/clef-qa/
5. http://trec.nist.gov/data/qamain.html

Question : *Quel premier ministre français s'est suicidé ?*
Réponse : Pierre Bérégovoy.
Extrait du document A : Deux ans plus tard, Pierre Bérégovoy s'est suicidé après avoir été impliqué...
Extrait du document B : Le premier ministre français Pierre Bérégovoy a mis en garde Bill Clinton contre...

Dans le cadre d'évaluation classique, la réponse « Pierre Bérégovoy » sera très certainement considérée comme UNSUPPORTED si justifiée seulement par l'extrait du document A (on ne sait pas qu'il est premier ministre français) ou seulement l'extrait du document B (on ne sait pas qu'il s'est suicidé). Il faudrait pouvoir fournir plusieurs extraits afin d'expliquer le recoupement multi-document.

Toujours sur la question de ce passage justificatif exigé dans certaines campagnes, ce dernier est limité en taille (nombre de caractères) et nous verrons que cela peut poser problème si les informations justifiant la réponse sont réparties dans des phrases séparées par un grand nombre de caractères. De plus, la réponse et le passage doivent obligatoirement être du texte issu d'un document de la collection alors qu'il peut être parfois plus pertinent de renvoyer un élément structural (un tableau par exemple). En effet, ces éléments structuraux sont très présents dans les documents Web or, de toutes les campagnes évoquées jusqu'à présent, seule Quaero utilise une collection de documents Web tout en imposant toutefois un format de réponse identique à celui des autres campagnes [Quintard *et al.*, 2010].

1.2.1.3 *Traitement des questions-listes par les systèmes de question-réponse*

Les SQR ayant participé à des campagnes comportant des questions-listes ont deux types d'approches : soit adapter leur traitement de questions factuelles aux questions-listes, soit développer un traitement spécifique.

L'adaptation à des questions-listes peut 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 au sein du SQR lors de la phase d'ordonnancement [Kaisser et Becker, 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 la question *the Gaza Strip* : *What were the settlements that were evacuated ?* est reformulée 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 dans lesquels a été développé un traitement spécifique pour les questions-listes ont notamment utilisé la détection de doublons pour éviter la redondance de candidats-

réponses, par exemple à l'aide d'une mesure de recouvrement de candidats-réponses [Monz Christof, 2001], de distance d'édition (Levenshtein sur les candidats-réponses), 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 et Bowden, 2007]. À travers l'expansion de requête, la redondance des candidats-réponses (au niveau de la phrase ou du document) est également très fréquemment utilisée comme critère de validation [Razmara et Kosseim, 2008] [Razmara, 2008] [Wang *et al.*, 2008] [Figueroa et Neumann, 2008].

L'agrégation de réponses multiples a également été abordée par des SQR traitant de questions factuelles. Par exemple, [Webber *et al.*, 2002] ont défini quatre catégories de relation entre des réponses correctes : équivalence, inclusion, agrégation, alternative. [Dalmas et Webber, 2005] effectuent une agrégation de toutes les réponses trouvées (correctes et incorrectes) à travers un graphe pour les questions de localisation. [Moriceau, 2007] ajoute une cinquième catégorie de relation entre les réponses, la complémentarité, et effectue une agrégation des réponses pour les questions numériques et temporelles.

La plupart de ces SQR utilisent des ressources extérieures comme des bases de données ou le Web afin de trouver une réponse mais surtout à des fins de validation de la réponse. De plus, l'aspect multi-document des réponses n'est vu généralement qu'en phase de validation par la redondance [Harabagiu *et al.*, 2005].

1.2.2 *Définition en recherche d'information*

Les SQR cherchent à extraire une réponse depuis des documents alors que les moteurs de recherche cherchent à renvoyer des références de documents en rapport avec une requête. Ces références de documents sont notamment couramment présentées avec un *snippet* : il s'agit d'un extrait du document, l'extrait pouvant être continu ou composé de plusieurs extraits discontinus, voire incomplets. Les figures 1.5 et 1.9 présentent les références des documents sous la forme d'un lien vers ce document (sous la forme de son titre) accompagné d'un court snippet où les mots-clés de la requête sont en gras. On observe donc une différence importante entre les SQR et les moteurs de recherche puisque dans le premier cas, une seule réponse est proposée alors que dans le second, des centaines de références de documents peuvent être pertinents par rapport à la requête. Malgré cette différence, on retrouve des similarités dans les deux domaines, et notamment pour le traitement de la métonymie.

Ainsi, le moteur de recherche Exalead propose par exemple dans une colonne de droite (figure 1.5) les termes associés les plus fréquents dans les documents retournés par la requête fournie. Ainsi, en les sélectionnant, il est possible de désambiguïser la requête puisqu'elle sera reformulée en intégrant les termes associés sélectionnés, on parle alors d'expansion de requête. Par exemple, une requête sur le terme « taupe » renverra

des documents sur plusieurs thématiques (correspondant à plusieurs réponses possibles) comme l'animal, le film de ce nom, une franchise, l'espionnage, etc. Une expansion de requête avec les termes associés adéquats permettra de restreindre à une thématique particulière comme par exemple au personnage de fiction George Smiley du roman « La taupe » (figure 1.5)

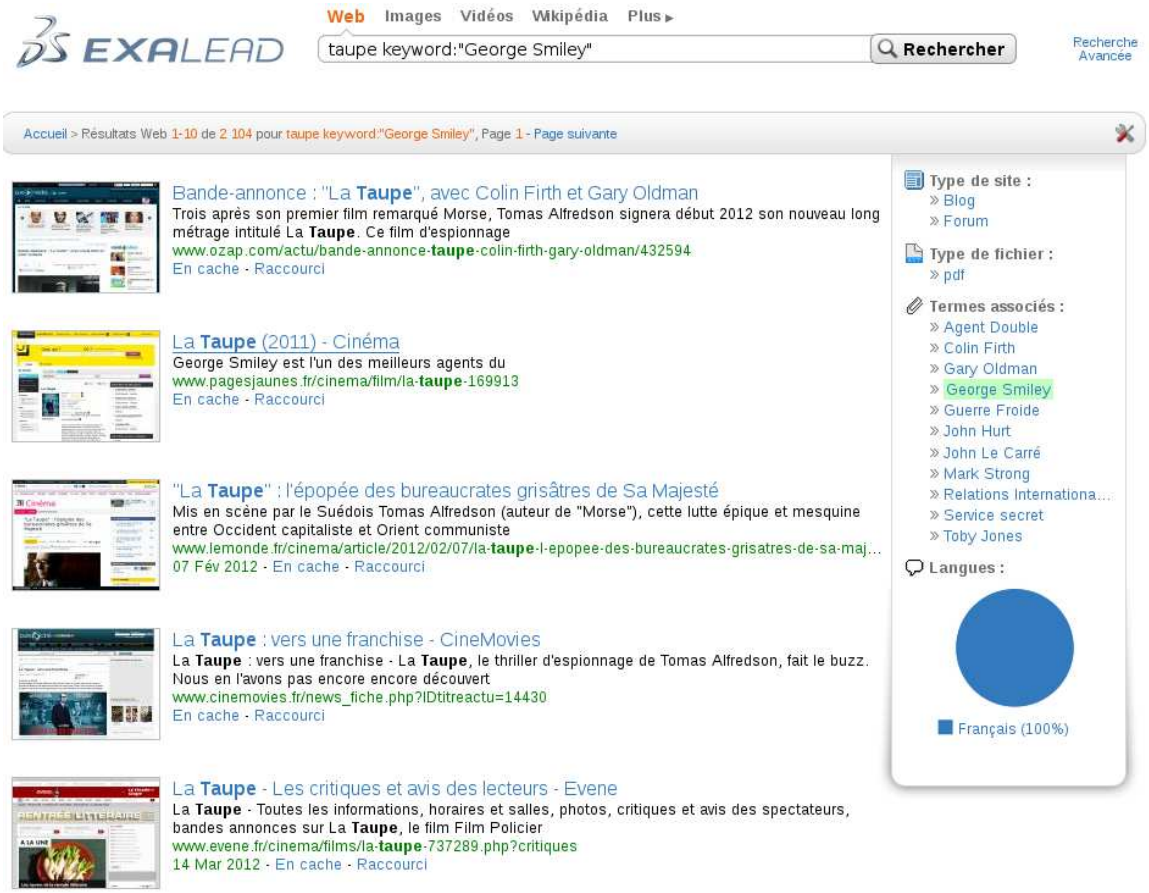

FIGURE 1.5 : Exemple d'expansion de requête avec le moteur de recherche Exalead.

La recherche d'information permet donc à travers l'expansion de requête plusieurs tours du point de vue Dialogue Homme-Machine. Cette possibilité permet également de dialoguer avec l'utilisateur pour lui proposer des corrections d'orthographe ainsi que d'autres requêtes (saisies par d'autres utilisateurs auparavant) qu'il pense en rapport avec sa requête. Cette proposition peut même commencer dès la formulation de la requête puisque des moteurs de recherche l'effectue au fur et à mesure que la requête est saisie au clavier.

Par exemple, à la saisie de la requête sous forme de question *Quel est le président de*, le moteur de recherche Google ouvre une liste de choix :

- la Chine ?
- l'Allemagne ?
- l'Assemblée Nationale ?

1.2.3 *Applications utilisateur*

Les SQR dédiés à un cadre utilisateur apportent de nouvelles dimensions à la tâche en commençant par l'interaction avec l'utilisateur : il devient possible pour l'application de demander à l'utilisateur de préciser sa demande (désambiguïsation de termes ou sélection thématique des documents par exemple).

De plus, puisqu'il ne s'agit pas ici pour les SQR de produire un fichier au format défini par la campagne contenant les réponses et compte-tenu du fait que plusieurs SQR fonctionnent sur le Web, de nouvelles formes de présentation des réponses et de leurs justifications sont apparues. Par exemple, le SQR WolframQA [6] utilise également les images, les tableaux et les chronologies pour présenter plusieurs réponses à l'utilisateur. On retrouve les tableaux dans *Google Squared* [Crow, 2010] et des chronologies dans *Google News* et ChronoZoom [7] ainsi que dans les travaux de [Llorens *et al.*, 2011] qui s'intéressent à l'annotation temporelle de textes à des fins de visualisation ergonomique pour l'utilisateur (figure 1.6). [Teissèdre, 2012] effectue une recherche d'information sur critère temporel (par exemple pour la requête : *le vote des femmes depuis 1900*) et présente également ses résultats sous forme de chronologie (figure 1.7).

Le SQR Wolfram Alpha [6] interroge une base de données pour proposer un tableau de toutes les informations à propos du film Inception lorsqu'on lui pose la question *Who directed Inception ?*, ou la liste des réalisateurs des six films sur Batman pour *Who directed Batman ?* (pour ce dernier exemple, le SQR explique à l'utilisateur son interprétation de la question en affichant *Batman movies*, *director*). L'utilisation d'une base de données est un handicap, en contre-partie pour intégrer des données d'autres langues ou chercher des informations qu'il ne possède pas : la question *Who won the French presidential election ?* reste ainsi sans réponse, Wolfram Alpha ne disposant apparemment dans sa base de données que d'informations à propos des élections américaines (figure 1.8).

Bien que ne détaillant pas son fonctionnement, il est très probable que l'application *akinator* [8] repose également sur une base de données. Dans cette application de Dialogue Homme-Machine, l'utilisateur choisit une personne, un objet, un animal ou un concept et le système doit le deviner par une succession de questions booléennes auxquelles l'utilisateur répond. L'inverse est également possible : l'utilisateur doit alors poser des questions

6. http://www.wolframalpha.com
7. http://research.microsoft.com/en-us/projects/chronozoom/
8. http://fr.akinator.com

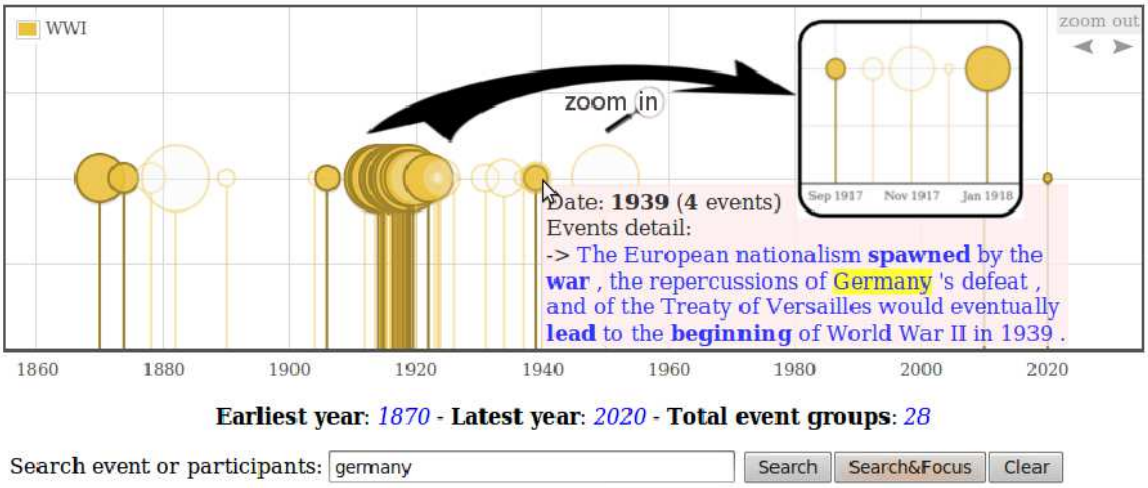

FIGURE 1.6 : [Llorens *et al.*, 2011] Affichage des résultats avec une chronologie.

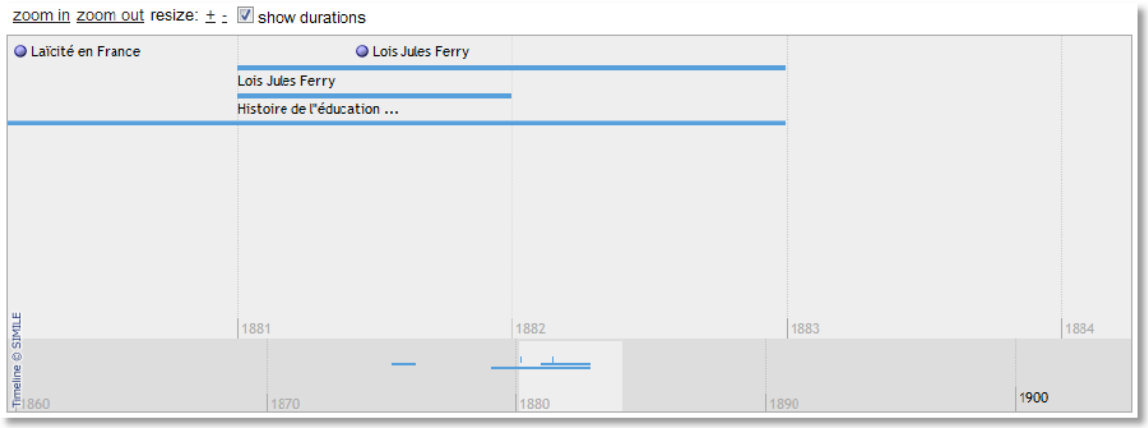

FIGURE 1.7 : [Teissèdre, 2012] Affichage des résultats avec une chronologie.

booléennes pour trouver la cible.

L'application *Google Squared* [Crow, 2010], qui n'existe plus aujourd'hui, regroupait également des informations dans des tableaux à partir de requêtes utilisateur. Leur approche était très proche de celle de WolframQA au sens où ils utilisaient en amont leur force de calcul pour extraire des centaines de millions de tableaux et de listes depuis leurs pages indexées ; l'extraction d'information de ces objets permettait alors d'obtenir des paires attribut/valeurs. Un travail linguistique avait permis d'extraire dix milliards de

FIGURE 1.8 : Exemple de question sans réponse pour Wolfram Alpha.

faits et de leur attribuer un score de confiance pour faire remonter les snippets contenant des réponses déjà connues.

Il existait également l'application *Google Question* fermée en 2006 mais qui subsiste encore partiellement sur la version anglaise du moteur de recherche : la formulation de questions factuelles telles que *Who directed the longuest day ?* est correctement interprétée par Google qui, en plus des résultats habituels (URL/titre/snippet), propose une réponse provenant de plusieurs sites de confiance (Wikipédia, Imdb). Il est alors possible pour l'utilisateur d'invalider certaines des réponses proposées (figure 1.9).

1.3 CONCLUSION

Dans ce chapitre, nous avons tout d'abord posé notre terminologie concernant les éléments structuraux susceptibles de contenir des réponses à des questions-listes dans les documents HTML et textuels, à savoir les énumérations et les tableaux. Nos définitions des énumérations s'appuient notamment sur les études existantes d'observations en corpus. Nous avons ensuite constaté que les questions-listes avaient été traitées par plu-

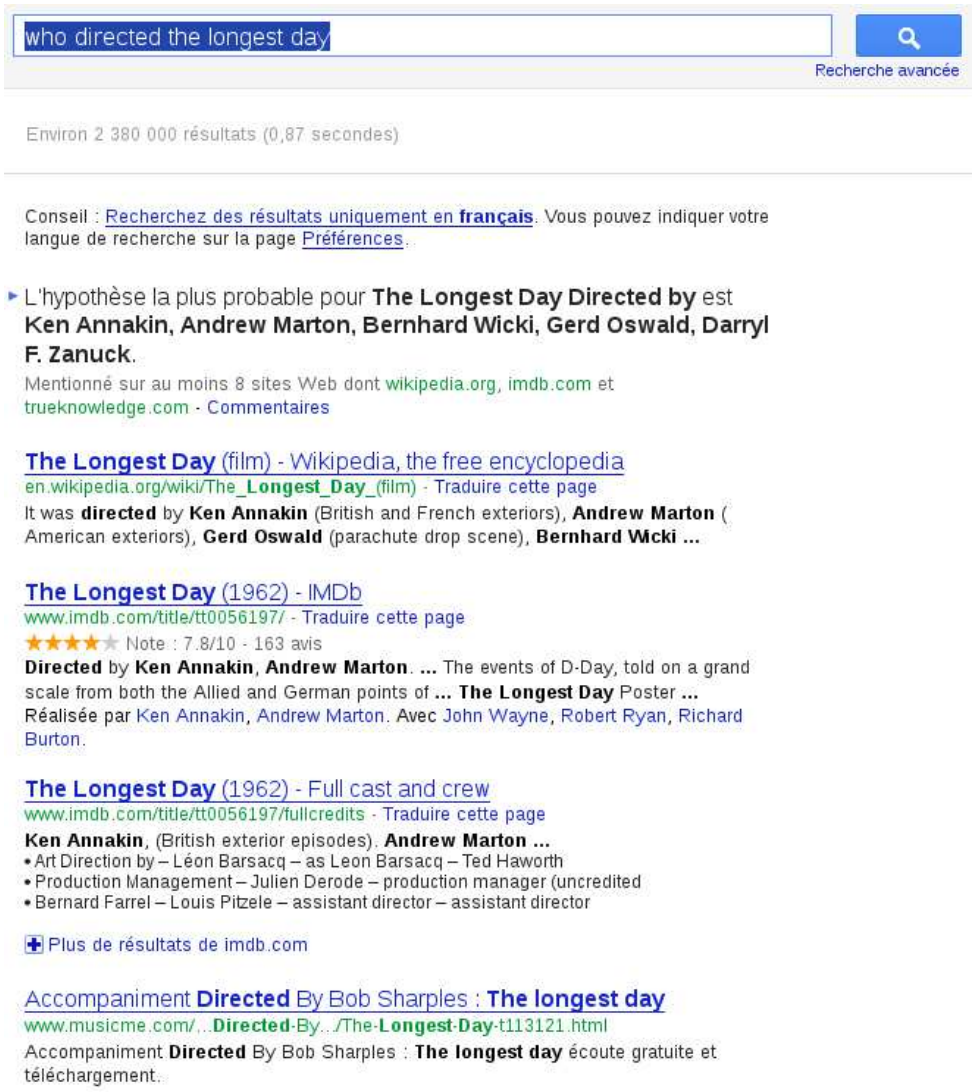

FIGURE 1.9 : Réponse à une requête formulée sous forme de question.

sieurs campagnes d'évaluation mais toujours avec le même schéma d'évaluation, schéma laissant transparaître quelques limitations, notamment pour la justification de réponse à partir de plusieurs documents. Ces questions-listes pouvaient être traitées spécifiquement par certains participants ou simplement bénéficier d'une légère adaptation par rapport aux questions non-listes. Enfin, nous avons vu plusieurs applications utilisateurs (question-réponse et recherche d'information) et les limites qu'elles rencontraient concernant le traitement des questions-listes, notamment dans le cas d'une approche s'appuyant sur une base de données.

Jusqu'ici, nous avons vu séparément les questions-listes et les éléments structuraux. Dans le chapitre suivant, nous présentons les observations que nous avons faites sur les formes que peuvent prendre les réponses aux questions-listes dans des corpus en français, notamment ceux utilisés lors des campagnes d'évaluation.

2

OBSERVATIONS EN CORPUS

Sommaire

Dans ce chapitre, notre but est d'observer en corpus les réponses aux questions-listes en français afin de pouvoir les caractériser. Notre première intuition est qu'une réponse à une question-liste est formulée sous forme d'énumération, et plus particulièrement d'une énumération verticale s'il s'agit de documents HTML. Nous allons donc chercher à vérifier cette intuition en catégorisant les réponses. Pour les réponses se présentant sous forme d'énumération, nous les confronterons à l'état de l'art. Notre deuxième intuition concerne la répartition des réponses dans la collection de documents : nous pensons qu'il est possible de recouper des informations provenant de plusieurs documents différents. Pour cela, pour toutes les réponses, nous étudierons leur répartition intra- et multi-document.

Après une étude de la constitution et des caractéristiques de ce corpus, nous présenterons nos principales observations sur les questions, les énumérations et les réponses. Ces observations justifieront la constitution d'un second corpus de questions dont nous présenterons les caractéristiques et nos principales observations.

Pour la suite, nous posons les définitions suivantes :
- une **question-liste** est une question qui attend plusieurs réponses, telle que définie dans les campagnes d'évaluation,
- une **réponse-liste** est la réponse à une question-liste : une réponse-liste est donc composée d'un ensemble exhaustif de **réponses individuelles** (ou réponses),
- un **passage-réponse** est un extrait de document qui contient la réponse-liste ou une ou des réponses individuelles.

Par exemple :
- **question-liste** : *Quels sont les 7 astres du système solaire visibles à l'œil nu ?*
- **réponse-liste** : *le Soleil, la Lune, Mercure, Vénus, Mars, Jupiter, Saturne*
- **passage-réponse** : *Les astres visibles à l'œil nu, le Soleil, la Lune, Mercure, Vénus, Mars, Jupiter et Saturne tiennent leur nom du monde romain.*

2.1 PREMIÈRES OBSERVATIONS SUR DES CORPUS DE CAMPAGNES D'ÉVALUATION DE SQR

Pour établir nos premières observations, nous nous sommes intéressés aux corpus des deux campagnes d'évaluation EQueR [Ayache *et al.*, 2006] et Quaero [Quintard *et al.*, 2010]. Ces deux campagnes sont les seules à réunir les trois conditions nécessaires à notre étude à savoir : travailler sur le français, proposer des questions-listes et pouvoir accéder aux collections de documents.

2.1.1 *Caractéristiques de ces corpus*

2.1.1.1 *Les documents*

Le corpus Quaero étudié ici a été constitué pour les campagnes d'évaluation annuelles des systèmes de question-réponse participant au projet Quaero dans le cadre de la thématique question-réponse [Quintard, 2010]. Pour le français, il se compose à l'origine de deux millions de documents collectés sur le Web par la société participante Exalead entre mai et juin 2008. Aucun autre fichier que celui référencé par l'URL n'a été téléchargé, il n'y a donc pas de fichier CSS, DTD ou image par exemple. Ce travail a été réalisé de manière automatique et le corpus ne peut donc garantir une homogénéité de style, source ou fenêtre temporelle. Il constitue toutefois un corpus suffisamment important pour permettre un travail en domaine ouvert. Nous utilisons ici le sous-corpus d'environ 500 000 documents, décrit par Exalead comme étant représentatif du corpus complet. Il comporte deux types de documents : ceux de type HTML (99,84 %) et ceux ayant dû être convertis dans un format XML basique ne comportant que des balises de segmentation de pages et de paragraphes (0,16 %). Pour ces derniers, il s'agit de documents au format fermé (animations Flash, Microsoft Office, etc.) ou libre (PDF, Open Document, etc.) mais néces-

sitant une conversion préalable pour être utilisables en tant que document texte.

La campagne d'évaluation EQueR a proposé deux tâches de recherche automatique de réponses pour le français [Ayache *et al.*, 2006] : une « tâche générique » sur une collection hétérogène de textes – en large partie des articles de presse – et une « tâche spécifique », liée au domaine médical, sur une collection de textes de cette spécialité. Le corpus EquER de la tâche spécifique, désormais désigné dans ce document par *Eq-Méd*, se compose principalement d'articles scientifiques et de recommandations de bonne pratique médicale, sélectionnés par le CISMeF (Catalogue et Index des Sites Médicaux Francophones) du Centre Hospitalier Universitaire de Rouen. Sa taille est de 140 Mo et il est composé de 5 623 fichiers. Le corpus EquER de la tâche générique, désormais désigné dans ce document par *Eq-Jour*, se compose d'articles de presse des journaux Le Monde et Le Monde Diplomatique, de dépêches de presse et de rapports d'information du Sénat français portant sur des sujets très variés. Une attention particulière avait été portée sur les fenêtres temporelles afin de garantir une couverture de sujet hétérogène avec plusieurs points de vue et types d'article (d'actualité ou de fond). Sa taille est de 1, 5 Go et il contient 557 300 documents.

2.1.1.2 *Les questions-listes*

La campagne Quaero de 2008 comporte 257 questions qui ont été formulées par des francophones à partir des documents de la collection. Elles se répartissent de la façon suivante : 148 factuelles, 56 définitions, 31 listes, 19 booléennes et 3 complexes (questions en comment et pourquoi). Seules les 31 questions-listes ont été étudiées ici.

La génération des questions EQueR a été diversifiée : une partie est dérivée de mots-clés qui accompagnaient les articles et les dépêches de presse, une autre partie a été créée par un groupe d'utilisateurs potentiels, dont certains connaissaient le domaine du TAL. Il est notamment affirmé que *la présence d'au moins une bonne réponse a été vérifiée dans le corpus pour chaque question proposée aux participants* [Ayache, 2005] mais nous verrons que certaines questions-listes n'atteignent pas le nombre de réponses attendues. Le corpus Eq-Jour comporte 30 questions-listes tandis que le corpus Eq-Méd en comporte 25 (pour ce dernier, nous avons toutefois dû éliminer une question-liste qui ne comportait pas de réponse dans la collection et son nombre sera donc 24). Les questions-listes pour ces deux campagnes se définissent comme des questions attendant un nombre bien précis de réponses, ce nombre étant toujours indiqué dans la question.

Le tableau 2.1 présente les principales caractéristiques de ces 3 corpus.

	Eq-Méd	**Eq-Jour**	**Quaero**
Domaine	médical	presse, politique	ouvert
Format des documents	texte	texte	HTML et XML
Nombre de documents	5 623	557 300	499 736
Taille de la collection	0,135 Go	1,5 Go	5 Go
Nombre de questions-listes	24	30	31

Tableau 2.1: Caractéristiques des corpus EQueR et Quaero.

2.1.2 *Collecte des données d'étude*

2.1.2.1 *Formatage des documents*

Le corpus Quaero est composé de documents HTML. Dans un premier temps, un prétraitement de conversion a été appliqué avec le programme Kitten [1] [Falco *et al.*, 2012] afin d'obtenir une collection de documents textes au format linéaire. Cette conversion est notamment nécessaire pour l'utilisation de nombreux outils de TAL comme les analyseurs syntaxiques. En effet, les documents HTML sont destinés à être interprétés sur un support numérique et utilisent très fréquemment des espaces pour délimiter visuellement des blocs de texte. Il n'y a donc pas forcément de continuité textuelle au sens d'un document texte classique, à savoir linéairement de gauche à droite puis de haut en bas : une extraction linéaire du fichier source HTML causerait une rupture dans le lien sémantique du bloc.

Les corpus Eq-Méd et Eq-Jour sont composés des documents d'origine ainsi que de leur version faiblement balisée au format XML : les balises sont surtout nombreuses pour les informations relatives à la provenance du document (en-tête de fichier) et basiquement structurelles pour le contenu textuel (titre, texte, paragraphe). Il a toutefois été nécessaire d'effectuer un traitement de formatage pour tout passage extrait d'un document du corpus Eq-Méd. En effet, la conversion par les organisateurs des documents d'origine (principalement depuis le format PDF) en documents XML avait conservé la mise en page d'origine. Ainsi, les espaces entre les mots s'en étaient donc trouvés multipliés, ce qui peut poser des problèmes en terme de calcul de distance sur le nombre de caractères et de longueur du passage. Toujours pour respecter la mise en page d'origine, des retours chariot avaient été insérés en milieu de phrase, ce qui peut fausser l'analyse syntaxique. Ces retours chariot ont été supprimés.

1. Nous présenterons Kitten au chapitre suivant.

2.1.2.2 *Méthodologie*

Nous avons effectué une première étude des documents contenant des réponses correctes : ces réponses et ces documents étaient fournis par les organisateurs des campagnes sous forme d'un fichier de référence. Nous considérons ici qu'une réponse est correcte si elle répond à la question : une première validation humaine avait déjà été effectuée par les organisateurs et une seconde a donc été appliquée par nos soins pour confirmer que ces réponses étaient bien correctes et également pour en ajouter d'autres. Notre validation considérait une réponse comme correcte même s'il existait des réponses correctes plus pertinentes (au sens de plus récentes par exemple, ou bien satisfaisant plus l'utilisateur dans un cadre applicatif), et même si la question attendait un nombre déterminé de réponses.

Nous avons utilisé le moteur de recherche Lucene[2] [Hatcher *et al.*, 2010] pour rechercher les documents contenant au moins une réponse aux questions-listes puisque les réponses des fichiers de référence n'étaient pas forcément exhaustives. Ceci nous a permis d'étudier les différentes formes d'une réponse-liste dans différents documents, ainsi que les répartitions des réponses individuelles correctes dans plusieurs documents le cas échéant. Les requêtes ont été formulées manuellement : soit à partir des termes de la question jugés importants, soit à partir des réponses des fichiers de références.

Nous avons étudié manuellement jusqu'à 50 extraits de documents par question puis les documents entiers si les snippets sélectionnés par Lucene étaient pertinents. Ensuite, d'autres requêtes ont été reformulées à l'aide de synonymes pour les termes de la question et des réponses nouvellement trouvées afin d'augmenter le nombre de passages-réponses. Enfin, certaines requêtes ont été formulées avec une réponse, les termes jugés cruciaux de la question et en excluant les documents contenant d'autres réponses. Nous souhaitions ainsi faire remonter les documents ne contenant pas la réponse-liste mais contenant au moins une réponse individuelle de la réponse-liste, ceci afin de voir quelle était la proportion de réponses multi-documents pour ces questions. Nous avons arrêté la collecte quand nous n'observions plus de nouvelles réponses ou de nouveaux phénomènes, ce qui se produisait avant cinquante documents maximum.

2.1.3 *Observations sur les questions*

Nous avons d'abord analysé toutes les questions qui avaient été explicitement typées comme de type liste par les évaluateurs de ces campagnes. Nous avons constaté qu'elles portaient toutes une marque de pluriel sous forme du nombre de réponses attendues et qu'elles ne se formulaient que sous quatre patrons différents (voir tableau 2.2). Si les formes syntaxiques sont similaires entre les trois corpus, il n'en est pas de même pour le type de réponse attendu (voir tableau 2.3). En effet, il y a deux types de questions-listes :

2. http://lucene.apache.org/core/

- celles attendant des réponses individuelles qui sont factuelles. Par exemple :
  - *Quels sont les 7 astres du système solaire visibles à l'œil nu ?* (Quaero, 241)
  - *Quelles sont les trois sociétés possédant le capital de Air Inter ?* (Eq-Jour, 377)
  - *Quelles sont les 4 localisations possibles des neuroblastomes ?* (Eq-Méd, ML134)
- celles attendant des réponses individuelles complexes [Moriceau *et al.*, 2010]. Par exemple :
  - *Citez 4* ***causes*** *possibles d'une infection du site opératoire.* (Eq-Méd, ML135)
  - *Quelles sont les 8* ***étapes*** *pour la fabrication de la chaux ?* (Quaero, 234)
  - *Quels sont les 7* ***objectifs*** *de la consultation de diététique ?* (Eq-Méd, ML143)

	Eq-Méd	**Eq-Jour**	**Quaero**
Citez X	12	5	0
Quels sont les X ?	12	22	31
Qui sont les X ?	0	2	0
Comment se prénommaient les X ?	0	1	0
Nombre de questions-listes	24	30	31

Tableau 2.2: Nombre de questions-listes par forme syntaxique (X est le nombre de réponses attendues).

	Eq-Méd	**Eq-Jour**	**Quaero**
Nombre de questions-listes	24	30	31
Nombre de questions à réponses factuelles	18	29	30
Nombre de questions à réponses complexes	6	1	1

Tableau 2.3: Type des questions-listes.

2.1.4 *Observations sur les énumérations horizontales et intra-phrastiques*

La difficulté concernant les énumérations intra-phrastiques est de réussir à segmenter correctement les items, notamment lorsqu'ils sont reliés par une coordination. L'exemple ci-dessous montre d'abord l'utilisation de la conjonction *et* pour coordonner les différents items à l'intérieur d'une SE mais aussi pour introduire l'amorce d'une seconde SE enchaînée.

Question : *Quelles sont les trois tailles du Teckel ?* (Quaero, 15)
Passage-réponse :
Il faut tout d'abord préciser que la race teckel présente la particularité de trois tailles : Standard, nain *et* kaninchen, *et* de trois textures de poil : dur, long *et* ras, ce qui donne neuf types de chiens.

Dans l'exemple suivant d'énumération horizontale, la conjonction *et* est utilisée à l'intérieur même des items pour regrouper plusieurs actions appartenant à une même étape :

Question : *Quelles sont les 8 étapes pour la fabrication de la chaux ?* (Quaero, 234)
Passage-réponse :
Pour l'essentiel, les procédés de la chaux passent par les étapes fondamentales suivantes, illustrées à la Figure 2.3 : extraction du calcaire, stockage et préparation du calcaire, stockage et préparation des combustibles, cuisson du calcaire, broyage de la chaux vive, hydratation et extinction de la chaux vive, stockage, manutention et transport.

Pour le cas de réponses se trouvant dans des énumérations intra-phrastiques ou horizontales, le problème essentiel qui va se poser est donc celui de la délimitation des items.

2.1.5 *Observations sur la forme des énumérations verticales*

Durant l'étude des paires questions-réponses, nous nous sommes naturellement intéressés aux énumérations verticales rencontrées afin de les confronter à l'état de l'art et ainsi savoir si certains formalismes tirés d'autres corpus vus aux chapitres précédents (encyclopédies, journaux, notices techniques) peuvent également s'appliquer à nos corpus. Nous avons ici complété manuellement les passages-réponses en extrayant les énumérations verticales complètes depuis les documents.

2.1.5.1 *Exemples conformes aux descriptions attendues*

Les SE avec énumérations verticales trouvées dans le corpus et conformes aux descriptions vues dans le chapitre précédent répondent à la définition d'une SE : une amorce, des items et une conclusion facultative. Elles possèdent notamment toutes une amorce claire apportant une définition du type de chaque item et le type syntaxique de chaque item de la SE est identique comme le montrent les exemples suivants :

Question : *Citez les 15 livres sélectionnés pour concourir au prix Femina en 1992 ?* (Eq-Jour, 374)
Passage-réponse :
Le jury du prix Femina a rendu publique sa première sélection en vue du prix qui sera décerné le 16 novembre. Quinze romans figurent dans cette sélection :
Myriam Anissimov, Dans la plus stricte intimité (L'Olivier) ; Patrick Besson, Julius et Isaac (Albin Michel) ; Bruno Bontempelli, l'Arbre du voyageur (Grasset) ; Patrick Chamoiseau, Texaco (Gallimard) ; Régine Detambel, la Quatrième Orange (Julliard) ; Jean Echenoz, Nous trois (Minuit) ; Anne-Marie Garat, Aden (Seuil) ; Franz-Olivier Giesbert, l'Affreux (Gallimard) ; Guyette Lyr, la Petite Nudité (Calmann-Lévy) ; Marie Nimier, l'Hypnotisme à la portée de tous (Gallimard) ; Amélie Nothomb, Hygiène de l'assassin (Albin Michel) ; Jean-Claude Perpère, la Larme d'or (Plume) ; Marie Redonnet, Candy Story (POL) ; Christiane Singer, Une passion (Albin Michel) ; François Weyergans, la Démence du boxeur (Grasset).

Question : *Quels sont les 4 principaux symptômes du cancer de l'ovaire ?* (Eq-Méd, ML144)
Passage-réponse :
Les principaux symptômes du cancer de l'ovaire sont :
– une sensation de poids au niveau du ventre ;
– une augmentation du volume de l'abdomen liée soit au développement de la tumeur, soit à la présence d'ascite ;
– des douleurs pelviennes liées à l'ascite et à l'atteinte d'une partie du péritoine (l'épiploon) ;
– une douleur aiguë liée à certains mouvements qui déplacent la tumeur.

Question : *Quelles sont les 8 étapes pour la fabrication de la chaux ?* (Quaero, 234)
Passage-réponse :
Pour l'essentiel, les procédés de la chaux passent par les étapes fondamentales suivantes, illustrées à la Figure 2.3 :
– extraction du calcaire,
– stockage et préparation du calcaire,
– stockage et préparation des combustibles,
– cuisson du calcaire,
– broyage de la chaux vive,
– hydratation et extinction de la chaux vive,
– stockage,
– manutention et transport.

Une SE peut être récursive si elle est bien formée à tous les niveaux. Il s'agit donc d'un paramètre à intégrer pour gérer le cas où l'amorce définit le nombre d'éléments attendus et que ces derniers sont répartis dans des énumérations imbriquées. L'exemple suivant illustre ce point ainsi que l'utilisation du patron « terme : description du terme » étudié par [Kamel et Aussenac-Gilles, 2009] et [Laignelet *et al.*, 2011], patron utilisé fréquemment au début d'un item pour introduire une définition :

Question : *Quelles sont les trois tailles du Teckel ?* (Quaero, 15)
Passage-réponse :
Description (Standard FCI) Il représente à lui seul le 4e Groupe de la Classification FCI. On en distingue neuf variétés **dans la classification FCI** :
- Standard : Chien de plus de 35 cm de tour de poitrine mais ne dépassant pas 9 kg (avec un tolérance de 10 %)
  - teckel poil ras,
  - teckel poil dur,
  - teckel poil long.
- Nain : Chien ayant un tour de poitrine compris entre 30 et 35 cm (avec une tolérance de 2 cm) et il existe en
  - teckel poil ras,
  - teckel poil dur,
  - teckel poil long.
- Kaninchen : Chien ayant un tour de poitrine inférieur à 30 cm (avec une tolérance de 2 cm), et il existe en
  - teckel poil ras,
  - teckel poil dur,
  - teckel poil long.

Et Seulement 6 **en Angleterre, au Canada et au USA**, en effet, la taille kaninchen n'existe pas. Seuls les Standards et les Miniatures sont représentés. Dans les teckels miniatures, on retrouve les nains et les Kaninchens de la Classification FCI (voir les liens en fin d'articles).

L'amorce, les items ou la conclusion des SE peuvent comporter des informations définissant la portée de la réponse. Ainsi dans l'exemple précédent, on note que l'amorce précise qu'il existe neuf variétés dans le cadre de la classification FCI (soit trois tailles) tandis que la conclusion indique seulement 6 en Angleterre, au Canada et au USA (soit deux tailles). Ainsi, la question demandant les trois tailles se place dans le contexte de la classification FCI sans le préciser explicitement ; seule une analyse de la conclusion et un raisonnement *a posteriori* sur le nombre de tailles citées permet de le déduire.

2.1.5.2 *Exemples non conformes et problèmes qui se posent*

Nous avons rencontré dans notre corpus, plusieurs problèmes de mise en forme des SE, problèmes notamment dus au format HTML. Les énumérations verticales présentant ce type de problème ne sont pas traitées par l'état de l'art car elles sont la plupart du temps éliminées lors de la constitution des corpus d'étude.

Les usages des niveaux hiérarchiques en HTML

On trouve notamment un problème de mise en page dans des documents qui utilisent les énumérations verticales imbriquées à un niveau si profond qu'une simple mise à plat des phrases ferait perdre les informations sémantiques de la structure. Ce problème de

segmentation visuelle par niveau hiérarchique se retrouve dans l'exemple suivant mais également sur les forums ou les commentaires d'articles de journaux :

Généalogie :
I Guillaume de MOLIN, seigneur du Pont-de-Mars (entre Le Chambon-sur-Lignon F-43 et Saint-Agrève F-07),
ép. par contrat du 12 septembre 1475 Alasie d'ARLEMPDES ;
D'où probablement au moins :
II Pierre de MOLIN, né vers 1480, seigneur du Pont-de-Mars, testa le 5 septembre1547 (en faveur de Guilhot son petit-fils) ;
épousa par contrat du 2 novembre 1507 Anne VIALATTE ;
D'où :
III Jean de MOLIN, seigneur du Pont de Mars, né vers 1510 et décédé en 1547 ; ép. Ne...
D'où :

1. Guillaume qui suit.
2. Charles de MOLIN du PONT, seigneur du Pont-de-Mars et de Pelinac, né au Pont-de-Mars vers 1547 ;
   épousa à Saint-Jeures F-43 le 12 novembre 1577 Claudine de CHALENDAR de CORNILLON, née vers 1556 ;
   D'où :
   a) Louise de MOLIN du PONT de PELINAC.
   b) Jeanne de MOLIN du PONT de PELINAC.
   c) Charles Alexandre de MOLIN de PELINAC ;
      épousa Catherine de SENECROZE ;
      D'où :
      – - Suzanne ; épousa 1°) Charles de ROMANET puis 2°) Charles de LA ROCHENEGLY.
      – - Marguerite

Il arrive parfois que les énumérations verticales imbriquées soient toutes mises au même niveau. L'hypothèse d'une erreur du concepteur de la page est plausible mais le résultat est de toutes façons à prendre en compte. Il devient alors très difficile d'analyser correctement les items sans connaissance du monde. Ainsi, dans l'exemple suivant, l'absence d'indentation et d'amorce pour les items commençant par *à* rendent difficile la visualisation de la structure même si à la lecture, on comprend que les deux dernières énumérations sont en fait toutes deux imbriquées dans l'item *Par l'examen des incisives* bien que placées au même niveau hiérarchique.

Comment connaître l'âge d'une vache
- **Par l'examen des cornes**
  La croissance des cornes crée à leur base des bourrelets dus au fait que cette croissance est saisonnière. Le premier apparaît à l'âge de trois ans, puis il s'en forme un chaque année, avec une atténuation progressive. Une vache qui présente n bourrelets a donc n+2 ans, sous réserve que son propriétaire n'ait pas cherché à la rajeunir quelque peu en lui limant les cornes et... qu'elle ne soit pas de race Angus qui par suite d'une mutation naturelle n'a pas de cornes.
- **Par l'examen des incisives**
  La vache possède sur le maxillaire inférieur huit incisives : celles (deux) du centre sont les pinces (aussi appelées pelles), les suivantes (de part et d'autre des précédentes) les premières mitoyennes, puis viennent les deuxièmes mitoyennes et les coins. Les dents de lait sont progressivenment remplacées par des dents adultes puis celles-ci s'usent.
- **à** un an et demi, les pinces de lait sont remplacées par les dents définitives ;
- **à** deux ans et demi, c'est le tour des premières mitoyennes ;
- **à** trois ans et demi, c'est celui des deuxièmes mitoyennes ;
- **à** cinq ans enfin c'est les coins qui sont remplacés.
  Ensuite l'usure des dents apparaît :
- **à** six ans sur les pinces ;
- **à** sept ans sur les premières mitoyennes ;
- **à** huit ans sur les secondes mitoyennes ;
- **vers** neuf ans sur les coins.
  Ensuite, progressivement les dents ne sont plus jointives. Après douze ans l'usure des incisives ne laisse plus apparaître que les racines.

Le HTML permettant d'utiliser l'intégralité d'une page comme délimiteur de bloc de texte, des concepteurs de pages utilisent des tableaux pour placer des éléments sur une page. Par exemple, la figure 2.1 montre qu'avec l'utilisation de tableaux HTML, une SE peut posséder deux items sur une même ligne et qu'une amorce peut ne pas spécifier le type des items : ici, ce sont nos connaissances du monde qui nous permettent de savoir qu'il s'agit d'ingrédients pour une recette de cuisine. Extraire une réponse-liste pour répondre à la question *Quels sont les 10 ingrédients nécessaires pour cuisiner un « porc aux poivrons » ? (Quaero, 218)* nécessitera donc un important travail de détection et de traitement de cette SE particulière.

Les puces en HTML

Il existe dans le langage HTML deux balises permettant de créer des énumérations verticales : <ul> (listes non ordonnées) et <ol> (listes ordonnées). La balise <li> permet quant à elle de créer des items à l'intérieur d'une liste HTML. Il arrive cependant que les concepteurs de page HTML ne les utilisent pas et structurent les items « en dur », comme dans les deux passages[3] suivants :

3. identiques car il s'agit d'une blague reproduite en copier-coller sur de nombreux sites

FIGURE 2.1 : Deux items d'une SE sur une même ligne en HTML.

Question : *Quels sont les six ingrédients de la "dinde au Whisky" ?* (Quaero, 22)
Passage-réponse :
Recette de la dinde au whisky :
1 - Acheter une dinde d'environ 5 kg pour 6 personnes et une bouteille de whisky, du sel, du poivre, de l'huile d'olive, des bardes de lard.
2 - La barder de lard, la ficeler, la saler, la poivrer et ajouter un filet d'huile d'olive.

Passage-réponse :
La dinde au whisky :
* Étape 1 : Acheter une dinde d'environ 5 kg pour 6 personnes et une bouteille de whisky, du sel, du poivre, de l'huile d'olive, des bardes de lard.
* Étape 2 : La barder de lard, la ficeler, la saler, la poivrer et ajouter un filet d'huile d'olive.

Parmi ces énumérations verticales structurées « en dur », on trouve parfois l'utilisation d'une image comme puce. Cependant, si le concepteur a renseigné l'attribut `alt` de la balise <img> avec un texte alternatif, ce texte peut être utilisé pour guider l'extraction textuelle des items. Dans l'exemple suivant, le concepteur de la page a défini un texte alternatif à afficher à savoir la lettre *o* pour simuler une puce, illustrant la nécessité d'analyser ce texte alternatif afin d'identifier les items :

Baekenofe :
o Nb invités/Proportions : 6 personnes
o Cuisson : 4 h
o Ingrédients :
o 500 g d'épaule d'agneau désossée
o 500 g d'échine de porc désossée
(...)
o Pour la marinade :
  o 2 oignons
  o 2 gousses d'ail
  (...)

Dans cet exemple, la seconde SE comporte également un problème d'alignement de niveaux : les ingrédients concernant la marinade font partie des ingrédients de la recette mais le fait qu'ils soient séparés des autres est une information importante à conserver, notamment à des fins d'agrégation.

Référence à un item sous la forme d'un symbole

Le cas des renvois à des sections précédentes ou suivantes (par exemple, *dans la section précédente* ou *dans le paragraphe suivant*) a été évoqué dans les différentes études sur les SE. Le problème se pose lorsqu'il s'agit d'un renvoi à l'aide d'un symbole. L'exemple suivant montre des références à l'aide de puces :

Question : *Quels sont les 5 critères diagnostics de l'Algie vasculaire de la face selon l'International Headache Society ?* (E04-Méd, ML144)
Passage-réponse :
Les critères diagnostiques sont selon l'International Headache Society (IHS) :
A- Au moins 5 crises répondant aux critères **B** et **D**
B- Douleurs sévères unilatérales orbitaires, supraorbitaires ou temporales durant 15-180 minutes sans traitement
C- Céphalée associée à au moins un des caractères suivants survenant du côté de la douleur :
– injection conjonctivale
– larmoiement
– congestion nasale
(...)
D- Fréquence des crises de 1 à 8 par jour
E- Au moins un des caractères suivants :
– l'histoire, l'examen physique et neurologique ne suggèrent pas un désordre organique mais celui-ci est écarté par la neuro-imagerie ou toute autre investigation.
– un désordre organique existe mais les crises d'AVF ne sont pas apparues pour la première fois en liaison temporelle avec celui-ci.

Dans cet exemple, extraire le seul item *A* serait donc incomplet (au-delà du nombre de réponses attendues).

Présence de spécifieurs

Il arrive que des spécifieurs soient présents dans l'amorce ou dans une introduction à la SE. Il faut être capable de détecter et éventuellement interpréter ces spécifieurs si la question posée le demande. L'exemple suivant montre l'utilisation de synonymes dans l'introduction (*prouvés*) et dans l'amorce (*connus*) :

Question : *Quels sont les 6 facteurs de risque (supérieurs à 4,0) du cancer du sein ?* (Eq-Méd, 133)
Passage-réponse :
Les chercheurs scientifiques ne connaissent pas avec certitude les causes directes du cancer du sein, mais ils ont défini des facteurs de risque prouvés et des facteurs de risque possibles.
Facteurs de risque connus :
– Sexe : plus de 99 % des cancers du sein surviennent chez les femmes
– Âge : le risque augmente avec l'âge
– Début précoce des règles (avant 12 ans)
– Ménopause tardive (après 55 ans)
(...)
Facteurs de risque possibles :
– Alimentation pauvre en fruits et légumes
– Consommation excessive d'alcool
(...)

Ces observations sur les énumérations verticales révèlent la difficulté à identifier ces structures, et notamment à en délimiter ses composantes (amorce et items). Ces difficultés proviennent principalement de son utilisation récursive de l'espace vertical ainsi que du codage des listes en HTML.

2.1.6 *Observations sur les réponses*

Nous nous intéressons dans cette partie aux réponses-listes extraites du corpus. Il s'agit surtout de voir comment ces réponses peuvent être utilisées pour répondre à une question. Pour cela, nous allons étudier leur *unicitabilité*, leur longueur, leur forme et enfin leur répartition dans le corpus.

2.1.6.1 *Unicitabilité*

Lors de l'évaluation d'un SQR durant une campagne d'évaluation, le critère pour déterminer la validité d'une réponse repose entre autre sur le passage-réponse extrait du document, qui doit justifier la réponse précise qui en a été extraite. Le critère que nous appelons *unicitabilité* se rapporte à l'indépendance du passage au sens où il est à la fois compréhensible et justifie la réponse sans avoir besoin d'autres éléments contenus dans le document. Ce critère est très important puisqu'il pose les problèmes suivants : est-ce qu'une SE avec son amorce peut être extraite de son contexte et garder son sens ? Est-ce qu'une SE peut voir plusieurs de ces items extraits sans que l'amorce ne le soit ?

L'exemple suivant illustre ce phénomène d'unicitabilité à travers le problème des anaphores : la distance en nombre de caractères entre le référé et les réponses est trop importante pour que ce passage puisse être fourni comme passage justificatif, les campagnes

d'évaluation imposant un nombre limité de caractères (250 caractères pour Eq-Jour et Eq-Méd, 800 pour Quaero). La campagne Eq-Jour permettait de proposer jusqu'à 20 réponses individuelles pour une question-liste, avec un extrait de 250 caractères pour chacune de ces réponses mais cela se révèle insuffisant ici vu la distance trop importante entre le référé et les trois réponses attendues :

> Une fenêtre possible de 250 caractères est indiquée en italique. La chaîne de coréférence a été soulignée.
>
> **Question :** *Quels sont les trois auteurs que Guy Des Cars lisait en cachette ?* (Eq-Jour, 397)
> **Réponse-liste :** Zola, Balzac, Conan Doyle.
> **Passage-réponse :**
> Guy des Cars avait ce don de romancier populaire, une étonnante manière de vivre ses sujets. Ceux qui, un peu hautains, voient en lui un habile fabricant, ne comprennent pas que son succès époustouflant est venu de ce qu'il croyait à ses histoires, les vivant avec la naïveté d'un enfant qui, dans la nuit, s'invente des rôles, et les racontait avec la conscience professionnelle d'un journaliste. C'est avant tout à « Mademoiselle Marie », son « initiatrice », qu'il rendait hommage. Cette gouvernante bourguignonne, qui, dans *la lourdeur de la famille ducale des Cars, fut « sa seule alliée », lui racontait chaque soir une histoire pour l'endormir. Mais il restait éveillé tant qu'elle n'avait pas fini. En grandissant, il eut moins sommeil et lut* ***Zola***, ***Balzac*** *et* ***Conan Doyle***, en cachette des « bons Pères » du collège Saint-François-de-Sales à Evreux, où un professeur qui n'était autre que le Père Teilhard de Chardin cherchait à donner à ses élèves le goût de la littérature.

L'analyse des trois corpus montre que les réponses-listes sont très largement unicitables (voir tableau 2.4). Ce résultat était attendu étant donnée qu'une sélection manuelle avait été effectuée de façon à ce que les passages contiennent les réponses-listes. Les seuls cas de réponses-listes non-unicitables (2,93 %) sont dus à des distances en nombre de caractères extrêmement importantes (plus de 700 caractères).

Corpus	**Eq-Méd**	**Eq-Jour**	**Quaero**
Nombre de questions-listes	24	30	31
avec au moins une réponse-liste unicitable	23	30	31
avec au moins une réponse-liste non-unicitable	2 (8 %)	1 (4,35 %)	0 (0 %)
Nombre de réponses-listes total	54	101	124
Nombre de réponses-listes unicitables	51 (94,44 %)	100 (99,01 %)	124 (100 %)
Nombre de réponses-listes non-unicitables	**3** (5,56 %)	**1** (0,90 %)	**0** (0 %)

Tableau 2.4: Unicitabilité des réponses-listes dans les corpus des campagnes d'évaluation.

On constate dans le tableau 2.4 un nombre plus important de réponses-listes pour le corpus Quaero. Ce résultat provient du fait que le corpus Quaero est bien plus important en nombre de documents et propose donc plus de documents capables de fournir une des réponses attendues. La seule réponse non-unicitable de Eq-Jour est celle due à l'anaphore de l'exemple précédent. Les trois réponses non-unicitables de Eq-Méd illustrent les problèmes liés à la structure typique des documents médicaux puisque les documents entiers traitent du topic abordé par la question et que ce topic est de moins en moins répété au fil du document. Par exemple :

Question : *Quels sont les trois types d'examens à réaliser en cas de suspicion d'un neuroblastome ?* (Eq-Méd, ML132)
Passage-réponse :
Le Diagnostic Comment savoir si c'est un neuroblastome ?
2.1 Le bilan : quels sont les examens pratiqués pour établir un diagnostic ?
2.1.1 L'EXAMEN CLINIQUE
Le médecin interroge les parents sur la santé de l'enfant depuis sa naissance (maladies, accidents, etc.), ce qu'on appelle les antécédents. Le médecin interroge les parents et l'enfant pour connaître l'histoire de la maladie, leur demande la façon dont les signes et les symptômes sont apparus. Ensuite, le médecin examine l'enfant afin de rechercher d'éventuels signes anormaux : boule, hématome, endroit douloureux, difficulté à bouger les membres, etc. L'examen clinique permet de voir quels sont les autres examens à réaliser.
Généralement, le bilan comprend :
– des examens par **prélèvements** (sang, urines),
– différents examens **radiologiques** qui ont pour but de bien situer la tumeur, ses limites et sa taille (scintigraphie, échographie et/ou scanner et/ ou IRM),
– des examens **au microscope** que l'on appelle examens anatomo-pathologiques : un fragment de la tumeur et un peu de moelle des os sont observés au microscope.

2.1.6.2 *Longueur des passages-réponses*

À l'exception des rares réponses-listes non-unicitables dans notre corpus, le passage-réponse que nous avons retenu était suffisant à son interprétation. Nous avons donc étudié sa longueur, tout d'abord en nombre de caractères puis en nombre de mots (voir tableau 2.5). Les documents issus du Web semblent fournir des passages bien plus concis par rapport aux domaines médical et journalistique : les réponses attendues y seraient donc plus concentrées. On constate également que la longueur moyenne des passages pour les trois collections est supérieure à la limite de 250 caractères généralement utilisée durant les campagnes d'évaluation.

Nous avons ensuite compté, pour chaque passage-réponse, le nombre de retours chariot (symbole \n) utilisés ainsi que le nombre de phrases à l'aide de l'outil NLTK (Natural Language Toolkit)[4] [Loper et Bird, 2002] (voir tableau 2.6). On constate que le corpus Eq-

4. http://www.nltk.org/

	Eq-Méd	Eq-Jour	Quaero
Nombre de passages-réponses	54	101	124
Nombre moyen de caractères des passages-réponses	440,91	433,05	**256,45**
Nombre médian de caractères des passages-réponses	400,50	339	**202**
Nombre moyen de mots des passages-réponses	65,44	68,56	**43,53**
Nombre médian de mots des passages-réponses	64,50	53	**35**

Tableau 2.5: Nombre de caractères et de mots dans les passages-réponses.

Méd contient un grand nombre de retours chariot par phrase ce qui confirme l'importance du pré-traitement à effectuer pour ce type de documents.

	Eq-Méd	Eq-Jour	Quaero
Nombre moyen de \n	6,56	0,74	2,34
Nombre médian de \n	8,5	0	0
Nombre moyen de phrases (avec NLTK)	1,74	2,58	1,78
Nombre médian de phrases (avec NLTK)	1,5	2	1

Tableau 2.6: Nombre de retours chariot et de phrases dans les passages-réponses.

Le critère d'unicitabilité et les longueurs moyennes des passages-réponses nous montrent donc les limites des campagnes d'évaluation pour justifier une réponse-liste. Quand bien même est mise en place la possibilité de fournir plusieurs extraits par réponse composant la réponse-liste, cela peut malgré tout se révéler insuffisant dans certains cas. Ce bridage trop contraignant et inadapté pour certaines questions-listes doit donc être repensé.

2.1.6.3 *Format des réponses*

Nous nous sommes ensuite intéressés à la forme des réponses-listes. Nous avons recensé quatre types d'éléments structuraux (voir tableau 2.7). Plus précisément :

- les trois premiers éléments structuraux sont les trois types d'énumération définies précédemment (horizontale, verticale et intra-phrastique). Une tolérance a été appliquée concernant l'énumération verticale : la présence du « : » n'était plus obligatoire si la mise en page suggérait effectivement une volonté d'énumérer verticalement les items (décalage des items à l'aide d'une tabulation notamment) ;
- le quatrième élément structural est le tableau ;
- la phrase est considérée si un seul candidat-réponse est présent dans cette phrase ;

– le paragraphe est considéré dans deux cas. Tout d'abord lorsqu'aucun des éléments structuraux précédents n'est utilisé et que plusieurs candidats-réponses sont présents sur plusieurs phrases du même paragraphe. Par exemple :

> **Question :** *Quels sont les 4 stades du cancer de l'ovaire ?*
> **Passage-réponse :** On distingue quatre stades de cancer de l'ovaire. Ces stades sont déterminés par (—)). Stade 1 : le cancer est limité à un ou aux deux ovaires sans atteindre d'autres organes. Stade 2 : le cancer a atteint d'autres organes proches (—).

Ensuite lorsque la réponse ne peut être comprise qu'à l'aide d'informations mentionnées dans des phrases en amont ou en aval.

	Eq-Méd	**Eq-Jour**	**Quaero**
Nombre de questions-listes	24	30	31
Nombre de passages-réponses	54	101	124
Nombre de réponses-listes sous forme de :			
énumération intra-phrastique	**25 (46,30 %)**	**45 (44,55 %)**	**54 (43,55 %)**
phrase	6 (11,11 %)	25 (24,75 %)	13 (10,48 %)
paragraphe	2 (3,70 %)	23 (22,77 %)	3 (2,42 %)
énumération horizontale	3 (5,56 %)	8 (7,92 %)	29 (23,39 %)
énumération verticale	18 (33,33 %)	0 (0 %)	23 (18,55 %)
tableau	0 (0 %)	0 (0 %)	2 (1,61 %)

Tableau 2.7: Forme des réponses-listes.

Contrairement au Web et au domaine médical, on observe que le corpus journalistique n'utilise pas les énumérations verticales. Dans les trois corpus, la majorité des passages-réponses se trouve sous forme d'énumérations intra-phrasiques mais se concentrer sur les énumérations verticales serait très utile pour le domaine médical et le Web. Le corpus journalistique présente également une part plus importante de réponses sous forme de phrase à l'intérieur d'un document, il comporte donc plus de candidats-réponses isolés.

Le corpus Web Quaero est codé en HTML et les énumérations intra-phrastiques et horizontales y sont quasi-exclusivement utilisées à l'aide des balises textes <p>, , (voir tableau 2.8). De plus, seules 5 des 23 énumérations verticales sont codées avec une balise HTML prévues à cet effet (<ul>, <dt>), la grande majorité étant codée manuellement à l'aide de balise texte ou de tableau de formatage.

Forme de la réponse (nb total)	Forme du codage HTML de la réponse	Occurrences
énumération intra-phrastique (54)	balise texte	41 (33,07 %)
	balise texte dans un tableau de formatage	6 (4,84 %)
	image (attribut balise <img>)	3 (2,42 %)
	balise <meta>	3 (2,42%)
	balise <table>	1 (0,81%)
énumération horizontale (29)	balise texte	23 (18,55 %)
	balise texte dans un tableau de formatage	3 (2,42 %)
	item d'une balise)	3 (2,42 %)
énumération verticale (23)	formatée manuellement avec des balises textes	7 (5,65 %)
	formatée manuellement dans un tableau de formatage	9 (7,26 %)
	liste (balise ou <dt>)	5 (4,03 %)
	balise <table>	2 (1,61 %)
phrase (13)	balise texte	13 (10,48 %)
paragraphe (3)	balise texte	3 (2,42 %)
tableau (2)	balise <table>	2 (1,61 %)

Tableau 2.8: Code HTML utilisé pour coder un passage contenant la réponse (corpus Quaero).

Dans le corpus Quaero, on remarque que plusieurs réponses-listes sont sous forme de paragraphe car beaucoup de documents comportent des listes formatées manuellement à l'aide de balises paragraphe (<p>) et de sauts de ligne (
) au lieu d'utiliser des balises liste (<ul>). On constate également que la balise prévue pour les tableaux est également très utilisée à des fins de mise en page. Cela montre l'importance du prétraitement des documents HTML pour parvenir à extraire des réponses-listes depuis tout type de structure.

2.1.6.4 *Répartition des réponses*

Les réponses individuelles composant une réponse-liste peuvent se trouver toutes dans un même document ou réparties dans plusieurs documents. Cela amène donc un travail d'agrégation et/ou de spécification des réponses individuelles. Cette agrégation est également à effectuer à l'intérieur même d'un document lors que les réponses individuelles sont réparties sur plusieurs passages-réponses.

Plusieurs cas sont donc possibles (voir tableau 2.10) :

- Cas 1 : la réponse-liste n'est présente que dans un seul document ;
- Cas 2 : la réponse-liste est présente, sous la même forme ou sous des formes différentes, dans plusieurs documents ;
- Cas 3 : les réponses individuelles composant la réponse-liste sont réparties dans des documents différents ;
- Cas 4 : les réponses individuelles composant la réponse-liste sont réparties dans plusieurs passages d'un même document.

CAS 1 : UN SEUL DOCUMENT CONTIENT LA RÉPONSE-LISTE. On constate qu'un tiers des questions dans chacune des collections ne trouve leur réponse-liste que dans un seul document (voir tableau 2.9). Il s'agit quasi-exclusivement de questions générées à partir d'un document très précis, document qu'il faudra absolument trouver durant la recherche d'information pour pouvoir répondre correctement. Par exemple :

Question : *Quels sont les trois auteurs que Guy Des Cars lisait en cachette ? (Eq-Jour, 397)*
Réponse-liste : Zola, Balzac, Conan Doyle.
Passage-réponse : Guy des Cars avait ce don de romancier populaire, une étonnante manière de vivre ses sujets. Ceux qui, un peu hautains, voient en lui un habile fabricant, ne comprennent pas que son succès époustouflant est venu de ce qu'il croyait à ses histoires, les vivant avec la naïveté d'un enfant qui, dans la nuit, s'invente des rôles, et les racontait avec la conscience professionnelle d'un journaliste. C'est avant tout à « Mademoiselle Marie », son « initiatrice », qu'il rendait hommage. Cette gouvernante bourguignonne, qui, dans la lourdeur de la famille ducale des Cars, fut « sa seule alliée », lui racontait chaque soir une histoire pour l'endormir. Mais il restait éveillé tant qu'elle n'avait pas fini. En grandissant, il eut moins sommeil et lut **Zola, Balzac et Conan Doyle**, en cachette des « bons Pères » du collège Saint-François-de-Sales à Evreux, où un professeur qui n'était autre que le Père Teilhard de Chardin cherchait à donner à ses élèves le goût de la littérature.

CAS 2 : PLUSIEURS DOCUMENTS CONTIENNENT UNE RÉPONSE-LISTE. Pour plus de la moitié des questions de chacune des campagnes, il est possible de trouver une réponse-liste dans plusieurs documents différents (voir tableau 2.9). Cette multiplicité de réponses-listes provenant de documents différents implique d'effectuer un traitement sur les réponses individuelles qui les composent, notamment parce que les réponses-listes ne sont par forcément toutes complètes dans les documents d'où elles ont été extraites.

	Eq-Méd	**Eq-Jour**	**Quaero**
Nombre de questions-listes	24	30	31
Nombre de questions où la réponse-liste est dans un seul document	7 (29,17 %)	11 (36,67 %)	9 (29,03 %)
Nombre de questions où la réponse-liste est présente dans plusieurs documents	17 (70,83 %)	19 (63,33 %)	22 (70,97 %)
Nombre de questions où la réponse-liste est obligatoirement répartie dans plusieurs documents	3 (12,5 %)	1 (3,33 %)	0 (0 %)

Tableau 2.9: Répartition des réponses-listes dans les documents.

Lorsque le nombre de réponses individuelles correctes trouvées dépasse le nombre d'éléments attendus, il faut soit choisir d'en éliminer, soit choisir de donner plus de réponses que le nombre attendu (sanctionné dans une évaluation de question-réponse mais possible dans un cadre utilisateur). Une des possibilités de sélection de réponses est de se rapprocher le plus des spécifieurs de la question, s'ils existent.

En effet, dans les trois corpus, nous avons trouvé plusieurs questions-listes dont le nombre de réponses individuelles correctes excédait celui attendu mais sans posséder ce spécifieur discriminant. Par exemple :

> **Question :** *Quelles sont les trois sociétés possédant le capital de Air Inter ? (Eq-Jour, 377)*
> **Passage-réponse avec cinq sociétés actionnaires :** Actuellement **le holding d'Etat Groupe Air France SA** détient 72 % du capital d'Air Inter, les autres actionnaires étant, outre la **SNCF**, **le Crédit lyonnais**, **la Caisse des dépôts** et **les chambres de commerce**.

Certaines de ces questions-listes utilisaient toutefois le spécifieur *principal* pour annoncer que le nombre attendu de réponses pouvait être dépassé. Le choix parmi les réponses individuelles correctes doit alors se faire sur un critère implémenté dans le SQR, qui est traditionnellement celui de la fréquence d'apparition, à l'aide de l'information mutuelle notamment. Par exemple :

> **Question :** *Qui sont les 3 acteurs principaux du film "No Country for Old Men" ?* (Quaero, 228)
> **Trois acteurs à choisir parmi les acteurs revenant fréquemment :** Woody Harrelson, Tommy Lee Jones, Myk Watford, Josh Brolin, Javier Bardem.

Des réponses différentes provenant de documents différents peuvent aussi être dues à la présence de spécifieurs dans les réponses et un travail important est à mettre en place (repérer et vérifier les spécifieurs, dénombrer les occurrences, etc.). Ainsi, l'exemple suivant illustre l'importance de disposer de la date d'un événement pour différencier les trois réponses-listes :

> **Question :** *Quels sont les cinq nouveaux membres non-permanents du Conseil de Sécurité de l'ONU ?* (Eq-Jour, 391)
> **Passage-réponse du document 1 :**
> **Le Cap-Vert**, **la Hongrie**, **le Japon**, le **Maroc** et **le Venezuela** sont devenus, *le 1 janvier 1992*, membres non permanents du Conseil de sécurité des Nations unies pour une période de deux ans.
>
> **Passage-réponse du document 2 :**
> *20 OCTOBRE 1994*, ONU : cinq nouveaux membres au Conseil de sécurité. L'Assemblée générale des Nations unies, réunie en séance plénière à New-York, a élu jeudi 20 octobre **l'Allemagne**, **l'Italie**, **l'Indonésie**, **le Botswana** et **le Honduras** membres non permanents du Conseil de sécurité de l'ONU.
>
> **Passage-réponse du document 3 :**
> *01 JANVIER 1996*, Cinq nouveaux membres au Conseil de sécurité de l'ONU. Cinq pays, représentant leur zone géographique, ont fait leur entrée le 1er janvier au Conseil de sécurité de l'ONU. Les nouveaux promus sont le **Chili** (pour l'Amérique latine), **l'Egypte** et **la Guinée-Bissau** (Afrique), **la Corée du Sud** (Asie) et **la Pologne** (Europe orientale).

Parfois le document ne présente pas de spécifieur et il faut alors effectuer une agrégation des réponses avec celles provenant d'autres documents. Cette agrégation peut toutefois se montrer infructueuse comme le montre l'exemple suivant puisque plus de deux énergies renouvelables sont en développement :

Question : *Quelles sont les deux énergies renouvelables qui sont en développement ?* (Quaero, 144)
Passage-réponse du document 1 :
Le développement des énergies renouvelables, tels **le solaire** ou **l'éolien** permettra d'équiper de façon optimale de petites unités, non reliées au réseau, à faible consommation.

Passage-réponse du document 2 :
Pour réduire sa dépendance vis-à-vis du pétrole, diversifier ses sources d'énergie et renforcer la lutte contre l'effet de serre, la France s'est fixée des objectifs et un calendrier ambitieux de développement des énergies renouvelables. Le pays doit atteindre d'ici 2010 l'objectif de 21 % pour la part des énergies renouvelables dans la consommation brute d'électricité. Pour atteindre cet objectif, la France mobilisera plusieurs filières : **l'hydroélectricité**, mais aussi notamment la **biomasse** et **l'éolien**.

Enfin, certaines questions-listes attendent des réponses complexes exprimées le plus souvent dans les documents par une proposition, rendant alors extrêmement difficile une fusion des réponses. Le premier exemple montre l'unique question-liste complexe de Quaero et le deuxième montre la difficulté sémantique qu'une question bien que factuelle peut parfois revêtir :

Question : *Quelles sont les 8 étapes pour la fabrication de la chaux ?* (Quaero ; 34)
Passage-réponse du document 1 :
La fabrication traditionnelle de la chaux se fait en trois étapes. À la première étape, **on vérifie l'état du four, on construit une voûte dans sa partie inférieure et on amène les matières premières (pierre et bois) à proximité**. À la deuxième étape, **on procède au chargement du four en introduisant les pierres à partir du sommet du four, on allume le feu et on l'alimente de façon à ce que la chaleur augmente progressivement**. **La cuisson du calcaire terminée, on laisse refroidir le tout pendant quelques jours**. La dernière étape consiste à **vider le four**, soit à enlever la cendre, **faire descendre les briques de chaux**, puis **les entreposer dans des barils à l'abri de l'humidité**.

Passage-réponse du document 2 :
Pour l'essentiel, les procédés de la chaux passent par les étapes fondamentales suivantes, illustrées à la Figure 2.3 : **extraction du calcaire**, **stockage et préparation du calcaire**, **stockage et préparation des combustibles**, **cuisson du calcaire**, **broyage de la chaux vive**, **hydratation et extinction de la chaux vive**, **stockage**, **manutention et transport**.

Question : *Quelles sont les 3 feuilles d'imposition à remplir l'année d'un mariage ?* (Quaero ; 13)
Passage-réponse du document 1 :
Dans le cas d'un mariage, le foyer fiscal n'est imposé dans son ensemble qu'à partir de la date du mariage, **pour la période précédente chacun des époux remplit une déclaration séparée**. Ainsi les mariés devront réaliser trois déclarations pour l'imposition d'une année.

Passage-réponse du document 2 :
Mariage et impôt sur le revenu.Déclaration de revenus. **Pour les revenus acquis du 1er janvier 1999 au jour du mariage, votre conjoint et vous devez remplir une déclaration séparément. Vous devez utiliser la déclaration préimprimée que chacun de vous a reçue**. **Pour les revenus acquis du jour de votre mariage au 31 décembre 1999, vous devez remplir une déclaration commune**. Où envoyer la déclaration ?. Vous devez adresser les trois déclarations (les deux individuelles et la déclaration commune) au centre des impôts du domicile conjugal.

CAS 3 : PLUSIEURS DOCUMENTS SONT NÉCESSAIRES POUR CONSTITUER LA RÉPONSE-LISTE. Ce cas ne concerne que les corpus médical et journalistique, et que pour un très petit nombre de questions (voir tableau 2.9). Respectivement trois et une questions nécessitent plusieurs documents pour obtenir toutes les réponses individuelles et ainsi reconstituer la réponse-liste, mais pour deux d'entre elles, il est toutefois impossible de trouver le nombre de réponses attendues en combinant toutes les réponses-listes (y compris dans le document référence). Cette faible fréquence d'apparition est ici sans conséquence puisque notre but est d'observer les phénomènes présents. Ainsi, l'exemple suivant montre comment le nombre de réponses attendues peut être atteint après agrégation de trois passages provenant de trois documents différents alors qu'il aurait été impossible de l'atteindre avec un seul document :

Question : *Citez 5 critères diagnostics de l'aniridie.* (Eq-Méd, ML126)
9 réponses dans trois documents (8 réponses après fusion)
Passage-réponse du document 1 :
L'aniridie s'accompagne fréquemment d'une **microcéphalie**, d'une **hypotonie**, d'un **retard mental**.

Passage-réponse du document 2 :
Pourront ainsi être dépistées : - les malformations : microphtalmies, anophtalmies, colobome de l'iris et/ ou colobome choroio-rétinien, aniridie (**absence congénitale de l'iris**, qui peut s'associer à une **tumeur de type néphroblastome ou gonadoblastome** d'où l'indication d'une échographie abdominale systématique)

Passage-réponse du document 3 :
L'aniridie se définit comme l'**absence totale d'iris**. On peut néanmoins observer de véritables **aniridies avec persistance de minuscules collerettes** ou **brides iriennes très partielles**.

CAS 4 : LES RÉPONSES INDIVIDUELLES SONT RÉPARTIES SUR PLUSIEURS PASSAGES-RÉPONSES DANS UN MÊME DOCUMENT. Le nombre de questions pour lesquelles un même document contient plusieurs passages-réponses est significativement important pour Eq-Méd et Quaero. Cela montre que la technique de collecte des snippets permet de se centrer sur un passage contenant des réponses mais qu'il est tout de même primordial de regarder également le document de façon plus globale.

	Eq-Méd	**Eq-Jour**	**Quaero**
Nombre de questions-listes total	24	30	31
Nombre de questions avec plusieurs passages-réponses dans un même document	**8 (33,33 %)**	4 (13,33 %)	**12 (38,71 %)**
Pour ces questions à plusieurs passages-réponses dans un même document :	**Eq-Méd**	**Eq-Jour**	**Quaero**
Nombre total de passages-réponses	**20**	13	**39**
Nombre moyen de passages-réponses par document	2,22	2,17	**3**
Nombre médian de passages-réponses par document	2	2	2
Nombre total de passages-réponses	54	**101**	124

Tableau 2.10: Nombre de passages-réponses par document.

Le tableau 2.10 montre que, concernant le nombre de réponses individuelles, si la médiane est identique pour les trois corpus, la moyenne montre toutefois une redondance légèrement plus importante dans les documents Quaero. Les questions-listes Quaero sont quasiment toutes factuelles et se prêtent plus à une redondance intra-document des réponses que les questions-listes complexes. Dans le corpus Quaero, la redondance d'une même réponse dans un document se produit fréquemment sur les sites commerciaux ou les pages dédiées à une entité. Par exemple, pour la question *Qui sont les 2 réalisateurs du film « No Country for the Old Men » ?* (Quaero, 227), le billet d'un blog consacré au film *No country for old men* comporte huit occurrences visibles de la réponse (*Ethan et Joel Cohen*) sur le document HTML interprété (voir figure 2.2) .

FIGURE 2.2 : Huit occurrences de la réponse dans le même document pour la question *Qui sont les 2 réalisateurs du film « No Country for the Old Men » ?* (Quaero, 227).

Cette redondance ne se retrouve toutefois pas dans le corpus Eq-Jour alors que lui-aussi ne comporte pourtant quasiment que des questions factuelles. On remarque également que le corpus Eq-Jour comporte la proportion la plus faible de questions pour lesquelles un document contient plusieurs réponses individuelles. On trouve tout de même quelques articles qui décrivent les réponses dans un passage puis les mentionnent à nou-

veau plus loin dans le document. Par exemple, avec les deux passages suivants extraits du même document, on constate que le titre de l'article (passage-réponse 1) donne une réponse individuelle, puis la réponse est développée dans un paragraphe suivant (passage-réponse 2) :

Question : *Quelles sont les trois sociétés possédant le capital de Air Inter ? (Eq-Jour, 377)*
Passage-réponse 1 :
VIE DES ENTREPRISES Désormais concurrente de l'avion La **SNCF** envisage de vendre les actions d'Air Inter qu'elle détient encore.
Passage-réponse 2 :
Actuellement **le holding d'Etat Groupe Air France SA** détient 72 % du capital d'Air Inter, les autres actionnaires étant, outre la **SNCF**, **le Crédit lyonnais**, **la Caisse des dépôts** et **les chambres de commerce**.

Le domaine médical tend également, et dans une proportion plus significative, à présenter d'abord un passage contenant des réponses d'un niveau général puis à les spécifier ensuite comme le montre l'exemple suivant dont les trois passages-réponses sont tous issus d'un même document :

Question : *Quelles sont les 4 localisations possibles des neuroblastomes ? (Eq-Méd, 134)*
Passage-réponse 1 :
Les cellules de neuroblastome peuvent s'installer notamment dans **les os** ou dans **la moelle osseuse**.
Passage-réponse 2 :
LOCALISATIONS DES NEUROBLASTOMES 10 % au niveau du **cou** 30 % au niveau du **thorax** 50 % au niveau du **ventre** 10 % au niveau du **pelvis**
Passage-réponse 3 :
Le schéma ci-contre montre la localisation du **système nerveux sympathique**. Les neuroblastomes se développent au niveau de ce système nerveux. C'est la raison pour laquelle les neuroblastomes se trouvent le plus souvent au niveau du **ventre** (sur une glande située au-dessus du rein : la **glande surrénale**, ou **de chaque côté de la colonne vertébrale**). Parfois, ils apparaissent au niveau du **thorax** et, plus rarement, au niveau du **cou**.

Ce document fournit donc douze occurrences de réponses individuelles et neuf instances de réponses après élimination des doublons (*ventre*, *cou* et *thorax*). À l'aide de connaissances spécialisées, les deux réponses *la glande surrénale* et *de chaque côté de la colonne vertébrale* peuvent être considérées comme équivalentes à la réponse *au niveau du ventre*. On voit ici l'intérêt de se poser la question de l'agrégation des réponses et des liens qui les relient.

2.1.7 *Synthèse*

L'étude de ces corpus de campagnes d'évaluation ayant proposé des questions-listes pour le français nous a donc montré plusieurs points problématiques en vue du cadre applicatif que nous visons :

- les questions-listes spécifient le nombre de réponses attendues : dans un cadre utilisateur, cette situation est peu probable puisqu'elle suppose que l'utilisateur connaît le nombre de réponses avant même de poser sa question. De plus, si cette donnée peut effectivement beaucoup aider les systèmes dans la recherche des réponses, elle peut également conduire à de mauvais résultats si elle est erronée ;
- les structures énumératives rencontrées ne sont pas toutes conformes à l'état de l'art, certaines semblent même très difficiles à implémenter, notamment pour des documents HTML ;
- l'aspect multi-document est quasiment absent, ce qui semble peu réaliste si l'on travaille sur le Web et aussi problématique pour effectuer des recoupements multi-documents à des fins de validation de réponse.

Ces observations nous ont donc conduit à la constitution d'un nouveau corpus de questions-listes et de documents y répondant afin de nous rapprocher des conditions utilisateurs et également de nous assurer d'un aspect multi-document des réponses.

2.2 FRITES : UN NOUVEAU CORPUS D'ÉTUDE

Devant le peu de questions nécessitant de recourir à un traitement multi-document dans les trois collections étudiées, nous avons décidé de constituer un autre corpus d'étude : le corpus *Frites*. L'objet de cette partie est donc de présenter ce corpus en commençant par sa constitution puis en présentant les résultats de son étude.

2.2.1 *Constitution et caractéristiques du corpus*

2.2.1.1 *Les questions*

Nous nous sommes d'abord appuyés sur les questions-listes existantes et propices à cet aspect multi-document recherché. Pour cela, nous avons repris sept questions-listes rencontrées dans EQueR et Quaero en les modifiant très légèrement pour certaines : suppression du nombre de réponses attendues (*Qui sont les ~~huit~~ personnages de « Disney Princess » ?*) ou remplacement de certains termes (*Quels pays étaient candidats à l'organisation de la coupe du monde ~~2006~~ 2018 ?*).

Nous avons ensuite imaginé des questions en rapport avec des thèmes porteurs de réponses multiples : par exemple, *Qui a incarné Batman ?* (plusieurs acteurs ayant interprété ce rôle et certains l'ont même fait à plusieurs reprises), *Quand la France a-t-elle perdu son*

triple-A ? (il y a trois principales agences « globales » de notation qui occupent 94 % du marché ; plus de 100 agences locales existent mais elles ne sont pas à vocation internationale).

Ce nouveau corpus d'étude se compose donc de 100 questions créées manuellement sur des thématiques variées (sport, santé, politique, culture, économie, informatique) et de plusieurs types (ceux définis par les campagnes d'évaluation) :

- factuelle : *Quand la France a-t-elle perdu son triple-A ?*
  L'ayant perdu chez deux agences de notation à deux dates distinctes, il y a donc deux réponses possibles ;
- liste : *Quels pays étaient candidats à l'organisation de la coupe du monde 2018 ?*
  Au fil des mois, plusieurs pays se sont désistés pour n'être candidats qu'à la coupe du monde 2022 ;
- complexe : *Comment a évolué la croissance française en 2011 ?*
  L'INSEE a publié plusieurs chiffres officiels concernant la croissance de la France en 2011, au moins un par trimestre ;
- booléenne : *Pluton est-elle une planète ?*
  La définition du terme *planète* a été modifiée il y a quelques années, requalifiant Pluton comme une *planète naine* et non plus comme une *planète* ;
- définition : *Qu'est-ce que la croissance ?*
  Il existe plusieurs définitions selon les domaines (économie, mathématique, biologie).

La répartition de chaque type de question est donnée dans le tableau 2.11.

Type de question	**Nombre total de questions**	**Nombre de questions nécessitant un traitement multi-documents**
Factuelle	61	11
Liste	17	2
Complexe	10	3
Booléenne	8	3
Définition	4	0
Total	100	19

Tableau 2.11: Répartition des questions dans le corpus *Frites*.

Sur le même modèle et dans les mêmes proportions, un corpus de test de 100 questions a été constitué afin de procéder plus tard à une évaluation de notre SQR une fois

nos développements implémentés.

On le voit ici, des questions de tout type peuvent avoir plusieurs réponses correctes. Nous définissons donc la notion de **question à réponses multiples** (ou **question-ARM**) qui définit une question pour laquelle plusieurs réponses peuvent être correctes. Cette notion inclut évidemment les questions-listes telles que définies dans les campagnes d'évaluation (par exemple, *Quels sont les sept astres du système solaire visibles à l'œil nu ? le Soleil, la Lune, Mercure, Vénus, Mars, Jupiter, Saturne*) mais aussi les autres types de question (par exemple les questions factuelles comme *Quand le Paris Saint-Germain a-t-il été sacré champion de France de football ? 1986, 1994, 2013 et 2014*).

2.2.1.2 *Les documents*

En utilisant trois moteurs de recherche sur Internet (Exalead [5], Bing [6] et Google [7]), nous avons collecté pour ces questions 232 fichiers au format HTML contenant au moins une réponse individuelle correcte. Chacune des requêtes aux moteurs de recherche a été construite manuellement avec les termes jugés importants de la question accompagnés parfois d'une réponse attendue. Certaines requêtes excluent même une réponse attendue afin de sonder l'aspect multi- document des questions et éviter ainsi de reproduire ce qui avait été constaté dans le corpus Quaero, à savoir beaucoup de documents HTML contenant déjà toutes les réponses.

Ainsi un document peut ne contenir qu'une seule réponse individuelle correcte et non pas forcément la réponse-liste. Si un document contenait une ou des réponses dans un tableau ou une liste, il était ajouté au corpus au même titre que les autres documents. Tout comme la collection Quaero, seul le document HTML a été récupéré (pas de fichier CSS, DTD ou image par exemple) et la collection a été traitée avec Kitten [Falco *et al.*, 2012] pour obtenir des fichiers textes.

Comme vu dans la section précédente, seules 19 questions nécessitent obligatoirement un traitement multi-document pour composer la réponse-liste : cette faible proportion s'explique notamment par le fait que quelques pages très pertinentes (Wikipédia par exemple) contenaient effectivement toutes les réponses individuelles. Nous avons décidé de les conserver car Wikipédia est d'une part une source d'informations non négligeable sur le Web et d'autre part, les réponses étaient réparties sur l'ensemble du document (souvent de très grande taille), et leur identification pourra nous servir de baseline pour mesurer les résultats sans traitement multi-document.

5. http://www.exalead.fr/search/
6. http://www.bing.com/
7. http://www.google.fr

2.2.2 *Observation du corpus Frites*

Nous avons commencé par étudier les paires question-réponses de notre nouveau corpus afin de mieux caractériser les réponses aux questions-ARM. Ceci nous a permis d'ensuite proposer une typologie des questions-ARM qui permettra de faciliter l'analyse automatique des questions et de mieux cibler les difficultés à résoudre pour pouvoir y répondre.

2.2.2.1 *Étude des paires question-réponse(s)*

Nous avons observé les passages-réponses de 81,74 % des documents du corpus (189 documents). Le recensement des phénomènes (voir tableau 2.12) a fait émerger plusieurs catégories de problèmes dont certains récurrents aux SQR :

- la résolution d'anaphore, la réconciliation de référence [Lee *et al.*, 2011], [Recasens *et al.*, 2010], [Hickl *et al.*, 2007], [Chali et Joty, 2007], [Zhao *et al.*, 2005], [Krzysztof *et al.*, 2005], [Yang *et al.*, 2003] :

  Question : *Quand la France a-t-elle perdu son triple-A ?*
  Passage : Fitch dégrade cinq pays de la zone euro. L'agence de notation financière abaisse notamment la note de la Belgique, de l'Italie et de l'Espagne. Elle maintient en revanche celle de l'Irlande.

- le type métaphorique de la formulation de réponse :

  Question : *Quand la France a-t-elle perdu son triple-A ?*
  Passage : Le triple A, c'est une ligne Maginot. Il ne fallait pas chercher à la défendre.

  Question : *Quand la France a-t-elle perdu son triple-A ?*
  Passage : A côté des deux cow-boys américains, l'agence Fitch Ratings ferait presque figure de jeune fille rangée.

- le besoin de contexte :

  Question : *Quand la France a-t-elle perdu son triple-A ?*
  Passage : Pour mémoire : les agences de notations avaient noté AAA des dizaines de milliers de "papiers" qui étaient du "Junk" dès le départ.

- la densité de candidats-réponses dans un court passage textuel :

Question : *Dans quelles villes Sarkozy a-t-il présenté ses voeux ?*
Passage : Enfin, l'Elysée organise la séquence des vœux de début d'année, qui seront délocalisés dans une dizaine de villes. « Il prépare son atterrissage en France après une longue séquence internationale », décrypte-t-on à l'UMP. Il débutera le 1er janvier à Metz, en hommage aux personnels mobilisés à la Saint-Sylvestre. Une équipe de l'Elysée sera dépêchée sur place dès le 31 décembre... avec prime à la clé. Deux jours plus tard, direction Brest pour les vœux aux forces armées. Les vœux aux forces économiques, très attendus en pleine menace sur le triple A, se dérouleront à Lyon. Sarkozy se rendra aussi à Lille, chez Martine Aubry, Mulhouse, Marseille, dans le fief de Jean-Pierre Raffarin en Poitou-Charentes et peut-être en Guyane.

– les faux candidats :

Question : *Quand la France a-t-elle perdu son triple-A ?*
Passage : En France nous avons le quintuple A (Association amicale des amateurs d'andouillette authentique).

– les réponses se trouvant dans des tableaux de données :

Question : *Dans quels clubs a joué Nicolas Anelka ?*
Passage-réponse :

Saison	Club	Pays
95-96	Paris-SG	FRA
96-97(fév)	Paris-SG	FRA
96-97	Arsenal	ANG
97-98	Arsenal	ANG
98-99	Arsenal	ANG
99-00	Real Madrid	ESP
00-01	Paris-SG	FRA
01-02(déc)	Paris-SG	FRA
01-02	Liverpool	ANG

– les informations incertaines (par exemple, des rumeurs ou des passages utilisant le conditionnel) car ce facteur sera à prendre en compte pour estimer la validation d'une réponse :

Question : *Comment a évolué la croissance française en 2011 ?*
Passage-réponse : La vingtaine d'analystes interrogés sur la semaine écoulée dans le cadre des enquêtes trimestrielles sur les pays du G7 estiment que la croissance du produit intérieur brut (PIB) ne devrait pas dépasser 1,5 % en 2011, un chiffre proche de celui attendu pour 2010.

– les chronologies narratives : plusieurs réponses sont réparties chronologiquement sur le document autour d'un thème, par exemple la carrière d'un joueur de football :

> **Question :** *Dans quels clubs a joué Nicolas Anelka ?*
> **Passage-réponse :** Après 1 an et demi passé au **PSG**, sans avoir réellement eu sa chance, Nicolas Anelka s'engage avec les Gunners d'**Arsenal**. Il garde de très bons souvenirs de son passage à Londres et de son coach Arsène Wenger (—) Après 2 saisons et demie, 65 matchs et 23 buts marqués sous les couleurs d'**Arsenal**, Nicolas Anelka décide de quitter Londres pour Madrid. Il rejoint le **Real de Madrid** lors de la saison 99/2000. Son arrivée ne fit pas l'unanimité dans le vestiaire et on ne le mit pas dans les meilleures conditions. (—) Après un retour d'un an et demi au **Paris SG** (2000-02), période durant laquelle il inscrira 19 buts en 54 matchs, Nicolas Anelka est prété à **Liverpool** (—) A la fin de son prêt, Liverpool ne lui proposera pas de contrat pour un transfert définitif et Nicolas rejoindra **Manchester City** (—) Alors âgé de 25 ans, Nicolas Anelka décide de s'engager avec le club de **Fenerbahce**. (—)

Phénomène	**Nb d'occurrences**
critère variant	82
formulation de la réponse	72
ancre référentielle à chercher	53
faux candidats	46
information incertaine	21
chronologie narrative	20
tableau de données	18
terminologie dans la question	17
indice d'expansion de requête	13
nombreux candidats-réponses dans le passage	12
besoin de contexte	12
type métaphorique de la réponse	11
terminologie dans la réponse	10

Tableau 2.12: Phénomènes recensés (non mutuellement exclusifs) les plus fréquents dans le corpus *Frites*.

Nous présentons ici les 3 phénomènes auxquels nous avons choisi de nous intéresser par la suite car ce sont les plus fréquents dans notre corpus.

Le phénomène le plus fréquent est la variation des réponses selon certains critères : un critère de précision (que nous appelons **critère variant**) d'un élément permet d'obtenir plusieurs réponses correctes. Ici, la note souveraine de la France dépend de l'agence de notation :

Question : *Quelle est la note de la France sur les marchés financiers ?*
Passage-réponse 1 : L'agence de notation américaine Egan-Jones a abaissé aujourd'hui la note attribuée à la dette de la France à "A", cinq crans en dessous du "triple A" des trois grandes du secteur, Standard and Poor's, Moody's et Fitch.(—). La France avait perdu son "AAA" chez cette agence en juillet.
Passage-réponse 2 : "Moody's a maintenu le triple A de la France, la meilleure note possible", annonçait le matin une dépêche AFP, aussitôt reprise par une partie de la presse française.
Passage-réponse 3 : Peu après 16 heures, ce vendredi, une source gouvernementale a indiqué que l'agence de notation financière Standard & Poor's avait bel et bien décidé de dégrader la France en lui retirant sa note d'excellence triple A.

Le problème de **la formulation de la réponse** est aussi un problème habituel des SQR : la réponse, par la synonymie ou la paraphrase, peut prendre plusieurs formes :

Question : *Qui a incarné Batman ?*
Passage-réponse 1 : Après avoir usé **Michael Keaton**, **Val Kilmer** et **George Clooney** dans le rôle de Batman, les spéculations sur le prochain vengeur masqué de Gotham City se poursuivent.
Passage-réponse 2 : Le réalisateur chinois Zhang Yimou a choisi pour son prochain film l'acteur britannique **Christian Bale**, qui a incarné Batman, pour jouer le rôle d'un prêtre héroïque durant le sac de Nankin par les troupes japonaises en 1937.

L'ancre référentielle est le phénomène nécessitant un besoin de rattachement à une date précise. En effet, le temps est un critère variant et les réponses correctes ne le sont parfois que par rapport à un moment temporel précis. Par exemple dans les trois passages-réponses suivants, les réponses nécessitent de trouver la date absolue à partir des indices temporels relatifs (en gras) pour pouvoir être validées :

Question : *Quand la France a-t-elle perdu son triple-A ?*
Passage-réponse 1 : L'agence de notation américaine Egan-Jones a abaissé aujourd'hui la note attribuée à la dette de la France à "A", cinq crans en dessous du "triple A" des trois grandes du secteur, Standard and Poor's, Moody's et Fitch.(—).La France avait perdu son "AAA" chez cette agence **en juillet**.
Passage-réponse 2 : Moody's a maintenu le triple A de la France, la meilleure note possible", annonçait **le matin** une dépêche AFP, aussitôt reprise par une partie de la presse française.
Passage-réponse 3 : **Peu après 16 heures, ce vendredi**, une source gouvernementale a indiqué que l'agence de notation financière Standard & Poor's avait bel et bien décidé de dégrader la France en lui retirant sa note d'excellence triple A.

2.2.2.2 *Typologie des questions-ARM*

Nous avons étudié les 61 questions factuelles de notre corpus et en avons catégorisé 47 comme des questions-ARM ; en effet, certaines ne le sont pas de façon certaine. La figure 2.3 présente la typologie obtenue. Les types de question-ARM définis et leur répartition dans notre corpus sont les suivants (les classes ne sont pas mutuelllement exclusives) :

- **(1) pluriel non-quantifié,** 27, 87 % : *Quels sont les pays de l'UE ?*
  L'UE compte 28 pays en juillet 2013 ;
- **(2) pluriel quantifié,** 1, 64 % : *Quelles sont les neuf planètes du système solaire ?*
- **(3) critère variant rationnel,** 73, 77 % : *Qui a gagné la Ligue des Champions en 2011 ?*
  Ici, la réponse varie selon les catégories sportives : le FC Barcelone l'a remporté dans la catégorie UEFA homme, l'Olympique Lyonnais dans celle UEFA femme et l'Espérance de Tunis dans celle CAF homme ;
- **(4) critère variant subjectif,** 3, 28 % : *Quelle superbe victoire a remporté la France en 1998 ?*
  L'équipe de France de football a remporté onze victoires durant l'année 1998, c'est l'interprétation de *superbe* qui permet de déduire la réponse ;
- **(5) réponse temporelle,** 24, 59 % : *Quel jour Nicolas Sarkozy est-il devenu président de la République ?*
  Il a été élu le 6 mai 2007 puis investi officiellement le 16 mai 2007 ;
- **(6) réponse liée au temps,** 18, 03 % : *Qui sont les Disney Princess ?*
  Leur nombre et composition évoluent depuis plusieurs années.
- **(7) ancre contextuelle à calculer,** 19, 67 % : *Quand fut fêté le bicentenaire de la révolution française ?*
  Le bicentenaire a été fêté en 1989 mais à différentes heures selon les événements (défilé militaire sur les Champs-Élysées, commémoration), et également à différentes dates dans les écoles ;
- **(8) pluriel caché,** 54, 1 % : *Où/Quand/À qui Sarkozy a-t-il présenté ses vœux 2012 ?*
  Sarkozy avait effectué plusieurs cérémonies de vœux dans plusieurs villes, à plusieurs dates et les adressait à des groupes différents : *À Lille le 12 janvier aux fonctionnaires, À Lyon le 19 janvier au monde économique...* ;

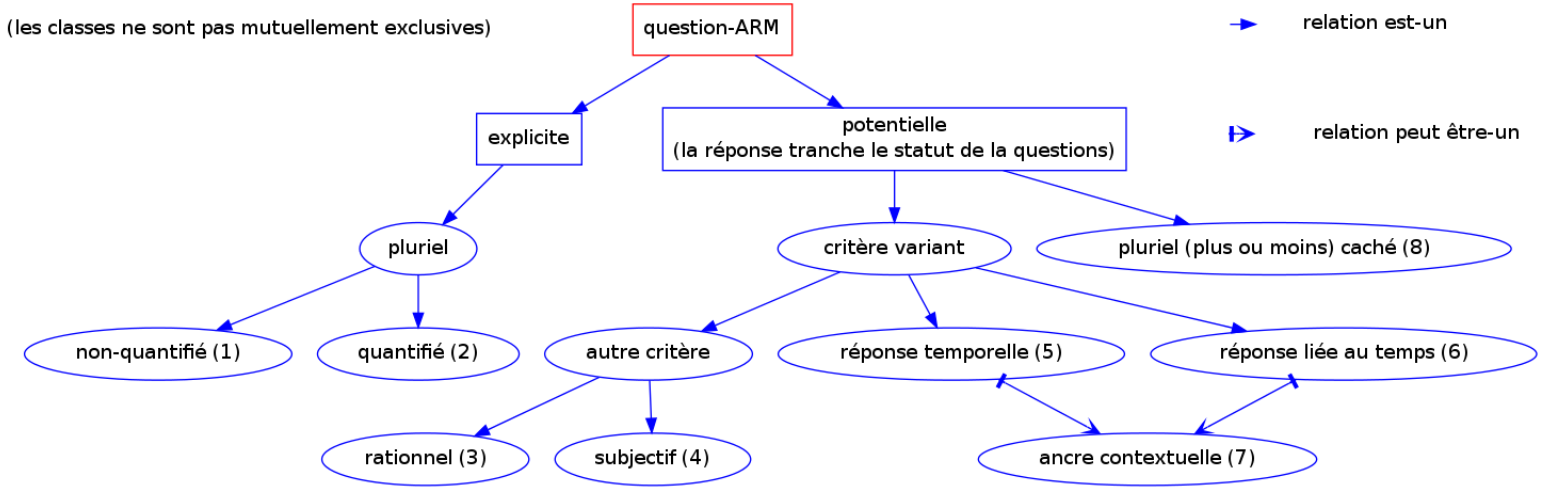

FIGURE 2.3 : Typologie des questions-ARM (questions à réponses multiples). Les chiffres correspondent aux exemples précédents.

Il est important de noter que ces classes ne sont pas mutuellement exclusives. Parmi les deux catégories principales, on trouve tout d'abord la catégorie *explicite* qui regroupe les questions comportant une marque de pluriel sur le focus de la question. La quantification indique incontestablement une pluralité tandis que la seule marque de pluriel ne sera toutefois pas fiable à cent pour cent comme le suggère l'exemple suivant : *Quels Jeux Olympiques se sont déroulés il y a 16 ans ?* où, bien qu'au pluriel, il n'y a *a priori* qu'une seule réponse correcte (on parle des jeux olympiques au pluriel pour désigner une seule édition et les jeux d'hiver et d'été n'ont pas lieu la même année).

La catégorie *potentielle* regroupe les questions potentiellement-ARM. La catégorie (8) regroupe les questions porteuses d'un pluriel possible du fait de la granularité de *qui*, *où* et *quand*. Elles peuvent potentiellement être des questions-ARM mais seules les réponses permettront au final de trancher.

Parmi les sous-catégories du *critère variant*, la catégorie (4) a été un peu forcée lors de la génération des questions, ce type de question subjectif est d'ailleurs exclu des campagnes d'évaluation [Fukumoto *et al.*, 2004] mais il nous intéresse car l'aspect subjectif apporte très fréquemment des réponses multiples : plusieurs personnes pourront penser à une autre superbe victoire remportée par la France en 1998 que celle contre le Brésil pendant la coupe du monde de football. Le critère variant peut être temporel ou plus général : les questions étant souvent courtes, le sens prototypique des concepts est fréquemment utilisé. Ainsi, parmi les exemples de questions illustrant les phénomènes de la typologie, la question de la catégorie (3) pour un français passionné de football fait communément référence à la Ligue des Champions de football masculine et européenne alors qu'aucun de ces deux termes n'est présent dans la question. La campagne EQueR [Ayache, 2005] avait d'ailleurs précisé que la question factuelle *Quel est le record du monde du 100 mètres ?* n'était pas précise puisqu'il pouvait s'agir de course à pied ou de natation ; il suffisait qu'un document justifie d'une réponse correcte pour qu'elle soit jugée correcte. De plus, au-delà de la catégorie sportive, le record évolue également en fonction du temps.

2.3 CONCLUSION

Nous avons vu dans ce chapitre que les campagnes d'évaluation des systèmes de question-réponse comportant des questions-listes étaient restrictives : (1) la forme des questions est peu naturelle puisqu'elles précisent toujours le nombre de réponses attendues, (2) les réponses doivent être justifiées par des passages trop courts pour pouvoir contenir des réponses multiples éventuellement exprimées sous forme de structure énumérative, et (3) l'aspect multi-document des réponses est peu présent. Nous avons également constaté que les structures énumératives que l'on trouve dans les documents HTML sont souvent peu conformes aux définitions de l'état de l'art.

L'étude de notre corpus de questions-ARM et de documents y répondant a confirmé la nécessité de mettre en place un traitement multi-document pour être en mesure de répondre le plus pertinemment possible à une question-ARM. Cette étude a également fait émerger, par leur fréquence, trois phénomènes caractéristiques des réponses multiples auxquels nous avons choisi de nous intéresser : le critère variant, la formulation de la réponse et l'ancre référentielle.

Dans le chapitre suivant, nous présentons les différents outils et hypothèses de travail issus de cette étude de corpus et qui vont permette de mettre en œuvre notre approche pour l'extraction et la validation multi-documents de réponses à des question-ARM.

3

PRÉSENTATION DES OUTILS ET HYPOTHÈSES DE TRAVAIL

Sommaire

Les SQR travaillant à partir d'une collection de documents textuels réalisent communément trois étapes : l'analyse de la question, la recherche des documents susceptibles de contenir la réponse par un moteur de recherche, l'extraction de la réponse [Kolomiyets et

Moens, 2011] ; une quatrième de validation des réponses s'imposant de plus en plus ces dernières années . Nous avons choisi de ne pas utiliser d'apprentissage automatique et de nous concentrer sur les méthodes symboliques pour l'extraction d'information. En effet, les approches par apprentissage automatique nécessitent de disposer de corpus annotés pour la phase d'apprentissage or il n'existe pas de tels corpus disponibles pour notre tâche et il serait trop coûteux de les construire nous-même.

Nous concentrons notre approche symbolique sur l'analyse syntaxique en dépendances pour plusieurs raisons. Nous n'utilisons pas de ressources sémantiques extérieures qui sont rarement disponibles pour le français et devons faire face à la difficulté de la variation syntaxique des formulations d'une réponse : la réécriture d'une phrase à l'aide de règles syntaxiques nous offre une solution à ce problème. De plus, nous disposons déjà du SQR FIDJI [Moriceau et Tannier, 2010] qui est fondé sur une approche syntaxique avec des résultat satisfaisants et que nous pourrons exploiter.

Nous présentons donc dans ce chapitre les outils et ressources qui sont utilisés en amont de *Citron*, notre système d'extraction de réponses multiples sur le Web, à savoir :

- Kitten, un outil de prétraitement de documents HTML,
- le moteur de recherche Lucene,
- l'analyseur syntaxique XIP,
- les types génériques d'items apparaissant dans des SE,
- le prétraitement de la Wikipédia utilisée pour la validation des réponses.

Enfin, nous présentons notre cadre de développement.

3.1 PRÉTRAITEMENT DES DOCUMENTS HTML AVEC KITTEN

Parmi les outils permettant de réaliser une extraction du texte d'un document HML, plusieurs fonctionnent par expressions régulières (par exemple, les scripts html2txt en Perl ou Python accessibles en ligne) ou par travail sur l'arbre HTML. Ces outils offrent un résultat rapide mais réalisent une extraction linéaire (BeautifulSoup [1]) ou une extraction du passage texte le plus dense (BoilerPipe [2]). Le navigateur Lynx [3] permet de créer dans un fichier texte conforme au rendu visuel d'un document HTML (voir figure 3.1) mais, tout comme l'extraction linéaire, nous verrons durant la présentation que Kitten que cette solution complique énormément la segmentation des phrases : nous n'utilisons cette extraction visuelle qu'en dernier recours, à savoir lorsque Kitten rencontre une exception durant le traitement du fichier HTML. BeautifulSoup possède quant à lui une méthode d'extraction linéaire du texte mais offre un cadre d'API Python permettant de travailler sur la structure d'arbre HTML.

1. http://www.crummy.com/software/BeautifulSoup/
2. https://code.google.com/p/boilerpipe/
3. http://lynx.isc.org

```
   #RSS FOAF Archive : tous les articles First Prev Next Last

Blog des Managers 2.0 (Intranet, SIC, 3D & co)

   B-R-ENT - Blog RSS et Entreprise
     * Actualités
     * Comment ça marche ?
     * Evènements
     * Fabrique du Futur - 6 nov 07
     * Innovation
     * Intranet, Blog et RSS
     * Virtual Life
     * Z Conf Stratégies 2007
     ____________________________________________________________

   S'identifier - S'inscrire - Contact

Discussions actives

     * Yulbiz Paris - et alors ... (5)
     * Intrablog 2.0 (2)
     * Voulons-nous apprendre à nos enfants à vivre comme avant ? (6)
     * Jury Prix Intrablog- Intraverse (1)
     * Participez à l'Odyssée de l'entreprise 2.0 (5)
     ____________________________________________________________

Recherche

   [Toutes rubriques_____________]
   __________ Chercher
     ____________________________________________________________

Vous et nous

   blogCloud

   [slot-frenchweb-180.jpg]
     ____________________________________________________________

Pour aller plus loin

   IFRAME:
   http://wwwdev.mob-it.com/mobthis/v2/3165/jerome223/8b3d-f9ec-37ab-ca76-
   d34d

   Un canal de com peut en imaginer un autre !
   Recommandé par des Influenceurs
   Effectuez vos paiements via PayPal : une solution rapide, gratuite et s
   écurisée
   Effectuez vos paiements via PayPal : une solution rapide, gratuite et s
   écurisée
     ____________________________________________________________
```

FIGURE 3.1 : Extraction d'un fichier HTML avec Lynx selon son rendu visuel.

Comme nous allons le voir, plutôt que de développer d'un côté un programme de normalisation de fichiers utilisant des API Java existantes et de l'autre un programme d'extraction de texte, nous avons préféré développer Kitten [Falco *et al.*, 2012] en y concentrant ces deux étapes. En effet, notre objectif étant de travailler en domaine ouvert, nous avons décidé de mettre en place une chaîne de traitement pouvant travailler à partir de fichiers HTML et pouvoir ainsi utiliser le Web comme ressource et collection de documents. Kitten avait simplement été mentionné dans les chapitres précédents comme un outil permettant d'obtenir des fichiers textes à partir de fichiers HTML, nous détaillons ici son architecture (voir figure 3.2) et son fonctionnement.

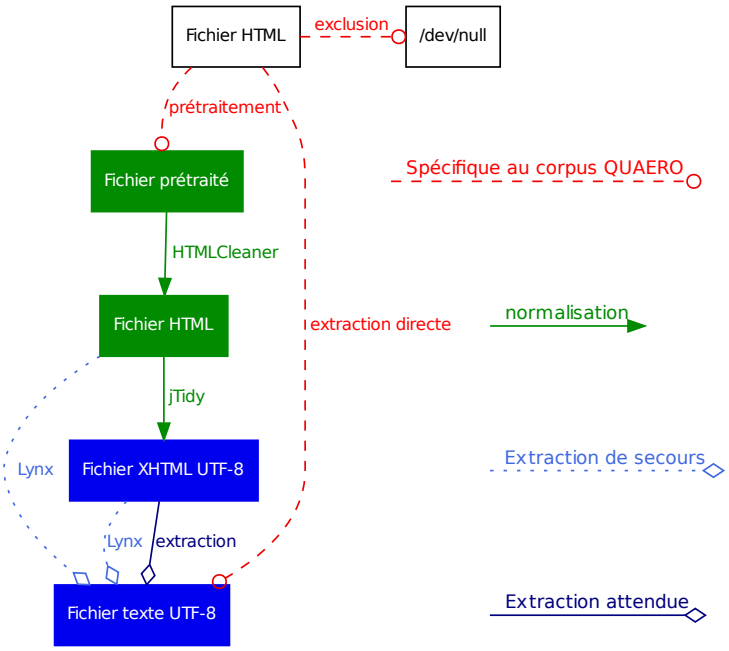

FIGURE 3.2 : Architecture de Kitten.

Kitten prend des documents HTML en entrée pour filtrer certains types de fichiers, normaliser ceux autorisés au format XHTML puis en extraire l'information dans un fichier texte UTF-8 le plus exploitable possible pour un analyseur syntaxique. L'extraction du contenu texte depuis les fichiers XHTML normalisés concernent principalement des segments de texte ainsi que l'extraction depuis les éléments structuraux que sont les énumérations verticales (marquées par les balises HTML pour les listes) et les tableaux (marqués par les balises HTML pour les tableaux).

3.1.1 *Filtrage et normalisation des documents HTML*

Il faut tout d'abord noter que Kitten n'a pas été développé pour détecter des fichiers SPAM et il va en conséquence extraire tout le contenu texte. Cette extraction peut toutefois être modulée par l'utilisateur, notamment concernant le choix des balises et attributs HTML à extraire ou rejeter.

A partir d'un ensemble de documents collectés sur le Web, Kitten applique un filtre uniquement sur le type de fichier, à savoir qu'il rejette les fichier binaires (Flash, format binaire propriétaire) ainsi que les fichiers « plan d'un site web » (sitemap). En effet, bien que ces derniers regorgent de fichiers de liens particulièrement utiles pour une tâche de classification de pages Web [Qi et Davison, 2009] ou de recherche d'information [Foucault, 2013], [de Groc, 2013], ils sont généralement extrêmement volumineux et très longs à traiter. Comme ils n'apportent que très peu pour de l'extraction d'information, Kitten élimine ce type de fichier. Dans le corpus Quaero français par exemple (499 736 fichiers), 0,003 % des fichiers sont des fichiers « sitemap ».

Les fichiers précédemment acceptés sont ensuite normalisés au format XHTML et encodés en UTF-8. Pour cela, Kitten utilise séquentiellement HTMLCleaner [4] [Nikic *et al.*, 2013] et jTidy [5] [Skarzhevskyy *et al.*, 2012] afin de corriger les erreurs HTML, principalement d'encodage et de structure.

Les erreurs d'encodage concernent notamment l'utilisation d'entités HTML codées « en dur » mais ne s'appliquant pas au format d'encodage déclaré, ainsi que l'utilisation d'entités valides en HTML mais invalides en XHTML (par exemple, ' doit notamment être remplacé par son caractère de simple apostrophe, de même que les entités numériques comprises entre € et Ÿ). La conversion en UTF-8 part de l'encodage détecté comme le plus probable par jChardet [6] [Castro et vdoss, 2013], un portage Java de l'algorithme de détection d'encodage utilisé par le moteur Mozilla [7].

Les erreurs de structure détectées concernent la structure d'arbre des balises qui peut être incorrecte (ordre d'ouverture et de fermeture des balises) ou incomplète. De plus, l'indentation éventuelle du code source HTML est supprimée afin d'éviter des ajouts de retours chariot coupant des segments de textes (contenu d'une balise paragraphe par exemple).

Cette étape de normalisation est très robuste puisque seuls cinq fichiers n'ont pu être normalisés sur le corpus Quaero.

4. http://htmlcleaner.sourceforge.net
5. http://jtidy.sourceforge.net
6. http://jchardet.sourceforge.net/index.html
7. http://www-archive.mozilla.org/projects/intl/chardet.html

3.1.2 *Extraction du contenu textuel*

La segmentation de texte est fondamentale pour un analyseur syntaxique qui a besoin de phrases correctement délimitées. Kitten ajoute donc un point si aucun symbole de fin de phrase (« . », « ? » ou « ! ») n'est présent à la fin d'un bloc de texte. Kitten reconstruit également les phrases en supprimant les retours chariot de formatage visuel. Par exemple, sur la figure 3.3, la balise de saut de ligne
 est utilisée pour imposer visuellement un retour à la ligne. Un être humain interprète selon le contexte ce retour à la ligne comme coupant ou ne coupant pas la phrase alors qu'une extraction linéaire du rendu HTML couperait la phrase au niveau du retour chariot.

FIGURE 3.3 : Utilisation de balise
 pour formater visuellement le rendu HTML.

Une extraction linéaire de ce fichier source insérerait donc un retour chariot après chacune des lignes. Par exemple, un extrait de la figure 3.3 produirait :

9 phrases avec une extraction linéaire :
Le secrétaire d'État à l'outre-mer,
ministre de l'intérieur par intérim,
Jean-Jack Queyranne
Le ministre de l'éducation nationale,
de la recherche et de la technologie,
Claude Allègre
Le ministre de l'économie,
des finances et de l'industrie,
Dominique Strauss-Kahn

alors que Kitten extrait 3 phrases reconstruites :

3 phrases avec Kitten :
Le secrétaire d'État à l'outre-mer, ministre de l'intérieur par intérim, Jean-Jack Queyranne. Le ministre de l'éducation nationale, de la recherche et de la technologie, Claude Allègre. Le ministre de l'économie, des finances et de l'industrie, Dominique Strauss-Kahn .

À l'aide d'expressions régulières, deux lignes séparées par un saut de ligne sont analysées pour décider du séparateur à éventuellement placer entre ces deux lignes ; elles sont rattachées sur une seule ligne quoiqu'il arrive :

- si la première des 2 lignes considérées se termine par un espace et que la deuxième commence par une majuscule, un point est inséré lors du rattachement. Par exemple : *Jean-Jack Queyranne* et *Le ministre de l'éducation nationale,*
- sinon si la seconde des 2 lignes considérées commence par une lettre minuscule ou, très rarement, par une ponctuation non terminale, alors les deux lignes sont fusionnées avec un espace comme séparateur. Dans l'exemple précédent :
  la ligne *Le secrétaire d'Etat à l'outre-mer,* et la ligne *ministre de l'intérieur par intérim,* sont fusionnées avec un espace.

Toutes les informations propres au HTML ne sont toutefois pas effacées puisque, selon les paramètres définis par l'utilisateur, les valeurs des attributs de balises peuvent être conservées, ce qui est fortement utile pour l'attribut *title* des balises <acronym> et <abbreviation> (la valeur de cet attribut apparaît en passant la souris sur le contenu de la balise). Pour la tâche de question-réponse qui nous intéresse, nous avons constaté que les attributs *alt* et *title* des balises <img> et <a> généraient trop de bruit et nous ne les avons donc pas conservés :

Exemple d'attribut *title* pour la balise <acronym> :
- *code source HTML :*
  Mozilla a étendu l'idée d'utiliser un navigateur pour avoir accès aux applications en reprenant certaines des technologies utilisées pour créer des sites Web, comme <acronym lang="en" **title="Cascading Styles Sheets"**>CSS</acronym> et JavaScript.
- *extraction par Kitten :*
  Mozilla a étendu l'idée d'utiliser un navigateur pour avoir accès aux applications en reprenant certaines des technologies utilisées pour créer des sites Web, comme CSS **(Cascading Styles Sheets)** et JavaScript.

Exemple d'attribut *title* bruité pour la balise <a> :
- <a **title="Afficher le lien vers ce message"** href="#">#1

Exemples d'attribut *alt* bruités pour la balise <img>
- les smileys : <img src="icon_redface.gif" **alt="Embarrassed"**>, <img src="../icon_sad.gif" **alt=" :("**>
- les boutons de menus :

Kitten génère ainsi un fichier texte sans balisage concernant les phrases : la seule balise présente est la balise <titleKitten> qui contient le contenu de la balise HTML <title> : cette information servira à Citron pour mesurer la pertinence et l'importance d'un document lors de la recherche de documents avec Lucene.

3.1.3 *Extraction du contenu depuis un tableau*

Pour extraire correctement le contenu d'un tableau, il est préférable de savoir si ce tableau est un tableau de données ou un tableau utilisé à des fins de formatage.

3.1.3.1 *Catégorisation de tableaux et de cases par apprentissage automatique*

Nous avons vu dans les chapitres précédents que les tableaux ne pouvaient être extraits linéairement depuis le code HTML puisque le code est linéaire (lu de gauche à droite et de bas en haut) alors que l'interprétation d'un tableau est normalement guidée par les cases entêtes. Kitten traite les tableaux en deux temps à l'aide de deux arbres de décision inspirés de [Wang et Hu, 2002] : tout d'abord il détermine s'il s'agit d'un tableau de données ou d'un tableau de formatage, puis, dans le cas d'un tableau de données, il identifie chaque case selon plusieurs catégories (principalement entête et donnée).

Un tableau de formatage (voir exemple de la figure 3.4) utilise la structure d'un tableau pour disposer visuellement des objets sur une page Web. Cette utilisation requiert fréquemment une utilisation récursive de la balise <table> et c'est pourquoi Kitten ne cherche des tableaux de données que sur une balise <table> ne contenant pas elle-même

FIGURE 3.4 : Exemple d'utilisation de la structure tableau à des fins de formatage.

d'autre(s) balise(s) <table>. Les tableaux de formatage sont eux extraits linéairement.

Quand un tableau est identifié comme tableau de données (figure 3.5, les cellules sont identifiées parmi six catégories :

- theme : titre du tableau ou information concernant toutes les cases du tableau ;
- entete : case explicitant sémantiquement une ou plusieurs cases données ;
- donnees : case contenant une information prédisposée à être reliée à une case entête. Une case données peut être vide ;
- neutre : case vide utilisée pour le formatage (et non suite à une absence d'information par rapport à une case entete) ;
- enteteFinaleInformative : information concernant toutes les cases du tableau mais située en bas de tableau ;
- enteteFinaleNonInformative : doublon d'une case entête. Ce cas se produit souvent pour les longs tableaux dont les entêtes sont situés sur une ligne horizontale : cette même ligne d'entête se retrouve dupliquée en bas de tableau.

Éric Cantona

Éric Cantona au Festival de Cannes 2009

Biographie	
Nom	Éric Daniel Pierre Cantona
Nationalité	France
Naissance	24 mai 1966 (47 ans)
Lieu	Marseille (Bouches-du-Rhône)
Taille	1,88 m (6′ 2″)
Période pro.	1983-1997
Poste	Attaquant

Parcours junior	
Saisons	**Club**
	Caillols
1981-1983	AJ Auxerre

Parcours professionnel [1]

Saisons	Club	M.	(B.)
1983-1988	AJ Auxerre	94	(29)
1985-1986	→ Martigues	15	(4)
1988-1991	Marseille	43	(14)
1989	→ Bordeaux	12	(6)
1989-1990	→ Montpellier	39	(14)
1990-1991	Nîmes Olympique	19	(4)
1991-1992	Leeds United	35	(13)
1992-1997	Manchester United	188	(82)
Total		**445**	**(166)**

FIGURE 3.5 : Extrait d'un tableau identifié comme un tableau de données (capture d'écran de l'article *Éric Cantona* sur Wikipédia.

Pour illustrer ces catégories, nous avons volontairement choisi un tableau dont l'identification n'a pas été parfaite. Le tableau de la figure 3.6 analysé par Kitten comprend :

- une case theme *Éric Cantona* qui est le titre du tableau ;
- une case enteteFinaleInformative comprenant une image et une description *Éric Cantona au Festival de Cannes 2009*. Il s'agit d'une erreur du point de vue de sa localisation (elle n'est pas en bas d'un tableau) mais correspond bien à une information concernant toutes les cases du tableau. Le schéma d'annotation du corpus de tableaux n'avait pas été suffisamment précis pour ce type de case comme nous allons le voir juste après dans la partie évaluation ;
- des cases entete : la première partie du tableau les place verticalement à gauche (*Taille*) et les parties suivantes du tableau les placent horizontalement sur une ligne complète (*Club*). Certaines cases occupent par fusion la totalité de la ligne comme pour *Parcours Professionnel*.
- une première case vide considérée à tort comme neutre : il s'agissait d'une case *donnees* et cette erreur a sûrement conduit à l'erreur suivante considérant la case *1981-1983* comme une case *entete* ;
- une deuxième case vide considérée à raison comme neutre : il n'y a pas de données en relation avec l'entête *Club*. Par contre la case *Total* aurait dû être identifiée comme entête.

FIGURE 3.6 : Exemple d'identification de cases d'un tableau de données par Kitten.

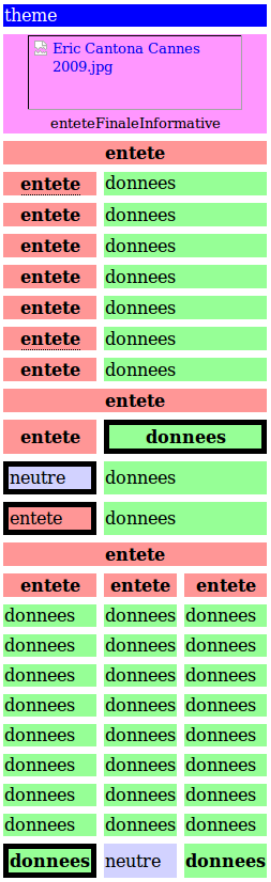

FIGURE 3.7 : Transcription des types trouvés par Kitten dans la figure 3.6. (les quatre erreurs sont entourées).

DESCRIPTION DU CORPUS D'APPRENTISSAGE. Pour pouvoir apprendre l'arbre de décision, nous avons constitué un corpus d'apprentissage. Nous avons repris une partie du corpus annoté de Wang et Hu [2002] [8] (118 documents) et non la totalité. En effet, il existait plusieurs instances d'un même site et elles comportaient des tableaux quasiment identiques. Afin d'éviter un surentraînement, nous n'avons gardé qu'une seule page par site. Nous avons également annoté des tableaux du corpus Quaero (294 documents) et des tableaux issus de Wikipédia (7 documents). En tout, 419 documents ont donc été annotés (type des tableaux et type des cases) (voir tableaux 3.1 et 3.2).

Au final, la phase d'apprentissage s'est déroulée sur 1 322 tableaux (661 de données et 661 tableaux de formatage choisis aléatoirement) par 10 validations croisées. Pour être

8. http://gsl.lab.asu.edu/doc/webtable.html

Corpus	**Nombre de documents annotés**	**Nombre de tableaux**	
		donnees	**formatage**
Corpus de Wang et Hu [2002]	118	282	548
Quaero	294	361	2 297
Wikipédia	7	57	59
Total	419	700 (661)	2 904 (661)

Tableau 3.1: Nombre de tableaux annotés. Entre parenthèses se trouve le nombre utilisé pour l'apprentissage.

Corpus	**Nombre de cases annotées**					
	entete	**donnees**	**neutre**	**entete Finale Informative**	**entete Finale Non Informative**	**theme**
Corpus de Wang et Hu [2002]	1 935	14 403	8 281	52	0	71
Quaero	2 366	15 980	17 954	20	17	63
Wikipédia	375	2 591	589	3	0	26
Total	4 676 (4 638)	32 974 (31 297)	2 317 (2 317)	75 (75)	17 (17)	160 (153)

Tableau 3.2: Nombre de cases annotées. Entre parenthèses se trouve le nombre utilisé pour l'apprentissage.

pris en compte dans l'apprentissage, un tableau devait comporter au moins deux lignes et au moins deux colonnes.

Nous avons utilisé WEKA [Hall *et al.*, 2009] pour les deux arbres de décision sur ces données sources (nous avons utilisé sa fonction d'échantillonnage de données par instance). Nous avons testé deux stratégies pour la catégorisation d'un tableau et trois pour la catégorisation d'une case.

CATÉGORISATION D'UN TABLEAU. Pour catégoriser un tableau en tableau de données ou de formatage, nous avons testé deux approches. La première utilise les traits propres aux tableaux à savoir :

– les traits de Wang et Hu [2002],
– le nombre de balises <th> (balises entêtes),
– le taux d'images dans le tableau par rapport au nombre de cases,
– le taux de liens dans le tableau par rapport au nombre de cases.

La deuxième stratégie testée utilise les traits de la première auxquels s'ajoutent le taux de chacun des types de cases (entete, donnees, neutre, enteteFinaleInformative, enteteFinaleNonInformative, theme). Ces taux sont calculés après étiquetage des cases par l'arbre destiné à cet usage. Notre but était de vérifier si un tableau de formatage étiqueté comme un tableau de données allait générer des taux discriminants pour certains types de case (neutre notamment).

Les traits les plus pertinents dans la catégorisation d'un tableau sont :
– la visibilité de la bordure du tableau ;
– la longueur moyenne du texte dans les cases ;
– le taux de cases vides ;
– l'utilisation de la balise <th>.

CATÉGORISATION D'UNE CASE. Pour la catégorisation d'une case entête ou donnée, nous avons utilisé des traits comme l'homogénéité (Wang et Hu [2002]) :
– l'homogénéité de la longueur du texte d'une case par rapport aux cases de sa ligne/colonne (LT_i). Par exemple, pour une case d'une ligne i dont la longueur de texte est lt_i et la longueur moyenne du texte des cases de sa ligne est m_i :

  $LT_i = 0,5 - min(\frac{|lt_i - m_i|}{m_i}, 1.0)$
– l'homogénéité de la hauteur d'une case par rapport aux cases de sa ligne/colonne ;
– l'homogénéité des attributs de la case par rapport à sa ligne/colonne. On pondère chaque attribut de la case par son nombre d'occurrences dans la ligne/colonne :

  $\frac{1}{\sum p_x} \sum d_x p_x$

  où p_x est le nombre d'occurrences de l'attribut sur la ligne/colonne et où d_x vaut 1 si la case possède l'attribut a_x, 0 sinon .
– le type de case dominant sur la ligne/colonne ;
– le taux de cases fusionnées sur la ligne/colonne.

Nous avons également utilisé les traits suivants :
– le numéro de ligne/colonne est transformé en pourcentage du nombre de lignes/-colonnes total du tableau ;
– la case est-elle vide ? fusionnée ? en gras ? se termine-t-elle par une ponctuation ? ;
– la ligne/colonne de cette case possède-t-elle une case entête ? ;
– la case est-elle sur la première/dernière ligne/colonne ;
– le nombre de liens, d'images, de balises de saut de ligne (
, <p>) dans la case.

Nous avions intégré un trait prenant en compte le taux de confiance lors de la catégorisation du tableau en tant que tableau de données, ainsi que le taux de confiance d'identification de la case mais ces scores étaient toujours quasiment à 1. Ils n'ont donc pas été conservés.

Les traits les plus pertinents dans la catégorisation d'une case d'un tableau de données sont :

- l'homogénéité de la hauteur d'une case par rapport aux cases de sa ligne/colonne ;
- l'homogénéité des attributs de la case par rapport à sa colonne ;
- le taux de cases fusionnées sur la ligne.

Nous avons testé trois stratégies de catégorisation des cases :

- stratégie sans contexte : utilisation uniquement des traits de la case à catégoriser ;
- stratégie avec 4 contextes : quatre traits sont ajoutés :
  - catégorie de la case au-dessus de la case à catégoriser,
  - catégorie de la case en-dessous de la case à catégoriser,
  - catégorie de la case à droite de la case à catégoriser,
  - catégorie de la case à gauche de la case à catégoriser.

  Il y a donc eu une première catégorisation de toutes les cases du tableau puis une seconde qui utilise les traits déjà calculés ainsi que ces traits contextuels ;
- stratégie avec 8 contextes : lors de la seconde catégorisation, le contexte entier de la case est pris en compte à savoir les quatre cases décrites précédemment auxquelles s'ajoutent les cases au-dessus à gauche/au-dessus à droite/en-dessous à gauche/en-dessous à droite.

Par exemple sur la figure 3.8, pour l'identification de la case *Leeds United*, la stratégie avec 4 contextes utilisera les 4 cases en jaune (*Nîmes Olympique*, *1991-1992*, *35 (13)* et *Manchester United*) et la stratégie avec 8 contextes utilisera les 4 cases en jaune et les 4 cases en rose (*1990-1991*, *19 (4)*, *1992-1997* et *188 (82)*).

1990-1991	Nîmes Olympique	19 (4)
1991-1992	Leeds United	35 (13)
1992-1997	Manchester United	188 (82)

FIGURE 3.8 : Exemple d'utilisation du contexte des 4 cases (en jaune) et 8 cases (en jaune et rose) environnantes pour la case *Leeds United*.

3.1.3.2 *Évaluation de la catégorisation de tableaux et cases.*

CATÉGORISATION D'UN TABLEAU. Il s'agit ici d'évaluer la catégorisation d'un tableau en tableau de données ou de formatage.

Nous avons testé différentes combinaisons concernant le nombre de catégories de cases à reconnaître. En effet, l'annotation a révélé peu d'instances pour les catégories theme, enteteFinaleInformative et enteteFinaleNonInformative et peut conduire à des erreurs d'apprentissage. Afin de réduire les types, nous avons considéré les cases theme, enteteFinaleInformative et enteteFinaleNonInformative comme des cases entete.

Les résultats obtenus (voir tableau 3.3) montrent que la deuxième stratégie (utilisation de l'étiquetage des cases avec 6 types possibles) apporte un gain de 0,013 points (1,38 %) sur notre corpus par rapport à la première stratégie (pas d'étiquetage de case). Ce gain correspond à 21 tableaux de données supplémentaires reconnus (sur 668 tableaux de données[9]) et semble confirmer notre intuition. Toutefois, ce gain restant modeste, nous avons préféré rester pour le moment sur la première stratégie s'en tenant aux traits propres aux tableaux pour des raisons de temps de calcul. De nouvelles expériences seront menées afin de confirmer cette voie d'amélioration.

Type	**Précision**	**Rappel**	**F-Score**
formatage (pas d'étiquetage de case)	0,937	0,950	**0,943**
données (pas d'étiquetage de case)	0,950	0,937	**0,943**
formatage (2 types de cases : E, D)	0,950	0,950	0,950
données (2 type de cases : E, D)	0,951	0,951	0,951
formatage (3 types de cases : E, D, N)	0,947	0,951	0,949
données (3 types de cases : E, D, N)	0,952	0,948	0,950
formatage (4 types de cases : E, D, N, T)	0,941	0,941	0,941
données (4 types de cases : E, D, N, T)	0,942	0,942	0,942
formatage (6 types de cases : E, D, N, T, EFI, EFNI)	0,967	0,942	**0,956**
données (6 types de cases : E, D, N, T, EFI, EFNI)	0,945	0,969	**0,956**
A titre indicatif : score de Wang et Hu [2002] sur leur corpus	0.942	0.973	**0.957**

Tableau 3.3: Catégorisation des 661 tableaux de formatage et des 661 tableaux de données. E : entete, D : donnees, N : neutre, T : theme, EFI : enteteFinaleInformative, EFNI : enteteFinaleNonInformative

CATÉGORISATION D'UNE CASE. Nous ne nous intéressons principalement qu'aux cases entêtes et données, c'est pourquoi le tableau 3.4 ne présente les résultats que pour ces

9. Le nombre diffère des 661 de départ suite à l'échantillonnage effectué (le pourcentage est calculé sur le nombre d'instances de tableaux de données échantillonnés).

deux catégories. Notre intuition sur l'aide que pouvait apporter le contexte d'une case n'a pas été confirmée sur notre corpus. Pour des raisons de temps d'exécution, nous utilisons donc la catégorisation des cases sans contexte. De plus, nous avons constaté expérimentalement que l'utilisation de deux types (entete et donnees) semblaient donner les meilleures résultats globalement mais que celle à quatre types (entete, donnees, neutre, theme) donnaient de meilleurs résultats sur les tableaux pertinents, notamment ceux de Wikipédia.

Types reconnus	E, D		E, D, N		E, D, N, T		E, D, N, T, EFI, EFNI	
	E	D	E	D	E	D	E	D
Sans contexte								
Précision	0,976	0,995	0,975	0,995	0,978	0,994	0,972	0,995
Rappel	0,960	0,997	0,970	0,996	0,968	0,997	0,971	0,997
F-Score	0,970	0,996	**0,973**	0,996	**0,973**	0,995	0,971	0,996
4 contextes								
Précision	0,976	0,995	0,974	0,996	0,978	0,994	0,972	0,995
Rappel	0,965	0,997	0,972	0,996	0,968	0,997	0,972	0,997
F-Score	0,971	0,996	**0,973**	0,996	**0,973**	0,996	0,972	0,996
8 contextes								
Précision	0,974	0,995	0,975	0,995	0,978	0,994	0,973	0,995
Rappel	0,966	0,996	0,970	0,996	0,967	0,997	0,972	0,997
F-Score	0,970	0,996	0,972	0,996	**0,973**	0,995	0,972	0,996

Tableau 3.4: Catégorisation des cases pour les 661 tableaux de données.
E : entete, D : donnees, N : neutre, T : theme, EFI : enteteFinaleInformative, EFNI : enteteFinaleNonInformative

L'évaluation des deux types de classification (tableaux et cases) a donc donné des résultats satisfaisants mais, malgré une diversification très forte des tableaux annotés manuellement pour le corpus d'apprentissage, certains tableaux ne sont pas correctement extraits.

3.1.3.3 *Extraction à partir d'un tableau de données*

L'extraction du contenu d'un tableau reconnu comme un tableau de données est guidée afin de relier les cases entêtes aux cases données selon une syntaxe prédéfinie facilitant ainsi le repérage et l'analyse syntaxique des informations. Le patron d'extraction

utilisé est le suivant :
(theme ? ; ; ? entete* ;* entete ; donnees / ?)+

Ce patron utilise la syntaxe des expressions régulières dans sa présentation :

- ? signifie 0 ou 1 occurrence : si un tableau possède plusieurs cases *theme* alors leurs contenus sont concaténés ;
- + signifie une ou plusieurs occurrences du contenu délimité par les parenthèses.

Ce patron associe donc toujours une case *donnees* avec une ou plusieurs cases *entete*, ce qui est notamment le cas dans le tableau 3.6 avec par exemple l'entête *Parcours professionnel* précédant *SaisonsPro*, *Club* et *M.(B)*.

Enfin, la présence des symboles « ; ; » et « / » a pour but de permettre la mise en place de règles syntaxiques pour analyser correctement l'extraction textuelle du tableau. En effet, nous verrons plus loin dans ce chapitre que l'analyseur syntaxique XIP que nous utilisons permet une telle configuration par la conception de règles d'analyse.

Ce patron d'extraction ne contient naturellement pas les cases neutres et enteteFinaleNonInformative mais il ne contient pas non plus les cases enteteFinaleInformative. En effet, la confrontation aux données réelles nous a montré qu'il valait mieux ne pas l'extraire : il s'agit souvent de légende d'image (comme dans l'exemple de la figure 3.6 avec la photo du festival de Cannes). Kitten l'identifie donc mais ne l'extrait pas.

Par exemple, un extrait du contenu du tableau de la figure 3.6 est donné ici :

```
(—)
Éric Cantona ; ; Biographie ; Période pro. : 1983-1997 .
Éric Cantona ; ; Biographie ; Poste : Attaquant .
Éric Cantona ; ; Parcours junior ; SaisonsJunior : Club .
Éric Cantona ; ; Parcours junior ; SaisonsJunior : Caillols .
Éric Cantona ; ; Parcours junior ; 1981-1983 : AJ Auxerre .
Éric Cantona ; ; Parcours professionnel 1 ; SaisonsPro : 1983-1988 / Éric Cantona ; ;
Parcours professionnel 1 ; Club : AJ Auxerre / Éric Cantona ; ; Parcours professionnel 1 ;
M. (B.) : 094 0(29) .
Éric Cantona ; ; Parcours professionnel 1 ; SaisonsPro : 1985-1986 / Éric Cantona ; ;
Parcours professionnel 1 ; Club : Martigues / Éric Cantona ; ;
Parcours professionnel 1 ; M. (B.) : 015 00(4) . (—) Éric Cantona ; ; Parcours professionnel
1 ; SaisonsPro : Total / Éric Cantona ; ;
Parcours professionnel 1 ; Club : / Éric Cantona ; ;
Parcours professionnel 1 ; M. (B.) : 445 (186) .
```

Nous verrons dans le chapitre suivant comment Citron analyse ces données.

3.1.4 *Extraction du contenu depuis une énumération verticale (liste)*

Kitten extrait les listes codées à l'aide des balises <ul> (liste non ordonnée), <ol> (liste ordonnée) ou <dl> (liste de définition). Ces balises HTML codent une énuméra-

tion verticale que Kitten va « aplanir » pour en faciliter l'analyse syntaxique. Les listes codées « en dur » à l'aide de retour chariot ou de succession de balises <p> sont extraites linéairement et restent donc des énumérations verticales.

Les énumérations verticales sont difficiles à analyser syntaxiquement du fait des retours chariot et Kitten les transforme donc en énumérations horizontales ou en une séquence de phrases complètes syntaxiquement. L'aplanissement d'une liste consiste donc à transformer une énumération verticale en énumération horizontale ou en une séquence de phrases autonomes où chaque item de la liste est relié à l'amorce.

Deux cas sont possibles selon la configuration de l'amorce de la SE :

- soit l'amorce est incomplète syntaxiquement : l'amorce est alors répétée devant chacun des items pour former autant de phrases que d'items ;
- soit l'amorce est complète syntaxiquement : les retours chariot sont alors remplacés par de la ponctuation, une virgule ou un point selon la longueur de l'item.

Voyons comment Kitten aplanit ou séquence les énumérations verticales et délimite deux informations fondamentales pour un moteur de recherche et pour Citron : l'amorce avec la balise <SEamorce> et la structure énumérative complète avec la balise <structEnum>.

3.1.4.1 *Séquencement : cas d'une amorce incomplète syntaxiquement*

Dans le cas d'une amorce incomplète, on se retrouve avec une dépendance manquante dans l'amorce puisque celle-ci est reliée syntaxiquement à chacun des items. Il devient alors indispensable pour obtenir une analyse en dépendances correcte de rattacher l'amorce à chacun des items en effectuant au passage les modifications de ponctuation nécessaires. Nous considérons qu'une amorce est incomplète syntaxiquement dans les cas suivants :

- l'amorce se termine par une préposition ou un verbe ;
- chacun des items commence par une préposition, par un article indéfini ou par un même mot.

Kitten s'appuie sur l'analyse en dépendances de la phrase repérée comme introduisant une liste HTML et de chacun des items. Cette analyse en dépendances est réalisée à l'aide de l'analyseur syntaxique XIP que nous présenterons dans la section 3.3. Ainsi, les quatre items de l'énumération verticale suivante vont amener à construire quatre phrases :

Question : *Quelles sont les architectures possibles d'un système de télécommunications ?*
Énumération verticale (dans le document source HTML) :
Un système de télécommunications peut avoir une architecture :
– **de** type « point à point », comme par exemple un câble hertzien ou optique, ou une liaison radiotéléphonique. Des répéteurs peuvent y être inclus pour amplifier et corriger les signaux ;
– **de** « diffusion », comme en télévision où un émetteur est reçu par des milliers de récepteurs ;
– **de** « collecte », comme en surveillance océanographique, où des centaines de capteurs sont reçus par un système central ;
– **en** structure de réseau, où un ensemble d'émetteurs et de récepteurs communiquent entre eux par des liaisons « étoilées » (topologie en étoile) ou « point à point ».

Séquencement en quatre phrases (dans le document texte en sortie de Kitten) :
<structEnum>
<SEamorce>*Un système de télécommunications peut avoir une architecture* </SEamorce> de type « point à point », comme par exemple un câble hertzien ou optique, ou une liaison radiotéléphonique. Des répéteurs peuvent y être inclus pour amplifier et corriger les signaux.
<SEamorce>*Un système de télécommunications peut avoir une architecture*</SEamorce> de « diffusion », comme en télévision où un émetteur est reçu par des milliers de récepteurs.
<SEamorce>*Un système de télécommunications peut avoir une architecture*</SEamorce> de « collecte », comme en surveillance océanographique, où des centaines de capteurs sont reçus par un système central.
<SEamorce>*Un système de télécommunications peut avoir une architecture* </SEamorce> en structure de réseau, où un ensemble d'émetteurs et de récepteurs communiquent entre eux par des liaisons « étoilées » (topologie en étoile) ou « point à point ».
</structEnum>

Nous avons choisi de rattacher chacun des items à l'amorce dans une phrase complète syntaxiquement plutôt que de créer une énumération intra-phrastique de tous les items du fait que les items de ce type d'énumération sont souvent extrêmement longs et seraient sources d'erreurs lors d'une analyse syntaxique.

Nous verrons au chapitre suivant comment Citron gère ce type d'énumération, notamment dans le cas où l'amorce précise le nombre d'items qu'elle introduit.

3.1.4.2 *Aplanissement : cas d'une amorce complète syntaxiquement*

Lorsqu'une amorce est complète syntaxiquement, Kitten transforme l'énumération verticale en énumération horizontale en se basant sur le critère de la longueur des items. Kitten remplace le retour chariot soit par un espace ou un point, soit par une virgule :

– la longueur médiane des items est inférieure à 60 caractères [10], alors une virgule est insérée pour délimiter les items (cas 1). Si un item se termine par un symbole de ponctuation, ce symbole est alors supprimé.
– la longueur médiane est supérieure à 60 caractères, alors un point-virgule est inséré pour délimiter les items (cas 2). Si un item se termine par une ponctuation finale, cette dernière est supprimée.

Les deux exemples suivants montrent l'extraction réalisée par Kitten dans ces deux cas :

(Cas 1) Longueur médiane des items inférieure à 60 caractères
Transformation en une phrase avec ajout de virgules pour séparer les items :

Énumération verticale (dans le document source HTML) :
Voici une liste des aéroports allemands avec plus de 1 000 000 passagers par an :
– Aéroport de Francfort-sur-le-Main
– Aéroport international Franz-Josef-Strauss de Munich
– Aéroport international de Düsseldorf
– Aéroport de Berlin-Tegel, Aéroport de Hambourg

Aplanissement en énumération horizontale (dans le document texte en sortie de Kitten) :
<structEnum>
<SEamorce>Voici une liste des aéroports allemands avec plus de 1 000 000 passagers par an : </SEamorce> Aéroport de Francfort-sur-le-Main, Aéroport international Franz-Josef-Strauss de Munich, Aéroport international de Düsseldorf, Aéroport de Berlin-Tegel, Aéroport de Hambourg.
</structEnum>

10. Seuil déterminé expérimentalement sur une partie du corpus Quaero

(Cas 2) Longueur médiane des items supérieure à 60 caractères
Transformation en une phrase avec ajout d'une ponctuation non finale (;) pour séparer les items :

Énumération verticale (dans le document source HTML) :
Distinctions :
– 1993 : Troisième du classement du Ballon d'or
– 1994 : élu meilleur footballeur de l'année du championnat anglais par les joueurs adhérents à l'Association anglaise des footballeurs professionnels (PFA)
– Joueur du mois du Championnat d'Angleterre de football en mars 1996

Aplanissement en énumération horizontale (dans le document texte en sortie de Kitten) :
<structEnum>
<SEamorce>Distinctions : </SEamorce> 1993 : Troisième du classement du Ballon d'or ; 1994 : élu meilleur footballeur de l'année du championnat anglais par les joueurs adhérents à l'Association anglaise des footballeurs professionnels (PFA) ; Joueur du mois du Championnat d'Angleterre de football en mars 1996 ;
</structEnum>

3.1.5 *Performances*

Performances applicatives

Nous avons testé l'apport de Kitten en tâche applicative lors d'une première expérience avec le SQR QAVAL [Grappy *et al.*, 2011] puis lors d'une seconde expérience avec le SQR FIDJI [Falco *et al.*, 2012]. FIDJI repose principalement sur l'analyse syntaxique profonde (dépendances) des documents pour la recherche, l'extraction et la validation des candidats-réponses alors que QAVAL s'appuie sur une analyse syntaxique de surface pour l'extraction des candidats- réponses puis procède à une validation par apprentissage automatique.

Le corpus Quaero présenté au chapitre précédent a été traité par 3 systèmes différents :
– Kitten ;
– BoilerPipe [Kohlschütter *et al.*, 2010], un programme sous licence Apache s'étant révélé extrêmement performant lors des tâches d'extraction d'articles de journaux depuis une page HTML (sa stratégie repose notamment sur la densité textuelle) ;
– un système « baseline » qui consiste en une extraction linéaire totale des balises contenant du texte dans les documents HTML.

Pour ces deux expériences différentes, les données provenaient de campagnes d'évaluation Quaero :

– les 147 questions factuelles de Quaero 2009 [Ayache *et al.*, 2006] pour QAVAL ;
– les 500 questions (factuelles, définition, booléennes, complexes) de Quaero 2010 pour FIDJI.

La première expérience avec QAVAL (tableau 3.9) montre que le prétraitement baseline permet d'obtenir un nombre important de snippets contenant une réponse correcte. Cependant le MRR montre bien que le SQR QAVAL arrive extrêmement mieux à extraire ces réponses correctes avec le prétraitement effectué par Kitten.

Kitten semble également permettre à Lucene une meilleure sélection de documents puisque le système QAVAL renvoie de 7 à 14 % de passages supplémentaires contenant une réponse correcte par rapport à la baseline et BoilerPipe (voir tableau 3.9). De plus, 6 à 10 % de questions supplémentaires possèdent une réponse correcte dans les documents renvoyés par Lucene sur le corpus traité par Kitten (tableau 3.10).

Corpus Quaero traité par	**Nombre de snippets contenant une réponse correcte**	**MRR**
Baseline	114 (77 %)	0,28
Boilerpipe	121 (82 %)	0,32
Kitten	130 (88 %)	0,43

FIGURE 3.9 : Résultats de QAVAL pour les 147 questions factuelles de Quaero 2009.

Nombre maximum de passages par document	1		2	
Nombre de documents par questions	150	300	150	300
Corpus Quaero baseline	0.70	0.77	0.80	0.84
Corpus Quaero BoilerPipe	0.67	0.75	0,75	0.82
Corpus Quaero Kitten	**0.77**	**0.84**	**0.86**	**0.90**

FIGURE 3.10 : QAVAL : Ratio de questions contenant une réponse correcte dans les documents sélectionnés par Lucene (sur 210 questions factuelles Quaero 2009).

Le tableau 3.11 montre le nombre de réponses correctes obtenues par FIDJI à partir du corpus prétraité par les 3 systèmes.

Pour ces deux expériences, Kitten permet un apport au niveau performance applicative pour les SQR par rapport à une extraction linéaire totale avec :
– pour FIDJI : environ 25 % de réponses supplémentaires en rang 1 et 20 % pour les trois premiers rangs,

Corpus Quaero traité par	Nbre de réponses correctes au premier rang	Nbre de réponses correctes dans les trois premiers rangs
Baseline	69	109
BoilerPipe	75	116
Kitten	**86**	**131**

FIGURE 3.11 : Résultats de FIDJI pour les 500 questions de Quaero 2010 : nombre de questions correctement répondues.

– pour QAVAL : environ 38 % de réponses supplémentaires en rang 1 et 33 % pour les trois premiers rangs.

Performances informatiques

Kitten est également robuste : seuls 2 565 (0,51 %) documents ont nécessité d'utiliser l'extraction de secours, à savoir l'extraction brute par le navigateur *Lynx*.

Le prétraitement complet du corpus Quaero (environ 500 000 documents) par Kitten nécessite 24,5 heures de traitement à cinq processeurs en parallèle sur un serveur « classique » (50 Go de RAM, 15 processeurs). Kitten stocke sous forme de fichier chaque étape de traitement : si un utilisateur décide de changer une option, notamment concernant la segmentation du texte, il peut s'appuyer sur les fichiers intermédiaires existants. Cette étape ralentit le temps de traitement mais nous permet ensuite de faire une nouvelle extraction à partir de n'importe quelle étape.

La répartition du temps de traitement de Kitten sur le corpus Quaero par étape est en moyenne :

– normalisation (htmlCleaner et jTidy) : 23,90 % ;
– extraction du contenu : 76,10 % dont :
  – arbre de décision sur les tableaux : 31,42 %,
  – extraction des tableaux : 9,88 %,
  – extraction des listes : 4,62 %,
  – segmentation de phrases : 36,27 %.

Plus de la moitié de ce temps de traitement concerne donc la normalisation et les arbres de décisions sur les tableaux. Il y a un axe d'amélioration certain au niveau de la segmentation de phrases, notamment sur nos expressions régulières. Du point de vue extraction, Kitten est moins rapide que BoilerPipe (qui n'extrait cependant pas tout le document mais seulement la partie texte la plus dense) et moins rapide que BeautifulSoup (qui réalise une extraction linéaire).

3.2 MOTEUR DE RECHERCHE LUCENE

Comme dans tout SQR, nous avons besoin d'un moteur de recherche qui sélectionne des documents pertinents par rapport à une requête. Nous avons pour cela utilisé Lucene [Hatcher *et al.*, 2010] dans sa version 4 [11] que nous utilisons déjà pour les SQR (FIDJI [Moriceau et Tannier, 2010] et QAVAL [Grappy *et al.*, 2011].

3.2.1 *Indexation des documents avec Lucene*

Nous avons paramétré Lucene pour réaliser plusieurs indexations d'une collection de documents afin de tirer partie des balises introduites par Kitten durant son prétraitement.

Nous avons choisi de ne pas utiliser Tika [Mattmann et Zitting, 2011] en prétraitement de la collection de documents : Tika est un programme intégré à Lucene permettant d'extraire du texte depuis des documents HTML notamment. Le texte est extrait depuis des blocs d'information mais après avoir étudié son code source, nous avons constaté qu'il était plus pratique de développer Kitten plutôt que de coder nos modules directement dans Tika.

INDEXATION CLASSIQUE. Nous utilisons le racinisateur Snowball [Porter, 2001] configuré pour le français intégré à Lucene pour construire un premier index des termes racinisés. Nous utilisons également sa liste de mots-vides, liste à laquelle nous avons ajouté les quatre balises introduites par Kitten afin de ne pas fausser les pondérations (<titleKitten> pour le marquage du titre de la page, <tableauKitten> pour le marquage des tableaux, <structEnum> et <SEamorce> pour le marquage des énumérations).

INDEXATION DE BALISES. L'utilisation de balises comme attribut de recherche est prévue par Lucene. Il devient alors possible d'indexer et d'effectuer des recherches uniquement sur les valeurs de ces champs. En plus de l'index classique, nous utilisons donc Lucene pour construire quatre autres index soit un pour chacune des balises ajoutées par Kitten ce qui permettra des recherches sur certains types de structures.

3.2.2 *Recherche des documents avec Lucene*

Notre but est de sélectionner les passages les plus pertinents pouvant répondre à une question. Nous utilisons Lucene de deux façons différentes pour cela :

- sélection dans l'index classique (le document est pris dans sa globalité) : la méthode vectorielle de Lucene trouve des documents pertinents par rapport à une requête puis l'algorithme BM25 de Lucene extrait les meilleurs snippets (passages) de ces documents ;

11. http ://lucene.apache.org/

– sélection dans les index de balises : le contenu entier de la balise est vu comme un snippet.

Nous utilisons les snippets plutôt que le document complet car analyser syntaxiquement la totalité d'un document serait très coûteux en temps et n'apporterait pas forcément de gain. Nous avons effectivement vu au chapitre précédent la répartition des passages-réponses à l'intérieur d'un même document (tableau 2.10 : pour 38,17 % des questions-listes portant sur les documents issus du Web, il y avait en moyenne 3 passages-réponses par document (médiane de 2). Si nous arrivons à extraire les quelques snippets pertinents d'un document, il n'est pas nécessaire d'analyser le document entier.

Le but des deux types d'indexation est d'aboutir à une recherche fonctionnant comme une partition du document. Chaque document peut être vu comme une succession de parties considérées soit comme du texte, soit comme des structures énumératives, soit comme des extractions de tableau. La partie texte se compose du document complet alors que les deux autres parties sont délimitées par leurs balises respectives. Les balises <titleKitten> et <SEAmorce> ne sont pas utilisées durant la recherche.

Partant d'une requête, trois recherches séparées sont effectuées sur les trois champs de recherche (texte, SE, tableaux). Nous empêchons qu'un même passage ne soit sélectionné depuis des index différents. Du fait du prétraitement de Kitten, les balises <structEnum> et <tableauKitten> sont par nature mutuellement exclusives donc les passages sélectionnés à partir des index correspondant le sont également. Nous devons donc seulement nous assurer qu'un passage renvoyé par l'index des tableaux ou des SE ne sera pas contenu dans un passage renvoyé par l'index classique. Les passages textes renvoyés depuis le document entier se voient donc amputés des balises <structEnum> et <tableauKitten>. Nous considérons ces deux objets comme unicitables comme nous l'avait montré l'étude en corpus du chapitre précédent.

3.2.2.1 *Sélection de snippets dans l'index classique*

Dans la version 4 de Lucene, il est possible d'utiliser l'algorithme BM25 [Robertson et Zaragoza, 2009], [Pérez-Iglesias *et al.*, 2009] pour sélectionner les meilleurs *passages* d'un document. L'algorithme découpe le document en plusieurs parties de longueur paramétrable et chaque partie se voit attribuer un score relatif à la requête : le document entier est considéré comme un corpus et chaque partie comme un document. Toutefois, avec cet algorithme, Lucene ne renvoie que les meilleures segments de phrases des meilleures parties et non directement les meilleure parties segmentées. Nous devons donc constituer un snippet continu couvrant les meilleurs segments et répondant au critère de longueur paramétré précédemment.

Nous avons constaté expérimentalement que Lucene renvoyait généralement de un à quatre segments et que ces segments appartenaient à un même snippet une fois le snippet de 300 caractères constitué (les ponctuations et casse des lettres sont analysées

pour s'assurer que la première et la dernière phrases sont bien complètes, dans la limite de 300 caractères maximum pour chacune de ces deux phrases). Nous nous retrouvons donc fréquemment avec uniquement un, voire deux snippets de type texte par document.

3.2.2.2 *Sélection de snippets dans les index de balises*

L'algorithme BM25 fonctionne sur un document entier et n'est pas applicable pour sélectionner les meilleures balises <StructEnum> et <tableauKitten> d'un document. Un formatage des snippets sélectionnés à partir de l'index des balises <structEnum> est nécessaire. En effet, une SE peut être récursive et une balise <structEnum> peut donc être contenue dans une autre balise du même type, Citron supprime donc les SE de niveau inférieur dans le snippet <structEnum> renvoyé par Lucene. Si une structure énumérative imbriquée est pertinente, elle sera également renvoyée par Lucene.

Concernant la longueur des snippets sélectionnés, un filtrage est effectué sur la base d'un seuil propre à chaque balise. En effet, certains tableaux peuvent être extrêmement longs mais pertinents alors qu'une structure énumérative longue sera plus difficilement pertinente. Analyser et traiter une structure énumérative de cinquante commentaires d'un article de journal en ligne est très coûteux en temps et une grande source de bruit (plusieurs interlocuteurs, plusieurs thématiques). Nous avons ainsi fixé à 10 000 caractères la longueur maximale d'un snippet provenant d'un tableau et à 1 500 celle d'un snippet provenant d'une SE.

Actuellement, nous ne limitons pas le nombre de snippets sélectionnés depuis ces deux index de balises.

3.3 ANALYSEUR SYNTAXIQUE ET DÉTECTEUR D'ENTITÉS NOMMÉES XIP

Pour l'analyse des questions et des documents, nous utilisons l'analyseur syntaxique XIP [Aït-Mokhtar *et al.*, 2002] qui produit une analyse en dépendances et détecte aussi les entités nommées.

3.3.1 *Analyse syntaxique*

XIP permet la création de lexiques et de règles de réécriture. Pour notre travail, nous utilisons les règles définies pour le SQR FIDJI [Moriceau et Tannier, 2010] qui permettent entre autres d'extraire des relations de définition auxquelles nous avons ajouté des règles pour mieux détecter et extraire des réponses dans des structures énumératives et identifier leur type.

TYPAGE DES ITEMS D'UNE STRUCTURE ÉNUMÉRATIVE. Nous avons ajouté le trait *enumAmorce* aux éléments du lexique *voici* et *parmi* qui vont alors jouer le rôle de déclen-

cheur pour détecter l'enumeraTheme dans l'amorce d'une SE. Dans l'exemple suivant, les mots *distributions* qui sont en relation de dépendance avec *voici* et *parmi* vont pouvoir être identifiés comme enumeraTheme et ensuite permettre de typer les items de la SE :

Question : *Quelles sont les distributions Linux ?*
Enumération intra-phrastique 1 : Parmi les plus célèbres *distributions*, on peut citer (—).
Enumération intra-phrastique 2 : Par exemple voici quelques *distributions* spécialisées « environnement de bureau » : (—).

Pour le moment, XIP ne repère syntaxiquement que les structures énumératives horizontales (les énumérations verticales s'étalant sur plusieurs lignes sont trop difficiles à analyser syntaxiquement de façon automatique). Plutôt que de rajouter des règles directement dans XIP comme l'avaient fait [Gala, 2003] et [Aït-Mokhtar *et al.*, 2003], et ainsi quitter l'analyse au niveau phrastique pour entrer dans le niveau discursif, nous avons préféré prétraiter les documents avec Kitten pour transformer les SE verticales en SE horizontales.

AUTRES RÈGLES POUR LE TYPAGE DE NOMS. Il est possible avec XIP de définir des règles qui créent des dépendances en fonction d'un certain contexte : ce contexte peut imposer la présence de dépendances syntaxiques, de syntagmes ou de traits sur des lemmes. Par exemple, une règle définie dans FIDJI consiste à détecter un attribut du sujet dans les dépendances pour procéder à la création des relations ATTRIBUT-NN et DEFINITION entre les arguments pour finalement permettre de typer une réponse candidate :

Question : *Dans quels clubs a joué Nicolas Anelka ?*
Passage-réponse : Chelsea est un club de football très riche.

- **Analyse de XIP** : SUBJ(être[copule], Chelsea), OBJ(être, club[noun])
- **Résultat de la règle de réécriture FIDJI** :
  ATTRIBUT-NN(Chelsea, club), DEFINITION(Chelsea, club)

L'exemple suivant montre comment typer une réponse en cas d'incise :

Question : *Quelles villes ont été capitale de l'Allemagne ?*
Passage-réponse : Berlin, capitale réunifiée de l'Allemagne en 1991, fait partie des villes les plus visitées en Europe avec Londres, Paris ou Rome.

- **Analyse en constituants de XIP** : voir la figure 3.12
- **Contexte d'application de la règle (traduction synthétique)** : une relation de type DEFINITION est créée entre deux noms si ceux-ci se trouvent en début de phrase et séparés par une virgule éventuellement suivie d'une incise.
- **Résultat d'application de la règle** : DEFINITION(Berlin, capitale)

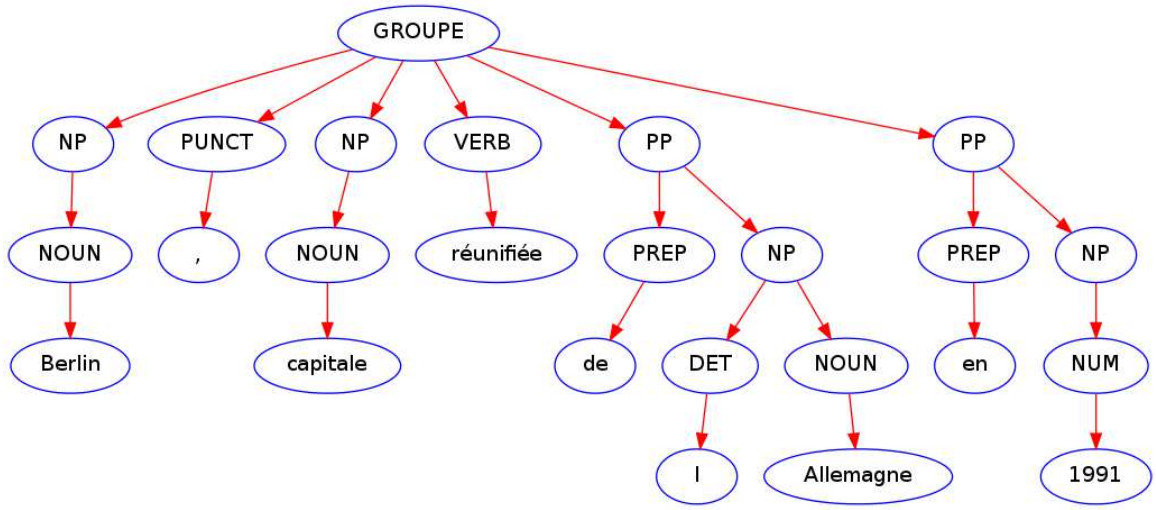

FIGURE 3.12 : Extrait de l'analyse en constituants de XIP pour la partie *Berlin, capitale réunifiée de l'Allemagne en 1991* de la phrase *Berlin, capitale réunifiée de l'Allemagne en 1991, fait partie des villes les plus visitées en Europe avec Londres, Paris ou Rome.*

Nos règles dépendent donc à la fois de l'analyse en dépendances de XIP mais également, pour certaines, de l'analyse en constituants. Notre version de XIP ne nous permet malheureusement pas de corriger l'analyse syntaxique en constituants lorsque nous identifions des problèmes.

3.3.2 *Détection d'entités nommées*

L'analyseur XIP intègre un détecteur d'entités nommées déjà très performant sur les types les plus rencontrés (personne, lieu, organisation). La version de XIP développée pour le projet ANR Chronolines dont nous disposons a été enrichie pour la détection des entités nommées temporelles et détecte donc ces entités aussi bien sous la forme d'événement (*la coupe du Monde de football*) que sous leur forme absolue (*jeudi 4 avril 2013*) ou relative (*mardi prochain*). Enfin, au sein du SQR FIDJI, plusieurs lexiques ont à leur tour été enrichis dont notamment ceux des couleurs et des animaux.

3.4 ÉTUDE DES AMORCES DES STRUCTURES ÉNUMÉRATIVES

L'enumeraTheme (Ho-Dac *et al.* [2010]) est l'élément le plus important de l'amorce, c'est lui qui type chacun des items, notamment si les items ne correspondent pas à une entité nommée (un pays, une ville, etc.). Il faut donc réussir à l'extraire correctement pour valider le type des items candidats-réponses. L'enumeraTheme ne doit pas être confondu

avec le *focusSE*, terme que nous définissons comme le thème sur quoi porte la SE, alors que l'enumeraThème est le type de chacun des items et correspond donc, dans notre cadre applicatif de question-réponse, au type des candidats-réponses.

Nous avons mené une étude sur le corpus Quaero afin de disposer d'une liste d'enumeraThemes génériques. En effet, nous allons voir qu'il existe des amorces où l'enumeraTheme n'est pas le type attendu de la question mais type tout de même correctement les items. Nous nous sommes également intéressé à la forme des amorces qui annoncent le nombre d'items de la SE. Les résultats de cette étude vise à pouvoir disposer d'un outil supplémentaire pour mieux analyser l'amorce des énumérations verticales et horizontales et ainsi mieux en extraire les items candidats-réponses.

3.4.1 *Absence de focusSE ou focusSE incomplet*

Nous avons constaté qu'il existe des SE ne possédant pas de focusSE ou un focusSE pas entièrement défini. Dans les exemples suivants, l'enumeraTheme est facile à repérer mais impossible à rattacher au focusSE sans résolution anaphorique, ce que ne fait pas Citron. Citron s'attache à identifier l'enumeraTheme dans l'amorce mais pas à rechercher le focusSE dans les phrases précédant l'amorce. La résolution anaphorique, ici nécessaire, est une tâche à part entière et source potentielle de beaucoup de bruit, nous sommes donc restés sur notre choix d'unicitabilité des SE. Il se peut donc que le moteur de recherche renvoie la SE de l'exemple A du fait de la présence du focus de la question dans l'amorce mais peut-être pas celle de l'exemple B (absence du focus) :

Exemple (A) Absence du focusSE (souligné) dans l'amorce mais présence dans la SE (l'enumeraTheme est en gras) :
Question : *Quels sont les atouts du langage PHP ?*
Focus de la question : PHP
Passage-réponse : PHP est un langage interprété (un langage de script) exécuté du côté serveur (comme les scripts CGI, ASP, ...) et non du côté client (un script écrit en Javascript ou une applet Java s'exécute sur votre ordinateur). La syntaxe du langage provient de celles du langage C, du Perl et de Java.
<StructEnum>
<SEamorce> Ses principaux **atouts** sont</SEamorce> La gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL).
<SEamorce> Ses principaux **atouts** sont</SEamorce> La simplicité d'écriture de scripts.
<SEamorce> Ses principaux **atouts** sont</SEamorce> la possibilité d'inclure le script PHP au sein d'une page HTML (contrairement aux scripts CGi, pour lesquels il faut écrire des lignes de code pour afficher chaque ligne en langage HTML).
<SEamorce> Ses principaux **atouts** sont</SEamorce> La simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés, mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur les plateformes Unix, Linux, et Windows)).
<SEamorce> Ses principaux **atouts** sont</SEamorce> L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, etc.).
</StructEnum>.

Exemple (B) Absence du focusSE (souligné) dans l'amorce et dans la SE :
Question : *Quels sont les fonctionnalités des daemondtools ?*
Focus de la question : daemondtools
Passage-réponse :
Les daemontools.
Présentation.
Les daemontools servent à superviser des services.
<StructEnum>
<SEamorce> Ils permettent de </SEamorce> démarrer un service.
<SEamorce> Ils permettent de </SEamorce> surveiller la vie de ce service.
<SEamorce> Ils permettent de </SEamorce> relancer un service qui meurt accidentellement.
<SEamorce> Ils permettent de </SEamorce> arrêter volontairement et redémarrer un service, si besoin est.
<SEamorce> Ils permettent de </SEamorce> gérer les logs de ces services.
</StructEnum>

3.4.2 *Généricité de l'enumeraTheme*

Comme vu dans les exemples précédents, en plus d'être présent sans le focusSE, l'enumeraTheme peut se révéler être trop générique dans une amorce. En effet, il va être difficile voire impossible de valider le type d'un candidat-réponse si l'enumeraTheme indique que le candidat-réponse est une *caractéristique* ou une *ligne*.

Pour identifier les enumeraThemes génériques, nous avons recensé les amorces de toutes les SE verticales balisées par Kitten dans le corpus Quaero (499 736 fichiers) et indexées par Lucene. Il est important de noter que ce recensement n'est pas un recensement sur un corpus de référence parfaitement annoté car Kitten peut s'être trompé dans l'identification des amorces de SE. Bien que ne portant que sur les énumérations verticales, le résultat de cette étude est aussi applicable lors de la recherche d'enumeraTheme dans des SE horizontales.

Lucene recense 325 475 balises d'amorce dans l'index de la collection Quaero traitée par Kitten réparties sur 107 234 documents, soit environ 21 % des documents de la collection.

Nous avons regroupé toutes les amorces vides (amorces composées uniquement d'espace(s), tabulation, etc.) en un seul type qui compte 19 444 occurrences. Sur le total des 325 475 occurrences d'amorces, nous recensons alors 137 192 amorces différentes, dont 115 283 amorces (35,42 %) ne comptent qu'une seule occurrence :

- nombre de documents avec au moins une amorce : 107 234
- nombre d'occurrences total d'amorces : 325 475
- nombre d'amorces différentes récupérées avec Lucene : 137 192
- nombre d'amorces à une seule occurrence : 115 283 (35,42 %, des occurrences totales, 84,03 % des amorces)
- nombre d'amorces à plus d'une occurrence : 21 909 (64,58 % des occurrences totales, 15,97 % des amorces)
- nombre d'occurrences d'amorces vides : 19 144

Dans ce corpus Web, les dix amorces les plus fréquentes sont :

- 19 444 occurences : amorce vide
- 5 702 occurences : *Pages : 1*
- 5 673 occurences : *Sommaire*
- 5 401 occurences : *Catégories*
- 3 616 occurences : *Liens*
- 2 479 occurences : *Derniers commentaires*
- 2 061 occurences : *Rubriques*
- 2 012 occurences : *Dans la même rubrique*
- 1 841 occurences : *À retenir*

– 1 722 occurences : *Derniers billets*

Nous voyons donc que les amorces les plus fréquentes comportent peu de mots et quasiment pas de verbe. Lucene recense 1 652 018 mots au total dans toutes les amorces, ce qui donne effectivement une moyenne peu élevée de 5,4 mots par amorce (en ne comptant pas les occurrences des amorces vides). Parmi ces amorces les plus fréquentes, deux cas sont possibles :

– l'amorce est vide : cela peut être du à une erreur de Kitten ou du site Web qui utilise des listes sans amorce à des fins de formatage ;
– l'amorce est annonciatrice d'une liste de liens mais Kitten ne réussit pas à l'identifier comme telle (par exemple dans la liste précédente, *Dans la même rubrique*, *Catégories*).

AMORCES ANNONÇANT LE NOMBRE D'ITEMS D'UNE SE. Afin d'étudier les amorces qui annoncent le nombre d'items de la SE, celles-ci ont toutes été analysées par XIP afin d'identifier celles possédant une entité numérique supérieure à 1 (XIP repère ces entités en lettres et en chiffres) dépendant d'un nom. Un filtre a été appliqué pour éliminer notamment les dates, horaires, mesures, nombres romains, nombres ordinaux et les adresses. Sur un total de 137 192 amorces, nous obtenons :

– 8,39 % (11 509) des amorces possèdent une entité numérique supérieure à 1,
– 91,74 % (125 863) des amorces n'en possèdent pas.

En regardant de plus près ces amorces contenant une entité numérique, de très nombreux cas ne comportent pas d'enumeraTheme mais concernent des entrées de tables des matières (*1.1.1 Principe*) ou des références (*Vol. 1, no 4, octobre 1985*) : nous les avons éliminées. Parmi les énumérations verticales identifiées par Kitten, il y en a donc peu possédant une amorce qui annonce le nombre d'items. Il s'agit d'un point important du point de vue question-réponse puisque cela tendrait à montrer que le cas d'une SE possédant toutes les réponses à une question-liste et le précisant dans son amorce est peu fréquent. C'est pourquoi, lorsque le nombre de réponses attendues est précisé dans la question, Citron va naturellement utiliser cette information mais ne va pas se limiter à chercher cette entité numérique dans une amorce.

On trouve également 55,28 % d'amorces (75 846) possédant un verbe. Nous n'avons conservé que les amorces comportant une entité numérique supérieure à 1 et un verbe (7 845 amorces au total) car elles nous semblaient porteuses de ce que nous recherchions durant notre survol manuel. Par exemple avec les amorces suivantes :

– <SEamorce>Au moins *trois possibilités* s'offrent à nous pour diminuer les impacts négatifs de nos activités domestiques sur la qualité de l'eau</SEamorce>
– <SEamorce>Cette question est induite par la position de la Direction, qui veut réduire le temps de travail annuel par *deux moyens* simultanés</SEamorce>
– <SEamorce>*Deux types* de pays du sud ont échappé à ces phénomènes</SEamorce>

En dénombrant les occurrences des noms dans les couples (entité numérique, nom) de ces amorces, nous sommes arrivés à 1 651 noms, dont 753 ayant plus d'une occurrence : ce sont ces 753 noms que nous avons manuellement évalués. Nous avons ainsi constitué deux listes d'enumeraThemes : une comprenant ceux attestés (128 noms, en annexe page 210) et une comprenant ceux que nous considérons comme douteux. Les cinq enumeraThemes les plus fréquents sont : *type* (235 occurrences), *commentaire* (148), *point* (143), *partie* (122) et *étape* (111). Parmi les trente noms les plus fréquents, trois sont supprimés (*bit*, *$* et *g* qui n'avaient pas été filtrés par XIP comme des mesures) et seuls trois sont jugés douteux du fait de la forte présence de ces mots dans les documents issus du Web : *commentaire*, *article* et *page*. Ces derniers signaleront à Citron que, s'il extrait un tel enumeraTheme, la SE est peut-être à considérer comme indigne de confiance. Ces cas douteux concernent notamment les titres générés automatiquement sur les blogs comme :

Exemples fréquents de noms n'étant pas des enumeraThemes :
– 2 *commentaires* pour Compiler son noyau Linux sous Debian GNU/Linux :
– Il y a 50 autres *articles* dans cette rubrique :
– Les 4 *pages* les plus visitées :

Par cette étude en corpus de l'enumeraTheme, nous disposons désormais d'une liste de 128 enumeraThemes génériques. Ces enumeraThemes ont été extraits depuis les amorces des énumérations verticales balisées par Kitten et sélectionnées ensuite par Lucene. Seules les amorces possédant un verbe et une entité numérique en relation avec un nom ont été conservées après l'analyse par XIP. De ces amorces ont été extraites les occurrences de noms qui ont été finalement validées manuellement comme étant un enumeraTheme à caractère générique.

Cette liste permettra à Citron non seulement d'invalider des candidats-réponses à cause d'un type trop générique mais également de repérer des SE utilisant un de ces enumeraThemes en relation avec une entité numérique particulière, notamment lorsque le nombre de réponses attendues est spécifié dans la question.

3.5 WIKIPÉDIA POUR LA VALIDATION DE RÉPONSES

Comme nous le verrons au chapitre suivant, nous utilisons Wikipédia pour valider le type des candidats-réponses. 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.

Nous utilisons cette ressource uniquement pour la validation du type des candidats. Cette utilisation nous parait cohérente avec notre approche qui se veut indépendante de toute ressource extérieure : en effet, Wikipédia étant accessible sur le Web et ses pages ressortant très fréquemment parmi les meilleurs résultats d'un moteur de recherche, notre SQR sur le Web a lui-aussi de fortes chances de devoir traiter ces pages.

3.5.1 *Prétraitement d'un dump de la Wikipédia*

Nous avons utilisé le *dump* de la Wikipédia du 28 octobre 2012 en le prétraitant de manière à obtenir un index des termes possédant une page d'homonymie, un index des termes de redirection et un index des articles.

Le dump est un fichier de 10,4 Go dont les méta-données des articles sont balisées en XML. Le contenu de chaque article est écrit avec la syntaxe Wikipédia mais nous ne nous intéressons qu'à l'information textuelle puisqu'elle sera traitée par XIP :

Début de l'article sur l'Autriche avec la syntaxe Wikipédia :
L'''Autriche'''ou '''République d'Autriche ''', en forme longue, ({{lang|de|''Österreich''}} et {{lang|de|''Republik Österreich''}} en [[allemand]]), est un [[Liste des pays du monde|pays]] d'[[Europe centrale]], [[Accès à la mer|sans accès à la mer]].

Sortie texte pour XIP :
L'Autriche ou République d'Autriche, en forme longue, (Österreich et Republik Österreich en allemand), est un pays d'Europe centrale, sans accès à la mer.

Pour obtenir cette sortie texte, nous avons écrit un script Python qui repère l'introduction à l'aide de la balise du premier titre de l'article et formate l'introduction à l'aide d'expressions régulières. Ce script effectue le prétraitement en 6 heures et n'extrait pas les pages qui nous sont peu utiles à savoir les pages *Catégorie*, *Projet*, *Aide*, etc.

Des 3 millions d'articles ont été construits des index de 2 541 698 articles, 65 430 homonymes et 866 redirections.

3.5.2 *Étude du type dans les introductions*

Nous avons analysé syntaxiquement avec XIP toutes les introductions d'articles afin d'obtenir un recensement des types d'entités présents (l'entité étant le titre de l'article). Cela nous a permis de nous assurer de la bonne application de nos règles (nous présentons ces règles de détection des définitions en détail au chapitre suivant).

En regardant le détail des types les plus fréquents, nous avons procédé à l'exclusion de quelques types, dont :

– en rang 20 se trouvait le type *thé* avec 17 197 occurrences. Nous avons constaté que ce n'était pas dû à un nombre important d'articles sur le thé mais à XIP que nous

utilisons pour le français et qui transforme l'article anglais *the* en nom français *thé*. Par exemple dans l'article *The Ecstasy of Gold*, *thé* se retrouve donc identifié comme un type car il est vu comme modifieur de nom ;
– quelques participes passés sont extraits comme type suite à la relation *ATTRIBUT* de XIP. Par exemple pour l'article *Jean-Jacques Goldman* :
Jean-Jacques Goldman, né le 11 octobre 1951 à Paris, est un auteur-compositeur-interprète français → ATTRIBUT(né, Jean-Jacques Goldman) ;

Au final, nous obtenons un top-50 (voir en annexe page 212) qui nous rassure sur la bonne application des nos règles DEFINITION, avec notamment des termes géographiques (*ville*, *commune*), des nationalités (*français, américain*), des métiers (*joueur*, *acteur*, *footballeur*) et des déclencheurs de définitions (*nom*, *espèce*, *genre*, *catégorie*).

3.6 CADRE DE DÉVELOPPEMENT

Pour le développement de notre système Citron, nous nous sommes tout d'abord placé dans ce que nous avons appelé des « conditions idéales ». L'utilisation d'un cadre en conditions idéales a pour objectif de permettre de se concentrer sur une tâche particulière en traitant de façon certaine le ou les cas idéalement attendus. Ainsi, ce cadre de développement permet :

– de nous concentrer sur une tâche précise : pour la stratégie de détection et traitement des SE, il est logique de vouloir disposer de façon certaine dans les documents des différents types de SE recensés durant nos observations en corpus ;
– de ne pas être bruité par une erreur se produisant en amont de l'extraction de la réponse. Ces erreurs sont souvent irrattrapables et brideraient le développement. Elles peuvent se produire notamment durant le prétraitement des documents, l'analyse de la question, la recherche du document et l'analyse syntaxique du document [El Ayari, 2009]. Afin de traiter ces erreurs, ce n'est que dans le deuxième cycle de développement que la robustesse sera apportée ;
– d'avoir une certitude sur la mesure du rappel : le corpus étant de taille « humaine », il est possible de constituer manuellement un fichier de référence contenant les réponses correctes relativement rapidement en parcourant la totalité des documents ;
– de définir un cahier de spécifications des données d'entrée pour le branchement de Citron à un SQR.

Notre cadre de conditions idéales concerne aussi bien les questions-ARM que les documents Web utilisés.

3.6.1 *Conditions idéales pour les questions*

Les questions-ARM que nous avons sélectionnées pour le développement présentent toutes des problématiques significatives découvertes durant notre étude du corpus Frites.

Nous avons choisi quatorze questions-ARM issues de cette étude [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 ?*, *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 [12] [Afantenos *et al.*, 2012], contenant des structures énumératives annotées : *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-clés. Par exemple :

Question : *Dans quels clubs joue Nicolas Anelka ?*
- Type de la question : liste
- Type de la réponse attendue : *club*
- Focus : *Nicolas Anelka*
- Verbe principal : *jouer*

3.6.2 *Conditions idéales pour les documents*

Pour chaque question présentée précédemment, des documents ont été récupérés depuis Internet au format HTML à partir de plusieurs moteurs de recherche (Bing [13],

12. http ://redac.univ-tlse2.fr/corpus/annodis
13. http://www.bing.com/

Exalead [14] et Google [15]) grâce à des requêtes générées manuellement à l'aide des éléments de l'analyse de la question (focus, verbe principal, mots-clés). Ensuite, une validation manuelle de la présence des réponses correctes dans les documents a été effectuée. Pour cette raison, le nombre de documents à récupérer a été volontairement restreint, un total de 67 documents a été récupéré pour les 14 questions-ARM sélectionnées.

De chaque document, les passages contenant une réponse correcte ont été manuellement extraits (un passage peut comporter plusieurs phrases mais est en large majorité composé d'une seule phrase). L'extraction consiste à copier le passage dans un fichier en y recensant la ou les réponses correctes. Lorsque le passage provient d'éléments structuraux (tableau, liste), nous l'avons formaté telle qu'attendue en sortie de Kitten. Par exemple pour le tableau suivant :

Question : *Dans quels clubs a joué Nicolas Anelka ?*
Passage-réponse HTML :

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

Fichier de référence :
- **Saison** : 95-96 / **Club** : Paris-SG / **Pays** : FRA .
- **Saison** : 96-97(fév) / **Club** : Paris-SG / **Pays** : FRA .
- **Saison** : 96-97 / **Club** : Arsenal / **Pays** : ANG .

Enfin, deux résolutions manuelles ont également été effectuées :
- réconciliation de référence : les coréférences ont été résolues en remplaçant manuellement tous les référents anaphoriques par les référés ;
- résolution temporelle : la date de chaque document a été extraite puis affectée à chacun des passages provenant de ce document, y compris lorsque cette dernière ne pouvait être déduite qu'à partir de connaissances du monde. Les heures n'ont pas été conservées et si plusieurs dates existaient (date de parution et date de mise à jour), seule la plus récente a été prise en compte.
  Par exemple le document de la figure 3.13 devient le passage suivant :

Question : *Quand la France a-t-elle perdu son triple-A ?*
Passage-réponse : La France avait perdu son « AAA » chez l'agence de notation américaine Egan-Jones en juillet.
Date du passage-réponse : 2011-11-30

14. http://www.exalead.com
15. http://www.google.fr

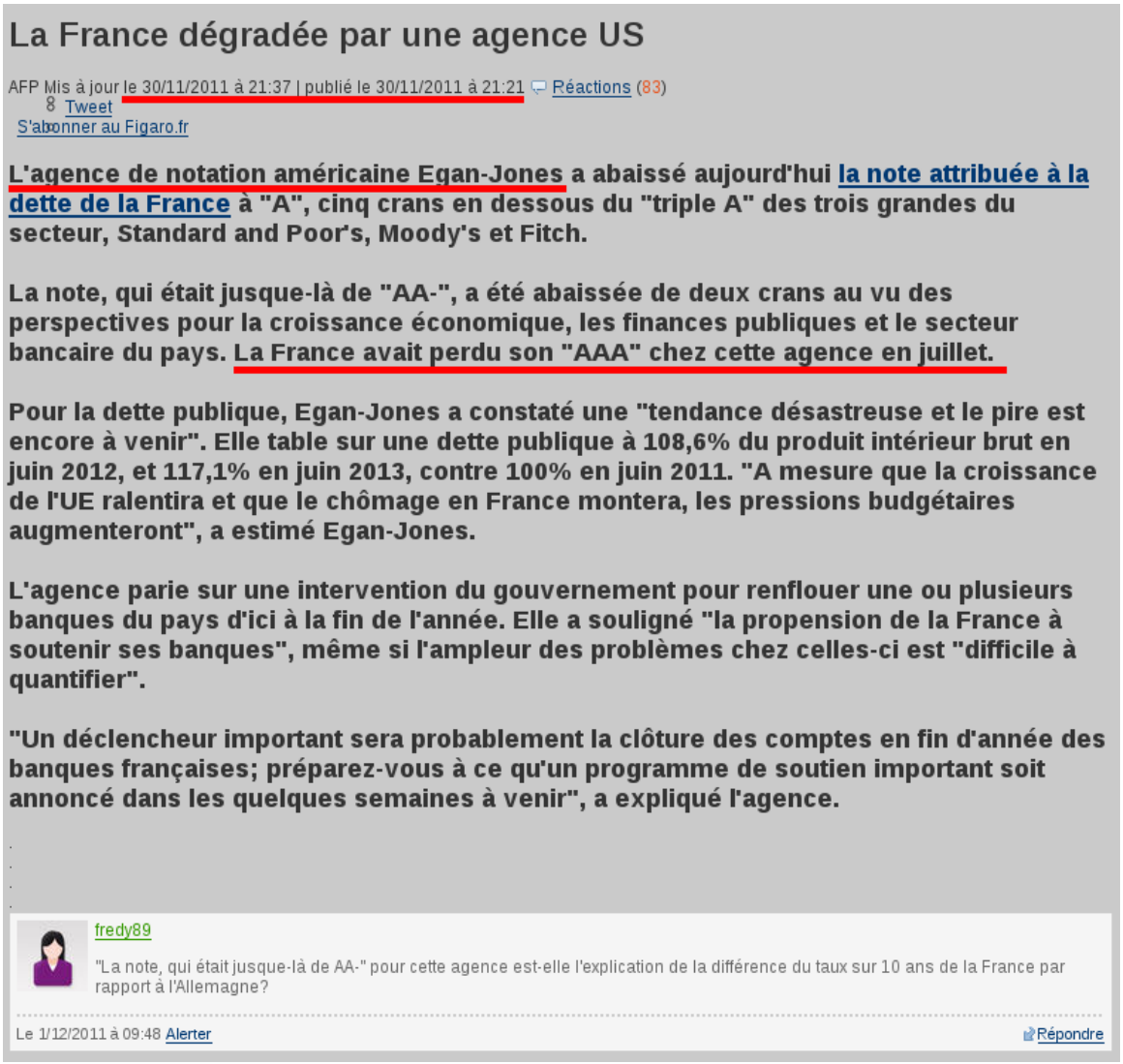

FIGURE 3.13 : Document HTML brut avec réconciliation de référence et résolution temporelle.

Enfin, un recensement des réponses correctes dans chacun des passages a été effectué pour procéder à une segmentation de chacun des passages. Un passage contient au moins une réponse correcte et peut en contenir plusieurs. La longueur d'un passage est généralement d'une phrase, trois maximum. Nous obtenons ainsi 422 passages contenant chacun au moins une réponse correcte à extraire, et sur ces 422 passages sont dénombrées 208 réponses correctes différentes (forme de surface).

3.7 CONCLUSION

Dans ce chapitre, nous avons présenté les différents outils utilisés en amont ou par Citron, notre système d'extraction de réponses multiples sur le Web. Kitten est ainsi utilisé pour prétraiter les documents HTML et notamment pour rendre exploitable les tableaux et structures énumératives de ces documents. L'analyseur syntaxique et détecteur d'entités nommées XIP est quant à lui utilisé pour réaliser l'analyse des questions et des passages sélectionnés par le moteur de recherche Lucene. Enfin, nous avons détaillé l'ensemble de questions et de documents que nous avons défini pour le développement de Citron, système que nous présentons au chapitre suivant.

4

CITRON : UN SYSTÈME D'EXTRACTION DE RÉPONSES MULTIPLES SUR LE WEB

Sommaire

Nous présentons dans ce chapitre le système *Citron* qui permet l'extraction de réponses multiples sur le Web ainsi que leur agrégation.

La figure 4.1 présente l'architecture générale de Citron. Il est codé en Java et a pour objectif de pouvoir être utilisable par tout SQR à des fins de validation de réponses. En

effet, il peut être branché à différentes étapes de fonctionnement d'un SQR en recevant l'entrée XML adéquate et peut donc procéder à l'extraction et à l'agrégation des réponses depuis une liste de documents au format texte, depuis des snippets au format texte et des candidats-réponses. Aux étapes classiques d'un SQR va s'ajouter pour Citron une étape préalable de prétraitement des documents HTML pour les transformer en documents textes.

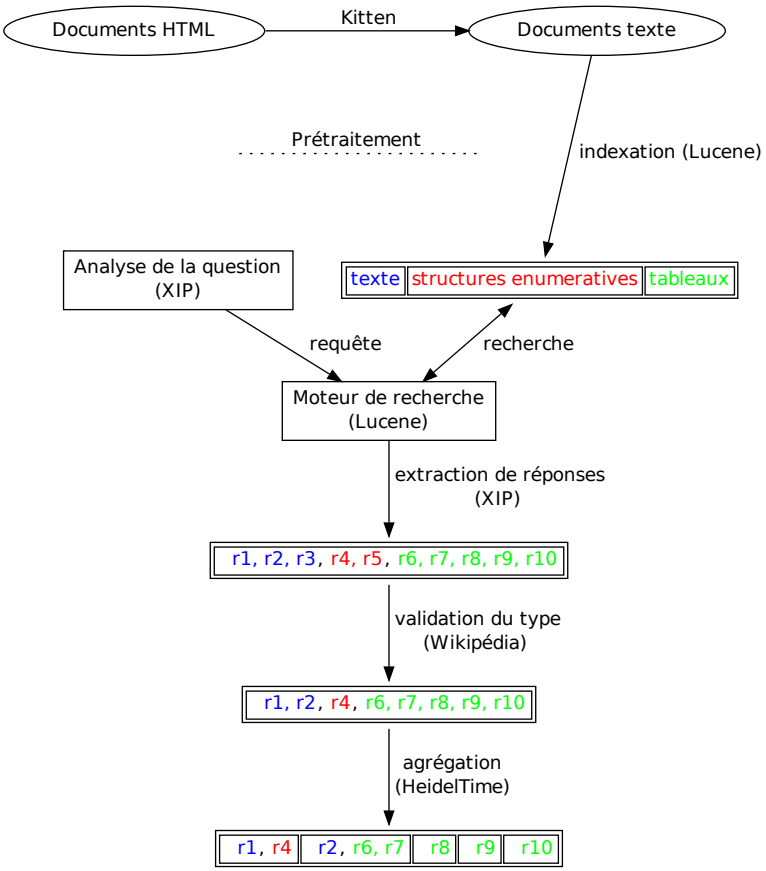

FIGURE 4.1 : Architecture du système Citron.

Pour l'extraction des 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. Après avoir analysé avec XIP les snippets sélectionnés par Lucene, Citron extrait des candidats-réponses du type attendu depuis les dépendances syntaxiques. Puis, il recherche des candidats-réponses indirectement depuis ces dépendances en se concentrant notamment sur la similarité contextuelle des phrases

afin d'extraire des noms et des verbes reliés au focus, au type de la réponse attendu et au verbe principal de la question. Ensuite, il recherche des candidats-réponses du bon type dans les éventuelles structures énumératives et les tableaux des documents sélectionnés. Enfin, Wikipédia est utilisée pour désambiguïser et valider si nécessaire le type des candidats recueillis. Du point de vue informatique, nous avons codé chacune des stratégies de Citron de manière indépendante, ce qui nous permet de pouvoir choisir au lancement celles qui doivent s'appliquer ou non. Pour résumer, les sept stratégies de Citron pour l'extraction de candidats-réponses (selon la provenance du snippet considéré) sont les suivantes (figure 4.2) :

- **Stratégies sur tous les types de snippets texte, SE et tableaux :**
  - **stratégie A** : si le type de la réponse est un type d'entité nommée spécifié dans la question, recherche de candidats du bon type d'entité nommée d'après XIP dans les snippets provenant de l'index classique ;
  - **stratégie B** : si le type de la réponse est spécifié dans la question, recherche avec nos règles XIP de candidats du bon type dans les snippets provenant de l'index classique ;
  - **stratégie C** : recherche des dépendances syntaxiques de la forme affirmative de la question dans les 3 index ;
- **Stratégies sur les snippets texte et SE :**
  - **stratégie D** : recherche de candidats du bon type à l'intérieur d'énumérations intra- phrastiques dans les snippets provenant de l'index classique et de celui des balises <StructEnum> ;
  - **stratégie E** : recherche de candidats du bon type à l'intérieur d'énumérations horizontales dans les snippets provenant de l'index classique et de celui des balises <StructEnum> ;
- **Stratégies sur les snippets SE :**
  - **stratégie F** : recherche de candidats du bon type à l'intérieur d'énumérations verticales formatées par Kitten dans les snippets provenant de l'index des balises <StructEnum> ;
- **Stratégie sur les snippets tableaux :**
  - **stratégie G** : recherche de candidats du bon type dans les tableaux formatés par Kitten dans les snippets provenant de l'index des balises <tableauKitten>.

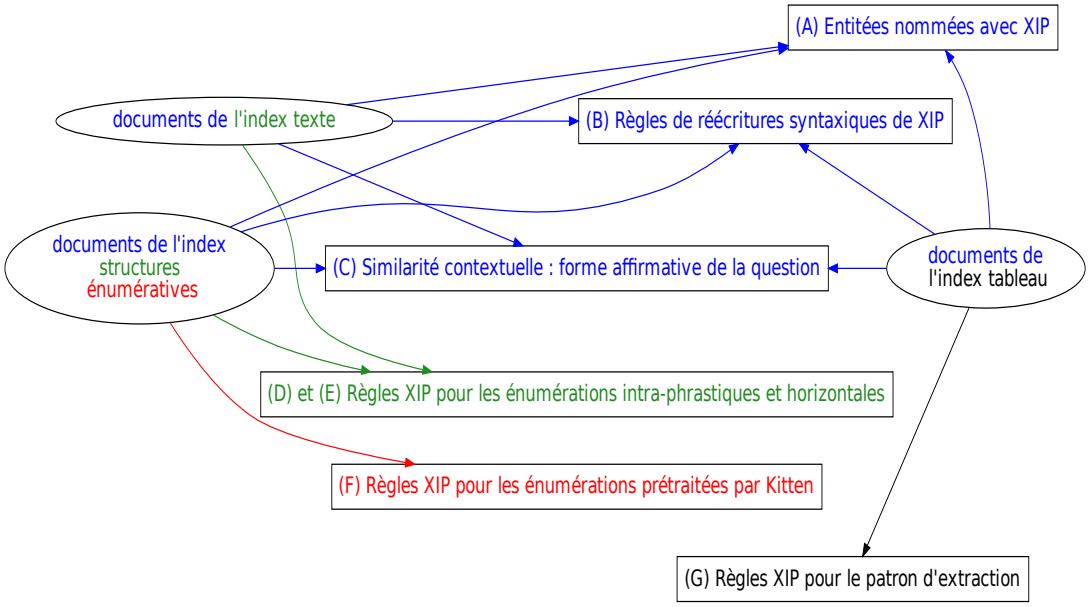

FIGURE 4.2 : Stratégies d'extraction du système Citron.

Nous allons détailler dans ce chapitre chacune des stratégies de Citron pour extraire les réponses, les valider et les agréger. Chaque stratégie a connu deux cycles de développement : un premier en conditions idéales, puis un second confronté à des conditions plus réalistes, ce sera l'objet du chapitre 5. Ces stratégies sont maintenant présentées de la façon suivante :

1. l'extraction de candidats-réponses depuis les trois types de snippets qui comprend :
   - la recherche de candidats-réponses du type attendu par la question,
   - la recherche de similarité contextuelle sur le focus et le verbe principal de la question,
   - la résolution temporelle,
2. la détection et l'extraction à partir de structures énumératives ;
3. la détection et l'extraction à partir de tableaux ;
4. l'utilisation de Wikipédia pour la désambiguïsation et la validation du type d'un candidat-réponse.

4.1 EXTRACTION DE RÉPONSES MULTIPLES À PARTIR DE TEXTE

4.1.1 *Recherche de candidats-réponses dont le type attendu est connu*

Quand le type précis de la réponse est explicitement spécifié dans la question (par exemple, *Quels aéroports...*), Citron applique deux recherches pour trouver des candidats-réponses de ce type : une première qui utilise la détection d'entité nommée (stratégie A), puis une seconde guidée par la présence du type attendu dans le document (stratégie B).

Quand le type précis de la réponse n'est pas explicite dans la question (par exemple, les questions en *où* (localisation) et *qui* (personne, organisation)), Citron recherche dans les snippets uniquement des entités nommées du type de la question. Citron utilise alors des lexiques et règles de détection d'entités nommées comme les villes, les pays, les lieux pour les localisations ou les métiers, les prénoms et organisations pour les personnes et les organisations. Les lexiques et règles de détection d'entités nommées sont définis pour le système de question-réponse FIDJI [Moriceau et Tannier, 2010] et ont été enrichis, notamment pour la production de règles de type DÉFINITION.

Recherche du type explicite par les entités nommées de XIP (stratégie A)

XIP est capable de détecter plusieurs types d'entités nommées donc Citron utilise cette fonctionnalité pour rechercher des candidats-réponses repérés par XIP comme étant du type d'entité nommée attendu par la question.

Pour les questions de type numérique ou temporel, Citron ne considère pas toutes les entités nommées de type NOMBRE et DATE de XIP car ces dernières entraînent l'extraction de beaucoup trop de candidats-réponses. Nous appliquons donc plusieurs filtres s'étant révélés pertinents durant le développement de Citron.

Pour les questions dont le type attendu est NOMBRE, les filtres sont les suivants :

- exclusion des nombres ordinaux et romains ;
- exclusion des nombres faisant partie d'une date ;
- exclusion des nombres désignant des mesures, par exemple la relation NOMBRE(50) pour *50 euros* est exclue et remplacée par ARGENT(50 euros). De même, pour *3 tonnes*, la relation NOMBRE(3) est remplacée par POIDS(3 tonnes) ;
- exclusion des collocations (par exemple, exclusion de NOMBRE(10) dans l'expression *top 10*).

Pour les questions dont le type attendu est DATE, les filtres sont les suivants :

- exclusion des entités NOMBRE vues comme des DATE : nos règles considèrent parfois à tort certains nombres comme des dates et il a fallu créer des règles pour supprimer ces relations erronées. Par exemple, nos règles suppriment les relations DATE suivantes pour ne garder que les relations NOMBRE sur ces entités :

les relations DATE(1806) depuis *N° 1806*, DATE(1649) dans *Page 1649* ou encore DATE(3189) dans *Recherche : 3189 connectés* sont supprimées ;
- exclusion des horaires : nous avons choisi de ne pas descendre jusqu'à cette granularité, notamment parce que nous n'avons pas la capacité de rattacher un horaire à sa date absolue ;
- exclusion des expressions temporelles relatives : XIP les identifie comme des entités temporelles (par exemple, *lendemain*, *désormais*, *ce jour-là*) mais nous n'arrivons pas encore à les résoudre notamment car, sur le Web, nous ne disposons que très rarement de la date qui pourrait permettre de réaliser la normalisation des expressions relatives. Les questions temporelles que Citron est capable de traiter ne portent donc que sur des dates absolues ;
- exclusion des dates postérieures à la date d'exécution de Citron, excepté pour les questions comportant une marque de futur (temps du verbe de la question, date absolue ou termes indiquant le futur dans la question) ;
- exclusion des dates provenant de forums : elles sont extrêmement nombreuses comme dans *Date d'inscription : 28-08-2011* et présentes pour chaque message du forum. Elles sont utiles pour situer temporellement une réponse à une question non-temporelle ou pour dater le document mais nous ne les considérons pas dignes de confiance pour une question temporelle. Ces dates sont donc constamment ignorées quel que soit le type de la question.

Recherche syntaxique du type explicite dans les documents (stratégie B)

Dans un premier temps, Citron recherche les candidats-réponses identifiés syntaxiquement par XIP comme étant du type attendu. Les dépendances syntaxiques recherchées sont celles identifiant :
- les compléments du nom : par exemple, dans *Anelka rejoint le club anglais de Manchester City*, le groupe nominal *Manchester City* est complément du nom *club* et sera un candidat-réponse à la question *Dans quels clubs a joué Nicolas Anelka ?*,
- les modifieurs du nom : par exemple, dans *Christopher Nolan a réalisé le film Inception*, XIP détecte une relation NMOD entre les deux noms *film* et *Inception*. Ainsi, *Inception* devient un candidat-réponse à la question *Quels films a réalisé Christopher Nolan ?*,
- les dépendances correspondant à l'expression de relations de définition. Pour cela, nous nous sommes appuyé sur la base de patrons définis pour le SQR FIDJI du type *DEFINITION(type attendu, candidat-réponse)*. Ainsi, des relations de définition existent :
  - entre la tête d'un syntagme nominal et le sigle qui le suit entre parenthèses, par exemple :
    L'organisation des Nations Unies (ONU) → DEFINITION(organisation, ONU) ;

– entre un nom propre et un nom commun du même syntagme, par exemple : *L'écrivain américain Charles Bukowsky* → DEFINITION(Charles Bukowsky, écrivain) ;
– entre deux noms reliés par les déclencheurs *comme*, *tel*, *nommé*, par exemple : *Jean, tel un chat, dort* → DEFINITION(Jean, chat).

Aux 15 patrons DEFINITION de FIDJI , nous en avons ajouté 40, dont par exemple [1] :
– lors d'une apposition propositionnelle :
Christian Bale, l'acteur qui incarne Batman, a fait une visite à Aurora
→ DEFINITION(Christian Bale, acteur) ;
– lorsqu'un mot désignant un groupe (*ensemble* dans l'exemple suivant) précède un nom dans un patron de définition :
L'ensemble wakizashi-katana s'appelle le daisho → DEFINITION(wakizashi- katana, daisho). Ainsi la relation DEFINITION(ensemble, daisho) n'est pas créée ;
– lorsqu'un verbe de définition (*correspondre*, *signifier*, etc.) est utilisé :
En botanique, un bourgeon désigne une excroissance apparaissant sur certaines parties des végétaux → DEFINITION(bourgeon, excroissance) ;
– lorsqu'un nom employé avec une relation de complètement de nom amène à faire une transitivité de la définition :
"Le Raton laveur commun est une espèce de mammifères omnivores."
→ DEFINITION(Raton, mammifère).

Enfin, nous appliquons une gestion basique de la négation pour ne pas créer des relations DEFINITION erronées. Une relation spécifique aux attributs du sujet est créée comme dans *Lyon n'est pas la capitale de la France* où la relation ATTRIBUTNN-NEGAT (Lyon, capitale) est ajoutée.

4.1.2 *Recherche de nouveaux candidats par similarité contextuelle*

Cette recherche correspond à la stratégie C. Comme nous n'utilisons pas de ressources sémantiques pour notre travail et qu'il est intéressant d'extraire un plus grand nombre de candidats-réponses en recherchant des synonymes ou paraphrases de ceux déjà extraits, Citron recherche les verbes et noms « reliés sémantiquement » aux éléments importants de la question. En effet, nous nous appuyons sur les nombreux travaux ayant montré que les mots apparaissant dans un même contexte ont un sens similaire [Miller et Charles, 1991]. reliés les éléments importants de la question que sont le focus, le verbe principal et le type attendu. En plus de nous fournir des éléments thématiques, nous allons voir que nous extrayons au passage des candidats-réponses selon le contexte de certains noms reliés. Citron utilise pour cela :

1. D'autres patrons DEFINITION sont présentés plus loin pour les SE, les tableaux et l'utilisation des introductions d'articles Wikipédia.

1. le type d'entité nommée d'un candidat-réponse déjà extrait à l'aide des stratégies A et B ;
2. le contexte du focus, du verbe principal de la question et du type de la réponse attendu ;
3. la forme affirmative de la question.

4.1.2.1 *Étape 1 : Utilisation du type d'entité nommée d'un candidat-réponse déjà extrait*

Dans un premier temps, nous regardons le type d'entité nommée (si il existe) des candidats-réponses étant du type attendu par la question que nous avons trouvés syntaxiquement depuis les documents (stratégie A et B). Si un de ces candidats-réponses est détecté par XIP comme une entité nommée alors toutes les entités nommées de ce type dans les documents sélectionnés par le moteur de recherche deviennent des candidats-réponses potentiels. Par exemple :

Question : *Dans quels clubs a joué Éric Cantona ?*
Passage-réponse : Éric Cantona ; ; Parcours professionnel 1 ; Club : Manchester United.

Candidat-réponse du type attendu (*club*) trouvé avec les règles XIP (stratégie B) :
DEFINITION(club, Manchester United)
Type d'entité nommée du candidat-réponse d'après XIP :
ORGANISATION(Manchester United).

Autres passages-réponses :
– L'Angleterre comme terre d'accueil : la renaissance à Leeds United (1992).
– Condamné en mars et suspendu neuf mois par la Fédération anglaise (suspension étendue au niveau international par la FIFA), Cantona fit un retour triomphal sur les pelouses anglaises au mois d'octobre suivant.
Nouveaux candidats-réponses du type *ORGANISATION* :
ORGANISATION(Leeds United), ORGANISATION(FIFA),
ORGANISATION(Fédération anglaise)

Il se peut que plusieurs candidats-réponses ainsi extraits l'aient déjà été à l'étape précédente. Une fois ces nouveaux candidats extraits, il reste bien sûr à valider leur type précis : dans l'exemple donné ici, il faudra vérifier que les nouveaux candidats sont bien des clubs, ce qui n'est pas le cas de *FIFA*, ni de *Fédération anglaise*. Cette approche permet toutefois d'extraire de nouveaux candidats-réponses pour les cas où le focus est bien présent dans la phrase avec l'entité nommée mais où il n'est ni sujet, ni objet. Ce cas se produit notamment dans les phrases complexes où XIP ne parvient pas à identifier certaines relations importantes comme dans l'exemple suivant où *Anelka* (focus de la question) est présent dans la phrase mais est impossible à relier au *Real Madrid* avec les dépendances obtenues par XIP :

Question : *Dans quels clubs a joué Nicolas Anelka ?*
Passage-réponse : Formé au club et revenu de 2000 à 2002 après avoir explosé à Arsenal et au Real Madrid, Anelka pourrait même recevoir une offre très rapidement.
Analyse en dépendances (extrait) :
– DEEPSUBJ(pouvoir, Anelka)
– DEEPOBJ(recevoir, offre)
– VMOD(exploser, Real Madrid)
– PERSONNE(Anelka)

4.1.2.2 *Étape 2 : Utilisation du contexte du focus, du verbe principal et du type attendu*

La deuxième étape recense les verbes et noms reliés par une dépendance au focus, au verbe principal de la question ou au type de réponse attendu lorsqu'il est défini explicitement dans la question. Durant les recherches de verbes, nous n'avons pas filtré les cas de négation. Nos règles XIP couvrent les cas de proposition infinitive (par exemple, *peut partir*, *espère remporter*). Le but de ce recensement est d'obtenir des informations qui serviront aux stratégies suivantes de Citron que nous détaillons dans les prochaines sous-sections : aide pour le traitement des amorces des SE (stratégie D) et aide à la validation du type de candidats-réponses extraits de tableaux (stratégie G).

RECENSEMENT DES VERBES RELIÉS DIRECTEMENT. Les verbes recensés à cette étape sont ceux qui se trouvent dans des dépendances liées directement au focus de la question ou au type attendu de la réponse lorsqu'il est spécifié. Les verbes ainsi recensés sont utilisés pour la recherche de la forme affirmative de la question et l'analyse d'une amorce de SE.

Question : *Dans quels clubs a joué Éric Cantona ?*
– Type attendu : club
– Focus : Éric Cantona
– Verbe principal : jouer

Recensement des verbes reliés par recherche des relations suivantes :
– en relation avec le focus : DEEPSUBJ / DEEPOBJ / VMOD(**VERBE**, Éric Cantona)
– en relation avec le type attendu : DEEPSUBJ(**VERBE**, club)
– en relation avec un candidat-réponse trouvé par la stratégie B : DEEPSUBJ / DEEPOBJ / VMOD(**VERBE**, Manchester United)

Exemples de passages contenant ces relations (pas forcément des passages-réponses) :
– Éric Cantona a mené une carrière itinérante qu'il doit autant à son talent qu'à son caractère ombrageux.
– À Beaupréau, Éric Cantona joue la discrétion.
– Il devient champion de France en 1989 avec l'Olympique de Marseille, qu'il quitte en cours de saison pour aller jouer à Bordeaux puis à Montpellier, club avec lequel il remporte la Coupe de France en 1990.

Exemples de verbes extraits en relation avec le focus ou le type attendu :
– DEEPSUBJ(mener, Éric Cantona)
– DEEPSUBJ(jouer, Éric Cantona)
– DEEPOBJ(jouer, club)

RECENSEMENT DES NOMS RELIÉS DIRECTEMENT. Les noms recensés à cette étape sont ceux qui se trouvent dans des dépendances liées directement au focus et au verbe principal de la question (ceux reliés au type attendu de la réponse ont déjà été recensés comme candidat-réponse avec la stratégie B précédente). On retrouve donc les relations citées précédemment auxquelles s'ajoutent les relations de modifieur du nom, complément du nom et définition. Ce qui donne pour ce recensement :

Question : *Dans quels clubs a joué Éric Cantona ?*
– Type attendu : club
– Focus : Éric Cantona
– Verbe principal : jouer

Recensement des noms reliés par recherche des relations suivantes :
– avec le verbe principal : DEEPSUBJ / DEEPOBJ / VMOD(jouer, **NOM**)
– en subordonnant du focus : DEFINITION / NMOD(Éric Cantona, **NOM**)
– en gouverneur du focus : DEFINITION / ATTRIBUT_DE / NMOD(**NOM**, Éric Cantona)
– avec un verbe ayant été en relation avec le focus (verbes recensés précédemment) : VMOD, OBJ(verbe, **NOM**) et au moins un relation DEEPSUBJ(verbe, **Éric Cantona**) dans les snippets

Exemples de passages contenant ces relations (pas forcément des passages-réponses) :
– Éric Cantona joue l'étalon sur la plage de M. Hulot. Éric Cantona a mené une carrière itinérante qu'il doit autant à son talent qu'à son caractère ombrageux.
– Il devient champion de France en 1989 avec l'Olympique de Marseille, qu'il quitte en cours de saison pour aller jouer à Bordeaux puis à Montpellier, club avec lequel il remporte la Coupe de France en 1990.
– Ferguson, Rooney, MU, City, son avenir : les confidences d'Éric Cantona.

Exemples de noms extraits en relation avec le verbe principal :
– DEEPOBJ(jouer, **étalon**)
– VMOD(jouer, **Bordeaux**)
– VMOD(jouer, **Montpellier**)
Exemples de noms extraits en relation avec le focus :
– NMOD(**news**, Éric Cantona)
– NMOD(Éric Cantona, **culture**)
– ATTRIBUT_DE(**confidence**, Éric Cantona)
– DEEPOBJ(mener, **carrière**)

Les noms en relation avec le verbe principal vont notamment nous servir à renforcer la validité du type attendu. Par exemple, les trois noms *étalon*, *Montpellier* et *Bordeaux* sont en relation avec le verbe principal de la question et sont donc potentiellement des clubs. Nous verrons ce point plus tard.

Pour des raisons de temps d'exécution, il n'a pas été possible d'étendre cette similarité contextuelle à des instances du même type que le focus.

4.1.2.3 *Étape 3 : recherche de la forme affirmative de la question*

Cette étape reconstruit la forme affirmative de la question puis cherche cette forme dans les documents, à savoir des dépendances syntaxiques entre le verbe principal, le focus de la question et un éventuel candidat-réponse. Par exemple :

Question : *Dans quels clubs a joué Éric Cantona ?*
Passage-réponse : Éric Cantona a joué à Montpellier en 89-90, il y a remporté la Coupe de France et inscrit 10 buts en championnat.

Dépendances syntaxiques trouvées dans le passage :
– DEEPSUBJ(jouer, Éric Cantona)
– VMOD(jouer, **Montpellier**)
Candidat-réponse extrait : Montpellier

Nous ouvrons également la forme affirmative de la question aux verbes recensés précédemment comme étant reliés au focus, ce qui ramène évidemment beaucoup de bruit mais il est normalement éliminé lors de la validation du type des candidats (à cette étape, le type des candidats-réponses n'est pas encore vérifié). Par exemple :

Question : *Dans quels clubs a joué Éric Cantona ?*
Verbes recensés à l'étape 2 : jouer, mener
Passage : Éric Cantona a mené une carrière itinérante qu'il doit autant à son talent qu'à son caractère ombrageux.

Dépendances syntaxiques trouvées dans le passage :
– DEEPSUBJ(mener, Éric Cantona)
– DEEPOBJ(mener, **carrière**)
Candidat-réponse extrait : carrière

Cette approche avait notamment été utilisée par [Dumais *et al.*, 2002] avec des n-grammes des termes importants des questions puis en extrayant des phrases les candidats-réponses correspondant au type d'entité nommée attendu (date, nombre). Par exemple pour la question *How many times did Björn Borg win Wimbledon ?* [2], leur méthode avait renvoyé les candidats-réponses 5 et 37 :

– **Björn Borg** blah blah **Wimbledon** blah blah **5** blah ;
– **Wimbledon** blah blah blah **Björn Borg** blah **37** blah ;
– blah **Björn Borg** blah blah **5** blah blah **Wimbledon** ;
– **5** blah blah **Wimbledon** blah blah **Björn Borg**.

Le système choisit ensuite le candidat le plus redondant sur le web.

4.1.3 *Résolution temporelle*

Nous avons vu précédemment durant l'étude du corpus Frites que le temps était une notion très fréquente dans les phénomènes recensés, aussi bien en tant que critère variant qu'ancre contextuelle. Citron utilise donc des règles syntaxiques pour repérer des désignations d'expressions temporelles dans un passage-réponse quel que soit l'index

2. Combien de fois Björn Borg a-t-il remporté Wimbledon ?

dont il provient. Il s'appuie notamment sur les dépendances temporelles identifiées par la version Chronolines de XIP à savoir les dates, dates-événements ou encore les durées :

- DATE-EVENEMENT(Jeux Olympiques d'été 2012, 2012) : *Les **Jeux Olympiques d'été 2012** ont eu lieu à Londres.*
- TIMEX3(du 18 mars 1871 au 28 mai 1871), DUREE (un peu plus de deux mois) : *La Commune de Paris est une période insurrectionnelle de l'histoire de Paris qui dura **un peu plus de deux mois**, **du 18 mars 1871 au 28 mai 1871**.*

Un passage-réponse est toutefois fréquemment insuffisant pour situer dans le temps la ou les réponses qui en sont extraites, c'est notamment le cas pour les descriptions temporelles relatives, autrement dit celles nécessitant une ancre contextuelle. Il faut alors être capable d'explorer le document dans sa totalité notamment, par exemple, pour les articles de journaux publiés en ligne où la date est toujours mentionnée ainsi que celle de sa dernière mise-à-jour. 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 ». Citron possède donc un module pour normaliser certaines dates relatives à l'aide d'une date absolue servant d'ancre. En conditions idéales, il utilise des règles de normalisation écrites manuellement dans un premier temps puis l'API Joda-time [3][Colebourne et O'Neill, 2013] qui permet une manipulation aisée et harmonisée de données temporelles selon la norme ISO-8601. En conditions réelles, ces règles ne sont plus efficaces (bruit et imprécision), nous avons alors utilisé HeidelTime [4] [Strötgen et Gertz, 2013] qui extrait les dates d'un document au format TIMEX3 et normalise également les dates relatives en absolues s'il possède la date du document. Joda-time est toujours utilisé ensuite pour les calculs entre dates.

Lorsque la date du document n'est pas accessible dans le texte (manquante ou non trouvée), il reste la possibilité que la date de dernière modification ait été précisée en balise <meta> dans le code source. Cette possibilité est toutefois fortement sujette à caution et cette information est plus fiable sur des documents récents que sur des documents plus anciens. Durant notre étude des dates de documents dans le corpus Quaero, nous avons trouvé des pages clairement rédigées en 2004 mais qui possédaient une date de modification plus récente. L'étude des balises <meta> a montré que seulement $0,57$ % des documents possédaient cette information de dernière modification. Récupérer la date de dernière modification du document depuis le code source HTML est aussi sujet à caution car la modification par un webmestre du document HTML n'est pas forcément en rapport avec le passage-réponse extrait nous intéressant. Des tests en petite quantité ont rapidement montré que peu de pages étaient concernées et, couplé à l'incertitude sur

3. http://joda-time.sourceforge.net
4. http://code.google.com/p/heideltime/

cette donnée, nous n'avons pas poursuivi ce processus.

Nous nous sommes limité dans nos conditions idéales de développement aux expressions temporelles absolues sans prendre en compte les désignations d'événements. Ainsi, il n'y a pas de recherche temporelle sur la date absolue de la prise de la Bastille dans l'exemple suivant :

Question : *Quand s'est déroulée la Commune de Paris ?*
Passage-réponse : La Commune de Paris existait en fait depuis **la prise de la Bastille**.

Durant les développements en conditions réelles, il s'est avéré que, hors du cadre des conditions idéales, il était trop difficile de compter sur l'information de la date du document car cette dernière se révélait souvent absente, présente mais fausse ou présente mais à extraire parmi beaucoup de candidats. Par exemple, les articles de journaux, forums et blogs comportent toujours la date de publication mais cette dernière est à différencier des autres nombreuses dates présentes sur la même page à savoir la date d'inscription d'un utilisateur et la date de publication d'un commentaire. Nous n'avons donc pas développé notre propre module d'extraction de date absolue d'un document qui se révèle être une tâche trop importante concernant notre problématique des réponses aux questions-ARM. Nous avons cependant prévu de tester l'outil proposé par [Tannier, 2014] qui détecte la date de création des pages Web, en vue d'une future intégration dans Citron si les résultats s'y prêtent.

4.2 EXTRACTION DE RÉPONSES DEPUIS DES STRUCTURES ÉNUMÉRATIVES

Nous nous intéressons dans cette partie à l'extraction de candidats-réponses depuis des SE. Nous rappelons qu'il en existe trois types :

- les énumérations intra-phrastiques (stratégie D) ;
- les énumérations horizontales (stratégie E) ;
- les énumérations verticales (stratégie F).

Comme nous l'avons vu durant le détail du prétraitement du corpus, l'objectif de Kitten pour une énumération verticale est soit de la séquencer en plusieurs phrases syntaxiquement correctes, soit de l'aplanir en une énumération horizontale d'une seule phrase. Nous traitons ces deux cas de façon différente.

Pour l'analyse des structures énumératives, nous utilisons XIP pour :

1. l'analyse et l'extraction de candidats-réponses dans des énumérations horizontales et intra-phrastiques ;
2. l'analyse et l'extraction d'énumérations verticales, en s'appuyant sur le formatage produit par Kitten lors du prétraitement des documents

4.2.1 *Extraction depuis des énumérations intra-phrastiques*

4.2.1.1 *Détection des énumérations intra-phrastiques*

Les énumérations intra-phrastiques ne comportent pas d'amorce délimitée par le symbole « : » et ne dépasse pas le cadre de la phrase. Citron les extrait depuis les balises <text> et <StructEnum>. La stratégie D permet d'analyser avec XIP chaque phrase contenant le focus ou un verbe relié au focus recensé durant la stratégie C afin de trouver une conjonction de coordination ou un élément déclencheur d'une énumération.

LES CONJONCTIONS DE COORDINATION. Les conjonctions que nous avons choisi de considérer comme déclencheur d'énumération sont *et* et *ou*. En effet, les autres conjonctions rencontrées dans notre corpus ont une utilisation discursive (*donc*, *car*, *or*) ou introduisent une négation (*ni..., ni...*), voire les deux à la fois (*mais*), et il est difficile de les traiter correctement.

LES DÉCLENCHEURS D'ÉNUMÉRATIONS. XIP ne reconnaît pas certains marqueurs potentiels d'énumération comme *ainsi que*, *outre*, *en plus de*, *tout comme*. Nous avons donc créé des règles pour détecter ces déclencheurs d'énumération dans un premier temps, puis propager ensuite les relations de dépendance adéquates. Par exemple :

Passage : *Il existe cependant un risque que la France (**ainsi que** l'Allemagne et le Royaume-Uni) ne cède aux pressions exercées par les États-Unis.*
Dépendance trouvée par XIP : DEEPSUBJ(cède, France)
Nouvelles dépendances créées après détection du déclencheur :
– DEEPSUBJ(cède, Allemagne)
– DEEPSUBJ(cède, Royaume-Uni)

Nous avons toutefois restreint l'étendue des déclencheurs.

Notre première restriction a été de ne pas traiter les déclencheurs fréquemment utilisés dans une SE dépassant le cadre de la phrase. Par exemple, une SE construite à l'aide des déclencheurs *d'abord*, *ensuite*, *enfin*, *dans un premier temps*, etc., a de forte chances de s'étendre sur plusieurs phrases, ce qui ferait sortir du cadre des énumérations intra-phrastiques.

Notre deuxième restriction a été d'éviter de détecter des déclencheurs manipulant des propositions, ainsi que ceux nécessitant un travail pouvant générer trop de bruit dans la propagation des relations de dépendances : *sans compter*, *par-dessus tout*, *finalement*. Notre objectif était de d'abord réussir à identifier et reconstruire les dépendances pour une énumération concernant des groupes nominaux ou prépositionnels. Pour cela, nous nous sommes concentrés sur certaines prépositions (*outre*), locutions conjonctives (*ainsi que*), locutions adverbiales (*en plus de*), tirées du Lexiconn [Roze *et al.*, 2012].

Nous avons recensé des exemples d'utilisations de ces déclencheurs puis écrit les règles XIP nécessaires pour ne détecter que les énumérations à base de groupes nominaux ou prépositionnels. Par exemple, nous avons exclu les utilisations introduisant une proposition. Ainsi, le premier exemple déclenche bien une énumération mais pas le second :

- *Elle aime les pommes, les poires, ainsi que les ananas.*
- *Ils ont été placé sous contrôle judiciaire, ainsi que l'avait demandé le parquet.*

Il existe toutefois une limitation à l'analyse syntaxique lorsque les éléments à coordonner nécessitent des ressources sémantiques pour désambiguïser le rattachement des items :

- (1) *En plus du PSG et de la ligue 1, Ibrahimović a conquis les arbitres.*
- (2) *En plus du PSG et de la ligue 1, les arbitres apprécient Ibrahimović.*

Si l'exemple (1) ne possède qu'une seule interprétation (*Ibrahimović a conquis les arbitres, le PSG et la ligue 1*), l'exemple (2) en possède deux :

- (a) Rattachement de la coordination à droite : *les arbitres apprécient Ibrahimović, le PSG et la ligue 1*
- (b) Rattachement de la coordination à gauche : *les arbitres, le PSG et la ligue 1 apprécient Ibrahimović*

Dans les configurations qui suivent, nous avons donc fait les choix de rattachement suivants selon la présence/localisation d'un adverbe (*aussi*, *également*) désambiguïsant le rattachement (les éléments entre parenthèses sont optionnels) :

- Rattachement de la coordination à droite :
  - *En plus du PSG et de la ligue 1, Ibrahimović a (aussi/également) conquis les arbitres.*
  - *En plus du PSG et de la ligue 1, les arbitres apprécient aussi/également Ibrahimović.*
- Rattachement de la coordination à gauche :
  - *En plus du PSG et de la ligue 1, Ibrahimović aussi/également a conquis les arbitres.*
  - *En plus du PSG et de la ligue 1, les arbitres (aussi/également) apprécient Ibrahimović.*

LIMITATIONS. Lors de l'écriture des règles XIP, nous n'avons pas considéré l'adverbe *respectivement* qui sert à relier des items d'une première énumération à ceux d'une seconde énumération. En effet, il nous est déjà difficile de segmenter correctement une énumération intra-phrastique, nous avons donc préféré ignorer ces cas, le risque de bruit généré en cas d'erreur nous paraissant trop important :

Exemple où la portée s'étale sur deux phrases :
En 2009, les dépenses de protection sociale étaient plus élevées en France qu'en Allemagne (écart de 1,7 point de PIB) : elles atteignaient 33,1 % du PIB français et 31,4 % du PIB allemand. La part de ces dépenses à la charge des administrations publiques était ***respectivement*** *de 23,9 % et 21,2 % du PIB.*

Enfin, toujours pour ces mêmes raisons de complexité, nous n'avons pas traité spécifiquement la coordination à redoublement [Mouret, 2007] (par exemple, *Paul a appris et l'espagnol, et l'italien, et le portugais*. *Paul aime et lire le journal et écouter la radio.*) mais elle est couverte par nos règles sur la coordination simple pour *et* et *ou*.

4.2.1.2 *Extraction des énumérations intra-phrastiques*

Une fois les énumérations intra-phrastiques détectées, Citron procède à l'extraction des items de deux façons : soit à l'aide des règles XIP implémentées si le patron d'une de ces règles est détecté, soit par la présence d'une conjonction de coordination et d'une amorce contenant l'enumeraTheme dans la phrase.

Concernant la première technique, la détection des énumérations intra-phrastiques propageant les dépendances amène certains candidats-réponses à être extraits par la stratégie C de similarité contextuelle, en l'occurrence l'étape qui consiste à rechercher la forme affirmative de la question :

Question : *Quels acteurs ont interprété le rôle de Batman ?*
Passage-réponse : Dans la série, Batman a été interprété successivement par Michael Keaton (Batman et Batman, le défi), Val Kilmer (Batman Forever), George Clooney (Batman & Robin) et Christian Bale (Batman Begins, The Dark Knight : Le Chevalier noir et The Dark Knight Rises).
Dépendances :
- DEEPOBJ(interpréter, Batman)
- DEEPSUBJ(interpréter, Michael Keaton) *// réécriture du passif en actif*
- COORDITEMS(Michael Keaton, Val Kilmer), COORDITEMS(Val Kilmer, George Clooney), ...
  // création de la relation par transitivité de la coordination
- DEEPSUBJ(interpréter, Michael Keaton), DEEPSUBJ(interpréter, Val Kilmer), ...

Candidats-réponses extraits : Michael Keaton, Val Kilmer, George Clooney, Christian Bale

La deuxième technique vise à extraire les groupes nominaux ou prépositionnels en présence d'une coordination. La ponctuation est gérée de la façon suivante :

- le contenu des parenthèses est ignoré dans une approche de simplification de phrases ;
- le symbole « : » conduit à l'arrêt de l'extraction.

Enfin, une fois ces groupes syntaxiques extraits, Citron recherche l'enumeraTheme puis, si nécessaire, le type d'entité nommée. La recherche de l'enumeraTheme est effectuée en priorité par la relation DEFINITION avec le premier item de l'énumération. S'il n'en existe pas et que le premier item est en relation objet avec un verbe de l'amorce, l'enumeraTheme est alors le sujet de ce verbe (cas du passif notamment suite à nos règles de réécritures).

L'exemple suivant est un exemple où nos règles XIP n'ont pas pu propagé la relation DEFINITION(apôtre, Jean) à *Luc*, *Marc* et *Matthieu* du fait de l'absence de coordination entre les deux derniers items. Cependant, notre approche permet ici d'analyser quand même cette phrase comme indiqué juste avant du fait de la dernière coordination *ou*. En effet, même si cette dernière ne s'applique pas aux apôtres mais aux participes *accompagné* et *représenté*, elle indique à Citron qu'il faut analyser cette phrase qui répond au patron d'une énumération intra-phrastique.

Question : *Quels sont les noms des Apôtres ?*
Passage-réponse : À noter dans les représentation religieuses et notamment sur les chaires les quatre apôtres évangélistes : **Jean** accompagné ou représenté par un aigle, **Luc** accompagné ou représenté par un taureau ailé, **Marc** accompagné ou représenté par un lion ailé - devenu le symbole de Venise où sont conservées ses reliques, **Matthieu** accompagné **ou** représenté par un ange.
Candidats-réponses extraits (tête du groupe nominal) : Jean, Luc, Marc, Matthieu
EnumeraTheme trouvé par la règle créant une relation définition entre un groupe nominal précédant un deux-point (ici la fin de l'amorce) et un groupe nominal le suivant (ici, le premier item) : DEFINITION(apôtre, Jean)

4.2.2 *Extraction depuis des énumérations horizontales*

Une énumération horizontale ne dépasse pas le cadre d'une phrase et comporte une amorce se terminant par le symbole « : » avec chacun de ses items délimité par une virgule ou un point-virgule. Elle peut être le fruit du prétraitement de Kitten ou non. Dans le premier cas, elle est balisée et indexée dans l'index des balises <StructEnum>. Dans l'autre cas, elle n'est pas balisée et est indexée dans l'index classique.

La stratégie E consiste pour Citron à rechercher, à l'aide de Lucene :

- les énumérations verticales devenues horizontales grâce à Kitten dans l'index des balises <StructEnum>. Pour ces SE, l'amorce est déjà détectée (balisée) et les items sont séparés par un point-virgule ou une virgule (selon la longueur médiane des items) ;
- les énumérations horizontales dans l'index classique. Pour les détecter, si une phrase contient un point-virgule (marque de fin d'un potentiel item) et que le symbole « : » est détecté avant le point-virgule (marque de fin d'une potentielle amorce), alors on considère que l'on a affaire à une SE et on la segmente : l'amorce est délimitée par le symbole « : » et les items par des point-virgules.

Ainsi, après cette étape, deux cas sont possibles : soit les items sont séparés par des points-virgules, soit ils sont séparés par des virgules.

Pour extraire les candidats-réponses des SE dont les items sont séparés par des points-virgules, les conditions de présence du focus et d'un enumeraTheme du type de réponse

attendu ou d'un enumeraTheme générique dans l'amorce sont appliquées. Pour cela, le premier groupe nominal (NP) de chaque item est extrait comme candidat-réponse.

Énumération horizontale provenant de l'index des balises <StructEnum> (horizontalisée par Kitten avec insertion de deux point-virgules) :
<StructEnum>
<SEamorce>*Autres villes*</SEamorce>*Cottbus (103 415 habitants) est la 2ème ville du Brandebourg après Potsdam. Elle est située à 100 km au nord-est de Berlin.* ; *Brandenburg an der Havel (73 339 habitants) est située à 100 km à l'ouest de Berlin et est traversée par la Havel. C'est une région de forêts et de lacs.* ; *Frankfurt an der Oder (62 392 habitants) est située à la frontière avec la Pologne, à laquelle elle est reliée par trois ponts. Elle est traversée par l'Oder et présente un paysage de prairies, de fôrets et de lacs.*
<StructEnum>

Énumération horizontale provenant de l'index texte :
Par exemple voici quelques distributions spécialisées environnement de bureau : *Ubuntu, éditée par Canonical Ltd qui est dérivée de Debian* ; *Mepis également basée sur Debian* ; *Zenwalk dérivée de Slackware* ; *Mandriva, dérivée de Red Hat, aujourd'hui éditée par la société française de même nom et impliquée dans plusieurs projets libres.*

Contrairement aux énumérations verticales avec répétition du lien syntaxique entre l'amorce et l'item, nous ne considérons pas pour le moment la possibilité d'extraire un verbe depuis une SE horizontale. En effet, la validation du type du candidat-réponse lorsque celui-ci est un verbe n'est pour le moment pas possible. Dans le cas des SE horizontales, pour extraire des verbes, il faut alors être certain de l'identification de l'enumeraTheme de la SE et de sa distribution aux items, en plus d'être certain qu'il s'agit bien d'une SE, ce qui n'est pas le cas pour le moment.

Il est toutefois intéressant de noter que beaucoup des enumeraTheme génériques (*étapes*, *principes*) peuvent introduire des verbes en item et donc répondre notamment à des questions complexes mais il s'agit de questions pour lesquelles Citron n'est pas encore suffisamment armé.

Dans le cas d'énumérations horizontales utilisant une virgule pour séparer les items, Citron ne considère que les snippets où le focus de la question est présent et où l'amorce contient un enumeraTheme du type de réponse attendu et non un enumeraTheme générique comme pour les items séparés par des points-virgules. En effet, nous avons constaté dans nos corpus que les potentielles énumérations horizontales dans l'index classique étaient extrêmement nombreuses (recherche du symbole « : »), notamment du fait de la présence nombreuse de menus, spams, listes de mots-clefs en HTML. Quand on est en présence de structures provenant de l'index classique contenant un « : » et des virgules, nous ne pouvons pas être certain d'être en présence d'une amorce (et donc d'une SE), nous avons donc préféré imposer la présence de l'exact enumeraTheme du type de réponse attendu. Pour cela, nous avons défini des règles XIP traitant la transitivité de la

relation DEFINITION. En effet, si une relation DEFINITION existe entre un type et un élément lui- même lié par une coordination à d'autres éléments, alors ce type est appliqué aux autres éléments. Comme précédemment, les candidats-réponses extraits par les règles XIP sont les têtes des premiers NP de chaque item.

Par exemple :

Énumération horizontale : *On connaît huit planètes : Mercure, Vénus, la Terre, Mars, Jupiter, Saturne, Uranus et Neptune.*
Dépendance XIP existante : DEFINITION(planète, Mercure)
Dépendances XIP créées par transitivité :
– DEFINITION(planète, Vénus)
– DEFINITION(planète, Terre)
– (...).
Candidats-réponses extraits : *Mercure*, *Vénus*, *Terre*, *Mars*, *Jupiter*, *Saturne*, *Uranus*, *Neptune*.

4.2.3 *Extraction depuis des énumérations verticales séquencées par Kitten*

Nous ne nous intéressons ici qu'aux énumérations verticales dont l'amorce et les items sont balisés grâce à Kitten.

Pour la stratégie F, Citron utilise le moteur de recherche Lucene pour trouver les énumérations verticales « séquencées » par Kitten dans l'index des balises <structEnum> : la SE balisée doit contenir le focus de la question et posséder plusieurs balises <SEAmorce> (la même amorce reliée à chaque item). Grâce au balisage de l'amorce dans les phrases séquencées, les items d'une SE sont facilement identifiables. Si l'item se compose de plusieurs phrases, seule la première est prise en compte :

– si l'item commence par un NP ne contenant pas de pronom, ce NP ext extrait comme candidat-réponse. Ce schéma a été rencontré dans la quasi-totalité des cas de notre corpus de développement et semble très fréquent pour les énumérations verticales ;
– si l'item commence par un verbe, la totalité de l'item est extrait. Comme le lien syntaxique est répété entre l'amorce et chaque item, on est ici certain que le verbe commençant l'item est un candidat-réponse pertinent.

Dans les exemples qui suivent, les candidats-réponses extraits des items sont soulignés :

Question : *Quelles sont les architectures possibles d'un système de télécommunications ?*
Énumération horizontale (dans le document source HTML) :
Un système de télécommunications peut avoir une architecture :
– **de** type « point à point », comme par exemple un câble hertzien ou optique, ou une liaison radiotéléphonique. Des répéteurs peuvent y être inclus pour amplifier et corriger les signaux ;
– **de** « diffusion », comme en télévision où un émetteur est reçu par des milliers de récepteurs ;
– **de** « collecte », comme en surveillance océanographique, où des centaines de capteurs sont reçus par un système central ;
– **en** structure de réseau, où un ensemble d'émetteurs et de récepteurs communiquent entre eux par des liaisons « étoilées » (topologie en étoile) ou « point à point ».

Séquencement en 4 phrases (dans le document texte en sortie de Kitten) :
<structEnum>
<SEamorce>*Un système de télécommunications peut avoir une architecture* </SEamorce> de type « **point à point** », comme par exemple un câble hertzien ou optique, ou une liaison radiotéléphonique. Des répéteurs peuvent y être inclus pour amplifier et corriger les signaux.
<SEamorce>*Un système de télécommunications peut avoir une architecture*</SEamorce> de « **diffusion** », comme en télévision où un émetteur est reçu par des milliers de récepteurs.
<SEamorce>*Un système de télécommunications peut avoir une architecture*</SEamorce> de « **collecte** », comme en surveillance océanographique, où des centaines de capteurs sont reçus par un système central.
<SEamorce>*Un système de télécommunications peut avoir une architecture* </SEamorce> en **structure de réseau**, où un ensemble d'émetteurs et de récepteurs communiquent entre eux par des liaisons « étoilées » (topologie en étoile) ou « point à point ».
</structEnum>

Dans cet exemple, l'amorce ne précise pas le nombre d'items de la SE mais le traitement reste identique lorsque cela se produit. Par exemple avec l'énumération verticale suivante :

Question : *Quels sont les cinq piliers de l'Islam ?*
Énumération verticale (dans le document source HTML) :
Le prophète de Dieu a dit : "L'Islam est basé sur ces cinq principes :
– **De** témoigner que nul autre que Dieu ne peut être adoré et que Mahomet est le prophète de Dieu,
– **D'**effectuer le prière obligatoire (consciencieusement et parfaitement),
– **De** jeûner pendant le mois de Ramadan,
– **De** payer la Zakat obligatoire (aumône),
– **D'e**ffectuer le Hajj (Pèlerinage à la Mecque)."

Séquencement en 5 phrases (dans le document texte en sortie de Kitten) :
<structEnum>
– <*SEamorce*>*Le prophète de Dieu a dit "L'Islam est basé sur ces cinq principes* </*SEamorce*> de témoigner que nul autre que Dieu ne peut être adoré et que Mahomet est le prophète de Dieu.
– <*SEamorce*>*Le prophète de Dieu a dit "L'Islam est basé sur ces cinq principes* </*SEamorce*> d'effectuer la prière obligatoire (consciencieusement et parfaitement).
– <*SEamorce*>*Le prophète de Dieu a dit "L'Islam est basé sur ces cinq principes*</*SEamorce*> de jeûner pendant le mois de Ramadan.
– <*SEamorce*>*Le prophète de Dieu a dit "L'Islam est basé sur ces cinq principes* </*SEamorce*> de payer la Zakat obligatoire (aumône).
– <*SEamorce*>*Le prophète de Dieu a dit "L'Islam est basé sur ces cinq principes* </*SEamorce*> d'effectuer le Hajj (Pèlerinage à la Mecque).
</structEnum>

Ici, si les cinq phrases séquencées sont toutes syntaxiquement complètes, aucune ne l'est toutefois sémantiquement si elles sont considérées individuellement. N'extraire qu'une seule de ces phrases (amorce et item) n'aurait pas de sens pour un utilisateur, par exemple :

Question : *Quels sont les cinq piliers de l'Islam ?*
Réponse : jeûner pendant le mois de Ramadan
Passage-réponse : Le prophète de Dieu a dit : "L'Islam est basé sur ces cinq principes de jeûner pendant le mois de Ramadan.

Pour ces énumérations verticales séquencées dont le nombre d'items est annoncé dans l'amorce, Citron extrait et valide tous les items à partir du moment où un seul des items valide le type attendu de la réponse et que le focus de la question est présent dans l'amorce.

Cet exemple illustre également l'utilisation des enumeraThemes génériques (cf. chapitre 4) : dans la question, le nombre de réponses attendues est de 5, le focus est *Islam* et le type attendu de la réponse est *pilier*. Citron recherche donc en priorité dans les amorces renvoyées par Lucene, la relation DETERM(piliers, cinq) mais trouve la relation DETERM(principes, cinq). Comme *principe* a été recensé comme un enumeraTheme générique et que le focus de la question est présent dans l'amorce, cette énumération verticale séquencée voit chacun de ses items être validé comme une réponse correcte. Nous reviendrons sur ces aspects de validation plus tard.

4.3 EXTRACTION DE RÉPONSES DEPUIS DES TABLEAUX

Nous souhaitons pouvoir utiliser le contenu de tableaux de données afin d'en extraire des candidats-réponses du type attendu par la question. Pour cela, nous avons vu au

chapitre précédent comment Kitten détecte ce type de tableau puis en extrait de façon guidée le contenu en une séquence de phrases balisée <tableauKitten>. Lucene indexe ensuite ces balises et Citron analyse à l'aide de XIP chaque phrase contenue dans ces balises pour en typer les cases données en fonction des cases entêtes et extraire ainsi des candidats-réponses du type attendu (stratégie G).

Nous avons vu au chapitre précédent l'exemple de la figure 4.3 dont une partie de l'extraction est :

(—)
Éric Cantona ;; Biographie ; Période pro. : 1983-1997 .
Éric Cantona ;; Biographie ; Poste : Attaquant .
Éric Cantona ;; Parcours junior ; SaisonsJunior : Club .
Éric Cantona ;; Parcours junior ; SaisonsJunior : Caillols .
Éric Cantona ;; Parcours junior ; 1981-1983 : AJ Auxerre .
Éric Cantona ;; Parcours professionnel 1 ; SaisonsPro : 1983-1988 / Éric Cantona ;;
Parcours professionnel 1 ; Club : AJ Auxerre / Éric Cantona ;; Parcours professionnel 1 ;
M. (B.) : 094 0(29) .
Éric Cantona ;; Parcours professionnel 1 ; SaisonsPro : 1985-1986 / Éric Cantona ;;
Parcours professionnel 1 ; Club : Martigues / Éric Cantona ;;
Parcours professionnel 1 ; M. (B.) : 015 00(4) . (—) Éric Cantona ;; Parcours professionnel
1 ; SaisonsPro : Total / Éric Cantona ;;
Parcours professionnel 1 ; Club : / Éric Cantona ;;
Parcours professionnel 1 ; M. (B.) : 445 (186) .

Éric Cantona		
Eric Cantona Cannes 2009.jpg Éric Cantona au Festival de Cannes 2009		
Biographie		
Nom	Éric Daniel Pierre Cantona	
Nationalité	France	
Naissance	24 mai 1966 (46 ans)	
Lieu	Marseille (Bouches-du-Rhône)	
Taille	1,88 m (6′ 2″)	
Période pro.	1983-1997	
Poste	Attaquant	
Parcours junior		
SaisonsJunior	**Club**	
	Caillols	
1981-1983	AJ Auxerre	
Parcours professionnel [1]		
SaisonsPro	**Club**	**M. (B.)**
1983-1988	AJ Auxerre	94 (29)
1985-1986	→ Martigues	15 (4)
1988-1991	Marseille	43 (14)
1989	→ Bordeaux	12 (6)
1989-1990	→ Montpellier	39 (14)
1990-1991	Nîmes Olympique	19 (4)
1991-1992	Leeds United	35 (13)
1992-1997	Manchester United	188 (82)
Total		**445 (166)**

FIGURE 4.3 : Exemple d'identification de cases d'un tableau de données par Kitten.

theme		
Eric Cantona Cannes 2009.jpg enteteFinaleInformative		
entete		
entete	donnees	
entete	donnees	
entete	donnees	
entete	donnees	
entete	donnees	
entete	donnees	
entete	donnees	
entete		
entete	**donnees**	
neutre	donnees	
entete	donnees	
entete		
entete	**entete**	**entete**
donnees	donnees	donnees
donnees	donnees	donnees
donnees	donnees	donnees
donnees	donnees	donnees
donnees	donnees	donnees
donnees	donnees	donnees
donnees	donnees	donnees
donnees	neutre	**donnees**

FIGURE 4.4 : Transcription des types trouvés par Kitten dans la figure 3.6 (les quatre erreurs sont entourées).

4.3.1 *Segmentation des phrases*

Les « phrases » du tableau sélectionné par Lucene sont composées quasi-exclusivement de groupes nominaux et c'est le formatage effectué par Kitten qui rend la phrase analysable. Nous utilisons XIP pour analyser chacune des phrases d'un tableau. En effet, nous avons créé des règles XIP guidant l'analyse grâce aux symboles de ponctuation ajoutés par Kitten : le « ; » pour repérer le thème, le « / » pour les relations entre cases entête et données et surtout le point final.

Nous avons toutefois rencontré un problème sur la segmentation en phrase de XIP qui considérait toujours un point à l'intérieur d'une parenthèse comme une fin de phrase. Nous avons contourné ce problème en supprimant les points à l'intérieur de parenthèses

pour les phrases extraites depuis des tableaux. Plusieurs modifications sont d'ailleurs appliquées pour permettre une meilleure analyse par XIP :

- suppression des crochets contenant uniquement un nombre : *[10]*. Il s'agit des notes de bas de pages et XIP ne distingue pas les parenthèses des crochets donc nous supprimons l'information ;
- suppression des informations *[vdm]* (légende d'un modèle Wikipédia pour *Voir, Discussions, Modifier*) et *modifier*.

Il reste une limitation de XIP que nous n'avons pas pu contourner à savoir la limitation à 150 mots maximum pour une phrase. Cette limite est très raisonnable avec une phrase structurée mais peut poser problème pour les tableaux possédant beaucoup d'informations. Nous n'avons toutefois pas rencontré ce problème pour le moment.

4.3.2 *Extraction des informations*

Nous avons créé 9 règles XIP pour procéder à l'extraction des candidats-réponses depuis une phrase issue d'un tableau de données. Ces règles ont pour seul objectif de typer les cases données avec la case entête, ainsi que la case thème si cette dernière est présente. Il y a 3 règles THEME et 6 règles DEFINITION dédiées au traitement de ces phrases qui permettent d'obtenir les relations suivantes :

Question : *Dans quels clubs a joué Éric Cantona ?*
Phrase extraite du tableau de données :
Éric Cantona ; ; Parcours professionnel 1 ; SaisonsPro : 1983-1988 / Éric Cantona ; ; Parcours professionnel 1 ; Club : AJ Auxerre / Éric Cantona ; ; Parcours professionnel 1 ; M. (B.) : 094 0(29) .
Dépendances XIP créées :
- THEME (Éric Cantona) ;
- DEFINITION (Club, AJ Auxerre) ;
- INTERVALLE (1983-1988) (comme il s'agit d'une date, la relation DEFINITION (SaisonsPros, 1983-1988) n'est pas gardée mais sert à créer la relation INTERVALLE).

Candidat-réponse extrait : *AJ Auxerre.*

4.4 VALIDATION DU TYPE DES CANDIDATS-RÉPONSES

Pour commencer, les candidats-réponses pour lesquels Citron ne valide pas le type sont :

- quand le type attendu de la réponse est un type d'entité nommée et que XIP a détecté un candidat-réponse de ce type (DATE, NOMBRE, PERSONNE, ORGANISATION et LOCALISATION) ;
- quand un candidat-réponse est issu d'une SE dont l'enumeraTheme a été validé.

Pour ces cas, nous considérons que les informations de type fournies par XIP ou par un enumeraTheme valide suffisent pour valider les types des candidats-réponses concernés.

Pour les autres candidats-réponses, leur type est validé de façon un peu différente selon qu'ils ont été extraits d'un tableau ou non. Nous commençons par présenter le cas général.

4.4.1 *Validation des candidats-réponses non-issus de tableaux*

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.

4.4.1.1 *Recherche de l'article Wikipédia*

Comme nous l'avons vu au chapitre 4, 27 index basés sur la première lettre du nom des articles Wikipédia ont été créés (les 26 lettres de l'alphabet et un index regroupant tous les articles qui ne commencent pas par une de ces 26 lettres).

Pour chaque candidat-réponse à valider, une recherche est effectuée à l'aide de l'outil *Grep* sur ces index. Nous avons choisi cette approche plutôt qu'une indexation/recherche avec un moteur de recherche pour nous assurer de trouver l'article dédié à ce candidat-réponse s'il existe.

La recherche d'un article peut toutefois se révéler infructueuse du fait de sa dénomination. Par exemple, pour la question *Dans quels clubs a joué Éric Cantona ?* dont le type de réponse attendu est *club*, la recherche d'un article *Manchester United* trouve bien la page *Manchester United Football Club* alors qu'une recherche de *Bordeaux* renvoie l'article sur la ville et on passe à côté de l'article sur le club qui permettrait de valider le type du candidat. Nous avons tenté de rechercher à la fois le type attendu (*club* dans cet exemple) et le candidat-réponse (*Bordeaux*) dans les titres de page mais ce procédé est à la fois très coûteux du fait du parcours de tous les index et génère beaucoup de bruit, confirmant ce que Lucene aurait probablement produit avec des requêtes sur le champ titre des articles.

4.4.1.2 *Analyse de l'introduction*

Si l'article Wikipédia correspondant au candidat-réponse existe, l'introduction est analysée syntaxiquement par XIP pour vérifier si le type du candidat est bien conforme au type attendu de la réponse. Pour valider le type d'un candidat, Citron utilise les mêmes

règles de recherche des relations DEFINITION que celles utilisées pour la recherche des candidats-réponses, auxquelles s'ajoutent 4 règles supplémentaires dédiées au format de définition souvent rencontré dans Wikipédia. Par exemple :

- cas d'une apposition :
  Saturne, planète correspondant en alchimie au plomb → DEFINITION(Saturne, planète)
- cas de plusieurs appositions :
  Mercure, première planète du système solaire, la plus proche du Soleil
  → DEFINITION(Mercure, planète)
- cas des incises :
  Venise ((Venezia), (Venexia)), surnommée la Cité des Doges ou la Sérénissime
  → DEFINITION(Venise, Venezia), DEFINITION(Venise, Venexia) ;
- cas des parenthèses vides suite aux prétraitements :
  Matthias (), apôtre qui... → DEFINITION(Matthias, apôtre)

Pour les premières phrases des articles, nous tolérons également une portée dans la chaîne de coréférence en cas d'utilisation de pronom personnel ou possessif de troisième personne ou d'un déterminant défini en position sujet, par exemple :

Question : *Quels sont les fruits à consommer en automne ?*
Candidat-réponse à valider sur le type *fruit* : ananas
Introduction de l'article *Ananas* sur Wikipédia : L'**ananas** (Ananas comosus) est une plante xérophyte, originaire d'Amérique du Sud (nord du Brésil), d'Amérique centrale, et des Antilles. **Il** est connu principalement pour **son** fruit comestible, qui est en réalité une intrutescence.
Dépendances DEFINITION extraites : DEFINITION(ananas, plante), DEFINITION(ananas, **fruit**)

Question : *Quelles villes ont été la capitale de l'Allemagne ?*
Candidat-réponse à valider sur le type *capitale* : Bonn (XIP a validé le type d'entité nommée *ville*)
Introduction de l'article *Bonn* sur Wikipédia : Bonn est une ville d'Allemagne située au bord du Rhin dans le sud du Land de Rhénanie-du-Nord-Westphalie, à 25 km au sud de Cologne et 54 km au nord de Coblence. Entre 1949 et 1990, **la ville** était la capitale de la République fédérale d'Allemagne.
Dépendances DEFINITION extraites : DEFINITION(Bonn, ville), DEFINITION(Bonn, capitale)

4.4.1.3 *Traitement de l'homonymie*

Si une page d'homonymie existe dans Wikipédia 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)*. Pour la question *Dans quels clubs ...*, c'est la définition contenant le mot « club » qui sera retenue pour valider le candidat « Chelsea ».

Quatre règles *DEFINITION-WIKIHOMONYMIE* ont été créées spécialement pour gérer les définitions sur les pages d'homonymie de Wikipédia. Ces règles produisant trop de bruit à cause de la simple juxtaposition d'un groupe nominal en début de phrase, les relations DEFINITION-WIKIHOMONYMIE ne sont pas récupérées lorsqu'une relation DEFINITION est recherchée :

Question : *Quels sont les planètes du systèmes solaire ?*
Candidat-réponse à valider sur le type *planète* : Mars
Extrait de la page d'homonymie *Mars* : Mars est un nom propre ou commun et un acronyme qui peut désigner :
– Mars, dieu de la mythologie romaine
– Mars, troisième mois de l'année dans les calendriers occidentaux (grégorien, julien, ...).
– Martin Mars, un hydravion.
– Mars, quatrième planète du système solaire (voir également les catégories Mars sur Wikipedia ou Mars sur Commons).
Dépendances présentes : DEFINITION-WIKIHOMONYMIE(Mars, dieu)
DEFINITION-WIKIHOMONYMIE(Mars, mois)
DEFINITION-WIKIHOMONYMIE(Mars, hydravion)
DEFINITION-WIKIHOMONYMIE(Mars, **planète**)

4.4.1.4 *Cas de plusieurs types à valider*

L'analyse de la question peut révéler plusieurs types à valider comme dans la question *Quelles villes ont été capitales de l'Allemagne ?* vue précédemment, où la réponse doit être une ville et une capitale. Chacun des types est alors vérifié à l'aide de Wikipédia pour voir si l'un est l'hyponyme de l'autre, ce qui est le cas ici puisqu'une capitale est une ville. Les candidats-réponses ne sont donc validés que sur le type *capitale*.

4.4.1.5 *Validation par les dépendances syntaxiques*

Si la question possède des spécifieurs concernant le type attendu de la réponse, ces derniers sont également recherchés dans l'introduction. Les adjectifs, les modifieurs de noms et les compléments de noms sont les trois catégories ciblées. Ainsi pour la question *Quelles villes ont été capitales de l'Allemagne ?*, le type à valider est *capitale* mais il faut aussi trouver dans l'introduction de l'article du candidat-réponse une validation pour *capitale de l'Allemagne* ou *capitale allemande*.

Citron recherche alors le spécifieur dans les dépendances du candidat-réponse, à savoir une relation adjectivale (*ATTRIBUTADJ(capitale, allemande)*) ou un complément de

nom (*ATTRIBUT_DE(capitale, Allemagne)*). Les lemmes *allemand* et *Allemagne* sont ensuite comparés au lemme spécifieur de la question *Allemagne* par la distance de Jaro-Winkler présentée dans la prochaine sous-section. Par contre, nous n'avons pas encore implémenté de réconciliation de référence à ce niveau : *capitale de la RDA* serait invalidé sans vérifier si RDA et Allemagne désignent la même entité.

4.4.2 *Validation des candidats-réponses issus d'énumérations*

Les stratégies D et E extraient des candidats-réponses issus respectivement des énumérations intra- phrastiques/horizontales et verticales séquencées par Kitten. Pour chaque SE détectée, si la validation par Wikipédia valide plus de la moitié des items, alors Citron valide tous les items de la SE. Nous considérons en effet que l'utilisation d'un enumeraTheme qui type chacun des items d'une SE permet de le faire : les cas d'énumérations comportant des items de type différents mais partageant un même enumeraTheme n'ont pas été rencontrés dans notre corpus.

4.4.3 *Validation des candidats-réponses issus de tableaux*

Nous avons vu au chapitre précédent que Kitten obtient de bons scores sur le traitement des tableaux grâce à l'apprentissage supervisé mais nous avons constaté qu'en conditions réelles, certaines erreurs se produisent dans l'étiquetage des cases ce qui peut fausser l'extraction et le typage des candidats-réponses.

Le type des candidats-réponses est donc soumis à une validation supplémentaire : ainsi, rencontrer un patron *Type : candidat-réponse* est donc nécessaire mais pas suffisant. Les candidats-réponses extraits depuis des tableaux sont donc soumis dans un premier temps à une validation de type en utilisant Wikipédia comme présentée précédemment.

Pour les candidats-réponses non validés par Wikipédia, deux techniques supplémentaires sont mises en œuvre pour tenter de valider leur type : (1) l'utilisation des verbes recensés par similarité contextuelle (stratégie C) et, pour les candidats non-validés par cette étape, (2) la comparaison syntaxique des phrases dont ils ont été extraits avec les phrases du même tableau contenant un candidat-réponse validé. La validation du type par une seule de ces deux techniques suffit à valider un candidat-réponse.

4.4.3.1 *Validation d'un candidat par similarité contextuelle*

Lors de la stratégie C, pour la question *Dans quels clubs a joué Eric Cantona ?*, Citron avait extrait comme candidats-réponses les noms étant objets du verbe principal *jouer* et avait trouvé *étalon*, *Bordeaux* et *Montpellier*. Puisque ces trois noms ont été recensés dans ce contexte, ils sont potentiellement des clubs mais l'utilisation de Wikipédia n'a pas permis de valider leur type.

Lors de la stratégie G, *Bordeaux* et *Montpellier* ont été détectés comme des *clubs* grâce à l'entête du tableau :

(—)
Éric Cantona ;; *Parcours professionnel 1* ; *SaisonsPro* : *1989* / *Éric Cantona* ;; *Parcours professionnel 1* ; *Club* : *Bordeaux* / *Éric Cantona* ;; *Parcours professionnel 1* ; *M. (B.)* : *12 0(6)* .

Éric Cantona ;; *Parcours professionnel 1* ; *SaisonsPro* : *1989-1990* / *Éric Cantona* ;; *Parcours professionnel 1* ; *Club* : *Montpellier* / *Éric Cantona* ;; *Parcours professionnel 1* ; *M. (B.)* : *39 0(14)* .

Les deux conditions étant réunies (les candidats *Bordeaux* et *Montpellier* apparaissent tous les 2 dans le contexte du verbe *jouer* et dans un tableau dont la case entête valide le type), Citron considère qu'ils sont valides. En revanche, le candidat-réponse *étalon* extrait lors de la stratégie C n'est pas validé puisqu'il n'est pas présent dans le tableau sous la forme *Club : étalon*.

4.4.3.2 *Validation d'un candidat par similarité de phrase extraite du tableau*

Une détection de similarité est effectuée sur toutes les « phrases » d'un tableau contenant des candidats- réponses : si une phrase contenant un candidat-réponse invalidé par Wikipédia est similaire à 95 % (distance de Jaro-Winkler que nous détaillons plus loin) à une phrase dont le candidat-réponse a été validé, alors le candidat- réponse invalidé dans un premier temps est validé. Cette approche repose sur le fait que si deux phrases extraites d'un même tableau possède les mêmes entêtes et que l'une de ces phrases a une case entête qui confirme le type d'un candidat-réponse déjà validé par ailleurs, alors tous les candidats-réponses associés à cette case entête sont du même type et peuvent être validés. Par exemple :

Question : *Dans quels clubs a joué Éric Cantona ?*
Phrases extraites du tableau de données :
Éric Cantona ;; Parcours professionnel 1 ; SaisonsPro : 1983-1988 / Éric Cantona ;; Parcours professionnel 1 ; Club : AJ Auxerre / Éric Cantona ;; Parcours professionnel 1 ; M. (B.) : 094 0(29).
Éric Cantona ;; Parcours professionnel 1 ; SaisonsPro : 1985-1986 / Éric Cantona ;; Parcours professionnel 1 ; Club : Martigues / Éric Cantona ;; Parcours professionnel 1 ; M. (B.) : 015 00(4).
Éric Cantona ;; Parcours professionnel 1 ; SaisonsPro : 1988-1991 / Éric Cantona ;; Parcours professionnel 1 ; Club : Marseille / Éric Cantona ;; Parcours professionnel 1 ; M. (B.) : 043 0(14).
Éric Cantona ;; Parcours professionnel 1 ; SaisonsPro : 1989 / Éric Cantona ;; Parcours professionnel 1 ; Club : Bordeaux / Éric Cantona ;; Parcours professionnel 1 ; M. (B.) : 012 00(6).
Éric Cantona ;; Parcours professionnel 1 ; SaisonsPro : 1989-1990 / Éric Cantona ;; Parcours professionnel 1 ; Club : Montpellier / Éric Cantona ;; Parcours professionnel 1 ; M. (B.) : 039 0(14).
Éric Cantona ;; Parcours professionnel 1 ; SaisonsPro : 1990-1991 / Éric Cantona ;; Parcours professionnel 1 ; Club : Nîmes Olympique / Éric Cantona ;; Parcours professionnel 1 ; M. (B.) : 19 00(4).
Éric Cantona ;; Parcours professionnel 1 ; SaisonsPro : 1991-1992 / Éric Cantona ;; Parcours professionnel 1 ; Club : Leed United / Éric Cantona ;; Parcours professionnel 1 ; M. (B.) : 35 0(13).
Éric Cantona ;; Parcours professionnel 1 ; SaisonsPro : 1992-1997 / Éric Cantona ;; Parcours professionnel 1 ; Club : Manchester United / Éric Cantona ;; Parcours professionnel 1 ; M. (B.) : 188 0(82).
Candidats-réponses dont le type est déjà validé par Wikipédia :
Manchester United, Leeds United, Nîmes Olympique, AJ Auxerre

Candidats-réponses extraits par similarité : Martigues, Marseille, Bordeaux

Nous verrons toutefois dans le chapitre suivant que cette approche peut parfois valider un candidat-réponse invalidé à raison par Wikipédia.

4.5 AGRÉGATION DE RÉPONSES ET CRITÈRE VARIANT

Comme nous l'avons vu au chapitre 2, les campagnes d'évaluation en question-réponse utilisent un protocole permettant de proposer plusieurs réponses individuelles par question, même pour les questions n'étant pas de type liste. La mesure couramment utilisée (MRR) récompense plus les SQR ayant renvoyé une réponse correcte aux premiers rangs. Les SQR utilisent donc un système d'ordonnancement de leurs candidats-réponses afin de mettre en tête de liste ceux dont ils sont le plus sûrs. Pour les SQR probabilistes, l'ordonnancement repose sur la probabilité [Schlaefer *et al.*, 2007], Ko *et al.* [2007]), pour les SQR symboliques, une pondération des candidats (sur les termes de la question présents dans le passage, la redondance, les dépendances) est généralement toujours effectuée ([Sun *et al.*, 2005], [Moriceau et Tannier, 2010]).

Mais avant un éventuel ordonnancement des réponses, Citron va les agréger c'est-à-dire regrouper les réponses faisant référence à une même entité, l'objectif de Citron étant de fournir le plus d'entités correctes différentes, et pour chacune d'elle le plus de réponses individuelles correctes différentes. La quasi-totalité des SQR ayant participé aux campagnes TREC comportant des questions-listes disent avoir utilisé un système de détection de doublons mais peu l'ont détaillé, ni même précisé ce qu'était un doublon. Nous posons la définition qu'un doublon concerne une forme de surface identique ou la référence à une même entité.

Citron utilise des algorithmes classiques en TAL pour agréger les réponses : nous n'avons pas voulu nous appuyer sur une base de connaissances ou le web sémantique afin d'effectuer de la réconciliation de référence profonde. Citron reste sur une réconciliation de référence surfacique en détectant les acronymes, les plus longues chaînes communes et les similarités lexicales.

4.5.1 *Réconciliation de référence surfacique*

Citron effectue cette réconciliation sur tous les candidats-réponses validés dans le cas de questions n'attendant pas un type DATE ou NOMBRE. En effet, les candidats-réponses de ces deux types d'entité nommées sont souvent très proches graphiquement (*11 millions* et *1 million*) or nous ne proposons ici que des méthodes d'agrégation des réponses qui s'appuient sur leur forme de surface. Ces méthodes risqueraient d'agréger des réponses de ce type qui ne devraient pas l'être.

Dans un premier temps, nous avons essayé de regrouper les candidats-réponses avant validation puisqu'une variation d'orthographe ou un acronyme pouvait ne pas être validé par Wikipédia du fait de l'absence de l'article avec une telle orthographe. Les résultats ont produit trop de bruit et nous avons ainsi choisi de n'agréger que les candidats validés. Pour cela, nous utilisons trois approches sur la totalité des candidats-réponses validés : la détection d'acronyme, la plus longue chaîne commune et la distance de Jaro-Winkler. Nous obtenons alors jusqu'à trois instances de recoupement au maximum [5] que nous agrégeons alors à nouveau.

DÉTECTION D'ACRONYME. La détection d'acronyme est une tâche très importante dans la réconciliation de références. Beaucoup de questions de type définition proposées durant les campagnes d'évaluation concernent d'ailleurs cette tâche, par exemple : *Que signifie l'acronyme OPAC ?*. Citron applique une approche classique à base d'expressions régulières sur tout candidat-réponse en majuscule qui ne contient pas de nombre et dont la longueur est inférieure ou égale à cinq caractères (pour éviter de chercher des acronymes sur des noms simplement écrits en majuscule, par exemple sur des forums

5. Par exemple, il peut ne pas y avoir d'acronyme ou bien deux chaînes peuvent ne pas dépasser le seuil imposé pour la distance de Jaro-Winkler.

de discussion). Quand un candidat-réponse est sous forme d'acronyme, Citron recherche donc parmi tous les candidats-réponses la ou les chaînes de caractères pouvant correspondre à cet acronyme[6]. Ainsi trois candidats-réponses sont agrégés à une même entité *PSG* : *PSG*, *Paris-Saint Germain*, *Paris SG*.

PLUS LONGUE CHAÎNE COMMUNE. La plus longue chaîne commune est la chaîne de caractères la plus longue contenant les éléments recherchés de deux chaînes de caractères différentes. Elle peut être stricte (la chaîne recherchée doit être exactement commune aux deux chaînes sources (sous-chaîne) ou plus souple (les éléments recherchés sont des mots, des stemmes ou des lemmes, peut tolérer des mots-vides). Elle est couramment utilisé par les SQR. Citron agrège ainsi *FC Trappes* et *FC Trappes-St Quentin*.

DISTANCE DE JARO-WINKLER. Lors de l'utilisation de données issues du Web, il est plus fréquent de rencontrer des fautes de frappe que dans des corpus de textes de lois ou d'articles de journaux. Cependant, des variations lexicales très légères peuvent également être dues à des usages culturels différents. Nous cherchons donc à rassembler des données très proches lexicalement et pour cela, il existe plusieurs mesures de similarité entre deux chaînes de caractères comme Levenshtein, Jaro, Jaro-Winkler, cosinus. Citron utilise la distance de Jaro-Winkler qui se prête mieux aux chaînes relativement courtes. Cette distance est calculée à partir de la distance Jaro d_j, par exemple pour deux chaînes de caractères s_1 et s_2 :

$d_j = \frac{1}{3}(\frac{m}{|s_1|} + \frac{m}{s_2} + \frac{m-t}{m})$

où :

- $|s_i|$ est la longueur de la chaine de caractères s_i ;
- m est le nombre de caractères *correspondants*. Si m est nul alors d_j est nul. On pose que deux caractères identiques des chaines s_1 et s_2 sont correspondants si la différence entre leurs positions dans leurs chaînes respectives ne dépasse pas $\lfloor \frac{max(|s_1|,|s_2|)}{2} - 1 \rfloor$;
- t est le nombre de transpositions.

La distance de Jaro-Winkler d_{jw} entre s_1 et s_2 est alors : $d_{jw} = d_j + (l_p(1 - d_j))$
où :

- d_j est la distance calculée précédemment ;
- l est la longueur du préfixe commun (maximum 4 caractères) ;
- p est un coefficient favorisant les chaînes avec un préfixe commun. Nous avons gardé la valeur de $0,1$ proposée par Winkler.

6. En autorisant entre chaque lettre majuscules tous les caractères sauf la virgule et le point. Si une chaîne de caractère correspond, son nombre d'espace (segmentation basique) ne doit alors pas excéder la longueur de l'acronyme plus un.

Nous avons fixé expérimentalement un seuil de similarité à $0,85$, ce qui donne par exemple une agrégation entre *Real de Madrid* et *Real Madrid*.

4.5.2 *Réconciliation de référence temporelle*

NORMALISATION. Citron effectue également une agrégation concernant les candidats-réponses à une question temporelle. Nous nous intéressons ici aux réponses de type DATE et non aux événements. Dans un premier temps ces candidats sont normalisés avec HeidelTime : si la date du document est disponible alors les dates relatives sont normalisées en dates absolues. Lorsque la date du document n'est pas disponible, la date relative est normalisée seulement en partie : *8 mai* devient *XXXX*-05-08.

AGRÉGATION SUR L'ANNÉE. Dans un second temps, un travail de réconciliation temporelle est effectué afin d'agréger les candidats-réponses. Cette agrégation est actuellement effectuée sur l'année puis, une fois que les différentes années sont identifiées, nous y attachons les dates relatives si aucune ambiguïté n'est possible. Par exemple, voyons le traitement de Citron pour la question *Quand s'est déroulée la Commune de Paris ?* : les entités temporelles (années) identifiées sont en gras et pour chaque entité, on donne les candidats-réponses agrégés correspondants. On y voit notamment que les dates délimitant un intervalle sont rattachées aux deux années correspondant aux deux bornes de l'intervalle. La date relative *26 mars* est rattachée à l'entité *1871* car une réponse (*du 26 mars au 20 mai 1871*) contient aussi cette date. S'il y avait eu plusieurs années possédant la réponse *26 mars*, chacune des entités l'aurait reçue, c'est le cas notamment pour les intervalles. Une date relative peut n'être agrégée à aucune entité temporelle.

Question : *Quand s'est déroulée la Commune de Paris ?*
Candidats-réponses : juillet 1789-octobre 1795, le 14 juillet 1789, du 21 mai 1790, en 1792, de l'été 1792 à l'été 1794, Juillet 1792, 26 mars, du 26 mars au 20 mai 1871, du 18 mars au 28 mai 1871, en 1871, de mars à mai 1871
Agrégation des candidats-réponses sur l'année :
- **1789** : *juillet 1789-octobre 1795*, *le 14 juillet 1789* ;
- **1790** : *du 21 mai 1790* ;
- **1792** : *en 1792*, *de l'été 1792 à l'été 1794*, *Juillet 1792* ;
- **1794** : *de l'été 1792 à l'été 1794* ;
- **1795** : *juillet 1789-octobre 1795* ;
- **1871** : *26 mars*, *du 26 mars au 20 mai 1871*, *du 18 mars au 28 mai 1871*, *en 1871*, *de mars à mai 1871*.

SEGMENTATION EN BLOCS TEMPORELS. Enfin, une fois les candidats-réponses agrégés en des entités temporelles (années) distinctes, ces dernières sont délimitées de façon à obtenir des blocs temporels pertinents sur l'année. Pour cela, Citron calcule la distance

moyenne entre les années et regroupe dans un même bloc deux entités/années si leur distance est inférieure à la distance moyenne. Nous obtenons ainsi deux blocs avec l'exemple précédent :

- **1789-1795** : juillet 1789-octobre 1795, le 14 juillet 1789, du 21 mai 1790, en 1792, de l'été 1792 à l'été 1794, Juillet 1792, de l'été 1792 à l'été 1794, juillet 1789-octobre 1795 ;
- **1871** : 26 mars, du 26 mars au 20 mai 1871, du 18 mars au 28 mai 1871, en 1871, de mars à mai 1871.

Il existe un cas pour lequel le calcul des blocs n'est pas effectué : lorsque tous les candidats-réponses à l'intérieur de tous les blocs sont plus précis que l'année. Nous considérons alors que le découpage en bloc est obtenu dès la phase d'agrégation temporelle. Par exemple pour la question *Quand la France a-t-elle perdu son triple-A ?*, deux entités temporelles correspondent à un jour précis d'une année et la troisième à un mois précis d'une année :

- **2012-01-13** : vendredi, vendredi soir
- **2011-07** : en juillet
- **2012-11-19** : lundi soir, le 19.11.2012, lundi, Novembre 19

4.5.3 *Critère variant*

Pour détecter un éventuel critère variant des réponses, Citron se concentre sur le type d'entité nommées présentes dans la phrase contenant le candidat-réponse, ainsi que sur ses spécifieurs.

Pour cela, Citron extrait d'abord toutes les entités nommées temporelles et leur applique la normalisation vue précédemment. S'il n'y a qu'une seule date et qu'elle est présente avant le premier candidat-réponse de la phrase, alors on considère que cette date s'applique à tous les candidats de la phrase. Lorsque plusieurs dates sont extraites, chaque date ne s'applique alors qu'au candidat-réponse le plus proche (avec une priorité à gauche lors que deux candidats-réponses lui sont équidistants).

Question : *Quels sont les pays de la zone Euro ?*
Passage-réponse : Au 1er janvier 2011 , les 17 pays de l' Union européenne ayant adopté l' euro et constituant la " zone euro " sont l' Allemagne , l' Autriche , la Belgique , Chypre (partie grecque) , l' Espagne , l' Estonie , la Finlande , la France , la Grèce , l' Irlande , l' Italie , le Luxembourg , Malte , les Pays-Bas , le Portugal , la Slovaquie et la Slovénie.
Réponses extraites accompagnées de l'entité nommée TIMEX3 :
- Allemagne (Au 1er janvier 2001) ;
- Autriche (Au 1er janvier 2001) ;
- (—).

Citron peut également extraire d'autres entités nommées (LOCATION) comme critère variant si elles ne sont pas du type du candidat-réponse. Les entités nommées du type NUMBER, PERSON ou ORGANISATION peuvent être identifiées comme critère variant seulement si leur nombre dans la phrase est égal à 1.

Question : *Quand la deuxième Guerre mondiale s'est-elle terminée ?*
Passage-réponse : La Seconde Guerre mondiale se termine officiellement en Europe le 8 mai 1945, à 23h01.
Passage-réponse : Le 2 septembre 1945, le Japon capitule marquant la fin de la Deuxième Guerre mondiale.
Critère variant géographique découvert par extraction de l'unique lieu mentionné dans chaque passage : Europe, Japon
Réponses proposées : Le 8 mai 1945, à 23h01 (Europe), le 2 septembre 1945 (Japon)

Citron extrait également les spécifieurs en relation syntaxique directe avec un candidat-réponse que sont les attributs adjectivaux et nominaux (souvent également des entités nommées : couleurs, nationalités, ordinaux) et les compléments de noms. Par exemple :

Question : *Quand s'est déroulée la Commune de Paris ?*
Passage-réponse : La **première** Commune de Paris en 1792 est un des phénomènes les plus intéressants de la Révolution française, d'un point de vue libertaire.
Passage-réponse : Le 9 août 1792, se forme la Commune **insurrectionnelle**, avec des délégués de toutes les sections parisiennes.
Passage-réponse : La **seconde** Commune de Paris, un gouvernement révolutionnaire de Paris du 26 mars au 20 mai 1871, en rébellion à la suite de la Guerre franco-prussienne de 1870 et la capitulation de l'autorité nationale en place.
Critère variant (adjectifs) : première, insurrectionnelle, seconde
Réponses proposées après agrégation temporelle sur l'année : 1792 (première, insurrectionnelle), du 26 mars au 20 mai 1871 (seconde)

Cette approche permet de récupérer de sérieuses pistes pour la détection du critère variant mais nous n'arrivons pas encore à bien les filtrer. L'un des deux problèmes à surmonter est la chaîne de coréférence : beaucoup d'informations sur le critère variant ne peuvent pas être extraites du fait de la non résolution des coréférences. Il faudrait alors pouvoir dépasser le cadre de la phrase. De plus, il est fondamental de cibler la portée d'une entité nommée puisqu'une date dans une phrase ne se rapporte pas forcément au candidat-réponse (XIP extrait seulement la date ou le nombre sans rattacher la dépendance à un segment précis de la phrase) :

Question : *Quels pays font partie de la zone Euro ?*
Passage-réponse illustrant le problème de la coréférence non résolue : Dix-huit pays de l'Union européenne, représentant près de 324 millions d'habitants font partie de la **zone euro** au 1e janvier 2014. **Elle** a été créée en 1999 par onze pays : Allemagne, Autriche, Belgique, Espagne, Finlande, France, Irlande, Italie, Luxembourg, Pays-Bas, Portugal, rejoints par la Grèce en 2001, par la Slovénie en 2007, par Chypre et Malte en 2008, par la Slovaquie en 2009, par l'Estonie en 2011 et par la Lettonie en 2014.

Question : *Combien de spectateurs ont vu Bienvenue chez les Ch'tis ?*
Passage-réponse illustrant le problème de la portée d'une date : Avec plus de 8 millions de spectateurs qui se sont rués dans les salles, Intouchables est déjà devenu le plus gros succès du cinéma français de l' année **2011** en à peine un mois et a désormais le record de Bienvenue chez les Ch'tis en ligne de mire.
Dépendances :
– TIMEX3(2011) (ne concerne que le film *Intouchables*)
– NUMBER(8 millions) (ne concerne que le film *Intouchables*)

Nous avons donc doté Citron d'approches pour détecter un critère variant et certaines se révèlent très efficaces comme celles du critère variant temporel et géographique. Il reste néanmoins à renforcer la qualité de filtrage concernant le critère variant par attribut et la résolution de la portée d'une entité nommée afin de pouvoir bénéficier pleinement de ces approches.

4.6 CONCLUSION

Pour synthétiser ce chapitre, nous présentons le traitement complet de la question-ARM *Quels acteurs ont incarné James Bond ?* réalisé par Citron[7].

*** Analyse de la question**

Question : *Quels acteurs ont incarné James Bond ?*
Focus : James Bond
Type de réponse attendu : acteur (type PERSON)
Nombre de réponse attendu : non spécifié
Dépendances :
– DEEPOBJ (incarner, James Bond)
– NMOD (James, Bond)
– PERSON (James Bond)

*** Recherche des snippets avec Lucene**

Requête Lucene (index texte) : *+"james bond" incarner acteur*
Requête Lucene (index SE et index tableau) : *+"james bond" incarner acteur*
Dépendance obligatoire dans le snippet (focus) : PERSON(James Bond)

7. Certains extraits de documents ont été légèrement retouchés de façon à montrer comment Citron les analyse.

*** Extraction des candidats-réponses**

Stratégie A : par type attendu trouvé si c'est une entité nommée

Entité nommée recherchée (type natif de XIP) : PERSON(X)
Passage-réponse : À cette liste s'ajoutent deux autres acteurs qui ont interprété James Bond dans des films ne faisant pas partie de liste des films produits par EON productions : **Barry Nelson** (1917-2007) : acteur américain, qui a joué pour la première fois James Bond à l' écran en 1954 dans une adaptation de Casino Royale pour la télévision américaine.
Entité nommée trouvée : PERSON(Barry Nelson)
Candidat-réponse extrait : *Barry Nelson*

Entité nommée recherchée (type ajouté à XIP par un lexique) : ACTEUR(X).
Passage-réponse : Présentation de la vie de **Sean Connery**, le plus connu des James Bond, et sa filmographie.
Lexique *Acteur* (n'existe pas actuellement, donné uniquement pour illustration) :
– (—)
– **Sean Connery** ;
– Sam Rockwell ;
– Bud Spencer ;
– (—).
Entité nommée trouvée : ACTEUR(Sean Connery)
Candidat-réponse extrait : *Sean Connery*

Stratégie B : par type attendu trouvé syntaxiquement

Passage-réponse : Roger Moore devient en 1973 et à 45 ans, le plus vieil acteur à débuter pour le rôle de James Bond et sera à 58 ans, en 1985 le plus vieil acteur à avoir joué James Bond.
Dépendance trouvée qui valide le type : ATTRIBUTNN(Roger Moore, acteur)
Candidat-réponse extrait : *Roger Moore*

Passage-réponse :David Niven (1910-1983) : acteur britannique ayant interprété un James Bond vieillissant dans un film parodique adapté de Casino Royale.
Dépendance trouvée qui valide le type : DEFINITION(David Niven, acteur)
Candidat-réponse extrait : *David Niven*

Stratégie C : par similarité contextuelle

- Par similarité contextuelle de type d'entité nommée :

Candidat-réponse trouvé précédemment : *David Niven*
Type d'entité nommée pour ce candidat-réponse : PERSON(David Niven)

Passage-réponse : Skyfall : James Bond incarné par Hugh Jackman après **Daniel Craig** ?
Entités nommées du même type que ce candidat-réponse : PERSON(Daniel Craig), PERSON(Hugh Jackman)
Candidats-réponses extraits : *Daniel Craig, Hugh Jackman*

- Par similarité contextuelle avec le focus, le verbe principal et le type attendu :

Les verbes recensés ici sont utilisés plus tard pour la recherche de la forme affirmative de la question et l'analyse des amorces de SE.

Passage-réponse : David Niven (1910-1983) : acteur britannique ayant **interprété** un James Bond vieillissant dans un film parodique adapté de Casino Royale.
Dépendances liant un verbe au focus et au type attendu :
– DEEPOBJ(interpréter, James Bond)
– DEEPSUBJ(interpréter, acteur)
Verbe recensé relié au focus : *interpréter*

Passage-réponse : Aussi, la venue d'un nouvel acteur pour jouer James Bond, Daniel Craig, a-t-elle suscité autant la curiosité que le scepticisme lors de la sortie de "Casino Royale" en 2006.
Dépendances liant un verbe au focus et au type attendu :
– DEEPSUBJ(jouer, acteur)
– DEEPOBJ(jouer, James Bond)
Verbe recensé relié au focus : *jouer*

- Par recherche de la forme affirmative de la question :

Verbes recensés reliés au focus : *interpréter*, *jouer*

Passage-réponse : James Bond a donc été interprété par un Écossais (**Sean Connery**), un Australien (George Lazenby), un Anglais (Roger Moore), un Gallois (Timothy Dalton), un Irlandais (Pierce Brosnan), et un Anglais à nouveau (Daniel Craig).
Dépendances trouvées avec des verbes reliés au focus :
– DEEPOBJ(interpréter, James Bond)
– DEEPSUBJ(interpréter, écossais) *// réécriture du passif en actif*
– DEFINITION(Écossais, Sean Connery)
// création de la relation par transitivité de la définition
– DEEPSUBJ(interpréter, Sean Connery)
Candidat-réponse extrait : *Sean Connery*

Passage-réponse : Javier Bardem a déjà refusé de jouer James Bond.
Dépendances trouvées avec des verbes reliés au focus :
– DEEPOBJ(jouer, James Bond)
– DEEPSUBJ(jouer, Javier Bardem)
– VMOD(refuser, jouer)
Candidat-réponse extrait : *Javier Bardem*

Stratégie D : depuis des énumérations intra-phrastiques

Passage-réponse : James Bond a donc été **interprété** par un Écossais (Sean Connery), un Australien (**George Lazenby**), un Anglais (**Roger Moore**), un Gallois (**Timothy Dalton**), un Irlandais (**Pierce Brosnan**), **et** un Anglais à nouveau (**Daniel Craig**).
Dépendances :
– DEEPOBJ(interpréter, James Bond)
– DEEPSUBJ(interpréter, écossais) *// réécriture du passif en actif*
// création de la relation par transitivité de la coordination
– COORDITEMS(Écossais, Australien), COORDITEMS(Australien, Anglais), ...
– DEEPSUBJ(interpréter, Australien), DEEPSUBJ(interpréter, Anglais), ...
– DEFINITION(Australien, George Lazenby), DEFINITION(anglais, Roger Moore, ...
// création de la relation par transitivité de la définition
– DEEPSUBJ(interpréter, George Lazenby), DEEPSUBJ(interpréter, Roger Moore), ...
Candidats-réponses extraits : *George Lazenby*, *Roger Moore*, *Timothy Dalton*, *Pierce Brosnan*, *Daniel Craig*

Passage-réponse : Par ailleurs, Skyfall sortira pour les 50 ans de James Bond. Michael G. Wilson voit donc les choses en grand et aimerait par exemple ouvrir les portes des studios Pinewood au public, ou bien réunir les six interprètes de James Bond (Connery, Dalton, Lazenby, Moore, Brosnan et Craig).
Dépendances :
– DEFINITION(James Bond, Connery)
– COORDITEMS(Connery, Dalton), COORDITEMS(Dalton, Lazenby), COORDITEMS(Lazenby, Moore), ...
// création de la relation par transitivité de la coordination
– DEFINITION(James Bond, Dalton), DEFINITION(James Bond, Lazenby), DEFINITION(James Bond, Moore), ...
Candidats-réponses extraits : *Connery*, *Dalton*, *Lazenby*, *Moore*, *Brosnan*, *Craig*

Stratégie E : depuis des énumérations horizontales

Passage-réponse : En 50 ans, le célèbre espion James Bond a été interprété à l'écran par six acteurs : Connery, Lazenby, Moore, Dalton, Brosnan ou Craig.
Dépendances :
– DEEPSUBJ(interpréter, acteur)
– DEEPOBJ (interpréter, James Bond)
– COORDITEMS(Connery, Lazenby), COORDITEMS(Lazenby, Moore), ...
// création de la relation par transitivé de la coordination
– DEFINITION(acteur, Connery), DEFINITION(acteur, Lazenby), ...
// création de la relation par transitivité de la définition
– DEEPSUBJ(interprété, Connery), DEEPSUBJ(interprété, Lazenby), ...
Candidats-réponses extraits : *Connery*, *Lazenby*, *Moore*, *Dalton*, *Brosnan*, *Craig*

Stratégie F : depuis des énumérations verticales séquencées par Kitten

Passage-réponse :
<structEnum>
<SEamorce>À ce jour, le rôle de James Bond a été interprété<SEamorce> par **Barry Nelson** dans la version téléfilm de Casino Royale de 1954.
<SEamorce>À ce jour, le rôle de James Bond a été interprété<SEamorce> par **Sean Connery** (6 films).
<SEamorce>À ce jour, le rôle de James Bond a été interprété<SEamorce> par **David Niven** dans la parodie de 1967 de Casino Royale.
<SEamorce>À ce jour, le rôle de James Bond a été interprété<SEamorce> par **George Lazenby** (1 film).
<SEamorce>À ce jour, le rôle de James Bond a été interprété<SEamorce> par **Roger Moore** (7 films).
<SEamorce>À ce jour, le rôle de James Bond a été interprété<SEamorce> par **Timothy Dalton** (2 films).
<SEamorce>À ce jour, le rôle de James Bond a été interprété<SEamorce> par **Pierce Brosnan** (4 films).
<SEamorce>À ce jour, le rôle de James Bond a été interprété<SEamorce> par **Daniel Craig** (2 films).
</structEnum>
Candidats-réponses extraits (premier NP de chaque item) : *Barry Nelson*, *Sean Connery*, *David Niven*, *George Lazenby*, *Roger Moore*, *Timothy Dalton*, *Pierce Brosnan*, *Daniel Craig*

Stratégie G : depuis des tableaux prétraités par Kitten

Passage-réponse :
James Bond ; ; Numéro : 15 / James Bond ; ; Titre français : Tuer n' est pas jouer / James Bond ; ; Titre original : The Living Daylights / James Bond ; ; Année : 1987 / James Bond ; ; Acteur : ***Timothy Dalton*** */ James Bond ; ; Box-office , France :* 1978347 .
James Bond ; ; Numéro : 21 / James Bond ; ; Titre français : Casino Royale / James Bond ; ; Titre original : Casino Royale / James Bond ; ; Année : 2006 / James Bond ; ; Acteur : ***Daniel Craig*** */ James Bond ; ; Box-office , France :* 3182602 .
Dépendances :
– THEME(James Bond) *// titre du tableau*
// relations de définition créées grâce à la case entête Acteur
– DEFINITION(acteur, Timothy Dalton)
– DEFINITION(acteur, Daniel Craig)
Candidats-réponses extraits : *Timothy Dalton*, *Daniel Craig*

*** Validation des réponses**

Les 16 candidats-réponses dont il faut valider le type (*acteur*) sont : *Barry Nelson*, *Sean Connery*, *David Niven*, *George Lazenby*, *Roger Moore*, *Timothy Dalton*, *Pierce Brosnan*, *Daniel Craig*, *Javier Bardem*, *Hugh Jackman*, *Connery*, *Dalton*, *Lazenby*, *Moore*, *Brosnan*, *Craig*.

Les 16 candidats sont validés sur le type *acteur* :
- soit directement par le type d'entité nommée trouvé par XIP si XIP dispose d'un lexique d'acteur (ce qui n'est pas le cas pour le moment) ;
- soit par l'article trouvé dans Wikipédia :
  - **DEFINITION(Barry Nelson, acteur) :** *Barry Nelson (né Robert Haakon Nielson, le 16 avril 1917 à San Francisco, Californie et mort le 7 avril 2007 dans le comté de Bucks, Pennsylvanie), est un acteur américain.*
  - **DEFINITION(Javier Ángel Encinas Bardem, acteur) :** *Javier Ángel Encinas Bardem est un acteur espagnol, né le 1_er mars 1969 à Las Palmas de Gran Canaria (aux Îles Canaries).*
- soit par la page d'homonymie :
  - **ATTRIBUTNN(Timothy Dalton, acteur) :** *Timothy Dalton (né en 1944) est un acteur britannique qui a tenu notamment le rôle de James Bond.*
  - **DEFINITION-WIKIHOMONYMIE(Sean Connery, acteur) :** *Sean Connery, acteur.*

*** Agrégation des réponses**

Détection des doublons et regroupement

Par plus longue chaîne commune :
- Sean Connery, Connery
- Barry Nelson
- David Niven
- George Lazenby, Lazenby
- Roger Moore, Moore
- Timothy Dalton, Dalton
- Pierce Brosnan, Brosnan
- Daniel Craig, Craig
- Javier Barden
- Hugh Jackman

Critère variant temporel

Passage-réponse : Roger Moore devient en 1973 et à 45 ans, le plus vieil acteur à débuter pour le rôle de James Bond et sera à 58 ans, en 1985 le plus vieil acteur à avoir joué James Bond.
Critère variant temporel : *Roger Moore (à 58 ans, à 45 ans, en 1973, en 1985)*

Passage-réponse : Barry Nelson (1917-2007) : acteur américain, qui a joué pour la première fois James Bond à l'écran en 1954 dans une adaptation de Casino Royale pour la télévision américaine.
Critère variant temporel : *Barry Nelson (en 1954)*

Passage-réponse : 1995-2002 : Pierce Brosnan reprend le rôle de James Bond après quelques années d'absence.
Critère variant temporel : *Pierce Brosnan (1995-2002)*

*** Présentation des réponses**

Les réponses finalement proposée par Citron sont donc les suivantes (elles comportent deux réponses incorrectes *Javier Bardem* et *Hugh Jackman*) :

- Sean Connery
- Barry Nelson (en 1954)
- David Niven
- George Lazenby
- Roger Moore (à 58 ans, à 45 ans, en 1973, en 1985)
- Timothy Dalton
- Pierce Brosnan (1995-2002)
- Daniel Craig
- Javier Barden
- Hugh Jackman

On le voit par cet exemple, il est important pour un système traitant ce type de question, de pouvoir détecter et analyser des structures comme les tableaux ou les structures énumératives. Ceci permet de mieux en extraire les informations pertinentes et de les présenter de façon compréhensible à un utilisateur réel.

Dans le chapitre suivant, nous présentons l'évaluation de Citron, aussi bien en termes de performances d'extraction des réponses qu'en termes de satisfaction des utilisateurs.

5

ÉVALUATIONS

Sommaire

Dans le chapitre 2, nous avons montré que les évaluations menées lors des campagnes d'évaluation en question-réponse ne permettent pas ou peu d'évaluer les points originaux d'un système comme Citron, à savoir la possibilité de fournir des réponses multiples, la possibilité de les extraire de plusieurs documents et/ou de structures (SE ou tableaux) ainsi que la possibilité de présenter les réponses de façon à faire apparaître des éventuels critères variants.

C'est pourquoi dans ce chapitre, nous nous intéressons à l'évaluation de Citron sous plusieurs angles : une évaluation « classique » qui mesure les performances de notre système en terme de sélection des documents et d'extraction des réponses mais aussi une

évaluation plus originale où les performances de Citron sont comparées à celles d'utilisateurs qui font aussi part de leur satisfaction vis-à-vis du système.

Nous commençons par donner quelques éléments d'état de l'art sur les évaluations, en particulier celles en cadre utilisateur puis décrivons nos évaluations et leurs résultats.

5.1 QU'EST-CE QU'UNE ÉVALUATION ? QU'EST-CE QU'UN UTILISATEUR ?

5.1.1 *Évaluation des SQR*

Les campagnes d'évaluation en général ont deux objectifs [Harman, 2013] :
- modéliser une tâche utilisateur réelle (correspondant à un besoin) ;
- simuler le jugement d'un utilisateur à travers l'évaluation d'un système réalisant la tâche.

Nous avons vu dans les chapitres précédents plusieurs mesures d'évaluation des SQR : le MRR, la F-mesure, la précision moyenne. Il existe également une mesure d'évaluation empruntée à la tâche de résumé automatique de textes qui mesure le recouvrement de mots entre la réponse d'un SQR et la réponse générée par un expert [Kolomiyets et Moens, 2011]. Cette mesure a été utilisée pour les questions de type *Autre* dans les campagnes TREC et ne s'applique pas pour nos questions-ARM. En effet, les questions *Autres* sont en réalité une tâche demandant au SQR d'extraire des phrases contenant des informations factuelles à propos d'un focus donné, avec la contrainte que ces informations soient nouvelles au sens de non mentionnées dans la série de questions fournies avec le focus.

Lors de la campagne TREC 2006 [Dang *et al.*, 2006], les participants disposaient d'une semaine pour répondre à 492 questions (403 factuelles et 89 questions-listes, nous ne comptons pas ici les questions *Autres*), ce qui laissait à un SQR en moyenne 20 minutes pour répondre à une question. Ce temps est intuitivement trop long pour un utilisateur qui attend une réponse et parait également trop long en comparaison du temps qu'un être humain mettrait à trouver manuellement une réponse dans la collection de documents. De plus, l'évaluation des SQR se fait à travers le statut des réponses (correct, incorrect), et comme nous l'avons vu précédemment, le format de réponses imposé par les campagnes peut limiter les possibilités des systèmes .

5.1.2 *Évaluation des SQR en cadre utilisateur*

Concernant l'évaluation de SQR en cadre utilisateur, des expériences ont notamment été menées avec des sites communautaires de question-réponse [Shah, 2011; Chua et Banerjee, 2013] ou par des SQR disposant d'une puissance de calcul phénoménale [Ferrucci *et al.*, 2010], rendant impossible une comparaison avec notre système (leur système fournit

une réponse en 2 à 4 secondes sur un ordinateur à 2880 cœurs alors qu'il faut environ deux heures sur un ordinateur à un seul processeur).

Plusieurs expériences en cadre utilisateur existent également dans le domaine de la recherche d'information afin de modéliser le comportement des utilisateurs, particulièrement à travers le parcours de clic [Barry et Lardner, 2011], le temps passé par document [Liu *et al.*, 2010; Guo et Agichtein, 2012] ou les requêtes formulées [Ageev *et al.*, 2011]. [Ageev *et al.*, 2011] a ainsi étudié le comportement de recherche de 159 utilisateurs (sur 200 recrutés avec le service Amazon Mechanical Turk [Paolacci *et al.*, 2010]) : le temps moyen d'un utilisateur pour répondre à une question (recherche des documents et extraction de la ou des réponses) était alors de 215 secondes.

5.1.3 *Évaluation de la satisfaction utilisateur*

[Lin *et al.*, 2003] s'est intéressé à ce qui définit une réponse satisfaisante pour un utilisateur en regardant d'abord les critères de présentation de la réponse dont notamment le type du document dont est extraite la réponse ou la taille du support. [Quarteroni, 2007] a mis en place une expérience de modélisation utilisateur afin de filtrer les documents susceptibles de contenir les réponses ne correspondant pas au profil de l'utilisateur (selon son âge, son niveau de lecture, ses centres d'intérêts) et de mesurer sa satisfaction par des questions graduées. Pour son application en domaine médical, [Cao *et al.*, 2011b] évalue également la satisfaction des utilisateurs selon les critères de vitesse ou de qualité des réponses. Enfin, le domaine de la recherche d'information s'est également intéressé à la satisfaction utilisateur, par exemple [Azzah et Sanderson, 2010] a confirmé par le cadre utilisateur les hypothèses intuitives suivantes :

- la satisfaction de l'utilisateur augmente avec la qualité des résultats de recherche renvoyés par le programme, ainsi qu'avec ses propres performances dans l'expérience ;
- plus l'utilisateur doit fournir d'efforts (les documents pertinents sont mal classés) et moins il est satisfait.

Toutefois leur dernière hypothèse intuitive n'a pas été confirmée : les utilisateurs familiarisés avec les aspects de la recherche d'information n'ont pas été plus satisfaits des résultats renvoyés que ceux moins familiarisés.

5.2 ÉVALUATION « CLASSIQUE » DE CITRON EN CONDITIONS IDÉALES ET RÉELLES

Nous détaillons dans cette section l'évaluation de Citron en conditions idéales (notre cadre de développement, voir chapitre 4), puis en conditions réelles. Après un bref rappel des raisons de cette approche en deux temps, nous présentons les résultats obtenus dans ces deux cadres.

5.2.1 *Évaluation en conditions idéales*

Nous avons choisi de développer Citron en conditions idéales afin de nous focaliser uniquement sur la tâche d'extraction d'information, et de nous concentrer sur des objectifs précis, notamment l'extraction à partir de SE, sans être bruité par des erreurs générées en amont (analyse de la question, recherche de documents).

Nous avons profité de ces conditions pour réaliser une étude visant à quantifier l'apport de chacun des paramètres de Citron (tous cumulables) que sont :

- la similarité contextuelle pour la recherche de candidats-réponses (stratégie C) ;
- l'utilisation des règles XIP (celles de FIDJI couplées à celles de Citron) ;
- les algorithmes d'extraction dédiés aux structures énumératives ;
- l'utilisation de Wikipédia pour la validation du type des candidats-réponses.

Les performances de Citron sont comparées à une baseline et au SQR FIDJI [Moriceau et Tannier, 2010] capable de traiter les questions-listes et qui obtient globalement de bons résultats sur les données des campagnes d'évaluation. 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 (stratégie A) ;
- les candidats-réponses dont le type peut être vérifié syntaxiquement dans la collection de documents à savoir les entités nommées trouvées par XIP et les relations DEFINITION (stratégie B).

5.2.1.1 *Performances globales de Citron*

Le tableau 5.1 montre les résultats (précision moyenne, rappel et F-mesure) obtenus par la baseline, Citron et FIDJI. Ces résultats sont calculés à partir du nombre de réponses correctes identifiées manuellement dans notre collection de documents, chaque candidat-réponse ne pouvant être proposé qu'une seule fois (recoupement sur la forme de surface). Pour Citron, les résultats montrent une F-mesure supérieure à celle de la baseline, notamment grâce à de meilleures performances pour les questions dont les réponses se trouvent dans des structures énumératives (questions 7 à 14) alors que les performances sont plus similaires pour les questions temporelles (questions 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 réponses correctes contenues dans le passage. En moyenne, la baseline extrait $0,33$ réponse par question contre $0,44$ pour Citron.

Citron obtient pour quasiment toutes les questions une F-mesure meilleure ou équivalente à celle de FIDJI. Pour chaque question, FIDJI a toutefois dû réaliser une analyse des questions et une recherche de documents alors que pour Citron, l'analyse de chaque question était manuelle et supposée parfaite. FIDJI qui utilise le moteur de recherche

Système	baseline			Citron			FIDJI		
Mesure / Question	P	R	F	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	0,92	0,80	**0,86**
(2) Quand la 2ième guerre mondiale s'est... ?	0,85	0,79	0,81	0,86	0,86	**0,86**	0,70	0,50	0,58
(3) Quand est sorti l'Ibook ?	0,78	0,78	**0,78**	0,47	0,78	0,58	0	0	0
(4) Quand se déroule la fête de la bière ?	1	1	**1**	0,86	1	0,92	1	0,33	0,50
(5) Quand la France a-t-elle perdu son triple A ?	0,71	0,38	**0,50**	0,63	0,38	0,48	0,50	0,08	0,13
(6) Quand le PSG a-t-il gagné la coupe de... ?	0,71	0,19	0,29	0,52	0,41	**0,46**	0,75	0,33	**0,46**
(7) Quels pays étaient candidats à... ?	1	0,93	**0,97**	1	0,93	**0,97**	1	0,60	0,75
(8) Quels sont les fruits à consommer... ?	0	0	0	0,91	0,61	**0,73**	0,79	0,29	0,43
(9) Dans quels clubs a joué Nicolas Anelka ?	0,33	0,18	0,24	0,80	0,55	**0,65**	0,71	0,23	0,34
(10) Quels sont les noms des sept nains ?	0	0	0	1	1	**1**	0	0	0
(11) Quelles sont les sept merveilles du monde ?	0	0	0	0,75	0,86	**0,80**	1	0,14	0,25
(12) Quelles sont les architectures possibles... ?	0,50	0,50	0,50	0,50	0,50	0,50	0,67	0,50	**0,57**
(13) Quels polluants ont été dispersés dans... ?	0	0	0	0,83	0,63	**0,71**	0	0	0
(14) Quelles sont les distributions Linux ?	0,50	0,10	0,17	0,78	0,7	**0,74**	0	0	0
Moyennes	0,52	0,40	0,44	0,76	0,72	**0,73**	0,57	0,27	0,35

FIGURE 5.1 : Résultats de la baseline, Citron et FIDJI en conditions idéales (P : précision, R : rappel, F : F-mesure.)

Lucene a trouvé les bons documents mais n'est pas parvenu à en extraire les réponses correctes. Il faut noter que FIDJI, utilisant une pondération sur le nombre de dépendances

syntaxiques de la question trouvées dans plusieurs documents, a pu être gêné par cette petite collection, d'autant plus que chaque document est également extrêmement court (seulement quelques phrases).

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.

5.2.1.2 *Apports de chacune des stratégies de Citron*

Nous avons également mesuré l'apport individuel de chacun des paramètres de Citron (voir tableau 5.2) : 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.

Paramètre utilisé seul	P	R	F	Nbre de réponses correctes différentes	Nbre de 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	Nbre de réponses correctes différentes	Nbre de 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

FIGURE 5.2 : Résultats détaillés de Citron.

Les résultats montrent que certaines stratégies sont inefficaces lorsqu'elles sont utilisées seules. Les résultats obtenus sont identiques à la baseline pour la résolution de la coréférence, la similarité contextuelle et la validation du type par Wikipédia. Cela s'explique par une interdépendance de ces paramètres comme le montre la seconde partie du tableau. L'utilisation de nos règles syntaxiques permet à la résolution de la coréférence de fonctionner, de même que l'extraction à partir des SE. Quant à l'utilisation de la Wikipédia, elle permet bien de filtrer des candidats-réponses incorrects mais en élimine un correct, tombant ainsi légèrement sous la baseline.

Les résultats montrent cependant l'importance de traiter les structures énumératives ainsi que l'apport de nos règles de réécriture syntaxique : à chaque fois que ces deux stratégies sont utilisées ensemble, on obtient une F-mesure allant de $0,71$ de $0,72$, soit un résultat quasiment identique à l'utilisation des cinq stratégies en même temps.

5.2.1.3 *Évaluation de l'agrégation de réponses par Citron*

Comme nous l'avons vu au chapitre **??**, Citron effectue une réconciliation de référence surfacique (distance de Jaro-Winkler, plus longue chaîne commune, détection d'acronymes) et un partitionnement temporel des réponses.

Nous avons évalué manuellement l'agrégation de surface et le partitionnement temporel sur les résultats de Citron. L'agrégation de surface a produit 13 regroupements corrects de réponses ($92,3$ %) pour un seul mauvais regroupement de réponses dû à une erreur d'analyse syntaxique :

Question : *Quels polluants ont été dispersés dans l'atmosphère lors de l'effondrement du World Trade Center le 11 septembre 2001 ?*
Mauvais regroupement de réponses :
– (regroupement) de la dioxine, de l'amiante, de la fibre ;
– du mercure ;
– du plomb ;
– de l'américium 241.

L'agrégation de surface a notamment été très utile pour les questions (8) (agrégation de *pêche de vigne* et *pêche*) et (9) (agrégation 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) *Quand s'est déroulée la Commune de Paris ?* :
– une période 1789-1795 contenant 8 candidats-réponses compris entre ces deux dates ;
– une période 1871 contenant 5 candidats-réponses désignant cette année-là.

Pour la question (5) *Quand la France a-t-elle perdu son triple A ?*, le partitionnement des candidats-réponses temporels fait ressortir un critère variant qui est l'agence de notation ayant dégradé la France :

- 2012-01-13 : *Standard and Poor's* ;
- 2011-07 : *Egan-Jones* ;
- 2012-11-19 : *Moody's*.

5.2.2 *Évaluation en conditions réelles*

Nous avons utilisé un jeu de 19 questions-ARM pour l'évaluation de Citron sur des conditions réelles. Ces questions sont présentées dans le tableau 5.1. Les conditions dites réelles consistent à l'utilisation de documents issus du Web sans prétraitement manuel comme cela était le cas en conditions « idéales ». Pour constituer la collection de documents, nous avons récupéré les 100 premiers documents renvoyés par le moteur de recherche Google pour la requête générée par Citron durant l'analyse de la question : en avoir récupéré 100 au départ nous garantit normalement d'en obtenir au moins 30 (exclusion de fichiers PDF, etc.) et nous permet également d'élargir la collection pour de futures expériences sur ces données. Puis, Kitten a prétraité ces 100 documents qui ont ensuite été indexés par Lucene dans trois index : texte, structures énumératives et tableaux. Sur ces trois index, nous avons utilisé une deuxième fois la requête générée par Citron durant l'analyse de la question afin de collecter une liste de snippets pertinents. Tous les snippets ont été analysés manuellement afin d'annoter avec WebAnnotator [Tannier, 2012] toutes les occurrences de réponses jugées correctes.

Le tableau 5.1 présentent les résultats encourageants de Citron avec une précision médiane forte $(0,86)$ et un rappel médian acceptable $(0,94)$ pour une F-mesure médiane de $0,74$.

Cette évaluation nous a surtout permis de régler les tailles de snippets afin d'obtenir un compromis entre couverture des réponses correctes existantes et temps de traitement. Ce temps de traitement était notamment long dès la phase d'analyse du snippet (avec par exemple un cas d'une SE de plusieurs dizaines de milliers de caractères). Les différents tests ont montré qu'en terme de résultats et de temps d'exécution de Citron, les valeurs optimales des paramètres de recherche étaient :

- jusqu'à 5 snippets maximum par document ;
- la présence obligatoire du focus de la question dans le snippet ;
- une taille maximale du snippet de 1 500 caractères pour une SE, 10 000 pour un tableau extrait par Kitten et 900 pour les snippets de l'index texte (voir chapitre **??**).

Question	Précision	Rappel	F-mesure
(1) Quand l'Italie a-t-elle remporté la coupe du monde de football ?	0,86	0,75	0,80
(2) Combien de spectateurs ont vu Bienvenue chez les Ch'tis ?	1,00	0,18	0,31
(3) Quelles villes ont été la capitale de l'Allemagne ?	1,00	1,00	1,00
(4) Quels ministères a occupé Michèle Alliot-Marie ?	0,71	1,00	0,83
(5) Dans quels clubs a joué Éric Cantona ?	1,00	0,67	0,80
(6) Quels acteurs ont incarné James Bond ?	0,67	1,00	0,80
(7) Quels sont les noms des apôtres ?	1,00	0,58	0,74
(8) Quelles sont les planètes du système solaire ?	1,00	0,67	0,80
(9) Comment s'appellent les Rois mages ?	0,60	0,75	0,67
(10) Quels pays font partie de la zone Euro ?	0,74	0,70	0,72
(11) Combien de spectateurs ont vu le film Intouchables ?	1,00	0,50	0,67
(12) Quelles villes ont été la capitale de la Finlande ?	1,00	0,50	0,67
(13) Qui a incarné le rôle de Batman ?	0,63	1,00	0,77
(14) Quelle est la prévision de croissance de la France pour 2013 ?	0,50	0,50	0,50
(15) Quels pays ont occupé la présidence de l'Union Européenne depuis 2007 ?	0,71	0,33	0,45
(16) Quels sont les cinq piliers de l'Islam ?	1,00	1,00	1,00
(17) Comment s'appellent les Dalton ?	1,00	0,75	0,86
(18) Où Sarkozy a-t-il présenté ses voeux ?	0,75	0,27	0,40
(19) À qui Sarkozy a-t-il présenté ses voeux ?	0,44	0,44	0,44
Moyennes	**0,82**	**0,66**	**0,70**
Médianes	**0,86**	**0,67**	**0,74**

Tableau 5.1: Résultats de Citron en conditions réelles.

5.3 PROTOCOLE EXPÉRIMENTAL POUR UNE ÉVALUATION EN CADRE UTILISATEUR

Nous avons souhaité comparer Citron à des êtres humains, bien évidemment d'abord sur le critère des performances mais également sur celui du temps d'exécution et des

stratégies choisies. Nous voulions notamment vérifier l'hypothèse des campagnes d'évaluation à savoir qu'une évaluation simule le comportement d'un utilisateur et également son attente. Nous avons donc réalisé deux expériences avec des utilisateurs : une première d'extraction de réponses (*extraction)* et une seconde de satisfaction utilisateur devant la présentation de réponses *(satisfaction)*.

5.3.1 *Données d'évaluation*

Nous avons généré deux jeux de questions-ARM (*jeuA* et *jeuB*), un pour chacune des deux expériences à réaliser : l'extraction de réponses (*extraction*) et la satisfaction de l'utilisateur devant les réponses (*satisfaction*). Pour chaque utilisateur, un jeu sert pour l'expérience *extraction*, l'autre pour l'expérience *satisfaction*. Le choix de l'un ou de l'autre pour commencer se fait selon le nombre de fois où ils ont été attribués aux utilisateurs précédents afin d'avoir un niveau d'utilisation homogène.

Ces deux jeux de 10 questions-ARM sont homogènes au niveau des types de questions proposés : 5 questions dont le type attendu est spécifié et 5 dont le type est général (la liste des 20 questions-ARM en annexe D). Chaque question-ARM est le miroir syntaxique d'une autre question-ARM dans l'autre jeu avec un changement de focus. Par exemple :

- type attendu spécifié :
  - jeu A : *Quels sont les signes du zodiaque ?*
  - jeu B : *Quels sont les péchés capitaux ?*
- type attendu général :
  - jeu A : *Qui a joué Knocking on Heaven's Door ?*
  - jeu B : *Qui a joué Where Did you Sleep last Night ?*

Pour collecter et prétraiter les documents, nous avons appliqué la même procédure que pour les conditions réelles décrites précédemment à savoir la récupération de 100 documents à l'aide de Google (pour être sûr d'en obtenir 30), leur prétraitement par Kitten puis les 3 indexations de Lucene. Pour chaque question, nous avons ainsi obtenu en moyenne 18, 4 documents et 2, 4 snippets par document.

Tous les snippets ont ensuite été analysés manuellement afin d'annoter avec WebAnnotator [Tannier, 2012] les occurrences de réponses que nous avons jugées correctes, soit au total 1 084 occurrences de réponses correctes. Nous avons ensuite agrégé manuellement ces réponses, notamment pour les formes de réponses correctes désignant une même entité comme *Olympique de Marseille* et *OM* pour finalement arriver à 280 entités différentes.

Grâce à ces réponses annotées, nous avons pu confirmer que les phénomènes souhaités étaient bien présents dans les snippets qui seront proposés pour chacune de ces questions-ARM. Nous souhaitions notamment des questions pour lesquelles :

– les réponses correctes sont très nombreuses (plus de 10) ;
– les réponses correctes ne sont pas toutes situées dans un seul extrait (obligeant ainsi à ouvrir plusieurs extraits pour toutes les obtenir) ;
– seul un ou plusieurs tableaux contiennent les réponses correctes (afin de voir si un être humain analyse correctement un tableau extrait par Kitten) ;
– il y a très peu d'occurrences de réponses correctes (obligeant là-encore à ouvrir plusieurs extraits), révélant non seulement la stratégie pour choisir les documents à ouvrir mais également le temps au bout duquel un utilisateur passe à la question suivante s'il ne trouve pas de réponse correcte ;
– des SE contiennent des réponses correctes ;
– un critère variant (temporel, géographique) se trouve près de la réponse correcte afin de voir si l'utilisateur l'utilise pour justifier sa réponse (et comment il l'utilise) ;
– les réponses doivent être agrégées.

Par exemple :

Phénomène à étudier : un critère variant (temporel, géographique) est proche de la réponse
Question : *Combien de spectateurs ont vu Avengers ?* (jeuA)
Passage-réponse : Maintenant que la première semaine de sortie de The Avengers s'est achevée, place aux chiffres de la deuxième semaine (notamment le démarrage US), dans ce nouvel article box office ! Historique ! The Avengers a réuni pour sa première semaine en France **2 041 362** spectateurs, signant ainsi le meilleur démarrage de l'année. Mais la performance est ailleurs.
Réponse correcte : 2 041 362
Critères variants : *en France*, *première semaine*

5.3.2 *Infrastructure*

Les deux expériences *extraction* et *satisfaction* se déroulent consécutivement. L'expérience *extraction* a pour objectif de comparer les performances de Citron par rapport à celles d'un être humain pour la tâche d'extraction de réponses multiples depuis une collection de snippets imposés issus du Web. La seconde *satisfaction* évalue la satisfaction d'un utilisateur devant des réponses extraites et formatées de deux façons différentes : tel que Citron l'aurait idéalement fait et tel qu'une campagne d'évaluation les présenterait.

L'évaluation par des utilisateurs se fait par l'intermédiaire d'une interface Web. Nous avons configuré une machine virtuelle pour héberger le site Web de l'expérience à l'aide d'un serveur Apache [1]. La totalité des interactions du site est gérée par le framework Django [2] dont nous ne mentionnons ici que les principales fonctionnalités :

1. http://httpd.apache.org/
2. http://djangoproject.com. Nous avons utilisé la version 1.5 afin de pouvoir utiliser la version 3 de Python qui offre une gestion des chaînes de caractères en unicode par défaut.

- stockage des informations de l'expérience dans une base de données : les données sources et les données de l'utilisateur durant les deux expériences ;
- intervention de code Python pour la manipulation des données ;
- génération des pages HTML à la volée.

Le framework Django nous a par exemple permis de mettre en place un archivage de chacun des clics de l'utilisateur afin de mieux cerner son parcours et ses stratégies. Chaque ajout de réponse, suppression de réponse, début/fin de pause, ouverture/fermeture de document est donc enregistré dans une base de données.

L'accès au site doit être authentifié, un utilisateur se connecte au site de l'expérience selon deux possibilités : soit sans supervision, soit en présence du responsable de l'expérience qui peut répondre à ses interrogations si besoin. La supervision était toutefois non intrusive : le responsable s'assurait seulement de la compréhension de la tâche par l'utilisateur entre le tutoriel et le début de l'expérience. La quasi-totalité des questions posées au responsable concernaient l'expérience *extraction* et plus précisément la définition de ce qu'était une réponse et un support ; le responsable donnait alors la même réponse que celle donnée dans le tutoriel. Nous pensons que la différence entre la supervision ou non d'un responsable ne concerne que la confiance d'un utilisateur dans la tâche à effectuer (respect des consignes) mais nous n'avons pas mesuré cet aspect. Les retours rédigés par les utilisateurs en fin d'expérience semblent toutefois bien la confirmer.

Du fait du nombre restreint d'utilisateurs simultanés, le temps de réponse du site est instantané. Jusqu'à quatre personnes ont passé l'expérience en même temps sans qu'aucun temps de latence ne soit observé. Du fait de l'utilisation possible sans supervision, et également pour anticiper un incident réseau ou électrique, chaque utilisateur se voit générer un identifiant unique lui permettant de reprendre une expérience interrompue.

5.3.3 *Profil des utilisateurs*

Nous avons recruté uniquement des contacts directs dont nous étions certains de la motivation à passer une expérience de question-réponse pouvant durer plus d'une heure. Nous avons préféré privilégier la qualité des utilisateurs par rapport au nombre ou au manque de transparence d'un service d'évaluation de tâche en ligne [Sagot *et al.*, 2011]. 32 utilisateurs ont ainsi passé la double expérience *extraction* et *satisfaction* : 59,37 % (19) l'ont passée sous la supervision d'un responsable de l'expérience, 40,63 % (13) l'ont passée à distance. Concernant la répartition des jeux de questions-ARM, 46,88 % (15) ont passé l'expérience *extraction* sur le *jeuA* et *satisfaction* sur le *jeuB* contre 53,12 % (17) pour l'inverse.

L'utilisateur commence l'expérience *extraction* en répondant à six questions d'un formulaire dans le but de définir son profil : sa tranche d'âge, s'il travaille dans la recherche, dans le TAL (Traitement Automatique des Langues), combien d'articles scientifiques portant sur les SQR il a lus, combien de requêtes il effectue sur un moteur de recherche quotidiennement et son niveau d'informatique. La répartition des profils est donnée dans le tableau 5.2 et représentée graphiquement sur la figure 5.3. Concernant le niveau informatique, nous n'avons eu aucun profil *débutant* : nous avons considéré le profil *normal* comme celui correspondant à l'utilisation quotidienne d'Internet et d'un logiciel de bureautique, le profil *intermédiaire* comme celui impliquant l'utilisation d'un programme plus poussé (retouche d'image, musique) et enfin le niveau *avancé* pour les programmeurs. Concernant la connaissance des SQR, nous avons considéré qu'une personne y était sensibilisée si elle avait répondu avoir lu au moins un article sur le sujet.

L'utilisateur peut également laisser son adresse courriel pour recevoir ses résultats mais il lui est précisé que cela est facultatif, notamment pour éviter un stress lié à l'aspect compétition de l'expérience *extraction*.

Attribut	**Valeur**	**Pourcentage**
Tranche d'âge	10-25	28,13 %
	26-40	53,12 %
	41 et plus	18,75 %
Dans la recherche	Oui	59,38 %
	Non	40,62 %
Dans le TAL	Oui	37,50 %
	Non	62,50 %
Lecture SQR	Oui	53,13 %
	Non	46,87 %
Requêtes quotidiennes	Moins de 10	48,39 %
	Plus de 10	51,61 %
Niveau informatique	normal	31,25 %
	intermédiaire	12,50 %
	avancé	56,25 %

Tableau 5.2: Répartition des profils utilisateurs.

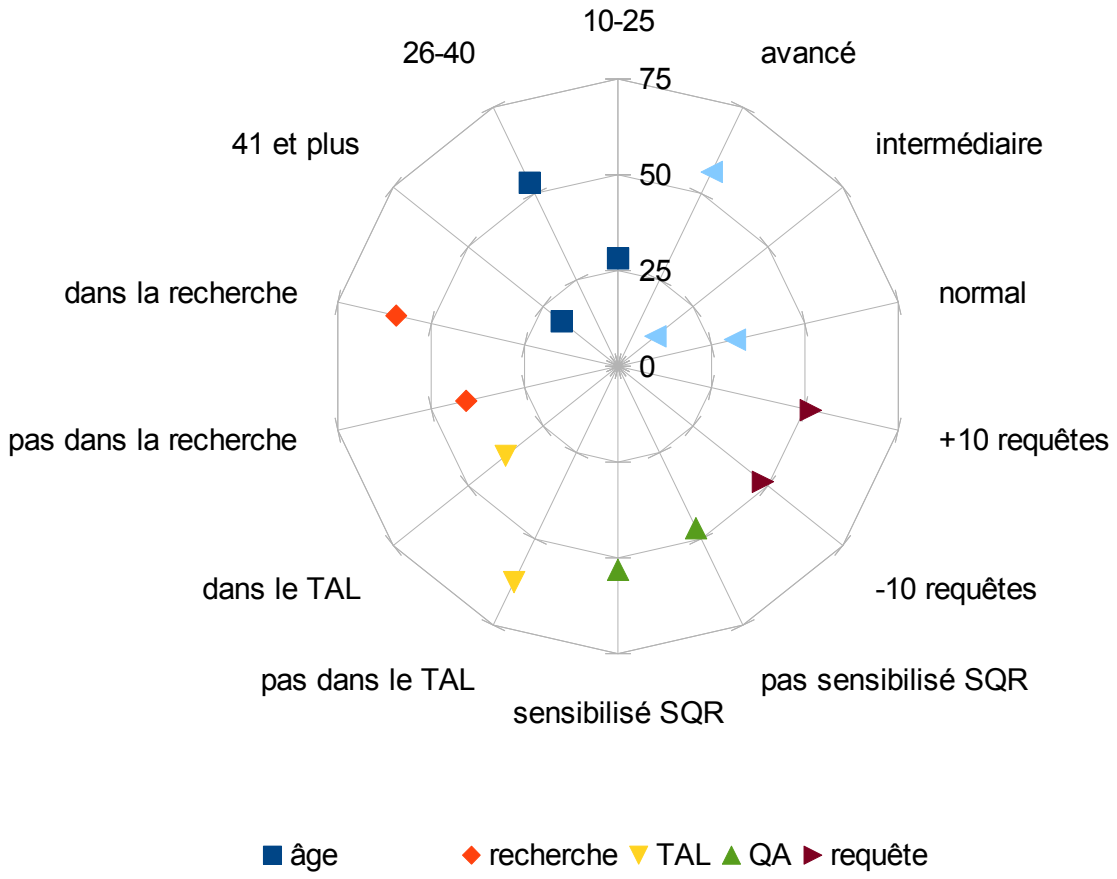

FIGURE 5.3 : Répartition en pourcentage des profils utilisateurs.

5.4 ÉVALUATION DE CITRON EN CADRE UTILISATEUR

Nous présentons ici d'abord l'expérience *extraction* puis l'expérience *satisfaction* et leurs résultats respectifs.

5.4.1 *Expérience d'extraction de réponses*

5.4.1.1 *Présentation de l'expérience*

Le but premier de cette évaluation est de voir si, en utilisant une base identique de documents imposés, un utilisateur est plus rapide que Citron pour extraire les réponses. En effet, nous avons développé Citron en cherchant à lui faire répondre à une question-ARM de façon rapide, si possible moins de deux minutes : les êtres humains ayant mis en moyenne 215 secondes durant l'expérience de recherche et d'extraction de réponses dans [Ageev *et al.*, 2011], nous avons souhaité que notre système soit au minimum deux fois plus rapide. de tester si un utilisateur parvient à être plus performant que Citron selon les mesures habituelles des SQR. L'utilisateur est donc informé qu'il est évalué sur ces deux critères : la rapidité et le nombre de réponses correctes fournies (précision et rappel). Comme nous nous intéressons au comportement de l'utilisateur, nous avons volontairement proposé une interface basique car nous ne souhaitions pas évaluer la qualité d'une interface mais le comportement de l'utilisateur face à cette tâche d'extraction d'information. Nous présentons ici l'interface de cette expérience.

L'utilisateur commence par suivre un entraînement avec une question ne servant qu'au tutoriel afin de se familiariser avec l'interface puis chaque utilisateur doit extraire les réponses pour un des 2 jeux de 10 questions (*jeuA* ou *jeuB*). La figure 5.4 montre l'interface au commencement de chaque nouvelle question.

Les différents éléments de cette interface sont (marqués par des nombres sur la figure) :

LE TEMPS. Le nombre 1 désigne le bouton de pause et le nombre 2 le temps actuellement passé par l'utilisateur sur cette question. Nous n'avons pas limité le temps passé par un utilisateur sur une question. En effet, cela aurait eu pour effet de mesurer la qualité de l'interface d'évaluation plutôt que la tâche d'extraction d'information. L'utilisateur peut choisir de passer beaucoup de temps sur une question ou même passer à la suivante après quelques secondes : s'il pense qu'il ne trouvera pas de réponse correcte pour une question, il est plus avantageux pour lui de la terminer le plus vite possible. Sur ces deux aspects, nous nous éloignons donc de la tâche CLEF [Gonzalo et Oard, 2005] où l'utilisateur était limité à cinq minutes et où il était obligé de fournir une réponse avant de passer à la question suivante. Puisque nous mesurons le temps passé sur une question, l'utilisateur peut mettre l'expérience en pause en cas de nécessité (ce temps de pause est déduit du temps total). Le fait de passer en pause efface de l'écran les documents à sa disposition pour qu'il ne puisse pas les lire sans être chronométré.

LES DOCUMENTS. Le nombre 3 désigne les informations disponibles pour chaque document. Au début de chaque question, l'utilisateur voit donc une liste de documents

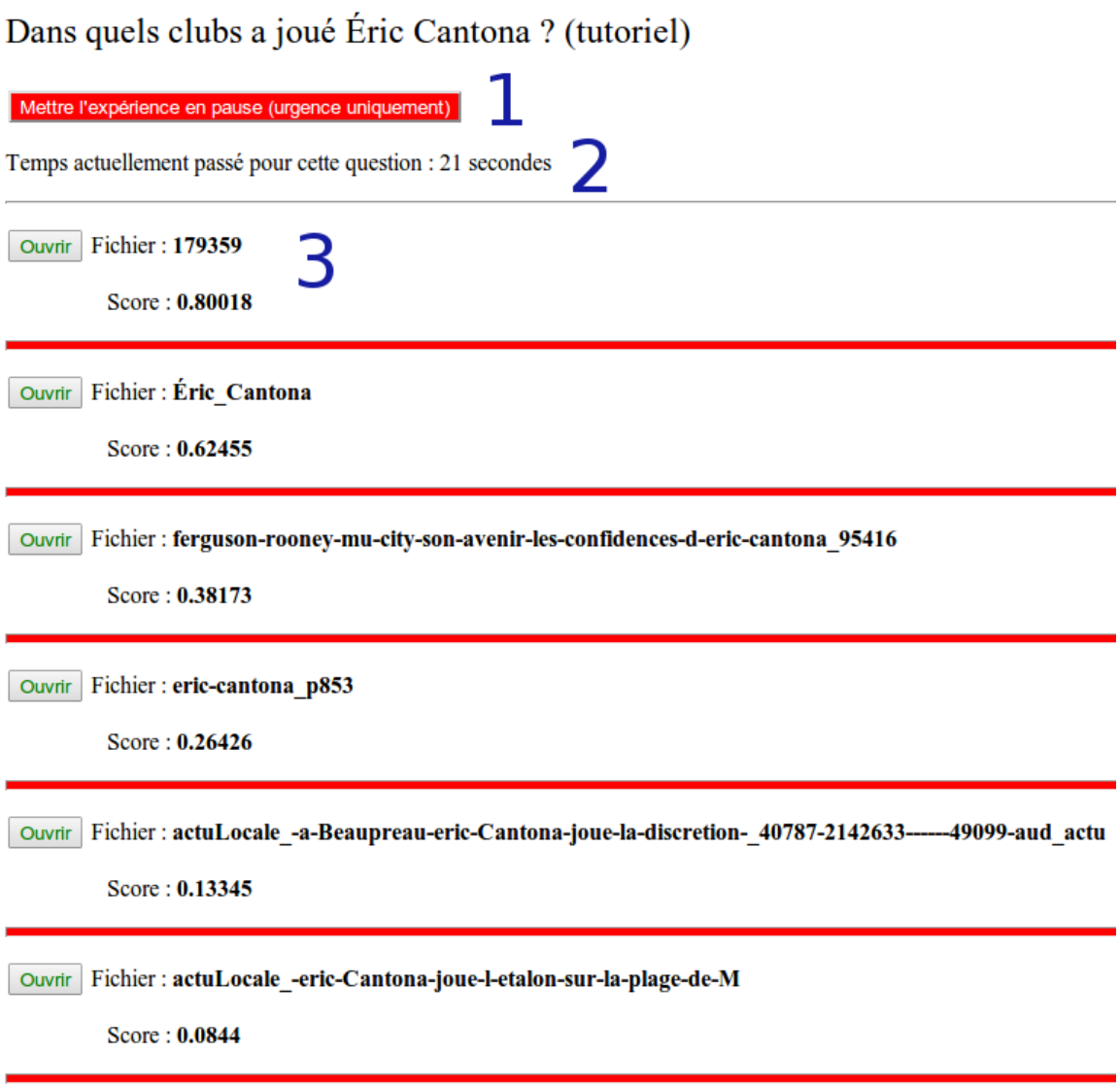

FIGURE 5.4 : Interface d'accueil pour chaque nouvelle question.

séparés par des traits horizontaux rouges. Il dispose d'un bouton pour ouvrir le document et en voir le ou les snippets qui en ont été extraits. Il dispose des deux informations suivantes pour choisir s'il ouvre le document ou non :

- le nom du fichier : il s'agit du fichier récupéré depuis les résultats du moteur de recherche. Un traitement a été effectué pour rendre chaque nom de fichier unique en y ajoutant en suffixe un nombre incrémenté, l'utilisateur en est informé durant le tutoriel ;

– le score de similarité calculé par Lucene entre le document dont provient le ou les snippets et la requête générée par Citron à partir de la question. L'utilisateur a été informé de la définition de ce score (similarité entre la question et le document) ainsi que des bornes de ce score sans toutefois lui présenter les détails arithmétiques : 0 (pas similaire) à 1 (très similaire).

Si l'utilisateur choisit d'ouvrir un document, un formulaire de saisie s'ouvre comme le montre la figure 5.5. Dans cette figure, lLe nombre 1 reprend les informations disponibles en haut de page à savoir la question en cours ainsi que le temps passé sur cette question, ces informations sont donc également disponibles juste avant le premier snippet d'un document après ouverture de ce dernier.

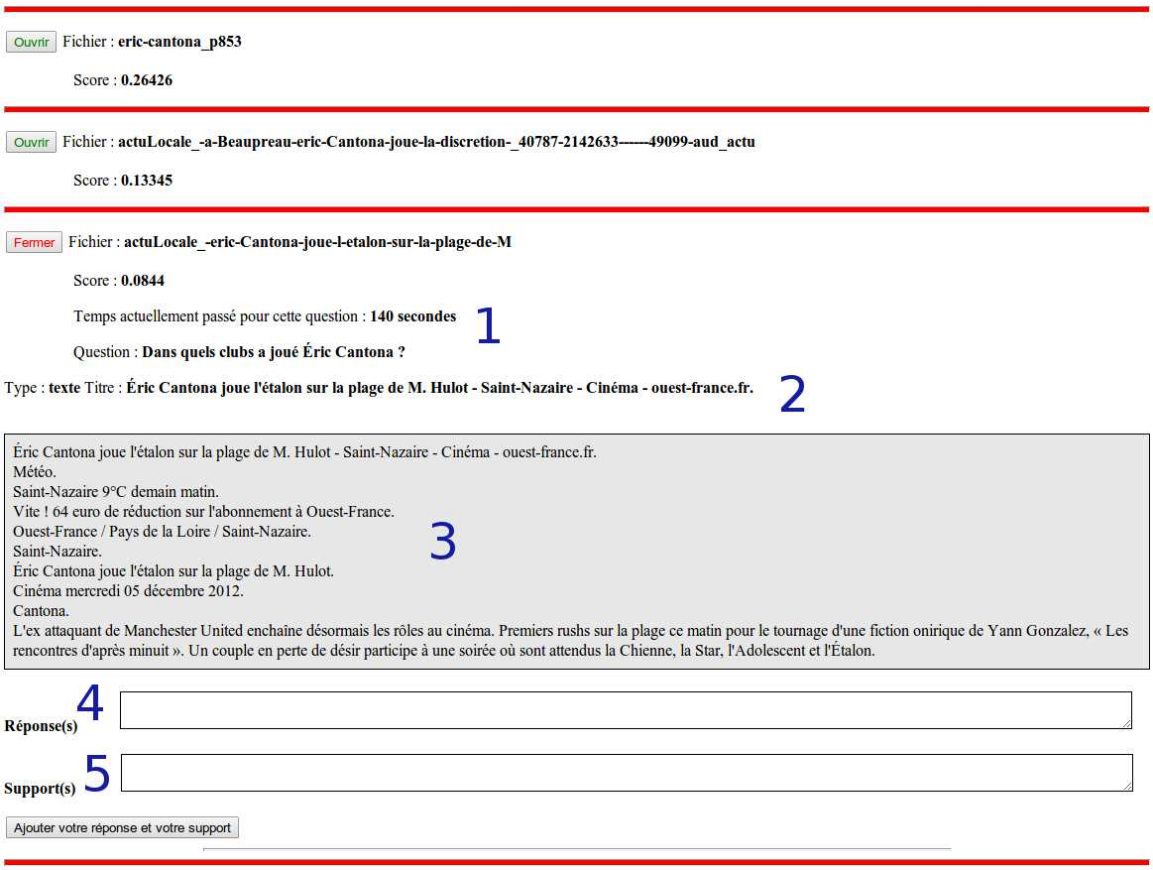

FIGURE 5.5 : Interface lors de l'ouverture des snippets d'un document.

LES SNIPPETS. Dans la figure 5.5, les nombres 2, 3, 4 et 5 représentent un snippet et ses informations : un seul est visible sur la figure mais lorsqu'il y en a plusieurs, ils sont alors

séparés par des traits horizontaux bleus. Les nombres 2 et 3 correspondent respectivement au type du snippet et au titre du document :

- le type du snippet. Un document peut contenir plusieurs snippets et chacun des snippets peut proveni de l'index texte, des SE ou des tableaux. Nous avons reformulé ces trois étiquettes en texte, liste et tableau pour ne pas troubler l'utilisateur. Un extrait de type liste est décrit comme *porteur d'une énumération* et un exemple complet d'extraction d'un tableau par Kitten a été détaillé à l'utilisateur ;
- le titre du document. Il s'agit du contenu de la balise HTML <title>. Quand il y a plusieurs extraits par document, ce titre est identique pour chaque extrait.

Le nombre 3 indique le cadre délimitant le contenu d'un snippet : l'utilisateur ne peut extraire une réponse que depuis l'intérieur de ce cadre, le nom du fichier et le titre du document ne peuvent être utilisés pour l'extraction. Si l'utilisateur pense que le snippet contient une réponse correcte (définie dans le tutoriel par *ce que vous pensez être la réponse*, il peut l'extraire dans la zone de saisie 4, obligatoirement accompagné d'un support non vide (défini par *ce que vous pensez être le texte justifiant la ou les réponses extraites*) dans la zone de saisie 5. Si un snippet contient plusieurs réponses correctes, l'utilisateur est libre de saisir une réponse à la fois ou plusieurs réponses en une seule fois.

LES RÉPONSES CORRECTES. L'utilisateur est informé qu'un snippet ne contient pas forcément de réponse correcte et que chaque question est une question-ARM, c'est-à-dire qu'elle attend au moins deux réponses/entités correctes dont la forme de surface est différente (au-delà de la variation en nombre) sur la totalité des snippets disponibles pour chaque question.

De plus, il est informé que la totalité des documents est à prendre à compte pour qu'une réponse soit correcte : en effet, une réponse peut être correcte dans un document mais l'information apportée par un autre document peut montrer que cette réponse est finalement incorrecte. Par exemple pour la question *Quels acteurs ont interprété le rôle de Batman au cinéma ?*, le deuxième document invalide la réponse *Bruce Thomas* du premier document puisqu'on y apprend qu'il s'agit d'une série télévisée et non d'un film :

- Premier document : *Bruce Thomas a joué le rôle de Batman en 2002 dans « Les anges de la nuit »* ;
- Deuxième document : *« Les anges de la nuit » est une série télévisée diffusée en 2002 et 2003.*

En conséquence nous permettons à l'utilisateur de supprimer une réponse précédemment saisie (un rappel de toutes les réponses déjà saisies est affiché en bas de page de l'interface), cette possibilité n'est toutefois accessible que pour la question en cours.

La réponse n'est pas limitée en nombre de caractères : nous souhaitons voir ce que l'utilisateur considérera comme étant une réponse correcte et quelle forme elle aura. L'utilisateur peut également ajouter des réponses correctes d'un même document en une seule fois (sous forme d'une énumération par exemple) ou une réponse à la fois.

LES SUPPORTS. L'utilisateur doit accompagner chaque réponse d'un support justifiant la réponse. Ce support n'est pas limité en nombre de caractères et aucune indication n'est donnée sur la façon de le segmenter. En effet, nous souhaitons voir ce que l'utilisateur fera de lui même pour justifier sa réponse : quelle sera la longueur du support, sera-t-il un extrait continu du document ou composé d'extraits de plusieurs documents, etc.

5.4.1.2 *Annotations des données*

Les utilisateurs ont saisi un total de 876 réponses dans la zone de saisie du formulaire [3], chacune de ces réponse pouvant contenir une ou plusieurs réponses individuelles. En effet, les utilisateurs étaient libres de saisir dans le formulaire une seule réponse à la fois (623 réponses saisies ne contenaient qu'une seule réponse, soit 71, 12 %) ou plusieurs en même temps (253 en contenaient plusieurs, soit 28, 88 %) tant qu'elles provenaient du même snippet. Dans le deuxième cas, nous avons du d'abord segmenter toutes les réponses fournies en réponses individuelles. Nous sommes ainsi arrivés à un total de 2 319 réponses/entités individuelles [4], soit une moyenne de 72, 47 réponses individuelles par utilisateur et de 7, 27 réponses par questions. Ces 2 319 réponses proviennent pour 41, 22 % de structures énumératives, 23, 83 % de tableaux et 41, 22 % de texte, confirmant non seulement l'importance de pouvoir les détecter et les analyser correctement mais aussi que le travail réalisé par Kitten sur les documents n'a pas gêné les utilisateurs dans leur compréhension.

Concernant les supports, 876 supports ont également été saisis puisqu'il n'était pas possible de saisir une réponse (ou plusieurs réponses à la fois) sans un support. Sur ces 876 supports saisis, on obtient 670 supports différents, ce qui veut dire que les utilisateurs ont fourni 206 supports identiques, soit 23, 52 %. Sur ces 876 supports, pour les 623 réponses saisies ne contenant qu'une seule réponse, 602 supports étaient continus, soit 96, 63 %. Cela peut être vu comme une confirmation des campagnes d'évaluation à demander aux SQR un support continu. Nous pensons toutefois que les SQR ne doivent pas chercher absolument à copier le comportement de l'être humain mais qu'ils doivent être efficaces en premier lieu. En l'occurrence ici, l'efficacité d'un support se juge à la satisfaction de l'utilisateur concernant la justification, et l'expérience suivante nous montrera que si les utilisateurs extraient des supports continus, ils préfèrent finalement des réponses justifiées grâce à des supports non continus. Enfin, ces 602 supports fournis étaient extrêmement courts puisque leur longueur moyenne étaient de 106 caractères.

3. En réalité, 877 réponses ont été saisies mais une réponse a été source d'une anomalie durant l'annotation et a été exclue. De plus, durant l'extraction, les utilisateurs ont choisi de supprimer 37 de leurs réponses saisies.

4. Il est à noter que parmi ces 2 319 réponses individuelles, 44 ne sont pas comptées pour l'évaluation. En effet, ces réponses résultaient d'une incompréhension de la consigne, certains utilisateurs ayant répondu qu'il n'y avait pas de réponse correcte dans l'extrait.

Nous avons annoté manuellement les réponses des utilisateurs en reprenant les statuts de réponses définis lors des campagnes d'évaluation, principalement ceux de Quaero qui étaient les plus complets [Quintard, 2010], que nous avons complétés ainsi (l'équivalent du statut Quaero est donné entre parenthèses) :

- (full) **Correct full** : la réponse est correcte, le support valide entièrement la réponse ;
- (full) **Correct OK** : la réponse est correcte, le support valide presque la réponse. Par exemple, pour *Quelles villes ont été la capitale des États-Unis ?*, la réponse *Philadelphie* associée au support *Philadelphie, capitale originelle* ne permet pas de savoir qu'il est question des États-Unis sans regarder le snippet en entier ;
- (right) **Correct** : la réponse est correcte, provient du snippet mais le support ne valide pas la réponse ;
- (unsupported) **Correct absente extrait** : la réponse est correcte mais n'a pas été extraite depuis le snippet. Cela s'est produit lorsque des réponses ont été extraites du titre du document ou qu'un utilisateur a inféré une réponse. Pour la question *Quand le Kärpät a-t-il remporté le championnat de Finlande de hockey sur glace ?*, l'extrait *En 2008, le Kärpät Oulu remporte son second titre consécutif, le quatrième en cinq ans* permet d'inférer les réponses *2004* et *2007* mais ces dernières sont absentes du document ;
- (inexact) **Correct non segmentée** : la réponse est correcte mais mal segmentée. C'est le cas lorsque l'on fournit une phrase complète sans extraire ce qui est considérée comme la réponse précise, par exemple pour la question *Quels sont les péchés capitaux ?*, la réponse suivante est non segmentée : *La Paresse est un amour du repos, qui nous pousse à omettre ou à négliger nos devoirs, plutôt que de nous faire violence pour les remplir.* ;
- (inexact) **Correct pb orthographe** : la réponse est correcte mais contient une faute d'orthographe ;
- (inexact) **Correct rédigée** : la réponse est correcte mais l'utilisateur l'a rédigée au lieu de simplement l'extraire. Par exemple : *C'est NIRVANA qui a joué "Where did you sleep last night"* ;
- (supported) **Incorrect mais support correct** : la réponse est incorrecte mais le support contient une réponse correcte ;
- (false) **Incorrect contradiction** : la réponse est incorrecte et il était possible de s'en rendre compte par recoupement avec un autre extrait proposé ;
- (false) **Incorrect** : la réponse est incorrecte.

La campagne Quaero ne comptait comme réponse valide que les réponses obtenant le statut *right* ou *full*. De notre côté, nous avons réalisé deux évaluations :

- évaluation *QR* : seules les réponses ayant les statuts *Correct full* et *correct OK* sont considérées comme valides. En effet, nous mesurons ainsi la tâche réelle d'extraction correcte de réponse ainsi que sa justification par le support. Les campagnes ont

évalué la validité d'une réponse quasi-exclusivement à l'aide du document complet, sans réellement juger du support, pourtant exigé ;
– évaluation *humaine* : toute réponse dont le statut commence par *Correct* est valide. Ce sont les réponses qui peuvent être jugées acceptables par des utilisateurs. En effet, ce cadre plus large tolère les erreurs d'orthographe ou de segmentation dont l'humain arrive à s'affranchir pour comprendre la réponse (ici, c'est d'ailleurs l'humain qui a produit ces « erreurs »).

Le tableau 5.3 présente les occurrences moyennes des statuts des réponses extraites par Citron et par les 32 utilisateurs pour les deux jeux *jeuA* et *jeuB* (le tableau 5.4 présente les occurrences de statuts pour chacun des deux jeux).

Source	**Moyenne *jeuA* et *jeuB*** (% (#))	
Statut	Citron	Utilisateurs
Correct full	72,34 (17)	46,74 (1084)
Correct OK	0	22,42 (520)
Correct	0	16,95 (393)
Correct absente extrait	0	0,26 (6)
Correct non segmentée	0	7,98 (185)
Correct pb orthographe	0	0,69 (16)
Correct rédigée	0	1,85 (43)
Incorrect mais support correct	17,02 (4)	1,34 (31)
Incorrect contradiction	0	0
Incorrect	10,64 (2,5)	1,77 (41)

Tableau 5.3: Répartition des réponses individuelles des 32 utilisateurs et de Citron pour la moyenne des *jeuA* et *jeuB*.

Nous y constatons que par une évaluation *humaine*, il n'y a quasiment pas de réponses incorrectes parmi celles extraites par les utilisateurs (3, 11 %). Ce résultat montre que l'être humain réussit presque parfaitement sa tâche d'extraction de réponses à des questions sur des thématiques courantes ou spécialisées. Les questions sur le football et le hockey sur glace [5] ont obtenu des taux élevés de réponses correctes, y compris de la part d'utilisateur ayant dit exécrer le sport et ne rien en connaître. La reconnaissance de ce qu'est un club ou une date associée à un titre de champion n'a pas posé de problème à ces utilisateurs malgré leur manque de connaissance du monde en la matière.

5. *Dans quels clubs a joué Laurent Blanc ?*, *Quand le Kärpät a-t-il remporté le championnat de Finlande de hockey sur glace ?*

Source	***jeuA*** (% (#))		***jeuB*** (% (#))	
Statut	Citron	Utilisateurs	Citron	Utilisateurs
Correct full	72 (18)	53.77 (556)	72,73 (16)	41.09 (528)
Correct OK	0	11.99 (124)	0	30.82 (396)
Correct	0	25.53 (264)	0	10.04 (129)
Correct absente extrait	0	0.19 (2)	0	0.31 (4)
Correct non segmentée	0	4.93 (51)	0	10.43 (134)
Correct pb orthographe	0	0.29 (3)	0	1.01 (13)
Correct rédigée	0	0.10 (1)	0	3.27 (42)
Incorrect mais support correct	50 (5)	0.58 (6)	13,63 (3)	1.95 (25)
Incorrect contradiction	0	0	0	0
Incorrect	8 (2)	2.61 (27)	13,63 (3)	1.09 (14)

Tableau 5.4: Répartition des réponses individuelles des 32 utilisateurs et de Citron pour le *jeuA* et le *jeuB*.

On constate également peu de problème d'orthographe et que seules quelques réponses ont été rédigées, principalement pour apporter des précisions, notamment en terme de critère variant. Par exemple pour le nombre de spectateurs ayant vu Skyfall (figure 5.6), des utilisateurs ont ajouté la date ou le lieu comme Citron le fait. Ce comportement est intéressant puisque, si peu de réponses ont été rédigées (1, 85 %), en retirant les réponses d'un seul utilisateur, on constate que 65 % d'entre elles (11 sur 17) possèdent un critère variant ajouté naturellement par les utilisateurs tout comme le propose Citron.

Enfin, du point de vue de l'évaluation *QR*, les utilisateurs ont extrait 69, 16 % de réponses valides uniquement avec leur support, ce qui est en dessous mais tout de même proche des performances de Citron (72, 34 %). Contrairement aux utilisateurs, Citron fournit une proportion importante de réponses incorrectes (27, 66 %), les causes de ces erreurs sont détaillées dans la sous-section suivante.

5.4.1.3 *Résultats de l'extraction de réponse*

Nous avons utilisé la F-mesure pour évaluer la qualité de l'extraction et nous avons également cherché à comparer les performances de Citron en temps d'exécution par rapport à celles des utilisateurs. Nous avons pour cela mesuré la durée moyenne passée par l'utilisateur sur chaque question et nous l'avons comparée en regard du temps mis par un utilisateur (tout profil confondu) pour fournir la réponse correcte la plus rapide pour chacune des questions :

$T = min(\frac{n_{rapide}}{n_{utilisateur}}, 1)$ où, pour une question :

- n_{rapide} est le nombre de secondes le plus petit nécessaire pour fournir une réponse correcte parmi tous les utilisateurs,
- $n_{utilisateur}$ est le nombre de secondes passées par l'utilisateur.

Nous prenons le minimum par rapport à 1 puisqu'il s'agit du temps le plus rapide parmi les 32 utilisateurs et que Citron peut avoir été plus rapide que ce dernier.

Les résultats par jeu montrent que l'on observe de meilleurs résultats sur le *jeuA* que sur le *jeuB* alors que le *jeuB* est plus rapide. Les mêmes tendances se retrouvent cependant dans les deux jeux et c'est pourquoi nous ne présentons ici que la moyenne des deux jeux dans le tableau 5.5.

Profil	$\mathbf{P_{hum}}$	$\mathbf{R_{hum}}$	$\mathbf{F_{hum}}$	$\mathbf{P_{QR}}$	$\mathbf{R_{QR}}$	$\mathbf{F_{QR}}$	$\mathbf{T_{QR}}$
TAL	0,91	0,61	0,68	0,87	0,59	**0,65**	**0,22**
Non TAL	0,86	0,55	0,62	0,77	0,51	0,57	0,21
Recherche	0,94	0,67	0,73	0,87	0,64	**0,69**	0,19
Non Recherche	0,81	0,45	0,52	0,71	0,42	0,48	**0,22**
Nb requête quotidienne <= 10	0,83	0,47	0,54	0,73	0,43	0,49	**0,21**
Nb requête quotidienne > 10	0,93	0,67	0,73	0,88	0,64	**0,70**	0,20
Niveau informatique débutant	0,80	0,47	0,53	0,76	0,45	0,52	**0,24**
Niveau informatique intermédiaire	0,86	0,38	0,47	0,68	0,33	0,40	0,21
Niveau informatique avancé	0,95	0,68	0,75	0,87	0,65	**0,70**	0,18
Âge 10-25 ans	0,89	0,59	0,65	0,76	0,54	0,59	**0,22**
Âge 26-40 ans	0,93	0,64	0,71	0,88	0,62	**0,68**	0,21
Âge 41 ans et plus	0,76	0,42	0,47	0,69	0,40	0,45	0,16
Aucun article QR lu	0,81	0,51	0,57	0,75	0,49	0,54	**0,23**
De 1 à 5 articles QR lus	0,91	0,63	0,69	0,82	0,59	0,65	0,20
>= 6 articles QR lus	0,95	0,66	0,73	0,92	0,65	**0,72**	0,19
Moyenne utilisateurs (sur la totalité des questions)	0,88	0,58	0.65	0,81	0.55	**0.61**	0.21
Moyenne Citron (sur la totalité des questions)	0,47	0.36	0.37	0,47	0.36	0.37	**0.89**

Tableau 5.5: Moyennes de *jeuA* et *jeuB* (*hum* pour l'évaluation des réponses selon des critères *humains* et *QR* pour l'évaluation des réponses selon les critères d'une campagne d'évaluation). P pour précision, R pour rappel, F pour F-mesure, T pour temps.

LA RAPIDITÉ DE L'EXTRACTION DES RÉPONSES.

Dans le tableau 5.5, la rapidité se lit à l'aide de la mesure T_{QR} et montre que plus les utilisateurs sont jeunes, plus ils ont été rapides (+31 et +38 % pour les 2 tranches d'âge les plus jeunes). Si les utilisateurs travaillant dans le TAL ont été plus rapides que ceux n'y travaillant pas, ceux travaillant dans la recherche ont été moins rapides que ceux n'y travaillant pas. La familiarité avec les moteurs de recherche n'influe que très peu, au contraire de celle avec les SQR puisque les experts en question-réponse y ont passé plus de temps. Enfin, plus les utilisateurs possédaient un bon niveau informatique et plus ils ont passé du temps sur les questions, ce qui corrobore certains retours utilisateurs où des personnes peu à l'aise avec l'informatique se sentaient perdues après avoir ouvert plusieurs extraits de documents et ont préféré passer rapidement à la question suivante. Il faut noter que la quasi-totalité des utilisateurs a choisi d'extraire le maximum de réponses différentes possibles pour chaque question avec une durée moyenne de $238,37$ secondes pour une médiane de $239,77$. Par exemple, seul 9 % des utilisateurs n'ont pas répondu à une question de leur jeu.

Citron a été conçu pour répondre de façon rapide aux questions et se retrouve sans surprise le plus rapide avec un score T moyen de $0,89$ contre $0,21$ pour les utilisateurs, Citron passant en moyenne 22 secondes par question contre $238,37$ pour les utilisateurs. Même pour les questions auxquelles Citron décide de ne pas répondre, le temps consommé pour extraire les candidats-réponses et finalement ne pas les proposer ne le désavantage pas au niveau temporel. Citron obtient le maximum de points pour T_{QR} sauf pour 4 questions nécessitant de valider le type d'énormément de candidats-réponses.

Le coefficient de corrélation de Bravais-Pearson calculé entre le temps passé sur l'expérience et respectivement la précision, le rappel et la F-mesure montre qu'il n'y a pas de corrélation entre le temps passé et la qualité de l'extraction.

LA QUALITÉ DE L'EXTRACTION DES RÉPONSES.

La qualité de l'extraction des réponses se lit à l'aide de la F-mesure. On constate que la F-mesure (F dans le tableau 5.5) est naturellement plus élevée selon les critères d'évaluation *humain* plutôt que *QR*. Les connaisseurs du TAL dominent ceux extérieurs au TAL ($+14,03$ %), tout comme ceux travaillant dans la recherche sur ceux extérieurs ($+43,75$ %). Sans surprise, les deux tranches d'âge les plus jeunes ainsi que le niveau informatique avancé obtiennent les meilleurs résultats. Le niveau informatique intermédiaire obtient toutefois les moins bons résultats mais cela vient surtout du fait que des utilisateurs se sont surévalués. Le nombre de requêtes effectuées quotidiennement influe énormément, ce qui pourrait confirmer une hypothèse d'entraînement régulier à lire des résultats sous forme d'extraits : non seulement à les parcourir vite mais également à en extraire rapi-

dement l'information pertinente. Enfin, on observe une corrélation entre la qualité des réponses et le nombre d'articles QR lus : la connaissance du domaine, même superficielle, amène notamment à anticiper et repérer beaucoup plus vite le type de la réponse recherchée.

Citron est nettement moins performant principalement parce qu'il ne répond pas à 7 questions sur les 20. Pour ces questions, Citron a rencontré les problèmes suivants :

- impossibilité de trouver des candidats-réponses au type valide (*Quels sont les quatre nobles vérité du bouddhisme ?* et *Quels sont les dix commandements ?*) ;
- problème de conversion HTML : Kitten n'est pas encore parfait et son extraction des SE des recettes de cuisine en a souffert. En conséquence, aucune de nos règles XIP n'a pu extraire d'ingrédients pour les questions de cuisine (*Quels sont les ingrédients d'un couscous/chili végétarien ?*) ;
- problème d'analyse syntaxique : le choix d'utiliser un focus de question en anglais était volontaire pour les questions *Qui a joué Knocking on Heaven's door / Where did you sleep last night ?* afin de tester la robustesse de Citron. Ce fut un échec : par exemple, le focus trouvé pour la première question était *s*, ce qui a empêché de cibler les phrases pertinentes pour en extraire des candidats-réponses ;
- trop forte contrainte pour la validation des réponses : la validation d'une réponse par Citron doit satisfaire énormément de contraintes dont celle de la présence du focus dans le snippet, ce qui fait que même lorsque Citron a extrait des candidats-réponses corrects, il ne les a finalement pas proposés du fait de cette absence.

Citron a rendu une copie conforme à sa nature à savoir une forte précision aux questions répondues : la moyenne sur un jeu n'est pas forcément représentative d'un score moyen et les résultats de Citron ici en sont l'illustration. Cette force précision ressort lorsqu'on ne prend en compte que les questions auxquelles Citron a répondu (voir tableau 5.6). Citron n'est alors en moyenne que de 10 points en F-mesure derrière les utilisateurs, obtenant même un meilleur score pour le *jeuB*. Même s'il est difficile de comparer ces résultats car les questions et documents sont différents de ceux des campagnes, le score de Citron est honorable car proche de certains scores obtenus par des SQR à des questions factuelles de campagne d'évaluation : sa F-mesure est bonne (0.56) en regard des résultats des dernières campagnes TREC avec des questions-listes (année : médiane/-meilleure F-mesure) : 2004 : 0,094/0,652 [Voorhees, 2004] ; 2005 : 0,053/0,468 [Voorhees et Dang, 2005] ; 2006 : 0,152/0,433 [Dang *et al.*, 2006] et 2007 : 0,09/0,433 [Dang *et al.*, 2007].

En conclusion, les résultats de cette expérience d'*extraction* confirme l'hypothèse intuitive qu'un être humain est plus performant que notre SQR, et plus globalement la quasi-totalité des SQR puisque la F-mesure obtenue est vraiment très élevée pour des questions-listes. Ces résultats confirment cependant que pour obtenir une F-mesure éle-

	P_{hum}	R_{hum}	F_{hum}	P_{QR}	R_{QR}	F_{QR}	T_{QR}
Moyenne utilisateurs (*jeuA*)	0,97	0,73	0,57	**0,94**	**0,72**	**0,77**	0,16
Moyenne Citron (*jeuA*)	0,75	0,53	0,52	0,75	0,53	0,52	**0,87**
Moyenne utilisateurs (*jeuB*)	0,86	0,54	0,61	**0,75**	0,50	0,55	0,19
Moyenne Citron (*jeuB*)	0,70	0,58	0,60	0,70	**0,58**	**0,60**	**0,90**
Moyenne utilisateurs	0,91	0,64	0,70	0,85	0.61	**0.66**	0.17
Moyenne Citron	0,72	0,56	0,56	0,72	0,56	0,56	**0,88**

Tableau 5.6: Moyennes de *jeuA* et *jeuB* pour les questions auxquelles Citron répond (*hum* pour l'évaluation des réponses selon les critères *humains* et *QR* pour l'évaluation des réponses selon les critères d'une campagne d'évaluation). P pour précision, R pour rappel, F pour F-mesure, T pour temps.

vée, les humains doivent consacrer beaucoup plus de temps qu'un système. Enfin, le point sans doute le plus important est que les utilisateurs ont majoritairement produit des réponses faisant apparaître des critères variants lorsqu'ils existent ce qui montre (1) l'intérêt pour un système comme Citron de reproduire ce comportement et (2) la nécessité d'évaluer les systèmes autrement que par les critères des campagnes d'évaluation.

5.4.2 *Expérience de satisfaction utilisateur sur la présentation des réponses*

Le but de cette seconde expérience *(satisfaction)* est de mesurer la satisfaction des utilisateurs devant la présentation des réponses.

5.4.2.1 *Présentation de l'expérience*

Après avoir extrait les réponses à un des deux jeux de questions-ARM pendant l'expérience *extraction*, l'utilisateur exprime son ressenti sur des réponses à l'autre jeu de question-ARM. Le but est ici qu'un utilisateur compare la façon dont sont présentées les réponses correctes à une question-ARM.

Ces réponses ont été extraites par des utilisateurs pour chacun des deux jeux : en effet, les deux premiers utilisateurs de l'expérience *extraction* ont eu une tâche supplémentaire par rapport aux autres puisqu'ils devaient fournir, pour chaque réponse extraite, un support ainsi qu'une *réponse globale*. Une réponse globale est un passage continu de 250 caractères maximum contenant une des réponses extraites, idéalement toutes (il ne peut y avoir qu'une réponse globale par document). Ces contraintes correspondent au guide de la campagne d'évaluation Quaero 2009 pour les questions-listes.

Le terme de *réponse globale* est également utilisé ici pour désigner le groupe de réponses individuelles tel que Citron les aurait agrégées, en ajoutant notamment le critère variant s'il existe.

La figure 5.6 montre l'interface qui est proposée aux utilisateurs. L'utilisateur voit pour chaque question-ARM deux colonnes présentant les mêmes réponses extraites par un être humain :

- format campagne d'évaluation Quaero : de 1 à 3 bloc-réponses. Un bloc-réponse se compose d'une réponse globale continue (250 caractères maximum), d'un support continu (8 000 caractères maximum) et de n réponses individuelles provenant du support ;
- format Citron : une réponse globale composée des n réponses individuelles choisies par les deux utilisateurs et une liste de supports (pas forcément continus). Chaque réponse individuelle peut être accompagnée de critères variants.

Chaque formatage est présenté aléatoirement cinq fois à droite et cinq fois à gauche.

Nous souhaitons ainsi vérifier si le fait de présenter des réponses agrégées apporte un plus au niveau de la satisfaction utilisateur par rapport à la présentation utilisée pour les campagnes d'évaluation.

Pour chacune des 10 questions du jeu concerné (soit *jeuA*, soit *jeuB*), cinq questions sont posées à l'utilisateur pour lui demander quel format de réponse il préfère : l'utilisateur doit choisir entre les deux colonnes. Ces questions sont :

1. *Est-ce que la totalité des réponses individuelles est correcte ?* Certaines réponses proposées par les utilisateurs sont incorrectes, nous souhaitons notamment voir quelle présentation aidera le plus l'utilisateur à s'en rendre compte.
2. *En ne prenant en compte que les réponses globales (pas les supports, ni les réponses individuelles), quelle colonne répond le mieux à la question posée ?* Nous voulons voir si les réponses globales de 250 caractères imposés par les campagnes sont suffisantes pour être considérées comme des réponses correctes malgré leur limitation de taille.
3. *Quelle colonne justifie le mieux ses réponses ?* Nous voulons voir si le fait que Citron ajoute des informations de critères variants améliore la justification des réponses pour l'utilisateur.
4. *Quelle colonne est la plus agréable à lire sur un écran d'ordinateur ?*
5. *Si vous aviez posé cette question par SMS depuis un téléphone portable, quelle colonne auriez-vous souhaité recevoir par SMS ?* Ces deux dernières questions sont des préférences de ressenti visant notamment à contrôler si la longueur est déterminante selon le support de lecture utilisé. Elles permettent également d'en savoir plus quant à un cadre applicatif d'un SQR.

Le choix neutre a été volontairement supprimé afin de obliger les utilisateurs à prendre position : *très nettement la gauche*, *plutôt la gauche*, *plutôt la droite*, *très nettement la droite*. Cette obligation empêche notamment l'utilisateur de répondre *aucune* ou *les deux*, notamment pour les cas où les deux colonnes sont presque identiques.

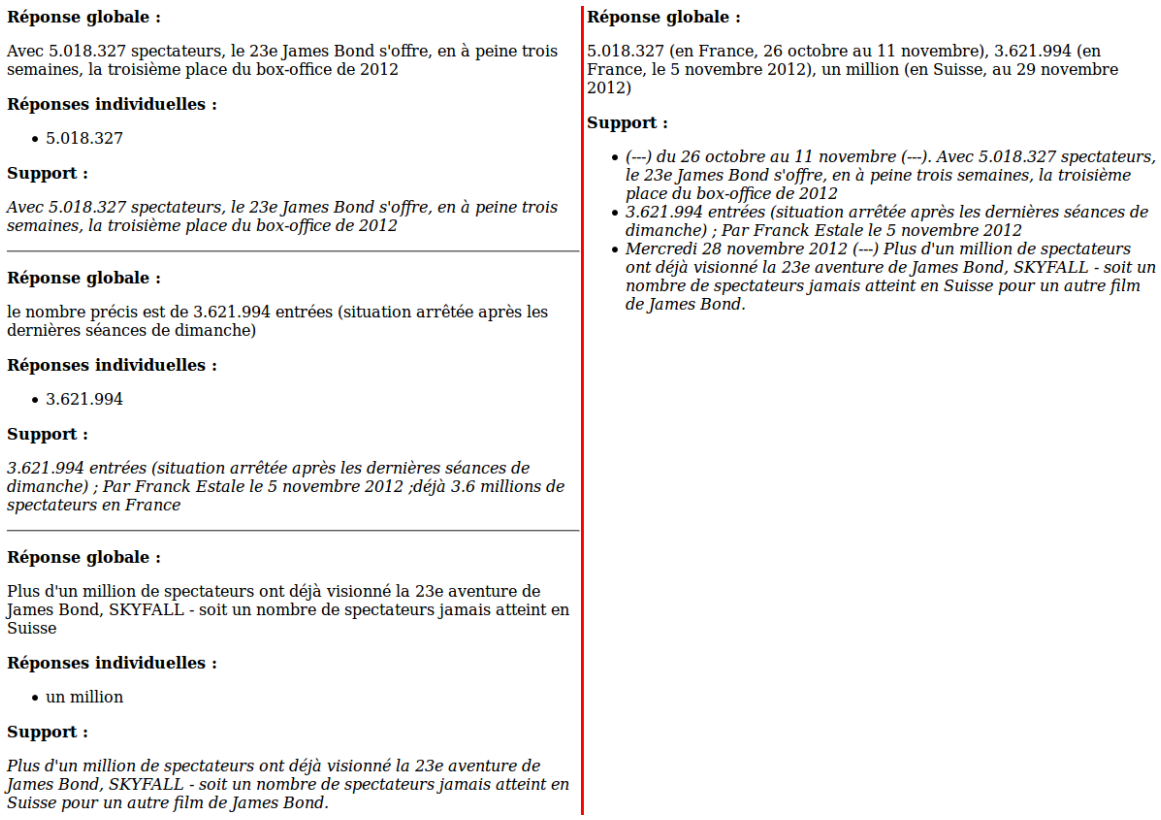

FIGURE 5.6 : Interface de présentation des réponses en deux colonnes sur la question *Combien de spectateurs ont vu Skyfall ?* (Formatage campagne d'évaluation à gauche, formatage Citron à droite.

5.4.2.2 *Résultats*

Le tableau 5.7 montre que 71,39 % des utilisateurs ont trouvé que la présentation des réponses *à la* Citron répondait mieux à la question, ce qui semble confirmer que la présentation sous forme de réponses globales n'est pas l'approche idéale. La présentation *à la* Citron est vue comme justifiant le mieux ses réponses à 61,79 %, ce qui semble aussi confirmer l'intérêt d'ajouter le critère variant aux réponses extraites quand il existe et que l'utilisation de supports non continus est plus attractive qu'un passage continu. La présentation en mode campagne d'évaluation a toutefois été légèrement plus décisive pour prouver que des réponses étaient incorrectes (pour 3 questions). Enfin, les réponses présentées *à la* Citron sont plus agréables à lire à 65,5 % sur écran d'ordinateur et 75,26 % sur téléphone portable ; les utilisateurs ayant toutefois précisé que les deux colonnes auraient été indigestes sur un petit écran. Il s'agit ici de moyenne mais les deux questions sur les recettes de cuisine ont été préférées en présentation *à la* campagne d'évaluation :

les utilisateurs ont précisé que voir les listes d'ingrédients par bloc (trois blocs-réponses) plutôt que juxtaposés dans une unique liste les aidait.

Est-ce que la totalité des réponses individuelles est correcte ?				
	non (la colonne de gauche me le prouve)	oui	non (la colonne de droite me le prouve)	
Moyenne (jeux A et B)	19.67	**67.58**	12.75	
En ne prenant en compte que les réponses globales (pas les supports, ni les réponses individuelles,) quelle colonne répond le mieux à la question posée ?				
	très nettement la gauche	plutôt la gauche	plutôt la droite	très nettement la droite
Moyenne (jeux A et B)	7.27	21.34	**30.0**	**41.39**
Quelle colonne justifie le mieux ses réponses ?				
	très nettement la gauche	plutôt la gauche	plutôt la droite	très nettement la droite
Moyenne (jeux A et B)	8.46	29.74	**42.73**	**19.06**
Quelle colonne est la plus agréable à lire sur un écran d'ordinateur ?				
	très nettement la gauche	plutôt la gauche	plutôt la droite	très nettement la droite
Moyenne (jeux A et B)	10.35	24.16	**38.42**	**27.08**
Si vous aviez posé cette question par SMS, quelle colonne auriez-vous souhaité recevoir par SMS ?				
	très nettement la gauche	plutôt la gauche	plutôt la droite	très nettement la droite
Moyenne (jeux A et B)	8.14	16.6	**40.35**	**34.91**

Tableau 5.7: Préférences des utilisateurs pour la présentation des réponses : *gauche* est le formatage des campagnes d'évaluation et *droite* celui de Citron.

5.5 CONCLUSION

L'originalité de Citron est de pouvoir :

– extraire des réponses à des questions-ARM dans des documents Web contenant notamment des tableaux et des structures énumératives ;

– présenter des réponses agrégées et les accompagner d'éventuels critères variant pour une meilleure compréhension des réponses.

L'évaluation du SQR Citron dans différentes configurations a permis de confirmer ces points. Tout d'abord, nous avons notamment pu voir que même en se plaçant dans des conditions idéales, la tâche d'extraction de réponses à des questions-ARM restait difficile alors que l'agrégation de réponses devenait plus accessible. Cependant, les résultats obtenus sont encourageants et montrent l'intérêt pour un SQR de pouvoir traiter les éléments structuraux (tableaux, structures énumératives) dans l'utilisation d'une collection de documents issus du Web.

Ensuite, nous avons étudié le comportement d'utilisateurs pour la tâche d'extraction de réponses à des questions-ARM afin de pouvoir comparer leurs performances à celles de Citron. Les résultats nous ont permis de confirmer l'hypothèse qu'un être humain est plus performant que Citron en terme de qualité des réponses mais à un coût temporel très fort.

Nous avons également étudié la satisfaction des utilisateurs concernant la présentation des réponses et nous avons constaté que la présentation *à la* Citron qui agrège les réponses et ajoute les éventuels critères variants était préférée par les utilisateurs, notamment parce qu'elle aide mieux à comprendre la justification des réponses et leur multiplicité.

CONCLUSION ET PERSPECTIVES

Les systèmes de question-réponse actuels, ainsi que les campagnes d'évaluation font en général l'hypothèse qu'une seule réponse est attendue pour une question. Or nous avons constaté que, souvent ce n'était pas le cas, surtout quand on cherche les réponses sur le Web et pas dans une collection finie de documents.

Nous nous sommes donc intéressés au traitement des questions attendant plusieurs réponses à travers un système de question-réponse sur le Web en français. Pour cela, nous avons développé le système *Citron* capable d'extraire des réponses multiples à des questions factuelles en domaine ouvert. Nous avons montré grâce à notre étude de différents corpus que les réponses à de telles questions se trouvaient souvent dans des tableaux ou des structures énumératives mais que ces structures, par le codage HTML et la liberté qu'il offre aux rédacteurs de pages Web, sont difficilement analysables automatiquement sans prétraitement. Nous avons aussi fait le constat que les formats de réponses imposés généralement aux systèmes de question-réponse par les campagnes d'évaluation sont peu adaptés non seulement à une évaluation pour ce type de questions mais aussi aux attentes que pourraient avoir de vrais utilisateurs en terme de présentation des réponses.

CONTRIBUTIONS

Étant donné ces constats et nos objectifs, nos principales contributions portent sur :

– **Le prétraitement des documents HTML.** Pour qu'un système de question-réponse puisse analyser, notamment syntaxiquement, des documents HTML et en extraire des réponses, il est indispensable de les prétraiter pour en extraire du texte exploitable. Pour cela, nous avons développé l'outil *Kitten* qui permet non seulement d'extraire le contenu des documents HTML sous forme de texte mais surtout de repérer, analyser et formater les éléments structuraux (tableaux et listes) susceptibles de contenir des réponses aux questions à réponses multiples. Kitten est ainsi capable de différencier les tableaux de formatage des tableaux de données et, pour ces derniers, d'identifier les cases entêtes et données. Il est aussi capable de repérer les structures énumératives verticales et horizontales et d'en identifier l'amorce et les items. Tous ces prétraitements ont pour but de rendre ces éléments analysables syntaxiquement et de permettre une validation du type des réponses plus efficace. Les résultats ont montré une amélioration des performances de plusieurs systèmes

de question-réponse utilisant les documents prétraités par *Kitten*.

– **Citron, un système d'extraction de réponses multiples sur le Web.** Nous avons développé *Citron*, un système capable, pour une question factuelle, d'extraire des candidats-réponses à partir de textes, de tableaux ou de structures énumératives de différentes formes puis d'en valider le type, soit directement dans les documents, soit par Wikipédia, afin de proposer une liste de réponses multiples si nécessaire. *Citron* obtient des résultats encourageants dans un temps d'exécution raisonnable : les expérimentations ont montré un temps moyen de 22 secondes par question et une durée maximale de 2 minutes sur un ordinateur « classique ».

– **La présentation des réponses.** Nous avons montré par nos études de corpus que les différentes occurrences de réponses correctes trouvées dans des documents issus du Web ont très souvent besoin d'être agrégées, par exemple pour regrouper des entités identiques désignées sous des formes différentes. Nous avons aussi montré que ces réponses pouvaient être multiples à cause de la présence d'un critère variant. Nous pensons que l'agrégation des réponses correctes désignant une même entité ainsi que l'ajout du critère variant aident l'utilisateur à mieux comprendre la justification des réponses. Pour cela, *Citron* est donc capable de détecter des critères variants de type temporel et géographique et d'agréger les réponses sur leur forme de surface pour proposer les réponses aux utilisateurs sous une forme originale.

– **Un cadre d'évaluation.** Les formats de réponses imposés généralement aux systèmes de question-réponse par les campagnes d'évaluation sont peu adaptés aux questions à réponses multiples et la présentation des réponses que nous proposons à travers *Citron* est difficilement évaluable dans ce cadre. Nous avons donc réalisé une évaluation de ses performances pour l'extraction de réponses selon les critères des campagnes d'évaluation même s'ils sont peu adaptés, mais nous avons aussi évalué ces mêmes performances avec des critères « humains » (ce qu'un humain considère comme une bonne réponse). Enfin, nous avons surtout proposé un cadre d'évaluation par des utilisateurs par le biais de deux expériences. Une première expérience sur la tâche d'extraction de réponses multiples a permis de comparer les performances de *Citron* et de vrais utilisateurs en terme d'extraction des réponses. Les résultats ont confirmé que notre système était plus rapide que les êtres humains et que l'écart entre la qualité des réponses de *Citron* et celle des utilisateurs était raisonnable. Un résultat important est que les utilisateurs ont, comme *Citron*, proposé des réponses faisant apparaître les éventuels critères variants. La seconde expérience a permis d'évaluer la satisfaction des utilisateurs concernant la présentation de réponses multiples et a montré que les utilisateurs préfèrent la présentation *à la*

Citron, c'est-à-dire agrégeant les réponses et y ajoutant un critère variant lorsqu'il existe, ce qui aide à la compréhension de la justification des réponses.

PERSPECTIVES

Citron est un système de question-réponse jeune qui a ouvert de nombreuses perspectives durant son développement. À plus ou moins long terme, il est ainsi possible d'envisager :

- **l'amélioration des règles syntaxiques pour l'analyse des questions et l'extraction des réponses.** Cela permettrait non seulement d'élargir le champ des questions analysables par *Citron* (cas des questions complexes : *Pourquoi Internet est-il utile ? Comment perdre du poids ?* (Quaero 2009)) mais également d'améliorer l'extraction de réponses, par exemple dans les structures énumératives horizontales et intraphrastiques : par exemple savoir différencier lorsque l'enumeraTheme, ici la provenance géographique, n'est pas en début d'item : *Au XVIème siècle, chaque Roi mage reçoit une provenance géographique : Melchior au visage blanc vient d'Europe, Gaspard, au visage jaune d'Asie, Balthazar, au visage noir d'Afrique*.

- **la résolution de coréférences et la prise en compte de la date du document.** Nous l'avons vu, il existe de nombreux cas où les passages-réponses sont longs et contiennent des coréférences. Pouvoir les résoudre par des travaux comme ceux de [Cao *et al.*, 2011a] sur l'anglais [6] permettrait d'extraire plus de réponses. L'identification de critères variants temporels pourrait aussi être facilitée si l'on connaît la date de création des documents. Sur le Web, cette information est souvent manquante ou difficile à trouver. Intégrer des outils comme DCTFinder [Tannier, 2014] permettrait d'améliorer ce point.

- **l'amélioration de la validation des candidats-réponses.** Pour le moment, *Citron* utilise l'introduction de Wikipédia pour valider les candidats-réponses. Les résultats sont satisfaisants mais de nombreuses pistes sont à l'étude pour les améliorer (liens des pages entre elles notamment). Une ouverture au Web sémantique est à l'étude par exemple pour l'agrégation de réponses à l'aide d'une réconciliation de référence profonde.

- **l'extraction de critères variants.** Les expérimentations ont notamment montré l'importance du critère variant dans la satisfaction des utilisateurs pour la présentation des réponses. Nos premiers travaux sur les critères variants temporel et géographique sont à poursuivre sur d'autres types (attributs adjectivaux et nominaux) :

6. http://cogcomp.cs.illinois.edu/page/software_view/18

un important travail de filtrage doit être implémenté afin de différencier les attributs étant des critères variants de ceux n'en étant pas.

– **le passage à l'anglais.** *Citron* utilise des outils qui fonctionnent également sur l'anglais. Ainsi, *Kitten* est capable de prétraiter des documents en anglais puisque son fonctionnement est indépendant de la langue (il faudrait cependant évaluer la qualité de la détection des structures énumératives en anglais). Les règles XIP que nous avons définies pour l'analyse des questions et l'extraction des relations de définition existent aussi pour l'anglais. Ce passage à l'anglais pourrait donc se faire à coût raisonnable. Il pourrait également se faire pour d'autres langues à condition de disposer d'un analyseur linguistique en dépendances et de connaissances sur la langue afin d'adapter les règles existantes et d'en créer de nouvelles. En plus de l'anglais, nous pouvons par exemple également adapter *Citron* au finnois [Haverinen *et al.*, 2013], la question de la taille du Web se posera toutefois pour les deux langues : plus petit que le français pour le finnois, plus grand pour l'anglais.

– **l'analyse plus fine du comportement des utilisateurs**. Les résultats des expérimentations utilisateurs ont été très fertiles et nous pouvons notamment affiner les résultats obtenus, par exemple en fonction des utilisateurs ou des types de questions. Nous n'avons pas encore analysé toutes les données de l'expérience sur l'extraction de réponses : la redondance des réponses dans les documents ainsi que les parcours de clics des utilisateurs sont des informations à analyser prochainement.

– **la mise à disposition publique de *Citron*.** Ceci permettrait à de vrais utilisateurs de formuler leurs propres questions, répondant alors à un réel besoin.

Première partie

ANNEXES

A

TERMINOLOGIE

Sont ici recensés les termes définis pour ce manuscrit selon le patron suivant :

terme : courte définition (numéro de page de la définition plus détaillée)

amorce : phrase introductrice d'une structure énumérative (page 21).

candidat-réponse : candidat à la réponse pour une question donnée (page 31).

case en-tête : case d'un tableau spécifiant le sens des cases données de sa ligne ou de sa colonne (page 25).

case donnée : case d'un tableau pouvant contenir une donnée ou être vide. L'interprétation de cette donnée dépendra quasiment toujours d'une case en-tête, voire de deux (page 25).

éléments structuraux : terme regroupant les objets porteurs d'une structure dans un document comme par exemple les tableaux, les listes, les paragraphes ou encore les tables de matière (page 20).

énumération : élément structural porteur d'une séquence d'items (page 20).

énumération verticale : énumération débutant par une amorce délimitée par un « : » et dont les items sont séparés par un retour-chariot (page 24).

énumération horizontale : énumération débutant par une amorce délimitée par un « : » et dont les items sont séparés par un symbole de ponctuation comme un point-virgule ou une virgule (page 24).

énumération intra-phrastique : énumération ne comportant pas d'amorce délimitée par un « : » et ne dépassant pas le cadre de la phrase (page 24).

F-mesure : la F-mesure est la pondération de la précision et le rappel. Nous l'utilisons avec la formule suivante : $F = \frac{2.\text{Précision.Rappel}}{\text{Précision+Rappel}}$ (page 36).

focus : ce sur quoi porte la question (page 12).

focusSE : ce sur quoi porte la structure énumérative (SE) (page 114).

item : unité d'une énumération, par exemple une phrase ou un syntagme (page 21).

passage-réponse : extrait de document qui contient la réponse-liste ou une ou des réponses individuelles (page 47).

précision : la précision est le nombre de réponses correctes divisé par le nombre de réponses fournies (page 36).

puce : symbole précédant l'item d'une énumération verticale (page 24).

question-ARM (à réponses multiples) : question pouvant attendre plusieurs réponses correctes. Par exemple, les questions-listes font partie des questions-ARM (page 75).

question-liste : question de type liste. Elle attend donc plusieurs réponses sous la forme d'une liste (page 47).

rappel : le rappel est le nombre de réponses correctes divisé par le nombre de réponses correctes existantes dans la collection (page 36).

relié : un nom ou verbe est relié thématiquement à un nom ou verbe parce qu'il apparait dans son contexte (page 132).

réponse-liste : groupe se composant de candidats-réponses individuels (page 47).

réponse multi-document : réponses individuelles correctes pour une question et provenant de deux documents différents (page 50).

snippet : extrait d'un document renvoyé par un moteur de recherche (page 38).

SE (structure énumérative) : structure composée d'une amorce et d'une énumération composée d'items (page 21).

SQR (système de question-réponse) : système fournissant une réponse à une question formulée en langue naturelle (page 12).

tableau de formatage : tableau utilisant la structure d'un tableau pour disposer visuellement des morceaux de texte sans rapport avec des entêtes (page 93).

unicitable : un passage-réponse est unicitable s'il est compréhensible sans contexte supplémentaire provenant du document et s'il valide la réponse pour une question donnée (page 59).

B

ENUMERATHEME

Liste des 128 enumeraThemes gouvernés par une entité numérique et validés manuellement comme un enumeraTheme. Ces enumeraThemes ont été recensés dans les énumérations verticales du corpus Quaero à l'aide de Kitten et Lucene (détail de l'étude page 113) :

1. type
2. point
3. partie
4. étape
5. catégorie
6. chose
7. réponse
8. personne
9. axe
10. raison
11. élément
12. possibilité
13. cas
14. méthode
15. domaine
16. objectif
17. solution
18. groupe
19. façon
20. phase
21. condition
22. niveau
23. mode
24. question
25. principe
26. thème
27. critère
28. option
29. problème
30. manière
31. fonction
32. temps
33. forme
34. facteur
35. volet
36. sorte
37. projet
38. aspect
39. paramètre
40. classe
41. caractéristique
42. outil
43. contribution
44. version
45. zone
46. dimension
47. section
48. pôle
49. technique
50. programme
51. module
52. ligne
53. approche
54. règle
55. opération
56. activité
57. composante
58. avantage
59. champ
60. pts

61. pièce
62. exemple
63. choix
64. action
65. série
66. ordre
67. état
68. composant
69. format
70. valeur
71. pilier
72. notion
73. mesure
74. information
75. entrée
76. ensemble
77. attribut
78. truc
79. tendance
80. sujet
81. sous-catégorie
82. produit
83. mécanisme
84. direction
85. conséquence
86. référence
87. proposition
88. poste
89. partition
90. modèle
91. concept
92. cadre
93. bloc
94. argument
95. volume
96. structure
97. stratégie
98. phénomène
99. morceau
100. idée
101. fonctionnalité
102. couche
103. branche
104. standard
105. secteur
106. rubrique
107. procédé
108. piste
109. passage
110. paquet
111. inconvénient
112. cause
113. variable
114. thématique
115. terme
116. stade
117. relation
118. position
119. particularité
120. origine
121. effet
122. défaut
123. aire
124. variante
125. unité
126. tranche
127. signification
128. rôle

C

TYPE DES ARTICLES WIKIPÉDIA

Liste des types les plus fréquemment trouvés dans la Wikipédia en français à l'aide de nos règles DEFINITION (détails de l'étude page 119) :

1. français (88179)
2. commune (76 926)
3. ville (41 863)
4. nom (35 742)
5. film (31 873)
6. espèce (31 593)
7. joueur (27 443)
8. acteur (26 838)
9. village (26 135)
10. américain (25 401)
11. homme (23 753)
12. album (21 234)
13. ancien (20 352)
14. groupe (19 649)
15. footballeur (19 394)
16. genre (18 374)
17. championnat (17 809)
18. ~~thé (17 197)~~
19. série (15 924)
20. municipalité (14 884)
21. département (14 673)
22. famille (14 300)
23. italien (14 284)
24. écrivain (13 780)
25. catégorie (13 622)
26. ~~lister (12 535)~~
27. club (12 341)
28. titre (11 890)
29. édition (11 641)
30. allemand (11 508)
31. comté (11 389)
32. district (11 378)
33. chanteur (10 480)
34. auteur (10 291)
35. membre (9 893)
36. province (9 812)
37. fils (9 788)
38. état (9 766)
39. premier (9 510)
40. communauté (9 489)
41. île (9 476)
42. équipe (9 464)
43. église (9 185)
44. peintre (9 090)
45. grand (8 971)
46. localité (8 927)
47. football (8 863)
48. administratif (8 726)
49. personnage (8 593)
50. réalisateur (8 431)
51. division (8 249)
52. région (8 246)
53. jeu vidéo (8 154)
54. château (8 116)
55. histoire (8 006)
56. politique (7 968)
57. ~~jean (7 893)~~
58. chanson (7 869)
59. roman (7 710)
60. homonymie (7 617)
61. compositeur (7 542)
62. terme (7 460)
63. rivière (7 342)
64. société (7 223)
65. ~~l'un (7 122)~~
66. gare (6 916)
67. journaliste (6 895)
68. ~~pierre (6 610)~~

69. canton (6 526)
70. prix (6 508)
71. type (6 409)
72. coupe (6 284)
73. scénariste (6 260)
74. athlète (6 218)
75. liste (6 075)
76. maison (5 978)
77. professionnel (5 933)
78. britannique (5 892)
79. coureur (5 842)
80. ensemble (5 828)
81. rue (5 807)
82. suisse (5 781)
83. centre (5 761)
84. ~~xian (5 739)~~
85. jeu (5 664)
86. anglais (5 648)
87. champion (5 638)
88. producteur (5 630)
89. france (5 629)
90. petit (5 563)
91. siège (5 527)
92. voie (5 516)
93. belge (5 482)
94. historien (5 478)
95. entreprise (5 445)
96. ~~charles (5 393)~~
97. ~~françois (5 379)~~
98. cycliste (5 378)
99. ~~john (5 361)~~
100. artiste (5 303)

D

LISTE DES 20 QUESTIONS-ARM DE L'EXPÉRIENCE UTILISATEUR

Liste des deux jeux de questions-ARM utilisées pour l'expérience utilisateur (page 180) :

Les dix questions-ARM du *JeuA* :

1. Quels sont les signes du zodiaque ?
2. Quelles sont les quatre nobles vérités du bouddhisme ?
3. Comment s'appellent les Mousquetaires ?
4. Combien de spectateurs ont vu Avengers ?
5. Dans quels clubs a joué Laurent Blanc ?
6. Qui a joué Knocking on heaven's door ?
7. Où s'est déroulée la conférence CORIA ?
8. Quels sont les ingrédients d' un chili végétarien ?
9. Quand le Jokerit a -t-il remporté le championnat de Finlande de hockey sur glace ?
10. Quelles villes ont été la capitale du Japon ?

Les dix questions-ARM du *JeuB* :

1. Quels sont les péchés capitaux ?
2. Quels sont les dix commandements ?
3. Comment s'appellent les Simpson ?
4. Combien de spectateurs ont vu Skyfall ?
5. Dans quels clubs a joué Didier Deschamps ?
6. Qui a joué Where did you sleep last night ?
7. Où s'est déroulée la conférence TALN ?
8. Quels sont les ingrédients d' un couscous végétarien ?
9. Quand le Kärpät a-t-il remporté le championnat de Finlande de hockey sur glace ?
10. Quelles villes ont été la capitale des États-Unis ?

E

RETOURS DES UTILISATEURS DURANT LES EXPÉRIENCES D'EXTRACTION ET DE SATISFACTION DES RÉPONSES

Nous évoquons ici plusieurs retours des utilisateurs à des fins informatives, sans volonté d'en tirer des conclusions. En effet, nous avons laissé la possibilité aux utilisateurs de laisser leurs impressions sur les deux expériences qu'ils venaient de passer. Notre but n'était pas de réaliser une expérience dans les domaines de la cognition et de l'apprentissage mais puisqu'elle affleurait ces domaines, nous trouvons intéressant de mentionner ici le vécu dont nous ont fait part les utilisateurs, notamment du point de vue de ce qu'est une réponse correcte pour un être humain.

L'EXTRACTION DE RÉPONSES

L'extraction du support a été une tâche très complexe : plusieurs utilisateurs ne comprenaient pas concrètement ce qui pouvait justifier une réponse dans un extrait de document.

Les tableaux formatés par Kitten ont été bien analysés mais plusieurs utilisateurs les ont trouvés barbares et auraient préféré visualiser le tableau d'origine.

Plusieurs utilisateurs ont expliqué n'avoir pas compris la question en la lisant la première fois, puis avoir compris à l'aide des documents. Par exemple dans la question *Qui a joué Where did you sleep last night ?*, des utilisateurs ont d'abord pensé qu'il s'agissait d'un film avant de voir qu'il s'agissait d'une chanson.

Plusieurs utilisateurs ont eu des difficultés pour juger si une réponse était correcte du point de vue de leur connaissance du monde. Par exemple pour la question *Quelles villes ont été la capitale des États-Unis ?*, le candidat-réponse *Miami* a été saisi avec le support *Miami - Capitale des Caraïbes*. Pour la question *Comment s'appellent les Simpson ?*, des utilisateurs ont précisé ne pas avoir extraits les candidats-réponse *Cody* et *Jessica* dont le nom de famille était *Simpson* parce qu'il était question de la famille Simpson du dessin animé éponyme. Pour cette question, plusieurs utilisateurs ont toutefois retourné qu'ils se demandaient qu'elle était la portée de la notion de famille (grand-parents, demi-frère).

La supervision des utilisateurs a montré quatre façons de réaliser un copier-coller pour extraire les réponses :

1. sélection à la souris puis glissé déposé ;
2. sélection à la souris puis copier avec le raccourci clavier et coller avec le raccourci clavier ;
3. sélection à la souris puis coller avec le bouton de la molette de la souris (sous Linux) ;
4. sélection à la souris puis clic droit pour copier puis clic droit pour coller.

La première méthode nous semble la plus rapide mais n'a été utilisé en supervision que par deux utilisatrices : une programmeuse de la tranche d'âge 26-40 et une utilisatrice normale de la tranche d'âge des plus de 41 ans.

Malgré toutes les précautions prises au niveau de l'encodage UTF-8 avec respect des standards, un utilisateur a eu un problème d'affichage sur le tibétain sous le système d'exploitation Mac OS X.

LA SATISFACTION UTILISATEUR SUR LA PRÉSENTATION DES RÉPONSES

Plusieurs utilisateurs auraient aimé un choix *neutre* lorsque aucune des présentations de réponse ne leur plaisait.

Plusieurs utilisateurs auraient également aimé un bouton *les deux* lorsque les deux colonnes leur plaisaient.

BIBLIOGRAPHIE

Rafik Abbes, Arlind Kopliku, Karen Pinel-Sauvagnat, Nathalie Hernandez et Mohand Boughanem : Apport du Web et du Web de Données pour la recherche d'attributs. *In Conférence francophone en Recherche d'Information et Applications (CORIA), Neuchâtel, Suisse*, avril 2013. (Citée en page 29.)

Stergos Afantenos, Nicholas Asher, Farah Benamara, Myriam Bras, Cecile Fabre, Mai Ho-Dac, Anne Le Draoulec, Philippe Muller, Marie-Paul Pery-Woodley, Laurent Prevot, Josette Rebeyrolles, Ludovic Tanguy, Marianne Vergez-Couret et Laure Vieu : An empirical resource for discovering cognitive principles of discourse organisation : the annodis corpus. *In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7. (Citée en page 121.)

Mikhail Ageev, Qi Guo, Dmitry Lagun et Eugene Agichtein : Find It If You Can : A Game for Modeling Different Types of Web Search Success Using Interaction Data . *In SIGIR*, pages 345–354, 2011. (Citée en pages 173 et 185.)

S. Aït-Mokhtar, J.-P. Chanod et C. Roux : Robustness beyond shallowness : incremental deep parsing. *Nat. Lang. Eng.*, 8, 2002. (Citée en pages 26 et 111.)

C. Ayache : Evaluation en question-réponse,rapport final de la campagne EVALDA. *In Campagne EVALDA/EQueR :*, 2005. (Citée en pages 36, 48, et 82.)

C. Ayache, B. Grau et A. Vilnat : EQueR : the French evaluation campaign of question-answering systems. *In Proceedings of The Fifth International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy*, 2006. (Citée en pages 32, 47, 48, et 107.)

Al-maskari Azzah et Mark Sanderson : A Review of Factors Influencing User- satisfaction in Information Retrieval. *Journal of the American Society for Information Science and Technology. Published*, pages 1–06, 2010. (Citée en page 173.)

S. Aït-Mokhtar, V. Lux et E. Banik : Linguistic parsing of lists in structured documents. *In Proceedings of the EACL Workshop on Language Technology and the Semantic Web (3rd Workshop on NLP and XML, NLPXML-2003), Budapest, Hungary*, 2003. (Citée en pages 26 et 112.)

K. Balog, P. Serdyukov et A.P. de Vries : Overview of the TREC 2011 entity track. *NIST Special Publication : TREC*, 2011. (Citée en page 29.)

K. Balog, P. Serdyukov et A.P. Vries : Overview of the TREC 2010 entity track. Rapport technique, DTIC Document, 2010. (Citée en page 29.)

Krisztian Balog, Pavel Serdyukov, Arjen P. De Vries, Paul Thomas et Thijs Westerveld : Overview of the TREC 2009 Entity Track. *In TREC-18*, 2009. (Citée en page 29.)

Eva Banik, Salah Aït-Mokhtar et Veronika Lux : Linguistic Parsing of Structured Documents. Rapport technique 2002/054, Xerox Research Center Europe, 2002. (Citée en page 27.)

Chris Barry et Mark Lardner : A Study of First Click Behaviour and User Interaction on the Google SERP. *In Information Systems Development*, pages 89–99. Springer New York, 2011. (Citée en page 173.)

Guillaume Bernard : *Réordonnancement de candidats reponses pour un système de questions-réponses*. Thèse de doctorat, Université Paris-Sud, 2011. (Citée en page 32.)

Johan Bos, Edoardo Guzzetti et James R. Curran : The Pronto QA System at TREC 2007 : Harvesting Hyponyms, Using Nominalisation Patterns, and Computing Answer Cardinality. *In TREC-16*, 2007. (Citée en pages 31 et 37.)

Gosse Bouma, Ismail Fahmi, Jori Mur, Gertjan Van Noord, Lonneke Van der Plas et Jörg Tiedemann : Linguistic Knowledge and Question Answering. *Traitement Automatique des Langues*, 46(3):15–39, 2005. (Citée en page 32.)

JL Bouraoui et N. Vigouroux : Les marqueurs des structures énumératives sur le web : analyse pour la transmodalité. *In Conférence Internationale sur le Document Electronique (CIDE 6)*, pages 199–217, 2003. (Citée en page 27.)

Myriam Bras, Laurent Prévot et Marianne Vergez-Couret : Quelle(s) relation(s) de discours pour les structures énumératives ? *In CMLF (Congrès mondial de linguistique française)*, 2008. (Citée en page 21.)

Michael J. Cafarella, Alon Y. Halevy, Daisy Zhe Wang, Eugene Wu et Yang Zhang : WebTables : exploring the power of tables on the web. *PVLDB*, 1(1):538–549, 2008a. (Citée en page 29.)

Michael J Cafarella, Alon Y Halevy, Yang Zhang, Daisy Zhe Wang et Eugene Wu : Uncovering the Relational Web. *In WebDB*. Citeseer, 2008b. (Citée en page 28.)

Ling Cao, Xipeng Qiu et Xuanjing Huang : Question Answering for Machine Reading with Lexical Chain. *In CLEF (Notebook Papers/Labs/Workshop)*, 2011a. (Citée en page 204.)

Yong-Gang Cao, Feifan Liu, Pippa Simpson, Lamont Antieau, Andrew Bennett, James J Cimino, John Ely et Hong Yu : AskHERMES : An online question answering system for

complex clinical questions. *Journal of biomedical informatics*, 44(2):277–288, 2011b. (Citée en page 173.)

Pedroa CASTRO et VDOSS : jChardet, portage java de l'algorithme de détection d'encodage de Mozilla. http://jchardet.sourceforge.net/index.html, 2013. (Citée en page 89.)

Yllias CHALI et Shafiq R JOTY : University of Lethbridge's Participation in TREC 2007 QA Track. *In TREC*, 2007. (Citée en page 76.)

Jiangping CHEN, He GE, Yan WU et Shikun JIANG : UNT at TREC 2004 : Question Answering Combining Multiple Evidences. *In TREC*, 2004. (Citée en page 32.)

Jennifer CHU-CARROLL, Krzysztof CZUBA, John PRAGER et Sasha BLAIR-GOLDENSOHN : IBM's PIQUANT II in TREC2004. *In TREC-13*, 2004. (Citée en page 37.)

Alton Y. K. CHUA et Snehasish BANERJEE : So fast so good : An analysis of answer quality and answer speed in community Question-answering sites. *Journal of the American Society for Information Science and Technology*, 64(10):2058–2068, 2013. (Citée en page 172.)

Stephen COLEBOURNE et BS O'NEILL : Joda-Time, API Java de manipulation de données temporelles. http://joda-time.sourceforge.net, 2013. (Citée en page 138.)

Dan CROW : Google squared : Web scale, open domain information extraction and presentation. *In ECIR*, 2010. (Citée en pages 40 et 41.)

Hang CUI, Renxu SUN, Keya LI, Min-Yen KAN et Tat-Seng CHUA : Question answering passage retrieval using dependency relations. *In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 400–407. ACM, 2005. (Citée en page 32.)

T. DALMAS et B. WEBBER : Using information fusion for open domain question answering. *In Proceedings of KRAQ 2005 Workshop, IJCAI*, 2005. (Citée en page 38.)

Christine Clark DAN I. MOLDOVAN et Moldovan BOWDEN : Lymba's PowerAnswer 4 in TREC 2007. *In TREC-16*, 2007. (Citée en pages 31 et 38.)

Hoa Trang DANG, Diane KELLY et Jimmy LIN : Overview of the TREC 2007 Question Answering Track. *In TREC-16*, 2007. (Citée en pages 33 et 195.)

Hoa Trang DANG, Jimmy LIN et Diane KELLY : Overview of the TREC 2006 Question Answering Track. *In TREC-15*, 2006. (Citée en pages 33, 172, et 195.)

Clément de GROC : *Collecte orientée sur le Web pour la recherche d'information spécialisée*. Thèse de doctorat, Université Paris-Sud, 2013. (Citée en page 89.)

Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin et Andrew Ng : Web Question Answering : Is More Always Better ? *In SIGIR*, pages 291–298. ACM, 2002. (Citée en page 137.)

Sarra El Ayari : *Évaluation transparente du traitement de la variabilité linguistique des éléments de réponse à une question factuelle*. Thèse de doctorat, Université Paris-Sud, 2009. (Citée en page 120.)

Mathieu-Henri Falco : Typologie des questions à réponses multiples pour un système de question-réponse (typology of multiple answer questions for a question-answering system) [in french]. *In Actes de la conférence conjointe JEP-TALN-RECITAL 2012,volume 3 : RECITAL*, page 191–204, Grenoble,France, June 2012. ATALA/AFCP. (Citée en page 121.)

Mathieu-Henri Falco, Véronique Moriceau et Anne Vilnat : Kitten : a tool for normalizing HTML and extracting its textual content. *In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7. (Citée en pages 49, 75, 88, et 106.)

Renxu Sun Jing Jiang Yee Fan, Tan Hang Cui Tat-Seng Chua et Min-Yen Kan : Using syntactic and semantic relation analysis in question answering. *In TREC*, 2005. (Citée en page 32.)

Li Fangtao, Zhang Xian et Zhu Xiaoyan : Answer validation by information distance calculation. *In Coling 2008 : Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, IRQA '08, pages 42–49, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. (Citée en page 32.)

Jean-Philippe Fauconnier, Mouna Kamel, Bernard Rothenburger et Nathalie Aussenac-Gilles : Apprentissage supervisé pour l'identification de relations sémantiques au sein de structures énumératives parallèles. *In* Emmanuel Morin, éditeur : *Traitement Automatique des Langues Naturelles (TALN), Les Sables d'Olonne*, pages 1–14, http ://www.atala.org/, juin 2013. Association pour le Traitement Automatique des Langues (ATALA). (Citée en page 25.)

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager *et al.* : Building Watson : An overview of the DeepQA project. *AI magazine*, 31(3):59–79, 2010. (Citée en page 172.)

Alejandro Figueroa et Günter Neumann : Finding Distinct Answers in Web Snippets. *In In the 4th International Conference on Web Information Systems and Technologies*, pages 26–33. INSTICC Press, 5 2008. (Citée en page 38.)

Pamela Forner, Anselmo Peñas, Eneko Agirre, Iñaki Alegria, Corina Forăscu, Nicolas Moreau, Petya Osenova, Prokopis Prokopidis, Paulo Rocha, Bogdan Sacaleanu *et al.* : Overview of the CLEF 2008 multilingual question answering track. *In Evaluating Systems for Multilingual and Multimodal Information Access*, pages 262–295. Springer, 2009. (Citée en page 33.)

Nicolas Foucault : *Questions-Réponses en domaine ouvert : sélection de documents pertinents en fonction du contexte de la question*. Thèse de doctorat, Université Paris-Sud, 2013. (Citée en page 89.)

Jun-ichi Fukumoto, Tsuneaki Kato et Fumito Masui : Question Answering Challenge (QAC-1) : An Evaluation of Question Answering Tasks at the NTCIR Workshop 3. *In New Directions in Question Answering*, pages 122–133, 2003. (Citée en page 32.)

Junichi Fukumoto, Tsuneaki Kato et Fumito Masui : Question answering challenge for five ranked answers and list answers-overview of NTCIR4 qac2 subtask 1 and 2. *In Proc. 4th NTCIR Workshop Meeting*, 2004. (Citée en pages 32 et 82.)

Junichi Fukumoto, Tsuneaki Kato, Fumito Masui et Tsunenori Mori : An overview of the 4th question answering challenge (qac-4) at ntcir workshop 6. *In Proceedings of NTCIR-6 Workshop Meeting, Tokyo, Japan*, pages 433–440, 2007. (Citée en page 32.)

Nuria Gala : *Un modèle d'analyseur syntaxique robuste fondé sur la modularité et la lexicalisation de ses grammaires*. Thèse de doctorat, Université Paris-Sud, 2003. (Citée en pages 24, 27, et 112.)

Wolfgang Gatterbauer, Paul Bohunsky, Marcus Herzog, Bernhard Krüpl et Bernhard Pollak : Towards domain-independent information extraction from web tables. *In Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 71–80. ACM, 2007. (Citée en pages 28 et 29.)

Danilo Giampiccolo, Pamela Forner, Jesús Herrera, Anselmo Peñas, Christelle Ayache, Corina Forascu, Valentin Jijkoun, Petya Osenova, Paulo Rocha, Bogdan Sacaleanu *et al.* : Overview of the CLEF 2007 multilingual question answering track. *In Advances in Multilingual and Multimodal Information Retrieval*, pages 200–236. Springer, 2008. (Citée en page 33.)

Laurent Gillard, Patrice Bellot et Marc El-Bèze : Question answering evaluation survey. *In Language Resources and Evaluation Conference*, 2006. (Citée en page 34.)

Julio Gonzalo et Douglas W Oard : iCLEF 2004 track overview : pilot experiments in interactive cross-language question answering. *In Multilingual Information Access for Text, Speech and Images*, pages 310–322. Springer, 2005. (Citée en page 185.)

Arnaud GRAPPY : *Validation de réponse dans un système de question-réponse*. Thèse de doctorat, Université Paris-Sud, 2011. (Citée en pages 32, 118, et 151.)

Arnaud GRAPPY, Brigitte GRAU, Mathieu-Henri FALCO, Anne-Laure LIGOZAT, Isabelle ROBBA et Anne VILNAT : Selecting answers to questions from Web documents by a robust validation process. *In IEEE/WIC/ACM International Conference on Web Intelligence,*, 2011. (Citée en pages 106 et 109.)

Qi GUO et Eugene AGICHTEIN : Beyond dwell time : estimating document relevance from cursor movements and other post-click searcher behavior. *In Proceedings of the 21st international conference on World Wide Web*, pages 569–578. ACM, 2012. (Citée en page 173.)

Mark HALL, Eibe FRANK, Geoffrey HOLMES, Bernhard PFAHRINGER, Peter REUTEMANN et Ian H. WITTEN : The WEKA Data Mining Software : An Update ;. *In SIGKDD Explorations*, volume Volume 11,Issue 1, 2009. (Citée en page 97.)

S. HARABAGIU, D. MOLDOVAN, C. CLARK, M. BOWDEN, A. HICKL et P. WANG : Employing two question answering systems in TREC-2005. *In Proceedings of the fourteenth text retrieval conference*, 2005. (Citée en page 38.)

Sanda HARABAGIU, Dan MOLDOVAN, Marius PASCA, Dan MOLDOVAN, Mihai SURDEANU, Roxana GÎRJU, Rada MIHALCEA, Finley LACATUSU, Paul MORARESCU, Razvan BUNESCU et Vasile RUS : Answering complex, list and context questions with lcc's question-answering server. *In TREC-10*, 2001. (Citée en pages 32 et 37.)

Donna HARMAN : TREC-Style Evaluations. *In* Maristella AGOSTI, Nicola FERRO, Pamela FORNER, Henning MÜLLER et Giuseppe SANTUCCI, éditeurs : *Information Retrieval Meets Information Visualization*, volume 7757 de *Lecture Notes in Computer Science*, pages 97–115. Springer Berlin Heidelberg, 2013. (Citée en page 172.)

E. HATCHER, O. GOSPODNETIC et M. MCCANDLESS : *Lucene in action, Second Edition*. Manning Publications Co., 2010. (Citée en pages 50 et 109.)

Katri HAVERINEN, Jenna NYBLOM, Timo VILJANEN, Veronika LAIPPALA, Samuel KOHONEN, Anna MISSILÄ, Stina OJALA, Tapio SALAKOSKI et Filip GINTER : Building the essential resources for finnish : the turku dependency treebank. *Language Resources and Evaluation*, pages 1–39, 2013. (Citée en page 205.)

M.A. HEARST : Automatic acquisition of hyponyms from large text corpora. *In Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992. (Citée en page 25.)

Andrew Hickl, Kirk Roberts, Bryan Rink, Jeremy Bensley, Tobias Jungen, Ying Shi et John Williams : Question Answering with LCC's CHAUCER-2 at TREC 2007. *In TREC*, 2007. (Citée en page 76.)

L.M. Ho-Dac, M.P. Jacques et J. Rebeyrolle : Sur la fonction discursive des titres. *L'unité texte*, pages 125–152, 2004. (Citée en page 22.)

Lydia-Mai Ho-Dac : *La position initiale dans l'organisation du discours : une exploration en corpus*. Thèse de doctorat, Université Toulouse le Mirail, 2007. (Citée en page 21.)

Lydia-Mai Ho-Dac, Marie-Paul Péry-Woodley et Ludovic Tanguy : Anatomie des structures énumératives. *In TALN*, 19-23 juillet 2010 2010. (Citée en pages 21, 22, et 113.)

Christian Jacquemin et Caroline Bush : Fouille du Web pour la collecte d'Entités Nommées. *In TALN*, 2000. (Citée en pages 25 et 28.)

Michael Kaisser et Tilman Becker : Question Answering by Searching Large Corpora With Linguistic Methods. *In TREC-13*, 2004. (Citée en page 37.)

Mouna Kamel et Nathalie Aussenac-Gilles : Utiliser la structure du document dans le processus de construction d'ontologies. *In TIA (Terminology and Artificial Intelligence)*, 2009. (Citée en page 53.)

Mouna Kamel et Bernard Rothenburger : Elicitation de Structures Hiérarchiques à partir de Structures Enumératives pour la Construction d'Ontologie. *In* Alain Mille, éditeur : *Journées Francophones d'Ingénierie des Connaissances (IC), Annecy (F), 17/05/2011-20/05/2011*, pages 507–522. Presses Universitaires des Antilles et de la Guyane, mai 2011. (Citée en pages 24 et 25.)

Tsuneaki Kato, Junichi Fukumoto et Fumito Masui : Question answering challenge for information access dialogue-overview of ntcir-4 qac2 subtask 3. *In Proceesings of the 5th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, pages 291–297, 2004. (Citée en page 32.)

Tsuneaki Kato, Junichi Fukumoto et Fumito Masui : An overview of NTCIR-5 QAC3. *In Proceesings of the 5th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, pages 361–372, 2005. (Citée en page 32.)

Boris Katz et Jimmy Lin : Selectively Using Relations to Improve Precision in Question Answering. *In EACL-2003 workshop on natural language processing for question answering*, 2003. (Citée en page 31.)

Boris Katz, Gregory Marton, Sue Felshin, Daniel Loreto, Ben Lu, Federico Mora, Özlem Uzuner, Michael McGraw-Herdeg, Natalie Cheung, Alexey Radul, Yuan Kui Shen, Yuan Luo et Gabriel Zaccak : Question Answering Experiments and Resources. *In TREC-15*, 2006. (Citée en page 32.)

Jeongwoo Ko, Eric Nyberg et Luo Si : A probabilistic graphical model for joint answer ranking in question answering. *In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 343–350. ACM, 2007. (Citée en page 156.)

C. Kohlschütter, P. Fankhauser et W. Nejdl : Boilerplate detection using shallow text features. *In Proceedings of the third ACM international conference on Web search and data mining*, page 441–450. ACM, 2010. (Citée en page 106.)

Oleksandr Kolomiyets et Marie-Francine Moens : A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, 2011. (Citée en pages 85 et 172.)

Arlind Kopliku, Karen Pinel-Sauvagnat et Mohand Boughanem : Attribute Retrieval from Relational Web tables (regular paper). *In Symposium on String Processing and Information Retrieval (SPIRE), Pisa, Italy*, pages 117–128, http ://www.springerlink.com, octobre 2011. Springer. (Citée en page 29.)

Jennifer Chu-Carroll Krzysztof, Krzysztof Czuba, Pablo Duboue et John Prager : IBM's PIQUANT II in TREC2005. *In In Proceedings of the Fourthteen Text REtrieval Conference (TREC*, 2005. (Citée en page 76.)

Marion Laignelet : *Analyse discursive pour le repérage automatique de segments obsolescents dans les documents encyclopédiques*. Thèse de doctorat, Université de Toulouse - Le Mirail, 2009. (Citée en page 21.)

Marion Laignelet, Mouna Kamel et Nathalie Aussenac-Gilles : Enrichir la notion de patron par la prise en compte de la structure textuelle - application à la construction d'ontologie. *In TALN*, 2011. (Citée en pages 25 et 53.)

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu et Dan Jurafsky : Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. *In Proceedings of the Fifteenth Conference on Computational Natural Language Learning : Shared Task*, pages 28–34. Association for Computational Linguistics, 2011. (Citée en page 76.)

Q. Li : *Searching for Entities : when Retrieval Meets Extraction*. Thèse de doctorat, University of Pittsburgh, 2011. (Citée en pages 28 et 29.)

Jimmy Lin, Dennis Quan, Vineet Sinha, Karun Bakshi, David Huynh, Boris Katz et David R Karger : What makes a good answer ? The role of context in Question Answering. *In Proceedings of the Ninth IFIP TC13 (INTERACT 2003)*, pages 25–32, 2003. (Citée en page 173.)

Chao Liu, Ryen W White et Susan Dumais : Understanding web browsing behaviors through weibull analysis of dwell time. *In Proceedings of the 33rd international ACM SIGIR conference*, pages 379–386. ACM, 2010. (Citée en page 173.)

Hector Llorens, Estela Saquete, Borja Navarro et Robert Gaizauskas : Time-Surfer : time-based graphical access to document content. *In ECIR'11 : Proceedings of the 33rd European conference on Advances in information retrieval*, pages 767–771, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-20160-8. (Citée en pages 8, 40, et 41.)

E. Loper et S. Bird : NLTK : The natural language toolkit. *In Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics, 2002. (Citée en page 61.)

Christophe Luc : Une typologie des énumérations basée sur les structures rhétoriques et architecturales du texte. *In TALN*, 2001. (Citée en page 21.)

Bernardo Magnini, Danilo Giampiccolo, Pamela Forner, Christelle Ayache, Valentin Jijkoun, Petya Osenova, Anselmo Peñas, Paulo Rocha, Bogdan Sacaleanu et Richard Sutcliffe : Overview of the CLEF 2006 multilingual question answering track. *In Evaluation of Multilingual and Multi-modal Information Retrieval*, pages 223–256. Springer, 2007. (Citée en page 33.)

Bernardo Magnini, Simone Romagnoli, Alessandro Vallin, Jesús Herrera, Anselmo Penas, Víctor Peinado, Felisa Verdejo et Maarten de Rijke : The multiple language question answering track at CLEF 2003. *In Comparative Evaluation of Multilingual Information Access Systems*, pages 471–486. Springer, 2004. (Citée en page 33.)

Bernardo Magnini, Alessandro Vallin, Christelle Ayache, Gregor Erbach, Anselmo Peñas, Maarten De Rijke, Paulo Rocha, Kiril Simov et Richard Sutcliffe : Overview of the CLEF 2004 multilingual question answering track. *In Multilingual Information Access for Text, Speech and Images*, pages 371–391. Springer, 2005. (Citée en page 33.)

Chris Mattmann et Jukka Zitting : *Tika in Action*. Manning Publications Co., 2011. (Citée en page 109.)

George A Miller et Walter G Charles : Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991. (Citée en page 132.)

Teruko Mitamura, Eric Nyberg, Hideki Shima, Tsuneaki Kato, Tatsunori Mori, Chin-Yew Lin, Ruihua Song, Chuan-Jie Lin, Tetsuya Sakai, Donghong Ji *et al.* : Overview of the NTCIR-7 aclia tasks : Advanced cross-lingual information access. *In Proceedings of the Seventh NTCIR Workshop Meeting*, pages 16–19. Citeseer, 2008. (Citée en page 32.)

Maarten de Rijke Monz Christof : Tequesta : The university of amsterdam's textual question answering system. *In TREC-10*, 2001. (Citée en page 38.)

Véronique Moriceau : *Intégration de données dans un système question-réponse sur le Web*. Thèse de doctorat, Université Paul Sabatier - Toulouse III, 2007. (Citée en page 38.)

Véronique Moriceau et Xavier Tannier : FIDJI : Using Syntax for Validating Answers in Multiple Documents. *Information Retrieval, Special Issue on Focused Information Retrieval*, 13(5):507–533, octobre 2010. (Citée en pages 31, 36, 86, 109, 111, 130, 156, et 174.)

Véronique Moriceau, Xavier Tannier et Mathieu Falco : Une étude des questions "complexes" en question-réponse. *In Actes de la Conférence Traitement Automatique des Langues Naturelles (TALN 2010, article court)*, Montréal, Canada, juillet 2010. (Citée en page 51.)

Guillaume Mouret : *Grammaire des constructions coordonnées*. Thèse de doctorat, Université Paris 7, 2007. (Citée en page 142.)

Vladimir Nikic, , Scott Wilson et Pat Moore : HTMLCleaner, analyseur HTML java. http://htmlcleaner.sourceforge.net, 2013. (Citée en page 89.)

Gabriele Paolacci, Jesse Chandler et Panagiotis G Ipeirotis : Running experiments on Amazon Mechanical Turk. *Judgment and Decision making*, 5(5):411–419, 2010. (Citée en page 173.)

Anselmo Peñas, Pamela Forner, Richard Sutcliffe, Álvaro Rodrigo, Corina Forăscu, Iñaki Alegria, Danilo Giampiccolo, Nicolas Moreau et Petya Osenova : Overview of ResPubliQA 2009 : Question answering evaluation over European legislation. *In Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 174–196. Springer, 2010. (Citée en pages 33 et 34.)

Joaquín Pérez-Iglesias, José R. Pérez-Agüera, Víctor Fresno et Yuval Z. Feinstein : Integrating the Probabilistic Models BM25/BM25F into Lucene. *CoRR*, abs/0911.5046, 2009. (Citée en page 110.)

D. Pinto, A. McCallum, X. Wei et W.B. Croft : Table extraction using conditional random fields. *In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 235–242. ACM, 2003. (Citée en page 29.)

Martin F Porter : Snowball : A language for stemming algorithms, 2001. (Citée en page 109.)

Marie-Paul Péry-Woodley : Une pragmatique à fleur de texte : approche en corpus de l'organisation textuelle, 2000. HDR. (Citée en page 21.)

X. Qi et B. D. Davison : Web Page Classification : Features and Algorithms. *In ACM Computing Surveys,41(2),February.*, 2009. (Citée en page 89.)

Silvia Quarteroni : *Advanced Techniques For Personalized, Interactive Question Answering*. Thèse de doctorat, University of York, 2007. (Citée en page 173.)

L. Quintard, O. Galibert, G. Adda, B. Grau, D. Laurent, V. Moriceau, S. Rosset, X. Tannier et A. Vilnat : Question Answering on web data : the QA evaluation in Quæro. *In Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta*, 2010. (Citée en pages 32, 36, 37, et 47.)

Ludovic Quintard : Overview of the Quaero 2008 monolingual question answering track, 2010. URL http://www.lne.eu/en/r_and_d/quaero.asp. (Citée en pages 32, 47, et 190.)

Majid Razmara : *Answering List and Other questions*. Thèse de doctorat, Concordia University, 2008. (Citée en page 38.)

Majid Razmara et Leila Kosseim : Answering List Questions using Co-occurrence and Clustering. *In LREC*. European Language Resources Association, 2008. (Citée en page 38.)

Marta Recasens, Lluís Màrquez, Emili Sapena, M Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio et Yannick Versley : Semeval-2010 task 1 : Coreference resolution in multiple languages. *In Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 1–8. Association for Computational Linguistics, 2010. (Citée en page 76.)

Stephen Robertson et Hugo Zaragoza : *The probabilistic relevance framework : BM25 and beyond*. Now Publishers Inc, 2009. (Citée en page 110.)

Charlotte Roze, Laurence Danlos et Philippe Muller : Lexconn : A french lexicon of discourse connectives. *Discours*, 10:15, 2012. (Citée en page 140.)

Benoît Sagot, Karën Fort, Gilles Adda, Joseph Mariani, Bernard Lang *et al.* : Un turc mécanique pour les ressources linguistiques : critique de la myriadisation du travail parcellisé. *In TALN'2011-Traitement Automatique des Langues Naturelles*, 2011. (Citée en page 182.)

Yutaka Sasaki, Hsin-Hsi Chen, Kuang-hua Chen et Chuan-Jie Lin : Overview of the NTCIR-5 cross-lingual question answering task (CLQA1). *In Proceedings of the Fifth NTCIR Workshop Meeting*, pages 6–9, 2005. (Citée en page 32.)

Yutaka Sasaki, Chuan-Jie Lin, Kuang-hua Chen et Hsin-Hsi Chen : Overview of the NTCIR-6 cross-Lingual question answering (CLQA) task. *In Proc. of NTCIR*, volume 6, 2007. (Citée en page 32.)

Nico SCHLAEFER, Jeongwoo KO, Justin BETTERIDGE, Guido SAUTTER, Manas PATHAK et Eric NYBERG : SEMANTIC EXTENSIONS OF THE EPHYRA QA SYSTEM FOR TREC 2007. *In TREC-16*, 2007. (Citée en pages 32, 37, 38, et 156.)

Chirag SHAH : Measuring effectiveness and user satisfaction in Yahoo! Answers. *First Monday*, 16(2), 2011. (Citée en page 172.)

Dan SHEN et Dietrich KLAKOW : Exploring correlation of dependency relation paths for answer extraction. *In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 889–896. Association for Computational Linguistics, 2006. (Citée en page 32.)

Vlad SKARZHEVSKYY, Andy TRIPP, Fabrizio GIUSTINA, Gary L PESKIN, Sami LEMPINEN, Russell GOLD et ADITSU : JTidy, portage java de HTML Tidy. `http://jtidy.sourceforge.net`, 2012. (Citée en page 89.)

Jannik STRÖTGEN et Michael GERTZ : Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013. (Citée en page 138.)

Renxu SUN, Jing JIANG, Yee Fan TAN, Hang CUI, Tat-Seng CHUA et Min-Yen KAN : Using Syntactic and Semantic Relation Analysis in Question Answering. *In TREC*, 2005. (Citée en page 156.)

Mihai SURDEANU, Massimiliano CIARAMITA et Hugo ZARAGOZA : Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2):351–383, 2011. (Citée en page 32.)

Keishi TAJIMA et Kaori OHNISHI : Browsing large HTML tables on small screens. *In UIST*, pages 259–268, 2008. (Citée en pages 28 et 29.)

Xavier TANNIER : WebAnnotator, an Annotation Tool for Web Pages. *In LREC 2012*, Istanbul, Turkey, mai 2012. (Citée en pages 178 et 180.)

Xavier TANNIER : Extracting News Web Page Creation Time with DCTFinder. *In Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014), Reykjavik, Iceland*, 2014. (Citée en pages 139 et 204.)

Charles TEISSÈDRE : *Analyse sémantique automatique des adverbiaux de localisation temporelle : application à la recherche d'information et à l'acquisition de connaissances*. Thèse de doctorat, Université Paris Ouest Nanterre La Défense, 2012. (Citée en pages 8, 40, et 41.)

Jordi TURMO, Pere COMAS, Christelle AYACHE, Djamel MOSTEFA, Sophie ROSSET et Lori LAMEL : Overview of QAST 2007. *In CLEF*, pages 249–256, 2007. (Citée en page 33.)

Jordi TURMO, Pere COMAS, Sophie ROSSET, Lori LAMEL, Nicolas MOREAU et Djamel MOSTEFA : Overview of QAST 2008. *In CLEF*, pages 314–324, 2008. (Citée en page 33.)

Jordi TURMO, Pere R COMAS, Sophie ROSSET, Olivier GALIBERT, Nicolas MOREAU, Djamel MOSTEFA, Paolo ROSSO et Davide BUSCALDI : Overview of QAST 2009. *In Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 197–211. Springer, 2010. (Citée en page 33.)

Alessandro VALLIN, Bernardo MAGNINI, Danilo GIAMPICCOLO, Lili AUNIMO, Christelle AYACHE, Petya OSENOVA, Anselmo PEÑAS, Maarten DE RIJKE, Bogdan SACALEANU, Diana SANTOS *et al.* : Overview of the CLEF 2005 multilingual question answering track. *In Accessing multilingual information repositories*, pages 307–331. Springer, 2006. (Citée en page 33.)

Ellen M. VOORHEES : The TREC-8 Question Answering Track Report. *In TREC-8*, 1999. (Citée en page 33.)

Ellen M. VOORHEES : The TREC-9 Question Answering Track Report. *In TREC-9*, 2000. (Citée en page 33.)

Ellen M. VOORHEES : Overview of the TREC 2001 Question Answering Track. *In TREC-10*, 2001. (Citée en pages 32 et 33.)

Ellen M. VOORHEES : Overview of the TREC 2002 Question Answering Track. *In TREC-11*, 2002. (Citée en page 33.)

Ellen M. VOORHEES : Overview of the TREC 2003 Question Answering Track. *In TREC-12*, 2003. (Citée en page 33.)

Ellen M. VOORHEES : Overview of the TREC 2004 Question Answering Track. *In TREC-13*, 2004. (Citée en pages 33 et 195.)

Ellen M. VOORHEES et Hoa Trang DANG : Overview of the TREC 2005 Question Answering Track. *In TREC-14*, 2005. (Citée en pages 33 et 195.)

Richard C. WANG, Nico SCHLAEFER, William W. COHEN et Eric NYBERG : Automatic Set Expansion for List Question Answering. *In EMNLP*, 2008. (Citée en page 38.)

Y. WANG et J. HU : A machine learning based approach for table detection on the web. *In Proceedings of the 11th international conference on World Wide Web*, pages 242–250. ACM, 2002. (Citée en pages 28, 29, 92, 96, 97, 98, et 100.)

Bonnie WEBBER, Claire GARDENT et Johan BOS : Position statement : Inference in Question Answering. *In In Proceedings of the LREC Workshop on Question Answering : Strategy and Resources, Las Palmas, Gran Canaria*, 2002. (Citée en page 38.)

Min WU, Xiaoyu ZHENG, Michelle DUAN, Ting LIU et Tomek STRZALKOWSKI : Questioning answering by pattern matching, web-proofing, semantic form proofing. *In TREC-12*, pages 578–585, 2003. (Citée en page 37.)

Hui YANG, Hang CUI, Mstislav MASLENNIKOV, Long QIU, Min yen KAN et Tat seng CHUA : Qualifier in TREC-12 qa main task. *In In Proceedings of the 12th Text REtrieval Conference (TREC-12), Gaithersburgh*, 2003. (Citée en pages 32 et 76.)

Xuchen YAO, Benjamin VAN DURME et Peter CLARK : Answer extraction as sequence tagging with tree edit distance. *In Proceedings of NAACL-HLT*, pages 858–867, 2013. (Citée en page 32.)

Y. ZHAI et B. LIU : Web data extraction based on partial tree alignment. *In Proceedings of the 14th international conference on World Wide Web*, pages 76–85. ACM, 2005. (Citée en page 29.)

Yuming ZHAO, ZhiMing XU, Yi GUAN et Peng LI : Insun05QA on QA Track of TREC 2005. *In TREC*, 2005. (Citée en page 76.)