Réordonnancement de candidats reponses pour un système de questions-réponses

Guillaume Bernard

NOTES et DOCUMENTS LIMSI N^{O} : 2011 - 06
Juin 2011

RÉORDONNANCEMENT D'HYPOTHÈSES DANS UN SYSTÈME DE QUESTIONS-RÉPONSES

Guillaume BERNARD

Thèse soutenue le 6 Juin 2011 devant le jury composé de :

Rapporteurs	Patrice BELLOT
	Kamel SMAILI
Directrice	Martine ADDA–DECKER
Co-Directrice	Sophie ROSSET
Président du Jury	Pierre ZWEIGENBAUM
Examinateurs	Frédéric BÉCHET
	Jeanne VILLANEAU

Notes et Documents LIMSI N^{O} : 2011 - 06 **Juin 2011**

Auteurs (Authors) : Guillaume Bernard

Titre : Réordonnancement d'hypothèses dans un système de questions-réponses.

Title : Re-ranking of hypotheses in a question-answering system.

Nombre de pages (Number of pages) : 225

Résumé : *L'objectif de cette thèse a été de proposer une approche robuste pour traiter le problème de la recherche de la réponse précise à une question.*

Notre première contribution a été la conception et la mise en oeuvre d'un modèle de représentation robuste de l'information et son implémentation. Son objectif est d'apporter aux phrases des documents et aux questions de l'information structurelle, composée de groupes de mots typés (segments typés) et de relations entre ces groupes. Ce modèle a été évalué sur différents corpus (écrits, oraux, web) et a donné de bons résultats, prouvant sa robustesse.

Notre seconde contribution a consisté en la conception d'une méthode de réordonnancement des candidats réponses retournés par un système de questions-réponses. Cette méthode a aussi été conçue pour des besoins de robustesse, et s'appuie sur notre première contribution. L'idée est de comparer une question et le passage d'où a été extraite une réponse candidate, et de calculer un score de similarité, en s'appuyant notamment sur une distance d'édition.

Le réordonnanceur a été évalué sur les données de différentes campagnes d'évaluation. Les résultats obtenus sont particulièrement positifs sur des questions longues et complexes. Ces résultats prouvent l'intérêt de notre méthode, notre approche étant particulièrement adaptée pour traiter les questions longues, et ce quel que soit le type de données. Le réordonnanceur a ainsi été évalué sur l'édition 2010 de la campagne d'évaluation Quaero, où les résultats sont positifs.

Mots clés : Question-Réponse, Oral, Réordonnancement, Domaine ouvert

Abstract : *The objective of this work is to introduce a new robust approach to treat the problem of finding the correct answer to a question.*

Our first contribution is the design and implementation of a robust representation model for information. The aim is to represent the structural information of sentences of documents and questions structural information. This representation is composed of typed groups of words (typed segments) and relations between these groups. This model has been evaluated on several corpus (written, oral, web) and achieved good resultats, which proves his robustness.

Our second contribution consisted is the design of a re-ranking method of a set of the candidate answers output by the question-answering system. This re-ranking method is based on the structural information representation. The general idea is to compare a question and a passage from where a candidate answer was extracted, and to compute a similarity score by using a modified edit distance we proposed.

Our re-ranking method has been evaluated on the data of several evaluation campaigns. The results are quite good on long and complex questions. These results show the interest of our method : our approach is quite adapted to treat long question, whatever the type of the data. The re-ranker has been officially evaluated on the 2010 edition of the Quaero evaluation campaign, with positives results.

Keywords : Question-Answering, Oral, Re-ranking, Open domain

Remerciements

Une thèse est un travail de longue durée : outre ma modeste personne, un nombre élevé de personnes a participé à mon travail, des fois indirectement. C'est la raison pour laquelle je m'excuse par avance des oublis potentiels dans ces remerciements.

Je remercie tout d'abord mes deux directeurs de thèse, Martine Adda-Decker et Sophie Rosset qui m'ont soutenu tout au long de ces quatre longues années. Je retiens particulièrement toutes les discussions de travail enrichissantes, ainsi que le soutien moral apporté lors des moments de doutes qui m'ont assailli.

Je remercie Patrice Bellot et Kamel Smaili d'avoir accepté d'être les rapporteurs de ma thèse, et pour les rapports détaillés soulevant des questions très pertinentes.

Je remercie aussi l'ensemble des examinateurs de mon jury : Frédéric Béchet, Jeanne Villaneau et Pierre Zweigenbaum. Les questions posées à la suite de ma présentation et les divers échanges ont été particulièrement intéressants et enrichissants.

Je remercie aussi Aurélien Max, qui m'a encadré dans mon stage de Master Recherche qui préfigurera mes différents travaux de thèse.

Je remercie spécialement Anne Vilnat, qui m'a donné le virus du TAL et des systèmes de questions-réponses à la suite d'un projet TER en Licence, et qui est donc une des origines principales de ma thèse.

Je tiens aussi à remercier l'ensemble du personnel du LIMSI, et plus particulièrement les membres du groupe TLP de m'avoir accueilli et intégré.

Enfin, je remercie ma famille et tous mes amis de m'avoir soutenu tout au long de ce long travail de thèse.

Table des matières

Introduction

Systèmes de Recherche d'Information

De nos jours, si un utilisateur a besoin de trouver une information générale, il se tournera tout naturellement vers des systèmes de recherche d'information. Ainsi, pour la recherche sur Internet, on trouve des systèmes tels que Google ou Yahoo. Les utilisateurs entrent une requête sous forme de mots-clefs, et le système retourne alors une liste de documents contenant ces mêmes mots, chaque documents pouvant potentiellement contenir les informations recherchées par l'utilisateur. De telles approches sont particulièrement adaptées lorsque l'on cherche à se renseigner sur un sujet donné. Au contraire, si l'utilisateur cherche une information précise, il devra alors parcourir l'ensemble des documents jusqu'à trouver la réponse à sa requête. Par ailleurs, les requêtes par mots-clefs ne sont pas la forme structurelle d'une question la plus naturelle pour un être humain. Par exemple, si un utilisateur cherche l'âge de Nelson Mandela lors de sa sortie de prison, une requête possible est *Nelson Mandela âge sortie prison*. Une liste de documents sera retournée, et l'utilisateur devra y trouver lui-même la réponse à sa question. La figure 1 montre une partie de la liste des documents retournée par Google pour la requête *Nelson Mandela âge sortie prison*.

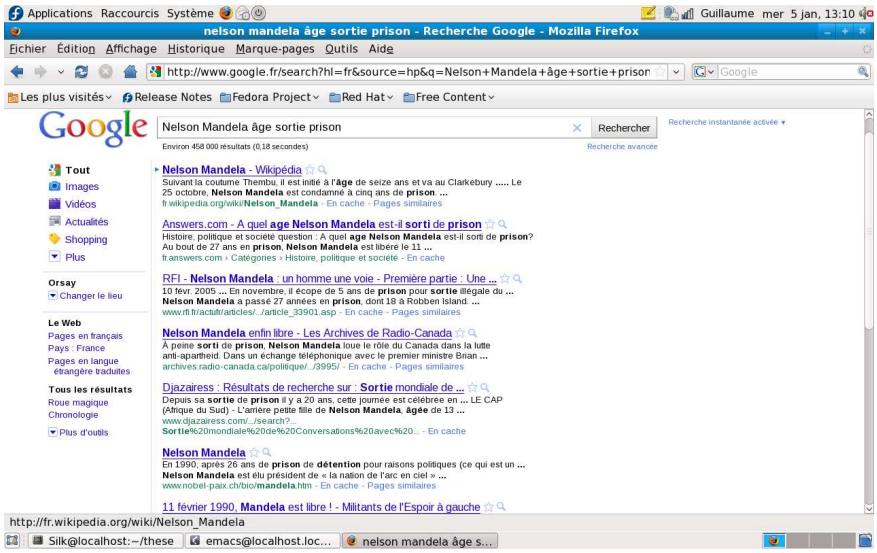

FIG. 1 – Documents renvoyés par Google à la requête *Nelson Mandela âge sortie prison*

Ces approches donnent de bons résultats. Cependant, elles rencontrent certaines limites, à partir du moment où l'utilisateur cherche une information plus précise. Ces systèmes ne retournent qu'un ensemble de documents, et l'expression des requêtes à partir de mots-clefs ne permet pas de définir précisément l'objet de sa recherche. L'utilisation la plus naturelle pour un utilisateur est de s'exprimer en langue, et non d'entrer une requête en mots-clefs. Une voie possible pour répondre à ces limites est celle des systèmes de questions-réponses.

Systèmes de Questions-Réponses

Les systèmes de questions-réponses (ou question-answering systems en anglais) cherchent à répondre à ces problématiques. Avec de tels systèmes, les utilisateurs peuvent poser leur question en langue, et une réponse précise leur sera retournée. Cette réponse peut être extraite par exemple d'un corpus de documents ou d'une base de données. Par exemple, pour trouver l'âge de Nelson Mandela à sa sortie de prison, l'utilisateur pose la question *"Quel âge avait Nelson Mandela à sa sortie de prison ?"*. Le système retourne alors une réponse. Cette réponse peut être construite sous la forme d'une phrase, comme *"Nelson Mandela avait 72 ans à sa sortie de prison."*. Néanmoins, les systèmes existants ont tendance à donner directement la réponse (*"72 ans"*). Par ailleurs, certains systèmes permettent même de gérer un dialogue entre l'utilisateur et la machine, lui permettant par exemple de poser des questions de précision par rapport à une réponse précédente. L'échange ci-dessous donne un exemple d'échange :

– Utilisateur : Quel âge avait Nelson Mandela à sa sortie de prison ?
– Système : 72 ans.
– Utilisateur : Quand est-il né ?
– Système : En 1918.
– ...

De tels systèmes donnent de bons résultats dans des domaines fermés [Lamel, et al. 2000 ; Walker, et al. 2002], comme la réservation de billets d'avions ou de trains, permettant même un dialogue relativement naturel entre l'homme et la machine. Cependant, le passage au domaine ouvert complexifie grandement la tâche, et le nombre et le type de problèmes rencontrés augmentent rapidement. Ainsi, les questions portent sur des sujets bien plus nombreux. Les structures des questions sont ainsi bien plus variées. De même, le système n'étant plus limité à un seul domaine, la forme et le type des réponses cherchées (âge d'une personne, prix d'un produit, définition d'un sujet ...) sont eux aussi plus variés. Par exemple, dans un système de réservation de billets de trains, les utilisateurs poseront des questions bien spécifiques : date de départ, prix des billets ... Par contre, en domaine ouvert, les questions portent sur n'importe quel sujet. Enfin, un système conçu pour la réservation de billets de train recherchera les réponses dans une base d'information structurée, comme une base de données. Les systèmes de questions-réponses en domaine ouvert traitent généralement des documents textuels, ce qui complique aussi l'extraction de la réponse.

Ainsi, une grande partie des systèmes de questions-réponses ont été conçus pour traiter dans un premier temps des questions factuelles, où une réponse courte est attendue, comme pour *"Quel âge avait Nelson Mandela à sa sortie de prison ?"*. De même, beaucoup de systèmes sont limités au traitement de textes issus de corpus journalistiques ou assimilés, et ne traitent donc pas de données issues de l'oral, ou même du web. Certains systèmes ont néanmoins été conçus ou adaptés à ce type de données dans le cadre de certaines campagnes d'évaluation : on peut citer QAst [Turmo, et al. 2008] pour l'oral, et la tâche questions-réponses de Quaero [Quintard, et al. 2010] pour le web.

Cadre du travail

Le projet Ritel [Rosset, et al. 2006], démarré en 2004 au LIMSI, a pour objectif de créer un système de dialogue, dont l'une des fonctionnalités est la recherche d'information en domaine ouvert. Comme on l'a vu précédemment, le domaine ouvert amène de nombreuses problématiques supplémentaires. Les auteurs ont choisi de s'orienter vers le domaine des systèmes de questions-réponses, et ont dû faire face à des contraintes inhabituelles à cause de l'environnement oral. En effet, la majorité des systèmes ont tendance à traiter des documents écrits, généralement extraits de sources journalistiques. Ces articles sont écrits dans une syntaxe que l'on peut qualifier de *"correcte"*. Par contre, les documents tirés de transcriptions de l'oral amènent un certain nombre de phénomènes généralement absents des articles journalistiques. On peut par exemple citer les répétitions bien plus nombreuses ou les hésitations inhérentes au dialogue. De même, les *"phrases"* d'une transcription sont en général bien plus longues, car il est difficile de détecter le début et la fin d'une phrase. Enfin, la reconnaissance automatique de la parole implique des erreurs supplémentaires. Le système se devait donc d'avoir une approche robuste. Par ailleurs, le contexte de dialogue impliquait aussi une gestion du temps de réponse : en effet, il n'est pas concevable de laisser l'utilisateur attendre trop longtemps sa réponse. Les auteurs ont donc conçu un moteur générique d'analyse de langue, s'appliquant aux documents et aux questions traitées par le système de questions-réponses. Ce moteur d'analyse fournit principalement de l'information sémantique, et offre de très bonnes performances en terme de robustesse et de vitesse [Galibert 2009]. Le système de questions-réponses s'appuie sur cette représentation des documents et questions, et permet de chercher et renvoyer une réponse précise à une question en un temps d'exécution rapide. Le système de questions-réponses utilise une approche qui exploite la répartition des éléments d'une question par rapport à un candidat réponse ainsi que la redondance de ce candidat dans l'ensemble des documents. Ce système donne de bons résultats [Turmo et al. 2008 ; Turmo, et al. 2009], comme le montrent ceux obtenus sur différents corpus de questions de campagnes d'évaluation, particulièrement dans un contexte oral.

Les systèmes de questions-réponses n'ont pas véritablement d'approche standard, mais on retrouve néanmoins une structure plus ou moins commune. Des documents prometteurs sont d'abord sélectionnés, avant d'en extraire des passages. Enfin, on extrait les réponses potentielles à une question de ces passages. La méthode utilisée par le système de questions-réponses pour extraire les réponses candidates a montré certaines limites. Les deux principales causes sont les ambiguïtés potentielles provoquées par l'utilisation de la redondance de l'information au sein de la base de documents, et le manque de représentation structurée au sein des questions et des phrases des documents. On peut notamment citer l'abscence de groupements de mots (groupe nominal, groupe verbal ...) et de relations entre ces groupes de mots.

Les réponses candidates retournées par Ritel sont ordonnées selon un score. L'objectif de cette thèse est de proposer une méthode pour réordonner ces réponses, en se basant notamment sur de nouvelles informations. Nous avons ainsi étudié la possibilité de rajouter une segmentation (*chunking*) typée des documents et questions en groupes de mots, en plus de l'analyse déjà existante de Ritel. En plus de cette segmentation, nous avons déterminé un formalisme simple de relations entre les segments. Cette

segmentation et ces relations permettent d'avoir une représentation simple de la structure. De plus cette représentation est adaptée aux différents types de documents traités (écrit et oral principalement, mais aussi web). En s'appuyant sur ce modèle de représentation, nous avons mis au point une méthode de réordonnancement des candidats réponses retournés par l'extracteur des réponses de Ritel. Cette méthode prend en compte les relations entre groupes de mots pour évaluer une réponse. Par ailleurs, si cette approche est conçue dans le cadre du système Ritel, l'idée est néanmoins de proposer une méthode généralisable à d'autres systèmes.

Problématique

L'objectif de ce travail était de proposer une méthode robuste de réordonnancement des réponses candidates à une question retournées par un système de questions-réponses. Nous voulons que cette méthode soit applicable à tous types de documents : écrits, oraux, web. Cette approche a comme cadre expérimental le système Ritel. Nous nous sommes d'abord intéressés à avoir un modèle de représentation des documents et des questions ajoutant de l'information structurelle par rapport à celui déjà existant fourni par l'analyseur de Ritel. Nous nous sommes ensuite appuyés sur cette représentation pour mettre au point une méthode de réordonnancement permettant de mieux traiter les cas ambigus pour l'approche actuelle de Ritel. Au delà de Ritel, notre ambition était de proposer une approche généralisable à d'autres systèmes de questions-réponses.

Principales contributions

Notre première contribution est un modèle de représentation robuste des documents et des questions pour un système de questions-réponses en domaine ouvert. Cette approche est basée sur un chunking typé simple. Des relations sont ensuite ajoutées entre les chunks pour représenter la structure des documents et des questions. L'idée principale est d'avoir un formalisme applicable quel que soit l'origine des documents ou questions en entrée : articles journalistiques, emissions de radio, pages web, etc. Ce formalisme reste volontairement simple dans un premier temps, de manière à fournir un terrain d'expérimentation pour notre deuxième contribution. L'un des objectifs est de complexifier ce formalisme en fonction des besoins.

Notre deuxième contribution exploitant ce modèle de représentation robuste a pour objectif d'améliorer les résultats d'un système de questions-réponses. L'idée est de procéder à un réordonnancement (re-ranking) des réponses potentielles retournées pour une question en appliquant une méthode basée sur le formalisme développé dans notre première contribution. Notre méthode de réordonnancement se base sur la structure fournie par ce formalisme de manière à traiter les cas ambigüs. L'idée est de prendre en compte les relations entre les chunks pour identifier la bonne réponse. Cette méthode de réordonnancement est évaluée dans le cadre du système de questions-réponses Ritel. Ce réordonnan-

ceur donne des résultats prometteurs sur des questions factuelles complexes comprenant un nombre important d'éléments.

Plan du document

Ce document est divisé en trois parties. La première partie présente le contexte de notre travail. Nous nous attachons tout d'abord à présenter dans le chapitre 1 le domaine des systèmes de questions-réponses, les problématiques impliquées, et notamment les approches principalement utilisées. Nous expliquons notamment les caractéristiques de ces approches, et leurs avantages et inconvénients par rapport à notre travail. Dans le chapitre 2, nous présentons le système de questions-réponses Ritel : quel est l'objectif de ce système, ses caractéristiques principales, mais aussi les limitations des méthodes utilisées pour répondre aux questions. Cette première partie a pour objectif de motiver les choix fait dans notre travail, et notamment d'expliquer les problématiques traitées. Nous procédons à la fin de cette partie à une discussion sur les avantages et inconvénients de chaque approche présentée dans le chapitre 1 par rapport à notre contexte de travail. Nous positionnons aussi le système Ritel par rapport à ces systèmes.

La seconde partie de ce document se concentre sur les apports de notre travail. Dans le chapitre 3, nous présentons un état de l'art s'appuyant sur les conclusions tirées dans la partie précedente. Cet état de l'art présente différentes approches de réordonnancement ou de sélection de réponses candidates. Nous détaillons par ailleurs les modèles de représentation des questions et des documents sur lesquels s'appuient ces méthodes. Les chapitres 4 et 5 détaillent nos deux contributions : un modèle de représentation robuste des questions et des documents, et une méthode de réordonnancement des réponses candidates. Nous nous appuyons sur les conclusions tirées dans le chapitre 3 et nous expliquons les approches utilisées pour nos deux contributions.

La troisième partie de ce document a pour objectif de détailler l'évaluation de notre approche. Dans le chapitre 6, nous présentons les différentes campagnes d'évaluation permettant d'évaluer nos méthodes. Le chapitre 7 présente les différentes évaluations effectuées et les résultats obtenus, et le chapitre 8 détaille l'analyse de ces résultats.

Enfin, nous discutons dans la dernière partie des conclusions amenées par ce document, et des perspectives ouvertes.

Première partie

Contexte du travail

Introduction

Nous nous intéressons dans cette partie aux différentes approches existantes pour trouver une information précise à une question. Ce domaine, dénommé *réponses aux questions*, ou de manière plus condensée *Question-Réponse* (QR), peut être défini par rapport à celui de la Recherche d'Information (RI). Les systèmes de recherche d'information proposent à l'utilisateur d'entrer une requête composée de mots-clefs pour trouver des informations à propos de ces mots-clefs. Ainsi, si l'utilisateur veut des informations sur une personnalité comme *Nelson Mandela*, il lui suffira d'entrer une requête composée de deux mots clefs, *Nelson* et *Mandela*. Le système de recherche d'information va alors retourner une liste de documents contenant ces mots-clefs, laissant l'utilisateur parcourir ces documents. Si ce dernier est à la recherche d'une information plus précise, comme l'âge de *Nelson Mandela*, il rajoutera un mot-clef à la requête, *âge*. Ce type de système est très efficace lorsque l'on cherche des informations sur un sujet général. Par contre, si l'on veut trouver des informations précises, ce type d'approche oblige l'utilisateur à parcourir les documents pour retrouver la réponse précise à sa requête. Trouver le nombre d'années que Nelson Mandela a passé en prison peut être exprimé sous la forme de la requête *Nelson Mandela années prison*. Une telle requête nécessitera de l'utilisateur une lecture des documents retournés.

Les systèmes de questions-réponses tentent de répondre à ces inconvénients en proposant des réponses précises aux questions (requêtes) de l'utilisateur. Les questions sont de plus formulées en langue et non en utilisant des mots-clefs. Nous avons expliqué dans l'introduction de ce document que l'on pouvait diviser les systèmes de questions-réponses en deux catégories : domaine fermé et domaine ouvert. Les systèmes de questions-réponses en domaine fermé permettent à l'utilisateur de poser des questions dans un contexte très spécifique, comme la réservation de billets de trains [Lamel et al. 2000]. Ce type de système donne de bons résultats : le contexte fermé permet de bien spécialiser le système par rapport aux types et formes de questions attendus (prix d'un ticket de train, horaires d'arrivées, etc). De plus, les réponses sont généralement stockées dans une base de données, rendant leur extraction relativement simple.

En domaine ouvert, les systèmes de questions-réponses ne traitent plus une base de données, mais une collection hétérogène de documents textuels. Ces documents doivent être analysés de manière à pouvoir extraire les informations qu'ils contiennent. Dans cette optique, ces documents sont généralement indexés, et l'information est ensuite extraite à partir d'un moteur de recherche. La taille du

corpus de documents ainsi que le type de documents du corpus a un fort impact sur le fonctionnement du système. La majorité des systèmes traitent des articles issus de corpus journalistiques. La structure et la syntaxe des phrases est classique permettant l'application d'analyses (syntaxiques par exemple) très complète, et facilitant ainsi la recherche des bonnes réponses. Cependant, il arrive fréquemment aux systèmes de devoir traiter des documents issus du web, voire de l'oral. De tels documents ont une structure différente des articles journalistiques, et amènent d'autres problématiques. Des approches développées spécifiquement pour des articles journalistiques ne donneront alors pas de bons résultats lors du traitement d'autres types de documents.

L'utilisateur peut poser des questions sur n'importe quel sujet, et n'est plus limité à une tâche précise. Le système de questions-réponses doit analyser la question de l'utilisateur, chercher des passages pertinents dans une base de documents, et retourner une ou plusieurs réponses satisfaisantes. Cette ouverture du domaine pose de nouveaux défis. Tout d'abord, les sujets des questions étant bien plus variés, l'analyse de la question effectuée par le système doit être bien plus générique. Il faut identifier à la fois le type de la question (factuelle, définition ...), et surtout le type ou la forme de la réponse recherchée : un âge, un lieu, ou une définition par exemple. Nous présentons ci-dessous différents exemples de questions traitées par des systèmes de questions-réponses.

– Question *factuelle* : *Quel est le nom du Président de la France ?*
– Question *oui/non* : *Est-ce-que l'Irlande a gagné le tournoi des VI Nations en 2010 ?*
– Question *définition* : *Qu'est ce que l'ONU ?*
– Question *pourquoi* : *Pourquoi De Gaule a-t-il démissionné ?*
– Question *comment* : *Comment construire un système de questions-réponses ?*
– Question *liste* : *Citer trois groupes de grunge.*

Ces questions ne peuvent pas être traitées de la même manière : elles font chacune appel à des techniques différentes pour trouver la bonne réponse. Il est à noter que ces différents types de questions se retrouvent aussi en domaine fermé.

Une autre difficulté dans les systèmes de questions-réponses vient du fait que les types de réponses à chercher sont très variables. Pour une question *factuelle*, les réponses à trouver sont en général des mots ou groupes de mots, comme dans l'exemple où une réponse correcte est *Nicolas Sarkozy*. Une réponse correcte pour les questions *oui/non* serait de valider ou invalider la question de l'utilisateur en proposant le passage justifiant la réponse. Mais pour des questions *définition*, il devient plus difficile d'évaluer ce qu'est la réponse la plus correcte : une définition simple de l'acronyme (*Organisation des Nations Unies*) ou une explication plus complète de l'ONU (création, rôle, etc ...) ? C'est le même problème avec les questions de type *pourquoi* ou *comment* : comment évaluer si la réponse apportée est suffisante ou non ? La majorité des questions posées dans les campagnes d'évaluation sont de type *factuelle*. Ce type de questions est généralement plus simple à traiter, et à évaluer dans le cadre de campagnes d'évaluation. De ce fait, la majorité des systèmes de questions-réponses ont choisi de traiter ce type de questions principalement.

Enfin, la ou les langues traitées est aussi une caractéristique très importante des systèmes. La majorité

des algorithmes et approches présentés dans ce document sont indépendants de la langue traitée par le système, si on laisse de côté les particularités de certaines langues. Cependant, ces approches ont tendance à s'appuyer sur des ressources linguistiques importantes [Baker, et al. 1998 ; Palmer, et al. 2005] selon les systèmes, ressources qui sont elles dépendantes de la langue. De ce fait, certaines méthodes ne sont tout simplement pas transposables d'une langue à une autre selon les ressources disponibles. C'est un problème classique lorsque l'on essaye de transposer une méthode adaptée pour l'anglais à une autre langue, les ressources étant plus complètes en anglais.

En gardant à l'esprit les différents problèmes soulevés précédemment, nous allons nous intéresser dans la partie suivante aux approches générales utilisées dans la conception des systèmes de questions-réponses. Dans le chapitre 1, nous présentons différents systèmes de questions-réponses, avec des caractéristiques différentes : systèmes utilisant des ressources linguistes, systèmes s'appuyant sur des méthodes syntaxiques, et systèmes intermédiaires. Nous discutons ensuite des avantages et des inconvénients de ces approches. Dans le chapitre 2, nous présentons le système de questions-réponses Ritel, qui sert de cadre expérimental à notre travail. Nous expliquons notamment comment ce système se place par rapport aux approches présentées dans le chapitre 1, ainsi que ses avantages et inconvénients.

Chapitre 1

Les systèmes de Questions-Réponses

1.1 Présentation générale des systèmes de questions-réponses

Il n'existe pas actuellement d'approche standard pour les systèmes de questions-réponses. De ce fait, les architectures des systèmes peuvent être assez variées. Néanmoins, on peut noter une structure commune, détaillée dans la figure 1.1. La première partie consiste au traitement des documents où seront cherchés les réponses aux questions. Les documents seront généralement prétraités : une analyse est effectuée de manière à extraire de l'information des documents. Cette étape peut ne pas exister. La complexité de l'analyse est variable selon les besoins du système. Par exemple, l'analyse peut être relativement simple, n'identifiant que les entités nommées ou les parties du discours, comme dans [Molla, et al. 2007 ; Comas & Turmo 2009]. Cette analyse peut aussi être beaucoup plus complexe, se basant alors sur des ressources complexes, et atteignant un niveau d'analyse syntaxique, voire sémantique très poussé [Laurent, et al. 2006 ; Hickl, et al. 2006c ; Tatu & Moldovan 2006]. Une fois analysés, les documents sont ensuite indexés. L'idée est de ne pas travailler sur des documents entiers : la tendance est donc à la segmentation de ces documents en phrases ou plus souvent en blocs de plusieurs lignes [Laurent et al. 2006 ; Reyes-Barragan, et al. 2009 ; Comas & Turmo 2009]. Lorsque l'on traite des documents où la notion de phrase n'est pas identifiée explicitement, comme dans le cas de la transcriptions de parole, il peut néanmoins être utile de construire là aussi des blocs équivalents, comme dans [Kürsten, et al. 2008 ; Reyes-Barragan et al. 2009 ; Comas & Turmo 2009].

Une fois l'indexation des documents faite, le système va pouvoir traiter les questions. L'analyse de ces questions va fournir deux informations : d'une part les éléments de la question devant être identifiés dans la base de documents, et d'autre part le type de la réponse attendu. Généralement, les éléments de la question recherchés dans le document sont des entités nommées, des mots-clefs, ou encore des expressions à mots multiples. L'analyse effectuée est assez proche, voir identique à celle faite sur les documents, de manière à faciliter la recherche d'information dans les documents. Le type de la réponse attendu dépend du type de la question traitée, ce type de réponse étant très difficile à déterminer

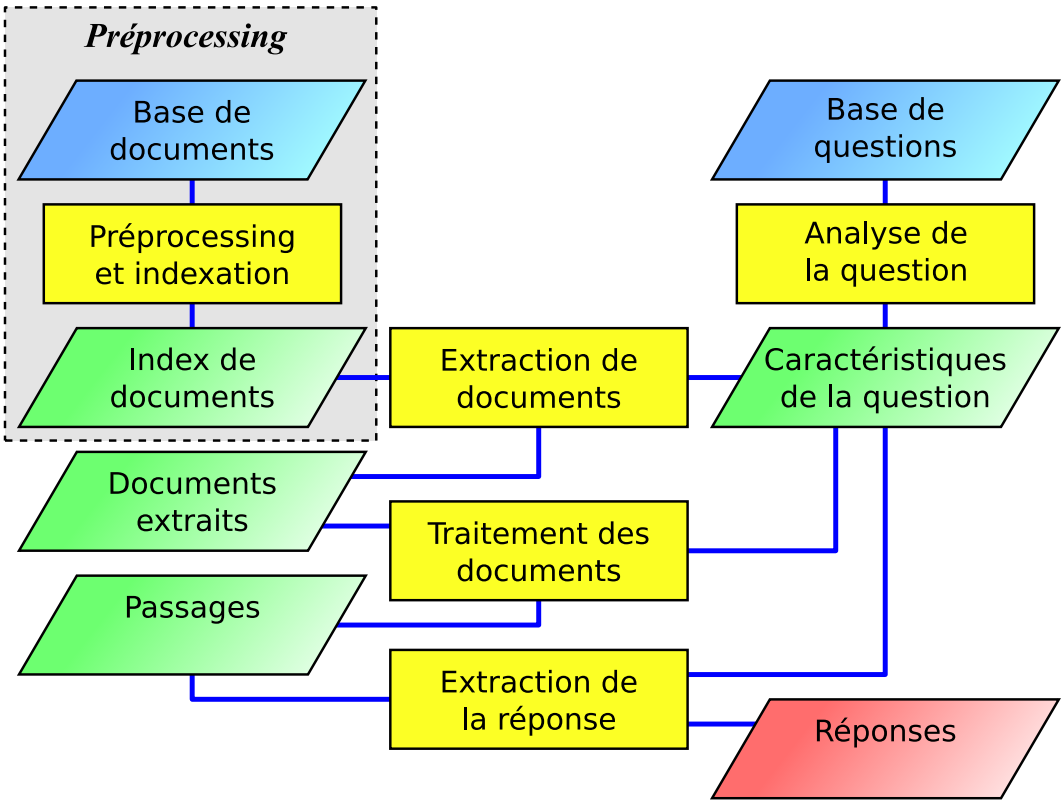

FIG. 1.1 – Structure générale des systèmes de questions-réponses (inspiré de [Ligozat 2006]).

dans le cas de certaines questions, notamment les questions pourquoi ou comment. Pour une question de type factuelle, il est relativement simple de déterminer le type de réponse attendu : en analysant le marqueur interrogatif, un système peut déterminer le type de réponse attendu. Pour la question *"Combien d'années Nelson Mandela a t-il passé en prison ?"*, l'analyse de *"Combien d'années"* permet de déterminer que l'on cherche une réponse numérique, chiffrée en années. Pour les questions factuelles, les types de réponse attendus sont généralement des entités nommées (personne, date, lieu, etc ...). Par contre, dans le cas d'une question *pourquoi*, comme *"Pourquoi De Gaule a t-il démissionné ?"*, il est beaucoup plus complexe de trouver, voir de *construire* une réponse convenable : doit-on donner la réponse la plus courte possible, ou au contraire donner un maximum d'informations ? Devant ces difficultés, la majorité des systèmes ont fait le choix de traiter dans un premier temps les questions factuelles.

A partir de l'analyse de la question, et des caractéristiques récupérées (éléments à rechercher et type de réponse attendu), le système va utiliser un moteur de recherche pour interroger la base des documents indexés et extraire les documents ou passages pouvant potentiellement contenir la réponse à la question. Pour la conception du moteur, il existe deux tendances : soit les concepteurs utilisent un moteur déjà existant, soit ils définissent et mettent au point leur propre moteur de re-

cherche. Le moteur de recherche Lucene [Apache 2007] est par exemple beaucoup utilisé dans le domaine [Comas & Turmo 2009 ; Tannier & Moriceau 2010 ; Quarteroni & Moschitti 2010]. Les systèmes utilisant leur propre moteur de recherche le font généralement pour pouvoir se servir de leurs analyses linguistiques pour la recherche des documents ou passages. On peut notamment citer [Rosset et al. 2006 ; Galibert 2009], ou encore [Laurent et al. 2006], qui s'appuie sur des analyses profondes. A partir des informations récupérées lors de l'analyse de la question, le moteur de recherche va interroger l'index des documents. Plusieurs étapes distinctes peuvent alors avoir lieu, généralement la sélection des documents candidats, puis des passages contenant potentiellement la réponse, et enfin l'extraction de la réponse ou de plusieurs candidats réponses.

Les réponses retournées pour une question ont un score qui leur est associé, permettant de donner un rang à la réponse. Les candidats réponses sont généralement les mots ou les groupes de mots dont le type correspond à celui identifié lors de l'analyse de la question. La méthode utilisée pour donner un score est très variable selon les systèmes : il peut être basé sur la distance entre la réponse et les mots clefs [Pardiño, et al. 2008] ou sur une mesure générale de densité [Gillard, et al. 2006 ; Comas & Turmo 2009]. Le score peut aussi tenir compte des similarités syntaxiques entre la question et le passage contenant la réponse [Tannier & Moriceau 2010]. Ainsi, la méthode utilisée pour calculer le score peut être très variable, et dépend bien entendu du contexte de travail : certaines approches n'auront pas la même efficacité selon la langue, le type de document traité, les ressources linguistiques à disposition, etc ... Par exemple, le système Chaucer du LCC [Hickl, et al. 2007] donne de très bons résultats sur l'anglais. Cependant, les approches utilisées dans ce système sont difficilement reproductibles dans d'autre langues, car Chaucer fait appel à des ressources linguistiques de grande envergure disponibles uniquement pour l'anglais pour le moment : FrameNet [Ruppenhofer, et al. 2006], Extended WordNet [Harabagiu, et al. 1999].

L'architecture des systèmes de questions-réponses présentée reste très générale. Comme on l'a vu précédemment, selon les approches, certains modules peuvent disparaître. Il est d'ailleurs intéressant de noter que de plus en plus de systèmes, comme [Moschitti & Quarteroni 2010] rajoutent un module de validation des réponses sélectionnées. L'idée est en effet d'appliquer les traitements les plus complexes en fin de chaîne de traitement. On parle souvent de réordonnancement des réponses candidates (ou *re-ranking*). Ce module intervient généralement après l'étape d'extraction de la réponse. On peut aussi citer [Grappy & Grau 2010] qui utilisent une approche basée sur la vérification du type des réponses candidates pour effectuer un réordonnancement.

L'architecture présentée est surtout pertinente dans le cas de systèmes s'appuyant sur des approches utilisant de l'information linguistique. Des méthodes alternatives existent bien évidemment. Ainsi, on peut citer certains systèmes s'appuyant considérablement sur des approches statistiques, et n'utilisant que peu d'information linguistique. On peut notamment citer [Berger, et al. 2000] et [Ittycheriah & Roukos 2002]. [Whittaker, et al. 2007] pousse le concept encore plus loin en éliminant toute connaissance linguistique. Enfin, certains systèmes choisissent une conception plus intermédiaire [Tannier & Moriceau 2010 ; Ligozat 2005], utilisant des connaissances linguistiques, mais en quantité limitée selon leur disponibilité.

1.2 CHAUCER, un système linguistique

Le système de la *Language Computer Corporation* [Hickl et al. 2006c ; Hickl et al. 2007] a obtenu d'excellents résultats sur les différentes évaluations auxquelles il a participé. Ce système fait appel à de très grandes ressources linguistiques.

CHAUCER traite d'abord la base de documents desquels sont extraits les réponses. Les documents sont traités par le parser Collins [Collins 2009] : ce parser statistique génère des arbres de dépendances syntaxiques. Trois parsers ajoutent ensuite des dépendances sémantiques à partir des ressources linguistiques suivantes : PropBank [Palmer et al. 2005], NomBank [Meyers, et al. 2004] et FrameNet [Ruppenhofer et al. 2006]. PropBank est une ressource linguistique dont l'objectif est d'indiquer les arguments associés à des prédicats verbaux. NomBank a le même objectif, mais cette fois consacrée aux noms. Enfin, FrameNet contient un ensemble de frames sémantiques. Les créateurs du système classent aussi les entités nommées se trouvant dans le corpus de documents en utilisant leur annotateur CICEROLITE. Enfin, une normalisation est effectuée sur les expressions temporelles des documents pour les convertir dans un format standard (ISO 8601). Les documents sont enfin indexés en utilisant Lucene [Apache 2007].

CHAUCER génère aussi une base de données composée de faits décrits dans les documents et des sources du web. Cette base de données est créée en utilisant l'extracteur d'information CICEROCUSTOM interne au LCC appliqué sur la collection de documents. Cette base de donnée est complétée par le biais d'heuristiques d'extraction d'information à partir de certains sites du web, comme *imdb* ou *wikipedia*. Chaque fait ajouté à la base de données est traité par un module de validation du contenu. Ce module utilise les sorties de systèmes d'inférence textuelle pour déterminer si un fait ajouté valide ou invalide les connaissances stockées dans la base de données. Par ailleurs, un ensemble de faits de la base de données est fourni à un module de générations de questions. L'objectif est de relier les informations de la base de faits à une question, et ainsi générer des paires questions/réponses. Ces paires sont ensuite utilisées par différents modules du système de questions-réponses.

Une fois les documents indexés, le système va traiter les questions. Le système commence par analyser chaque question. Les questions sont tout d'abord annotées par des informations lexicales et sémantiques : parties du discours, chunks, entités nommées. Une résolution des anaphores est effectuée. L'analyse de la question est utilisée pour la génération de la requête qui permettra de chercher les réponses dans les documents. Par ailleurs, les mots-clefs sont traités de manière à générer des formes proches pour améliorer la recherche (synonymes, forme lemmatisée du mot, etc ...). La question est ensuite traitée par un module de détection du type de la réponse. Ce module utilise un classifieur s'appuyant sur l'Entropie Maximale [Hickl et al. 2007]. Chaque question est associée à un type de réponse (environ 275 types sont utilisés). Le classifieur s'appuie sur trois éléments principaux : les marqueurs interrogatifs, les prédicats verbaux, et le mot permettant d'inférer le type attendu de réponse. Si l'on prend la question *"Combien d'années Nelson Mandela a t-il passé en prison ?"*, le marqueur interrogatif est *Combien*, qui indique que le type attendu de la réponse est numérique. Le prédicat verbal est *passé*, et le mot inférant le type attendu de la réponse *années*, qui est donc la spécification du nombre.

La forme analysée de la question ainsi que le type de la réponse cherchée sont traitées par le moteur de recherche qui va extraire les documents de l'index. Les passages sont extraits en se basant sur la densité de mots-clefs trouvés dans les documents et la distribution des types des entités correspondant à celle attendue pour la réponse. L'extraction des réponses candidates à une question va être traitée selon trois stratégies différentes : utilisation de la base de données créée précédemment, des patrons d'extraction, et enfin une stratégie basée sur la comparaison entre le type des entités et celui attendu pour la réponse.

Enfin, une fois les réponses candidates extraites, ces dernières vont être traitées par le module de validation des réponses. Ce module va réordonner les réponses en leur attribuant un score calculé par un système d'implication textuelle [Hick & Bensley 2007]. L'objectif de l'implication textuelle, tel qu'il est défini par les campagnes RTE (Recognizing Textual Entailment) [Bentivogli, et al. 2009], est de déterminer si le sens d'une *hypothèse* infère le sens d'un *texte*. Ces systèmes peuvent être utilisés pour identifier les relations d'implication entre des questions et des réponses [Harabagiu & Hickl 2006]. Dans l'exemple ci-dessous, le sens de l'hypothèse *H* infère le sens du texte *T*.

T : "Henri IV a été assasiné par Ravaillac."
H : "Henri IV est mort en 1610."

Dans le cas de *CHAUCER*, les auteurs utilisent le système *Groundhog* [Hickl, et al. 2006b] pour identifier les relations d'implication entre une question et un candidat réponse. Ce système extrait diverses informations linguistiques des paires hypothèse-texte, et utilise ces informations comme traits pour un classifieur SVM. *Groundhog* utilise notamment les données extraites d'un large corpus de paraphrases, des traits sémantiques, et des mesures d'alignement entre l'hypothèse et le texte. Par ailleurs, *Groundhog* s'appuie aussi sur les paires questions/réponses générées précédemment. Le classifieur est entraîné sur un très grand corpus de 200 000 paires hypothèse-texte générées automatiquement. Pour utiliser *Groundhog* dans le cadre du système de questions-réponses *CHAUCER*, les auteurs appliquent la méthode présentée dans [Harabagiu & Hickl 2006] : les candidats réponses qui ne sont pas impliqués par la question sont éliminés, puis le candidat réponse avec le meilleur score d'implication est choisi comme bonne réponse.

Le système a obtenu d'excellents résultats sur les campagnes d'évaluations auxquelles il a participé : il a trouvé une réponse correcte pour 71% des questions factuelles à TREC 2005, 58% à TREC 2006, et 56% à TREC 2007. Ce système était classé premier en 2005 et 2006 et second pour 2007. La tâche questions-réponses de TREC s'est arrêté en 2007.

1.3 Un système fortement statistique : le système des ATR Spoken Language Communication Research Laboratories

Le système proposé par [Sasaki 2005] est un exemple de système très statistique s'appuyant sur très peu d'informations linguistiques. L'idée des auteurs est de réussir à mettre en place une approche

non dépendante de la détection du type des questions. Leur choix est motivé par le fait que de telles approches obligent souvent la mise en place d'outils pour extraire les entités nommées, les expressions numériques ou les noms des classes. Cela rend problématique la mise en place de systèmes multi-lingues, car ces outils doivent être définis pour chaque langue. Si certains systèmes utilisent des approches par apprentissage, elles restent néanmoins limitées par le besoin d'annoter un corpus de paires de questions et de réponses pour déterminer le type de chaque question. De ce fait, si le typage doit être modifié, par exemple en introduisant de nouveaux types, l'annotation devra être recommencée. Le système décrit dans cette section a pour objectif de pallier ses problématiques en utilisant une approche par apprentissage non dépendante d'un typage des questions.

Ainsi, l'architecture du système s'éloigne de la structure classique d'un système de questions-réponses présentée dans la section 1.1. On peut diviser l'architecture en plusieurs modules : analyse de la question, sélection des documents, sélection des passages ... Le système proposé par l'ATR Spoken Language Communication Resarch Laboratories [Sasaki 2005] utilise une approche qui peut être divisée en deux composantes : la sélection des documents contenant potentiellement une bonne réponse, et l'application d'un modèle prenant en entrée certaines caractéristiques des documents et des questions, et retournant les meilleures réponses.

Le fonctionnement général est le suivant. Pour une question donnée, le système va sélectionner un certain nombre de documents à partir du corpus de documents. La sélection est relativement simple et utilise une méthode basée sur une mesure *idf* (inverse document frequency) : les mots de la question sont utilisés par la mesure pour sélectionner les documents. Les traits de chaque document ainsi que les traits de la question sont alors fournis au modèle défini *Question-Biased Term Extraction* (QBTE), créé par l'auteur.

QBTE utilise des modèles d'entropie maximale (Maximal Entropy Models - MEM). Pour chaque mot w_i d'un document $d = w_1, w_2, ..., w_m$, le classifieur va déterminer si le mot w_i fait partie de la réponse ou non en fonction des traits x^i qui lui sont associés. Plus précisément, trois labels sont utilisés pour classifier les mots : I si le mot est contenu dans le groupe de mots de la réponse, O s'il est à l'extérieur, et B si c'est le premier mot de la séquence réponse. Ce formalisme est basé sur IOB2 [Sang 2000]. Les données d'entrée x^i de chaque mot correspondent aux traits identifiés à partir des documents et des questions.

Avant d'extraire les données d'entrée x^i associées à un mot w_i, un certain nombre de traits sont identitifés dans la question traitée. Ces traits correspondent aux 4-grams présents, aux mots interrogatifs, et aux quatre types Parties du Discours de chaque mot de la question. Ces quatre types sont générés par l'analyseur morphologique japonais ChaSen. Par exemple, dans la question *Où est Tokyo ?*, les 4-grams sont *Où, est, Tokyo, Où-est, est-Tokyo* et *Où-est-Tokyo*. Les 4 types de Parties du Discours pour *Tokyo* seront *nom, nom propre, lieu* et *général*. Ces traits ne font pas directement partie des données d'entrée x^i du mot w_i.

Pour les documents sont extraits les *k* mots autour du mot w_i ainsi que leur 4 types de Partie du Discours. k est fixé à 3, ce qui donne une fenêtre avec une taille de 7. Dans la phrase suivante, *Tokyo*

est la capitale administrative du Japon depuis 1868, pour le mot *capitale*, les k-mots voisins seront *Tokyo, est, la, administrative, du* et *Japon*. L'ensemble de ces traits font partie des données d'entrée x^i associées au mot w_i

Enfin six traits supplémentaires sont générés en combinant ceux récupérés sur la question et le document. Le système regarde le nombre de mots identiques entre la question et les k-mots voisins du mot w_i, ainsi que les similarités entre type Parties du Discours. Pour la question *Où est Tokyo* et la phrase *Tokyo est la capitale administrative du Japon depuis 1868*, si on prend le mot *capitale*, alors *Tokyo* et *est* sont des mots que l'on retrouve dans la question. Le dernier trait correspond à l'ensemble des combinaisons possibles entre les mots de la question et les mots voisins de w_i (par exemple *Où & administrative*. Ces traits sont ajoutés aux données d'entrée x^i associées au mot w_i

A partir de ces traits, le modèle QBTE est généré à partir de données d'apprentissage. Le corpus utilisé est le CRL QA Data [Sekine, et al. 2002], qui est un corpus japonais de 2000 questions factuelles. Le corpus de documents est tiré d'articles journalistiques japonais. Le système est évalué selon deux mesures : le MRR (Mean Reciprocal Rank), qui correspond au rang moyen de la bonne réponse, et le top-5, qui est le pourcentage de questions avec une bonne réponse dans les 5 premiers candidats. Le système obtient un MRR de 0.36, et un top-5 de 0.47. Ces résultats peuvent être comparés à ceux obtenus par [Sasaki, et al. 2004], qui obtient un MRR de 0.4 et un top-5 de 0.55. Le corpus de questions n'étant pas disponible publiquement, il n'est pas possible de comparer directement leurs résultats sur ce corpus à d'autres systèmes. Néanmoins, l'auteur estime que les performances sont proches de celles obtenues par les approches plus conventionnelles. Par ailleurs, l'approche semble adaptable à n'importe quelle langue, étant donné qu'elle ne repose que sur un analyseur en Parties du Discours et un corpus d'apprentissage.

1.4 QALC, un système intermédiaire de LIMSI-ILES

Le système de questions-réponses QALC [Berthelin, et al. 2003], développé pour l'anglais (un système équivalent existe pour le français [Grau, et al. 2005]), est un exemple de ce que l'on pourrait appeler un système intermédiaire. Par intermédiaire, nous entendons les systèmes de questions-réponses faisant appel à des ressources ou connaissances linguistiques, mais de manière limitée. La majorité des systèmes utilisés dans les évaluations font partie de cette catégorie.

Un prétraitement est appliqué sur les documents : ils sont découpés en paragraphes, puis lemmatisés. Ces paragraphes seront ensuite indexés dans le moteur de recherche MG [Belle, et al. 1994].

Le traitement des questions est relativement complexe. L'analyse appliquée a pour objectif d'identifier un certain nombre de caractéristiques contenues dans les questions. Pour ce faire, un étiquetage morpho-syntaxique est effectué par le biais de TreeTagger [1] : le lemme et la catégorie lexicale de

[1] http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger

chaque mot sont ainsi déterminés. L'analyseur Cass [2] est ensuite appliqué pour segmenter la question en constituants. Une fois ces analyses effectuées, les caractéristiques de la question vont être déduites. Quatre types de caractéristiques sont recherchés. Le système identifie en premier lieu les mots de la question jugés comme pertinents pour la recherche de la réponse. Le deuxième type de caractéristique concerne la catégorie de la question, i.e. si la question est de type *factuelle* ou *définition*. La troisième catégorie correspond au type de réponse attendu, sous la forme d'un type général (animal, monnaie ...) ou d'un type d'entité nommée : personne, lieu, date ... Enfin, la dernière catégorie correspond au focus de la question. Cette notion relativement intuitive est définie comme étant l'objet sur lequel porte la question, comme une personne par exemple (*Henri IV* dans la question *"Qui a tué Henri IV ?"*).

A partir de ces caractéristiques, des requêtes vont être construites puis soumises au moteur de recherche pour obtenir les paragraphes des documents pertinents. Des relachements de contraintes (suppression de certains mots-clefs) ainsi que l'utilisation de certains synonymes ont lieu jusqu'à obtenir une centaine de paragraphes. Ces paragraphes vont être ensuite analysés. L'idée va être de chercher les mots clefs de la question présents dans les paragraphes. Du fait des différences linguistiques pouvant exister entre la question et les phrases des paragraphes, Fastr [Jacquemin 2004] est utilisé pour générer des variations des phrases des paragraphes. Ces dernières sont alors comparées avec les éléments importants et le focus de la question. Un score est calculé à partir de ces comparaisons, et ceux considérés comme étant les plus pertinents sont conservés. Les documents sont ensuite découpés en phrases, et seules celles contenant au moins un mot de la question ou une variante sont conservées.

Ces phrases candidates vont être ensuite classées en fonction d'une mesure de similarité établie entre la question et chaque phrase candidate. Deux mesures de similarité sont utilisées dans QALC [Ligozat 2005], une linéaire, et la deuxième dite syntaxique. La mesure linéaire est calculée en fonction du poids de chaque mot important présent dans la phrase, et de la proximité entre les termes. Les poids sont fixés en fonction de la présence du terme dans le corpus : plus le mot est fréquent, plus son poids sera important. Cette mesure étant très dépendante de l'ordre des mots, une mesure syntaxique a été mise en place pour pallier ce problème. Les phrases candidates sont transformées en arbre syntaxique à partir de l'analyseur de Charniak [Charniak 2000]. Un algorithme est ensuite appliqué pour remonter les têtes de chaque constituant. Par exemple pour un groupe nominal anglais, le dernier nom est choisi comme tête, et le reste des mots va dépendre de ce nom. Enfin, un sous-arbre est construit qui ne contient que les nœuds des mots de la question et ceux correspondant au type d'entité nommée attendu. La figure 1.2 donne un exemple de construction d'un sous-arbre. La question traitée est *"Who is Tom Cruise's wife"*, et la phrase candidate *"Tom Cruise met his wife Nicole Kidman on a movie set"*. Enfin, le type d'entité nommé attendu pour la réponse est une personne. La partie gauche de la figure représente l'arbre des dépendances syntaxiques construit à partir de la sortie de l'analyseur de Charniak, après détection des têtes des constituants. Le sous-arbre construit contient les mots correspondants de la question (*Tom*, *Cruise* et *wife*), ainsi que les mots du type d'entité nommée attendu (*Nicole* et *Kidman*).

[2] http ://www.sfs.nphil.uni-tuebingen.de/ãbney

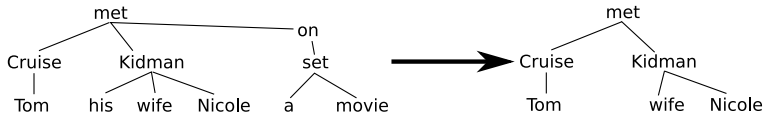

FIG. 1.2 – Elagage de l'arbre syntaxique de la phrase *"Tom Cruise met his wife Nicole Kidman on a movie set"* pour la question *"Who is Tom Cruise's wife ?"*.

La mesure syntaxique est basée sur deux composantes calculées à partir de ces sous-arbres : la composante syntagmatique et la composante paradigmatique. La première est basée sur la longueur du sous-arbre (nombre total de nœuds). La composante paradigmatique est la somme des poids des nœuds correspondant aux critères attendus. Ces poids dépendent des scores attribués par Fastr pour les nœuds correspondant à des mots ou des expressions de la question. Pour les nœuds correspondants à une entité nommée, le poids est maximum si l'entité nommée est du type attendu, et moins fort si c'est une généralisation du type attendu (type nombre alors qu'on attend un pourcentage).

Enfin, quelle que soit la mesure utilisée pour classer les phrases candidates, la méthode d'extraction des réponses est identique. Selon le type de réponse attendu, deux stratégies sont utilisées. Si le type est de catégorie *entité nommée*, les positions dans la phrase des éléments de la question sont relevés. Leur barycentre est calculé, et l'entité nommée la plus proche est sélectionnée comme réponse. Pour le cas des types *généraux* (monnaie, animal ...), un ensemble de patrons d'extraction est appliqué. Les patrons s'appuient à la fois sur les mots de la phrase et sur une annotation en parties du discours. Un score final est calculé pour trier les candidats réponses à partir de différents éléments : le score donné au paragraphe par le moteur d'indexation, la présence ou non des éléments de la question et enfin leurs scores Fastr associés ainsi que des poids attribués aux différents patrons.

QALC n'a pas participé à TREC 2005 ou 2006, ce qui empêche une comparaison avec CHAUCER. Cependant, le système obtient à TREC 2002 un CWS de 0.497, le classant neuvième, le premier obtenant un CWS de 0.856. Le système a aussi participé à CLEF 2005, mais sans utiliser la mesure syntaxique de scoring des phrases candidates. Le système obtient une précision de 28%. Il n'y avait pas d'autre système monolingue pour l'anglais sur cette évaluation. Sur les autres langues, les résultats vont de 23% à 64%.

1.5 Discussion

Dans ce chapitre, nous avons d'abord présenté une architecture classique pour un système de questions-réponses : analyse de la question, sélection des documents, sélection des passages ... Nous avons pu voir ensuite un ensemble de méthodes différentes, allant des approches utilisant de grandes ressources linguistes, à des approches entièrement statistiques. Les systèmes résultants ont chacun leurs avantages et inconvénients. Ainsi, les approches linguistiques s'appuient sur des bases de connaissances nombreuses et complexes, qui apportent ainsi beaucoup d'informations. Le problème est que ces

ressources ne sont pas toujours disponibles dans d'autres langues que l'anglais. Les approches statistiques sont relativement indépendantes de la langue, mais nécessitent un corpus d'apprentissage conséquent. Le choix de la méthode dépend souvent des ressources disponibles dans la langue traitée. Les approches intermédiaires, qui constituent la majorité des systèmes de questions-réponses, utilisent aussi des ressources et des analyses linguistiques, mais en couverture plus limitée.

L'anglais possède bien plus de ressources linguistiques que toutes les autres langues. Ainsi, une approche basée fortement sur des ressources ne sera pas adaptable à une autre langue non dotée des mêmes ressources. De même, les types de documents traités peuvent avoir un impact fort selon la méthode utilisée. Ainsi, beaucoup d'approches conviennent bien sur des documents rédigés dans une syntaxe "classique", tels que des articles de journaux. En revanche, traiter des transcriptions orales peut entrainer un certain nombre de problèmes. En effet, l'oral implique que les documents ont une syntaxe très différente de l'écrit. De ce fait de telles approches peuvent se révéler peu robustes par rapport au type des documents traités.

Le système du LCC donne de très bons résultats, d'après les évaluations TREC 2005, 2006 et 2007. Le système se base sur des ressources linguistiques peu disponibles dans d'autres langues que l'anglais. Des travaux sont en cours pour obtenir un WordNet français. Si EuroWordNet [Vossen 1998] est considéré comme insuffisant, on peut citer Wolf [Sagot, et al. 2009] qui tend à obtenir la même couverture que WordNet. Par contre, des ressources comme ProbBank ou NomBank n'ont actuellement pas d'équivalent en français. Par ailleurs, les analyses mises en œuvre dans ce système utilisent des traitements très poussés qui risquent de ne pas se révéler robustes sur des documents autres que du texte bien écrit.

A l'autre extrême, le système des ATR Spoken Language Communication Research Laboratories ne s'appuie sur aucune ressource ou analyse linguistiques, à part un annotateur en Parties du Discours. L'objectif du système est d'être indépendant d'une analyse du type de la question, quel que soit la langue. L'architecture traditionnelle des systèmes de questions-réponses est réduite à deux modules : la sélection des documents, et l'extraction de la réponse. L'extraction de la réponse est assimilée à un problème de classification. Cette approche permet d'être robuste en théorie quel que soit le type de question, voire de documents (aspect non évalué par l'auteur, le corpus de test étant tiré de documents journalistiques). Par contre, cette approche nécessite un corpus d'apprentissage conséquent pour obtenir des résultats convenables (voir les résultats obtenus par [Whittaker et al. 2007] sur un corpus de petite taille). De plus, il semble difficile d'améliorer les performances sur le moyen terme sans réintroduire de ressources linguistiques.

Nous avons aussi choisi de présenter le système QALC du LIMSI, qui se situe au milieu des deux approches. Il utilise des ressources et analyses linguistiques, mais en quantité bien moins importante que CHAUCER. Il se rapproche en ce sens de la majorité des systèmes existants. Néanmoins, le système fait appel à l'analyseur Charniak, qui a été conçu pour traiter avant tout des documents journalistiques. Le système a été conçu pour traiter principalement des documents écrits en langue naturelle, et n'a donc pas été évalué sur des transcriptions de l'oral, ou des documents issus du web.

Face au panel de systèmes de questions-réponses présentés, nous inscrivons nos travaux dans un système intermédiaire original, Ritel (voir chapitre 2), ayant les objectifs suivants :

– traiter des documents de sources hétérogènes : écrit classique, mais aussi des documents oraux (transcriptions manuelles et automatiques) et des documents du web ;
– la question de l'utilisateur peut être formulée à l'oral et de manière itérative en dialoguant avec le système.

Ces objectifs ont conditionné les approches utilisées, en partie pour les points suivants : pas d'utilisation d'analyse syntaxique, traitement du français, et pas de ressources linguistiques complexes. Nous présentons Ritel dans le chapitre 2. Nous y expliquons les choix effectués dans la conception du système de questions-réponses. Nous détaillons les traitements mis en œuvre pour choisir les réponses candidates à une question. Nous détaillons ensuite les différents problèmes rencontrés par le système, et ses limites actuelles.

Chapitre 2

Ritel : un système de questions-réponses oral en domaine ouvert

Ritel [Rosset et al. 2006] est un système de questions-réponses oral et interactif en domaine ouvert. De ce fait, un certain nombre de contraintes ont dû être prises en compte lors de l'élaboration d'un tel système. En effet, la plupart des systèmes traditionnels traitent des questions et documents dans une syntaxe de l'écrit classique. Le cadre oral et interactif implique de travailler sur des énoncés oraux dont la syntaxe est très différente de celle trouvée dans des documents écrits et des questions. Les questions posées par les utilisateurs amènent des problèmes spécifiques dus à la formulation de la question en interaction avec le système : des hésitations, des répétitions, des références à des éléments passés de la conversation avec le système ... L'aspect "dialogue" oblige le système à avoir une bonne réactivité lors des interactions avec l'utilisateur. Cela implique que les réponses à une question doivent être données rapidement.

Du fait de ce contexte d'application, un certain nombre de décisions ont été prises lors de la conception du système. Une analyse multi-niveaux, présentée dans la section 2.2, est utilisée. Son objectif est d'extraire les informations utilisées d'un document ou d'une question. L'analyse est la même pour les documents et les questions. Un certain nombre de traitements sont faits au niveau de l'indexation des documents pour que les questions puissent être traitées plus rapidement par la suite. De même, il a été décidé que les élements utilisés à chaque étape de la recherche d'une question seraient les mêmes : l'analyse de la question fournit une représentation de l'information réutilisée par les différents modules de la recherche de la réponse [Galibert 2009].

L'architecture de Ritel est relativement similaire à celle présentée dans la figure 1.1 du chapitre 1. Des pré-traitements sont d'abord appliqués sur la base de documents. Une normalisation est effectuée sur les documents de manière à uniformiser les textes. Les textes normalisés sont ensuite traités par l'analyseur multi-niveaux de Ritel, puis indexés. Le système de questions-réponses s'appuie ensuite sur cet index. Chaque question traitée par le système est d'abord normalisée puis analysée : ses carac-

téristiques sont fournies sous la forme d'un Descripteur De Recherche (DDR). Le DDR contient les informations nécessaires à la recherche des réponses que le système a extrait de la question. Les trois modules suivants s'appuient sur le DDR de la question. Le système sélectionne d'abord un ensemble de documents contenant potentiellement la bonne réponse à partir de l'index des documents. Cette liste de documents est ensuite fourni au module de sélection des passages, qui extrait un ensemble de passages des documents. Enfin, les réponses candidates sont sélectionnées puis extraites des passages.

Dans les sections suivantes, nous commençons par présenter la normalisation appliquée sur les documents et questions. Nous présentons ensuite l'analyse utilisée par le système, et les raisons des choix effectués lors de la conception de cette analyse. Nous détaillons ensuite le système de recherche des réponses à une question. Nous présentons notamment le formalisme de représentation de l'information, et comment il est utilisé dans l'indexation des documents et les différentes étapes de la recherche : sélection des documents, sélection des passages et sélection et extraction des réponses candidates. Enfin nous concluons par une analyse et une discussion des caractéristiques d'un tel système. Une présentation plus complète du système se trouve dans [Galibert 2009].

2.1 Normalisation

L'objectif de la normalisation [Adda, et al. 1997] est de convertir des textes bruts dans une forme où les mots et les nombres sont délimités sans ambiguïtés, les lettres majuscules ne sont laissées que sur les noms propres, la ponctuation est séparée des mots, et le texte est segmenté en phrases (tant que faire ce peut). Concrètement, 4 étapes de normalisation sont appliquées :

– Séparer les mots et les nombres de la ponctuation.
– Modifier la casse des caractères pour les mots.
– Ajout de la ponctuation.
– Segmentation du texte en phrases à chaque point.

Les documents sont normalisés une fois pour toutes lors de leur indexation par le système Ritel. Les questions sont normalisées au fur et à mesure de l'avancement du dialogue.

2.2 Analyse des documents et des questions

Comme on l'a vu précédemment, l'analyse est multi-niveaux [Rosset 2008] et s'appuie sur un typage hétérogène. L'analyseur commence par typer sémantiquement des mots ou des groupes de mots. Si aucun typage n'est possible, il retourne alors la catégorie morphosyntaxique. L'analyse est faite hiérarchiquement : un groupe de mots est décomposé en plusieurs sous-noeuds typés. Le tout est typé par un noeud global. On obtient en résultat des arbres textuels d'une profondeur moyenne de 2 à 3 ni-

veaux. Les types des noeuds appartiennent à différentes catégories : entités nommées (une personne, un lieu), entités grammaticales (des verbes, des prépositions), entités nommées étendues (couleurs). Il y a pour l'instant environ 300 types différents. Une vingtaine sont d'ordre linquistique : substantif, mot composé, adverbe, etc ...

Contrairement à certains systèmes de questions-réponses, qui utilisent des analyseurs syntaxiques, il a ici été fait le choix de privilégier une analyse plus sémantique. Il existe des types linguistiques, mais on ne retrouve pas par exemple une analyse avec des dépendances syntaxiques complexes. Il a en effet été montré [Paroubek, et al. 2008a ; Paroubek, et al. 2008b] que des analyses syntaxiques profondes donnent de mauvais résultats sur des documents dont le niveau de langue était éloigné de l'écrit classique. L'utilisation d'énoncés oraux, et plus généralement de différentes sources pour les documents, nécessite une approche plus robuste. De plus, cette analyse est basée sur un moteur de règles (Wmatch) permettant des traitements rapides, et donc de répondre aux problèmes de réaction cités plus tôt.

La figure 2.1 illustre le résultat de l'analyse multi-niveaux sur la phrase *Qui est le maire de Toulon depuis les élections municipales de 2008*. Nous pouvons ainsi observer que *maire de Toulon* est regroupé en un seul type nommé *_pers_fonc*. Puis les trois mots sont étiquettés individuellement, des fois avec plusieurs types. Ainsi, si *de* est juste annoté comme étant une préposition *_prep*, *maire* est d'abord étiquetté comme étant une fonction, puis une fonction publique.

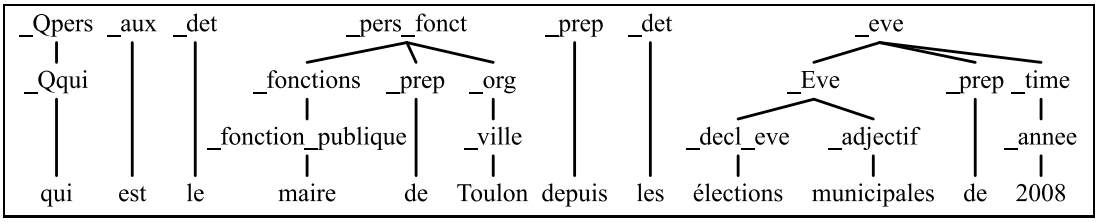

FIG. 2.1 – Exemple d'annotation issue de Ritel illustrant l'analyse multi-niveaux et le typage hétérogène

Dans les sections suivantes, nous parlerons souvent de paires type/valeur. Cette dénomination désigne pour chaque mot le type de l'analyse qui lui est associé. Dans la figure 2.1, *fonction_publique* et *maire* sont une paire type/valeur.

2.3 Système de questions-réponses

Une fois les documents et les questions normalisés et analysés, la recherche des réponses candidates à la question est effectuée. En s'appuyant sur la forme analysée de la question, le système de questions-réponses génère les informations nécessaires à la recherches des réponses. Le formalisme de représentation utilisé est appelé Descripteur de Recherche (DDR). Le DDR de chaque question

est utilisé par l'ensemble des modules du système de questions-réponses : sélection des documents, sélection des passages, et sélection et extraction des réponses.

Nous commençons par expliquer comment sont représentés les DDRs, et comment ils sont générés par le système. Ensuite, nous nous appuyons sur ces DDRs pour expliquer le fonctionnement des trois modules de recherche des réponses candidates.

2.3.1 Définition Descripteurs De Recherche (DDR)

Chaque question traitée par le système de questions-réponses est normalisée puis analysée. A partir de cette représentation de la question, le système génère une représentation des informations contenues dans la question, qui seront utilisées par les différents modules du système de questions-réponses. L'objectif est ainsi d'avoir une représentation uniforme des informations de recherche à chaque étape de traitement. Les auteurs ont adopté la terminologie de Descripteur de Recherche (DDR) pour définir cette représentation de l'information. Ces DDRs contiennent les élements considérés pertinents à trouver dans les documents, et les types de réponse attendus.

En premier lieu, le système analyse la question, et plus précisemment les marqueurs interrogatifs, pour déterminer le type de la question et donc le type de la réponse attendue. Ensuite, les différents éléments de la question sont identifiés et classés en deux catégories : critique et secondaire. Les éléments considérés comme critiques sont les mots ou groupes de mots devant être situés à proximité d'une réponse candidate. La présence des éléments secondaires est elle aussi souhaitée, mais pas indispensable. Les éléments ainsi que leur classification sont déterminés à partir des sous-arbres générés par l'analyse de la question. Les éléments critiques sont généralement des entités nommées. La figure 2.2 représente le DDR simplifié pour la question *"Quelle est la quantité de plutonium à recycler ?"*. Le mot *plutonium* est considéré comme étant critique, tandis que le verbe *recycler* est classé comme étant un élément secondaire. De plus, deux types de réponses sont cherchés, *masse* ou *volume*.

A chacun de ces éléments, qu'ils soient secondaires ou critiques, sont associées des transformations (ou variantes). L'idée est que les éléments d'une question peuvent se retrouver sous une autre forme dans les documents. Une transformation correspond aux différentes formes de l'élément que l'on va chercher dans les documents lors du processus de recherche des réponses. Par exemple un nom propre comme « Nelson Mandela » aura trois transformations : l'élément non transformé, que les auteurs nomment *identité*, et le nom ou le prénom tout seul, que les auteurs nomment *expansion*. De même pour d'autres mots on peut avoir des transformations telles que la lemmatisation ou la synonymie. Dans la figure 2.2, le verbe *recycler* a trois transformations associées : lemme, synonyme, et substantivation.

Aussi bien pour les types de réponses cherchés que pour les éléments et leurs transformations, des poids sont associés. Ces poids permettent de favoriser les types de réponse les plus probables ainsi que les éléments les moins transformés. Ces poids sont déterminés à la fois empiriquement et par le

- Élément critique
  - 1,0 *subs* identité(plutonium)
- Élément secondaire
  - 1,0 *action* identité(recycler)
  - 0,7 *action* lemme(recycle)
  - 0,5 *subs* verbe_subs(recyclage)
  - 0,5 *action* synonyme(traiter)
- Types de réponse
  - 1,0 *masse*
  - 1,0 *volume*

FIG. 2.2 – Exemple de Descripteur De Recherche pour la requête *Quelle est la quantité de plutonium à recycler ?*

biais d'expérimentations. Dans la figure 2.2, la transformation en synonyme de *recycler* a un poids de 0,5. Le poids des transformations est déterminé par la qualité des ressources linguistiques mises en oeuvre.

2.3.2 Recherche des réponses candidates

Nous avons présenté le modèle de représentation des informations à chercher pour trouver une réponse à une question, le Descripteur de Recherche (DDR). Les étapes de recherche de la question proprement dite vont s'appuyer sur les données contenues dans ce DDR. Nous présentons ces différentes étapes dans les sections suivantes : la sélection des documents, la sélection des passages, et enfin la sélection et l'extraction des réponses. L'idée derrière le découpage de la recherche en trois modules est de travailler sur des zones de textes de plus en plus petites, et ainsi appliquer des méthodes de plus en plus fines.

2.3.2.1 Sélection des documents

L'étape suivant généralement l'analyse de la question est celle de la recherche d'extraits pertinents. L'objectif est le suivant : extraire de la base des documents un ensemble de sous-parties (les passages des documents) pertinentes pour répondre à la question traitée. Le problème est de fournir assez de passages pour pouvoir trouver la réponse tout en garantissant une réponse rapide du système.

Une première étape avant d'extraire ces passages est de sélectionner les documents pertinents. Dans notre cas, un document correspond à un fichier. L'idée va être d'associer un score à chacun des documents pour juger de sa pertinence. Ce score va être calculé en se basant sur les éléments du DDR associé à la question. L'algorithme de calcul va compter le nombre d'occurences de chaque

transformation des éléments dans le document. On se servira à chaque fois des paires type/valeur pour effectuer la recherche.

Du fait de l'indexation effectuée préalablement sur la base de documents, chaque document est associé à une table contenant l'ensemble des paires type/valeur du document. A chacune de ces paires est associé le nombre d'occurences dans le document. Il faut aussi prendre en compte les transformations possibles de chacune des paires. Ces transformations sont donc aussi associées à leur paire type/valeur respective, tout comme leur nombre d'occurence dans le document. Ces tables permettent un traitement optimisé lors de la sélection des documents. Si l'on reprend le DDR présenté dans la figure 2.2, la figure 2.3 représente le résultat de la recherche des paires dans les documents satisfaisant les paires type/valeur et leur transformation.

Ligne du DDR	**Id document**	**Élément trouvé**
subs identité(plutonium)	45	*subs* plutonium
	134	*subs* plutonium
	141	*subs* plutonium
action identité(recycler)	4	*action* recycler
action lemme(recycler)	45	*action* recycle
action lemme(recycler)	65	*action* recylent
action lemme(recycler)	68	*action* recylent
action lemme(recycler)	134	*action* recylent
action lemme(recycler)	234	*action* recylent
action synonyme(recycler)	4	*action* convertir

FIG. 2.3 – Eléments de l'index des documents correspondant au DDR de la figure 2.2.

On peut voir qu'aucune transformation substantive du verbe *recycler* n'a pu être trouvée (absence dans la table). A partir des éléments trouvés, le score de chaque document va être calculé à partir des poids des entités correspondantes dans le DDR traité. Par exemple, pour le document 45, deux éléments sont trouvés, *plutonium* et *recycle*. Les entités correspondantes du DDR de la figure 2.2 ont respectivement un poids de 1.0 et 0.7. Le score du document 45 est calculé à partir de ces deux poids.

2.3.2.2 Sélection des passages

Une fois les documents sélectionnés, il faut alors les découper en passages. L'objectif est d'identifier les passages des documents contenant potentiellement la bonne réponse à la question. L'idée est de réussir à extraire des passages ni trop courts (perte potentielle de l'information liée à la réponse), ni trop longs (ajout d'informations sans lien avec la réponse).

Les lignes des documents sont analysées, de manière à déterminer leur distance d'influence. Une ligne correspond en général à une phrase, ou à son équivalent à l'oral. Le but va être de déterminer

quelles lignes sont validées par les éléments du DDR. Une ligne est validée si elle contient dans sa zone d'influence (fixée par une variable de paramétrage) tous les éléments critiques du DDR, ou au moins un élément secondaire s'il n'y a pas d'éléments critiques dans le DDR. Les ensembles de lignes valides vont permettre de créer un premier ensemble de passages (blocs de lignes). Nous illustrons ces traitements en nous appuyant sur le très court document présenté dans la figure 2.4. La question traitée est *"Quelle est la quantité de plutonium à recycler ?"*. Le DDR de la question est représenté dans la figure 2.2. Les mots en gras représentent les éléments critiques de la question, et les mots en italique les éléments secondaires. Nous fixons la zone d'influence d'une ligne à une phrase voisine. Par exemple, la zone d'influence de la ligne (4) comprend les lignes (3) et (5). Dans cet exemple, seules les lignes (6) et (7) ne sont pas validées : leur zone d'influence ne contient pas l'élément critique de la question, **plutonium**. Un passage est ainsi créé à partir de ce document, comprenant les lignes (1) à (5).

(1) Le **plutonium** est un métal *recyclable* lourd.
(2) Il est découvert en 1940 par des chimistes américains.
(3) Ces scientifiques s'en servirent pour créer la bombe atomique.
(4) Le dioxyde de **plutonium** est la forme la plus simple à manipuler.
(5) Il est utilisé pour le *recyclage*.
(6) Il s'agit d'une poudre de cristaux jaunes-verts.
(7) Il a longtemps été considéré comme inoxydable.

FIG. 2.4 – Exemple d'un document (très court) utilisé comme référence pour illustrer les traitements effectués par le module de sélection des passages. La question traitée est *"Quelle est la quantité de plutonium à recycler ?"* et son DDR est représenté dans la figure 2.2 ; les mots en gras correspondent aux éléments critiques de la question, et les mots en italique aux éléments secondaires.

Ces passages sont souvent très longs. De ce fait, un deuxième traitement va être appliqué sur ces passages. L'objectif sera de les subdiviser en plusieurs sous-passages. L'idée va être de changer un à un le statut des éléments secondaires en critiques et de tester de nouveau si les lignes valident le DDR, jusqu'à obtenir des sous-passages suffisamment petits. Dans l'exemple de la figure 2.4, la question traitée ne contient qu'un élément secondaire, *recycler*. En faisant passer cet élément en critique, la ligne (3) ne valide plus le DDR. Ainsi, deux sous-passages sont créés, comprenant respectivement les lignes (1) et (2) et les lignes (4) et (5).

Le problème de cette deuxième étape est qu'il peut arriver que les nouveaux passages de taille moindre ne contiennent plus les éléments critiques qui avaient rendu leur création pertinente. C'est le cas dans un DDR avec deux éléments critiques dont les entités les instanciant sont disposées sur deux lignes séparées : si ces deux lignes sont légérement plus éloignées que la distance d'influence, alors certaines lignes situées au milieu seront conservées mais les deux lignes contenant les entités ne le seront pas. De ce fait, une troisième étape est nécessaire où l'on fait grandir les passages de manière à réintégrer les éléments critiques perdus.

Une fois ces passages créés, il faut leur attribuer un score. Le calcul est très proche de celui utilisé

pour noter les documents : on se base aussi sur le nombre d'occurrences des éléments du DDR dans chacun des passages. Les seules différences étant que l'on ne compte pas plusieurs fois les inclusions d'entité, et que le score total est pondéré par le score du document d'où le passage est extrait.

2.3.2.3 Sélection et extraction des réponses

La dernière étape du système est d'identifier et d'attribuer un score aux réponses potentielles à une question. Le module récupère en entrée un ensemble de passages contenant des réponses potentielles à la question traitée. L'objectif est d'identifier les réponses potentielles des passages, puis de les classer selon un score. Ce score est basé sur une mesure de la distance entre le candidat réponse et les éléments de la question, similaire à [Plamondon & Kosseim 2003]. L'idée est que plus une réponse candidate est proche des éléments d'une question, plus cette réponse est potentiellement la réponse attendue à la question. Sachant qu'une réponse évaluée peut être retrouvée dans d'autres passages, la redondance est prise en compte dans le score de la réponse.

Nous définissons ci-dessous les traitements effectués pour sélectionner les réponses. Nous nous appuyons sur le passage représenté dans la figure 2.5 pour illustrer certains de ces traitements. La question traitée est *"Quelle est la quantité de plutonium à recycler ?"*. Le DDR de la question est représenté dans la figure 2.2. On retrouve deux éléments critiques de la question, qui sont indiqués en italique : *plutonium*, présent deux fois, et *recyclés*.

(1) Le *plutonium* est un métal lourd.
(2) **140 kilos** de *plutonium* doivent actuellement être *recylés* en France.

FIG. 2.5 – Exemple d'un passage utilisé comme référence pour illustrer les traitements effectués par le module de sélection des réponse. La question traitée est *"Quelle est la quantité de plutonium à recycler ?"* et son DDR est représenté dans la figure 2.2 ; les mots en italique correspondent aux éléments critiques et secondaires de la question ; la réponse candidate est **140 kilos**.

Le module détermine d'abord les candidats réponses de chaque passage. Le système considère les éléments du passage dont le type correspond à un des types de réponses attendus indiqués dans le DDR de la question. de la figure 2.2. Dans le passage, **140 kilos** correspond au type *masse*, et est donc considéré comme une réponse candidate. Pour chaque réponse candidate d'un passage, un score est calculé en plusieurs parties.

Une première partie du calcul détermine une distance entre la réponse candidate et les éléments du passage validant une entité du DDR. Cette distance n'est pas calculée en mots mais selon les groupements de mots effectués par l'analyse de Ritel. Nous définissons *d(e,a)* la distance entre l'élément du passage *e* validant une entité du DDR et la réponse *a*. Dans l'exemple de la figure 2.5, les deux occurences de *plutonium* ainsi que *recyclés* valident les entités suivantes du DDR :

– 1,0 *subs* identité(plutonium)
– 0,7 *action* lemme(recycle)

Néanmoins, tous les éléments d'un passage ne sont pas pris en compte dans le calcul, mais seulement ceux jugés pertinents. L'analyse employée par Ritel n'indique pas si les éléments du passage sont en relation avec la réponse candidate. Ainsi, la pertinence est déterminée en choisissant le sous-ensemble d'éléments du passage maximisant le score. Ces sous-ensembles sont créés en évitant les redondances des entités du DDR validées par les éléments. Dans l'exemple de la figure 2.5, les deux occurrences de *plutonium* valident la même entité. Ce passage a ainsi deux sous-ensembles d'éléments : les deux contiennent l'élément *recyclés* et une des deux occurrences de *plutonium*.

On définit donc *E*, un ensemble de paires *(e,l)* où *e* est un élément du passage et *l* l'entité du DDR validée, et dont les éléments sont jugés non redondants. On associe pour chacune de ces paires un score calculé à partir de la distance *d(e,a)* entre l'élément du passage et le candidat réponse, le poids associé à l'entité du ddr *w(l)*, et une variable α. Le score pour l'ensemble *E* est alors la somme des scores individuels.

$$S(E) = \sum_{(e,l)\in E} \frac{w(l)}{(1+d(e,a))^{\alpha}}) \tag{2.1}$$

Pour tous les ensembles *E*, nous cherchons celui donnant le meilleur score possible. Puis nous multiplions ce score par le poids du type du candidat réponse *a*. Dans l'exemple de référence de la figure 2.5, le candidat réponse a un poids de 1,0, indiqué dans le DDR de la figure 2.2.

$$S_1(a) = w(a) \max_E \sum_{(e,l)\in E} \frac{w(l)}{(1+d(e,a))^{\alpha}}) \tag{2.2}$$

Afin de prendre en compte le passage dans ce calcul, le score est affiné par le score du passage associé S_p et une variable γ.

$$S_2(a) = S_1^{1-\gamma} S_p^{\gamma} \tag{2.3}$$

De plus, la redondance est une part importante de cette approche. De ce fait, le score d'une réponse correspond à la somme des calculs obtenus pour chaque occurrence de cette réponse dans les passages. On note A_r l'ensemble des instances de candidats réponses ayant *r* comme paire type/valeur. Le score global est la somme des scores individuels.

$$S_1(r) = \sum_{a\in A_r} S_2(a) \tag{2.4}$$

Cependant, un dernier affinage est nécessaire. En effet, le score privilégie trop les réponses trop fréquentes. On compense donc cette valeur par rapport au produit du nombre d'occurrences dans les passages et dans les documents (respectivement $C_p(r)$ et $C_d(r)$. Par ailleurs, deux variables sont utilisées dans le calcul de ce score, β et δ.

$$S(r) = \frac{S_1(r)}{C_d(r)^\beta C_p(r)^\delta} \tag{2.5}$$

L'équation ci-dessous détaille la formule finale de calcul du score des réponses :

$$S(r) = \frac{\sum_{a \in A_r} (w(a) \max_{E_a} \sum_{(e,l) \in E_a} \frac{w(l)}{(1+d(e,a))^\alpha})^{1-\gamma} S_p(a)^\gamma}{C_d(r)^\beta C_p(r)^\delta} \tag{2.6}$$

Il est évident qu'évaluer l'ensemble des réponses potentielles peut être coûteux en temps de calcul [Galibert 2009]. De ce fait, une limitation sur le nombre de candidats possibles est fixée. L'ordre de calcul va dépendre du score du passage. Enfin, les différentes variables utilisées sont fixées via de multiples expérimentations. En effet, les interactions entre les différentes valeurs sont très difficiles à déterminer, et l'absence de convexité dans l'espace de recherche pose aussi problème. Enfin, ces types de valeurs sont différentes selon le type général de la question.

2.3.3 Résultats obtenus

Le système Ritel a participé officiellement à deux campagnes d'évaluation : QAst [Turmo et al. 2008 ; Turmo et al. 2009] (Question-Answering on Speech Transcripts) et Quaero [http ://www.quaero.org/ 2008 ; Quintard 2009 ; Quintard et al. 2010]. Ces deux campagnes évaluent les systèmes de questions-réponses respectivement sur des transcriptions de documents oraux, et un corpus construit à partir du web. Nous décrivons plus en détail ces deux campagnes d'évaluation dans le chapitre 7.

Le système Ritel a participé aux évaluations successives de QAst : 2007, 2008 et 2009. En 2007, les systèmes n'étaient évalués que sur l'anglais. A partir de 2008, les systèmes étaient évalués en plus sur le français et l'espagnol. Nous ne donnons que les résultats obtenus sur le français, le travail présenté dans ce document étant consacré à cette langue. De plus, les systèmes étant évalués sur des documents issus de l'oral, l'un des objectifs de la campagne était d'évaluer l'impact des erreurs engendrées par des systèmes de reconnaissance automatique de la parole sur les participants. Ainsi les documents du corpus étaient fournis en 4 exemplaires : une transcription manuelle, et trois sorties de systèmes de reconnaissance, avec différents niveaux de qualité. En 2009, le corpus de questions était construit en demandant à des utilisateurs de poser des questions. De ce fait, deux versions du corpus étaient fournies : une transcription manuelle, et une transcription automatique. Ritel est le seul système à avoir participé à l'évaluation QAst en français. Dans chaque édition, le nombre de réponses

retournées était de 5. Les résultats par questions obtenus par Ritel sur le français en 2008 et 2009 sont présentés dans le tableau 2.1.

Année	modalité	transcription	préc. (%)	MRR
2008	écrite	manuelle	45	0.50
		ASR A	41	0.49
		ASR B	25	0.28
		ASR C	21	0.24
2009	écrite	manuelle	28	0.39
		ASR A	26	0.31
		ASR B	21	0.25
		ASR C	21	0.24
	orale	manuelle	28	0.39
		ASR A	25	0.30
		ASR B	21	0.25
		ASR C	20	0.24

TAB. 2.1 – Précision (préc.) et Mean Reciprocal Rank (MRR) obtenus par Ritel sur les itérations 2008 et 2009 en français de QAst. ASR A, B et C correspondent à des transcriptions issues de trois systèmes de reconnaissance automatique de la parole.

Le projet Quaero contient une tâche consacrée aux systèmes de questions-réponses. Dans cette évaluation, les systèmes sont évalués sur des questions de type factuelle, mais aussi définition, oui/non, pourquoi, comment. Nous ne donnons que les résultats obtenus sur les questions factuelles. De plus, les participants sont évalués sur des documents extraits du web. Ritel a participé aux itérations 2008, 2009 et 2010 de cette évaluation. En plus de Ritel, deux autres systèmes ont participé aux itérations 2008 et 2009 : FIDJI [Tannier & Moriceau 2010], du groupe LIMSI-ILES, et QRISTAL [Laurent et al. 2006]. En 2010, un quatrième système a participé : QAVAL [Grappy & Grau 2010], du groupe LIMSI-ILES. Par ailleurs, deux sous-corpus de questions ont été ajoutés. Les utilisateurs devaient poser des questions spontanées, puis ces questions étaient réécrites. Chaque système devait retourner au maximum 3 réponses. Sur cette dernière édition, le système Ritel a été évalué en appliquant le réordonnanceur présenté dans ce document. Les résultats obtenus sont décrits dans le tableau 2.2.

Dans la section suivante nous analysons les résultats présentés, et nous discutons les avantages et inconvénients de l'approche employée par Ritel.

2.3.4 Analyse des résultats

Avant de discuter des avantages et inconvénients de Ritel, nous tenons à rappeler ses caractéristiques principales. Ritel est un système de questions-réponses oral interactif en domaine ouvert. Il a donc la particularité de traiter des textes dont la syntaxe n'est pas forcément celle que l'on retrouve dans des

	Ritel		FIDJI		QRISTAL		QAVAL	
Année	Préc. (%)	MRR	Préc. (%)	MRR	Préc. (%)	MRR	Préc. (%)	MRR
2008	19.3	0.204	11.9	0.143	30.9	0.337		
2009	27.5	0.284	33.0	0.372	50.2	0.540		
2010 normal	37.4	0.450	34.1	0.409	66.5	0.693	14.8	0.195
2010 spontanée	36.5	0.414	22.4	0.267	70.9	0.746	8.21	0.102
2010 réécrite	35.9	0.403	23.6	0.275	67.5	0.706	10.5	0.133

TAB. 2.2 – Précision (Préc.) et Mean Reciprocal Rank (MRR) obtenus par Ritel et les autres participants sur les itérations 2008, 2009 et 2010 de Quaero pour les questions factuelles.

textes considérés comme bien écrits. De plus l'interactivité apporte certaines contraintes supplémentaires, comme la rapidité du temps de réponse. Ces particularités ont guidé la définition des approches utilisées dans ce système. On peut notamment citer le système d'analyse des documents et des questions, et la génération des DDRs pour chaque question. Ces deux approches permettent de répondre aux différentes contraintes au contexte d'application.

Les résultats obtenus sur les campagnes QAst et Quaero permettent de tirer des conclusions positives sur la pertinence de cette approche. S'il est difficile d'analyser les résultats obtenus sur QAst en français étant donné l'absence d'autre participants sur cette langue, les résultats obtenus [Turmo et al. 2009] sur les autres langues montrent la robustesse de l'approche, le système se classant premier sur la majorité des tâches. Sur Quaero, si le système QRISTAL de Synapse obtient de bien meilleurs résultats, il est à noter que les résultats obtenus par Ritel sont très encourageants, et permettent au système de se classer deuxième sur les questions factuelles normales en 2008 et 2010. De plus, si les résultats de QRISTAL sont relativement peu modifiés par les sous corpus de questions spontanées, le changement de modalité entraîne une baisse significative pour FIDJI et QAVAL. Le système Ritel n'enregistre qu'une baisse d'environ 1%, ce qui est très encourageant.

Ceci dit, l'analyse des résultats obtenus montre une certaine limite de l'approche utilisée par le système sur certains types de questions, en particulier par la méthode de sélection et d'extraction des réponses. Lors de l'évaluation 2009 de Quaero, nous avons observé une perte significative entre le pourcentage des questions avec le passage contenant la bonne réponse (70%) et le pourcentage des questions avec la bonne réponse en premier rang (27.5%). Une analyse a été faite sur l'origine de cette perte des bonnes réponses. Parmi les différents points soulevés, deux concernaient directement la méthode de scoring des réponses par distance : l'importance de la redondance et l'absence de prise en compte de la structure des phrases et des questions.

Prenons la question suivante, *"Combien d'années Nelson Mandela passa-t-il en prison ?"*. Un des passages évalués pour répondre à cette question est « A 71 ans, Nelson Mandela est sorti de prison après 27 années. » , et deux réponses sont évaluées, *71 ans* et *27 années*, cette dernière étant la bonne réponse. Dans un tel cas, l'approche utilisée par Ritel se basera sur la distance entre les éléments de la question et les réponses évaluées, ainsi que les autres occurences de ces réponses dans les différents

passages traités. Or, les réponses *71 ans* et *27 années* ont une distance de respectivement 0 et 5 de *Nelson Mandela*, et une distance de 4 et 1 de *prison*. La distance moyenne par rapport aux éléments de la question est donc identique pour les deux réponses candidates. Pour peu que la réponse 71 ans apparaisse plus souvent dans le reste des documents, elle sera choisie comme étant le bon choix. Pourtant, une mise en relation peut permettre dans ce cas de déterminer la bonne réponse : on identifie le rattachement entre *27 années* et *prison*.

2.4 Hypothèses

Nous avons montré dans la section précédente certaines limites du système de questions-réponses de Ritel, notamment sur l'absence de structuration entre les éléments du DDR et les éléments de la phrase réponse, et l'importance accordée à la redondance dans les calculs. Les résultats obtenus sur la sélection des passages montrent une perte non négligeable lors de l'extraction des réponses. Néanmoins, on peut noter que si la bonne réponse n'obtient pas le meilleur score, elle se retrouve assez fréquemment dans les n-premières réponses. Ce n'est donc pas la sélection des réponses qui est l'étape la plus critique dans cette baisse des résultats, mais davantage le scoring des réponses, et donc le rang obtenu par chaque candidat.

Le travail présenté dans ce document a pour objectif de proposer une méthode robuste de réordonnancement des réponses candidates à une question. L'objectif d'une telle méthode est de calculer un nouveau score pour chaque réponse candidate retournée par un système de questions-réponses, et de réordonner ces réponses selon ce score. L'idée est d'utiliser des approches plus fines que celles utilisées par un sytème de questions-réponses en début de traitement. Nous voulons aussi que cette approche soit robuste quel que soit le type de documents traités : écrit, oral, web. Nous avons pris comme cadre expérimental le système Ritel.

L'un des principaux défauts de Ritel est l'absence de représentation de l'information structurelle contenue dans les documents et les questions. L'absence d'information sur les dépendances entre les différents groupes de mots conduit à certains cas ambigus où l'approche utilisée par Ritel montre ses limites. Ritel a vocation à être un système de dialogue, et se doit donc d'être robuste à des transcriptions de l'oral. Ce besoin de robustesse a pour conséquence la non-utilisation d'analyses syntaxiques profondes. Or il y a clairement un besoin d'informations structurelles pour résoudre certains cas ambigus.

Les informations proposées par Ritel étant insuffisantes, un de nos objectif est donc de définir un modèle de représentation de l'information structurelle contenue dans les documents et les questions. Ce modèle est ensuite utilisé par notre méthode de réordonnancement. Avant de passer à la seconde partie de ce document, intitulée *Contributions*, nous mettons en parallèle les hypothèses décrites dans cette section avec les conclusions tirées sur le domaine des questions-réponses. Nous en tirons certaines conclusions pour la suite de notre travail.

Discussion

Dans la précédente partie, nous avons présenté le domaine des questions-réponses. Nous avons commencé par définir ce domaine par rapport à celui de la *Recherche d'Informations*. Les systèmes de questions-réponses peuvent être soit conçu dans un cadre d'application fermé, soit ouvert. En domaine ouvert, de nouvelles problématiques doivent être traitées : structure des questions, type de réponse attendu, type de documents traités ...

L'architecture générale des systèmes de questions-réponses a été présentée. S'il n'existe pas vraiment d'approche standard pour trouver une réponse à une question, on peut néanmoins définir une structure générale qui est souvent utilisée : analyse de la question, sélection des documents, sélection des passages, et sélection et extraction de la réponse. L'analyse de la question va généralement permettre d'extraire les informations nécessaires à la recherche de la bonne réponse dans la base de documents. Les étapes suivantes vont alors avoir comme but de sélectionner et d'extraire la bonne réponse en effectuant des traitements sur des textes de taille de plus en plus réduite : d'abord des documents, ensuite des passages extraits de ces documents, et enfin l'extraction des réponses candidates. Le fait de travailler sur des textes de plus en plus réduits permet généralement d'appliquer des traitements de plus en plus fins, mettant en oeuvre des méthodes beaucoup plus complexes.

Le réordonnancement de réponses est une étape de traitement qui n'est pas propre à tous les systèmes de questions-réponses. Elle a pour objectif de réordonner les réponses retournées par un système de questions-réponses selon une nouvelle métrique. Les scores sont ainsi réévalués en utilisant des méthodes de plus en plus complexes. Ces méthodes font souvent appel à des analyses syntaxiques et sémantiques, ainsi qu'à des ressources linguistiques.

Le travail présenté dans ce document a pour objectif de définir une méthode de réordonnancement de réponses, avec deux contraintes fortes : robustesse de la méthode par rapport aux types de documents traités, et utilisation du français. Parmi les conséquences de ces deux contraintes on peut noter :

– le manque de certaines ressources linguistiques en français (par exemple FrameNet [Ruppenhofer et al. 2006] ou PropBank [Palmer et al. 2005]) ;
– l'inadéquation de nombreuses analyses syntaxiques ou sémantiques classiquement utilisées pour l'écrit lors du traitement de documents oraux ou web.

Nous avons présenté un état de l'art, détaillant ce que nous considérons comme les trois approches générales utilisées pour la conception de systèmes de questions-réponses : approches linguistiques [Hickl et al. 2006c], approches statistiques [Sasaki 2005], et approches intermédiaires [Berthelin et al. 2003 ; Gillard et al. 2006]. Nous avons expliqué les avantages et inconvénients de chaque approche en présentant pour chacune d'entre elles un système de questions-réponses. L'approche fortement linguiste du LCC [Hickl et al. 2006c] donne de très bons résultats mais repose sur un certain nombre de ressources généralement sans équivalent en français, comme par exemple FrameNet [Ruppenhofer et al. 2006] ou PropBank [Palmer et al. 2005]. Elle s'appuie en plus sur des analyses complexes difficilement applicables sur un contexte autre que l'écrit (particulièrement l'oral). L'approche fortement statistique de [Sasaki 2005] a l'avantage d'être très robuste : le classifieur utilisé pour extraire les réponses s'appuie sur des traits simples et applicables dans n'importe quel contexte. Il semble cependant difficile d'améliorer les résultats obtenus sans introduire plus d'information, notamment par le biais d'analyse syntaxique ou sémantique. Enfin, les systèmes hybrides comme [Berthelin et al. 2003] du LIMSI-ILES ont tendance à utiliser des ressources et des analyses linguistiques, mais de manière moins importante. Le système du LIMSI-ILES utilise une analyse syntaxique par dépendances non applicable dans notre cadre de travail incluant des documents de sources variées.

Notre méthode de réordonnancement s'inscrit dans le contexte expérimental du système Ritel [Rosset et al. 2006]. Ritel est un système de questions-réponses en domaine ouvert oral. De ce fait, on retrouve certaines contraintes et problématiques associées au travail présenté dans ce document, notamment le besoin de robustesse par rapport aux types de données traitées. De ce fait, les auteurs de Ritel ont fait le choix de ne pas s'appuyer sur une analyse syntaxique. La sélection et l'extraction des réponses est effectuée par le biais d'une métrique prenant en compte la répartition des éléments d'une question par rapport à un candidat réponse. La redondance du candidat réponse dans le reste de la base de documents est aussi prise en compte dans le calcul de ce score.

Si cette approche donne de bons résultats et a prouvé sa robustesse comme l'ont montré les résultats obtenus dans différentes campagnes d'évaluations impliquant différents types de documents (QAst pour l'oral, Quaero pour le web), elle montre néanmoins certaines limites. La non exploitation de l'information structurelle des documents et questions, notamment les dépendances entre les différents groupes de mots, ne permet pas à Ritel de résoudre certains cas ambigus.

Ceci suggère qu'un calcul plus complexe appliqué en fin de chaîne de traitements dans un système de questions-réponses pourrait contribuer à réduire les cas d'ambiguïté entre candidats réponses. L'approche de réordonnancement proposée dans ce document vise à s'appuyer sur une analyse et un format de représentation des données afin d'améliorer les performances du système tout en répondant aux deux contraintes principales de notre travail : robustesse et utilisation du français.

Dans la prochaine partie, nous présentons l'état de l'art concernant les différentes approches de réordonnancement proposées dans la littérature ainsi que les analyses sur lesquelles ces approches s'appuient. Nous présentons ensuite les contributions de ce document : un modèle de représentation des données et une méthode de réordonnancement, tous deux robustes à une grande variété de types de documents traités en entrée.

Deuxième partie

Contributions

Introduction

Le chapitre 1 a introduit le domaine des questions-réponses, en présentant une architecture générale pour de tels systèmes ainsi que les différentes approches utilisées. Nous avons notamment présenté différentes problématiques pouvant être rencontrées lors de la conception de tels systèmes, et ayant un impact direct sur les méthodes utilisées : ressources linguistiques à disposition, langue utilisée, types de documents traités ... Le système de questions-réponses oral du projet Ritel, servant de cadre expérimental, a été présenté dans le chapitre 2. Ce système a pour objectif d'être applicable en domaine ouvert et d'être robuste quel que soit le type de documents traités. Les choix effectués dans la conception du système montrent certaines limites, notamment sur la non utilisation de l'information structurelle des phrases des documents et des questions. Nous allons dans cette partie nous focaliser sur nos contributions, que nous divisons en deux composantes : un modèle de représentation des données et son implémentation, et un module de réordonnancement des candidats réponses d'un système de questions-réponses.

Dans le chapitre 1, il a été montré dans l'état de l'art sur les systèmes de questions-réponses que de nombreux systèmes ajoutaient un module de validation de réponses en fin de traitement. L'objectif d'un tel module est de réordonner les réponses scorées par les traitements antérieurs en utilisant des approches plus complexes et fines. Placer ces modules en fin de chaîne de traitements permet de ne travailler que sur un nombre réduit de réponses extraites, les premiers traitements servant à effectuer un premier *filtrage*. L'idée serait donc de concevoir un module de réordonnancement de réponses robuste à n'importe quel type de documents fournis.

Néanmoins, de tels systèmes nécessitent de s'appuyer sur un modèle de représentation de l'information pour mettre en œuvre des approches relativement complexes [Moschitti & Quarteroni 2010 ; Comas, et al. 2010]. Ces modèles de représentation s'appuient généralement sur des analyses sémantiques ou syntaxiques. Le cadre expérimental de notre travail étant le système Ritel, cette représentation utiliserait en partie l'analyse sémantique produite par Ritel. Néanmoins, il a été montré dans le chapitre 2 que l'analyse était insuffisante quant à la représentation de l'information structurelle des documents et des questions. Par ailleurs, nous avons comme objectif d'avoir une méthode de réordonnancement robuste à tous types de documents. De ce fait, le modèle de représentation doit être lui aussi défini en prenant en compte ces contraintes de robustesse.

Nous articulons cette partie en trois chapitres. Dans le chapitre 3, nous effectuons un état de l’art sur différentes approches de réordonnancement, ainsi que plusieurs méthodes s’en rapprochant (sélection et extraction de réponses, et implication textuelle). Nous présentons aussi dans ce chapitre plusieurs modèles de représentation de l’information. Nous nous appuyons ensuite sur cet état de l’art pour présenter les choix effectués dans notre travail. Dans le chapitre 4, nous détaillons notre modèle de représentation de l’information, et son implémentation. Le chapitre 5 présente la mise en oeuvre du module de réordonnancement.

Chapitre 3

Approches pour le réordonnancement de réponses

Ce chapitre a pour objectif de présenter différentes approches pouvant être appliquées dans le cadre du réordonnancement de candidats réponses pour un système de questions-réponses. Le réordonnancement de réponses est une phase optionnelle d'un système de questions-réponses qui arrive généralement en bout de chaîne de traitement. Ainsi, cette phase est généralement comprise dans la phase d'extraction des réponses, et de ce fait les techniques employées sont souvent très voisines. L'extraction de réponse est la phase la plus complexe d'un système de questions-réponses : si les systèmes arrivent généralement bien à sélectionner les documents puis les passages ou phrases contenant une réponse correcte à une question, l'extraction même de la réponse est plus problématique. Diverses approches sont utilisées [Moschitti & Quarteroni 2010 ; Comas et al. 2010], et il n'existe à ce jour pas vraiment de méthode standard. Le réordonnancement de réponses a pour but de fournir un nouveau classement par rapport à une liste de réponses candidates. L'exemple de la figure 3.1 présente un réordonnancement possible pour les réponses retournées à la question *"Question : En quelle année Thomas Mann a-t-il obtenu le prix Nobel ?"*, avec 1929 comme bonne réponse à la question.

Ces méthodes de réordonnancement peuvent être apperçues dans différents systèmes de questions-réponses [Hickl et al. 2007 ; Grappy & Grau 2010]. Par ailleurs, les modules de sélection et d'extraction de réponses emploient souvent des méthodes complexes qui nous semblent adaptables au réordonnancement [Tannier & Moriceau 2010 ; Stenchikova, et al. 2006]. Nous estimons donc que leur étude est importante pour mettre en place un module de réordonnancement. Enfin, le domaine de l'implication textuelle, qui consiste à déterminer si un segment de texte en implique un autre, nous semble lui aussi pertinent quant à son application dans le cadre du réordonnancement. Ce domaine est appliqué dans le cadre de systèmes de questions-réponses [Kouylekov & Negri 2010].

Ce chapitre est organisé de la manière suivante : dans un premier temps nous présentons un état de l'art effectué sur différentes méthodes de réordonnancement utilisées dans le cadre de systèmes de

Question : En quelle année Thomas Mann a-t-il obtenu le prix Nobel ?		
Avant réordonnancement		Après réordonnancement
1909	=>	1929
1946	=>	1909
1929	=>	1946
1935	=>	1944
1944	=>	1984
1984	=>	1935

FIG. 3.1 – Exemple de réordonnancement pour la question *En quelle année Thomas Mann a-t-il obtenu le prix Nobel ?* ; 1929 est la bonne réponse.

questions-réponses. Ces systèmes ont chacun une approche d'extraction ou de validation/réordonnancement de réponses avec des caractéristiques différentes. Dans un second temps, nous nous appuyons sur le modèle de représentation des documents et questions proposé par ces différents systèmes pour étudier plus en détail les approches possibles quant à la représentation des données dans un système de questions-réponses. Nous nous intéressons notamment à la segmentation en composants (segments, ou chunks) typés des documents et questions ainsi qu'aux relations typées entre mots ou groupes de mots.

3.1 Etude de différentes méthodes applicables pour le réordonnancement

Si certaines des approches présentées dans cette section ne sont pas utilisées dans des modules de réordonnancement (re-ranking) à proprement parler, elles ont toutes en commun d'appliquer des méthodes assez fines de calcul de score, contrairement à celles utilisées pour extraire les documents ou passages. C'est ce type de méthodes que nous voulons utiliser dans notre module de réordonnancement. Nous nous intéressons donc aussi à des approches de sélection et extractions de réponses, et à des approches d'implication textuelles.

Pour chaque méthode présentée, nous motivons son étude par rapport à nos objectifs et à notre contexte de travail. Nous détaillons aussi le système de questions-réponses associé. Nous estimons en effet qu'il est important de comprendre le contexte d'application pour présenter l'approche utilisée. Enfin, nous concluons sur les caractéristiques de la méthode, et son intérêt pour un module de réordonnancement. Par ailleurs, certaines de ces approches s'appuient sur un formalisme spécifique de représentation des documents qui est détaillé plus loin dans la section 3.2.

3.1.1 Utilisation de dépendances syntaxiques dans le cadre du web : le système FIDJI

FIDJI [Tannier & Moriceau 2010] est un système de questions-réponses développé au LIMSI au sein du groupe ILES. A l'origine, le système avait été défini pour traiter des collections de documents *propres*, comme des documents journalistiques. Ce système (tout comme Ritel) participe à la tâche questions-réponses du projet Quaero [Quintard et al. 2010], que nous décrivons plus loin dans ce document (chapitre 6). Cette tâche a pour objectif d'évaluer les systèmes participants sur un corpus de très grande taille (2 millions de documents) extrait du web par le biais d'un moteur de recherche. Les documents tirés du web amènent des caractéristiques assez particulières, que les systèmes doivent prendre en compte pour être performants sur ce type de collection.

L'objectif de FIDJI est de mettre au point une approche efficace ne reposant pas sur des ressources sémantiques, et avec un minimum de pré-traitements effectués sur les documents. FIDJI traite deux types de questions : factuelles et complexes (pourquoi et comment). Dans cette section, nous ne décrivons que la méthode utilisée pour résoudre les questions factuelles. FIDJI n'inclut pas de module de réordonnancement. Par contre, l'approche employée pour l'extraction des réponses candidates nous semble intéressante car elle s'appuie sur des dépendances syntaxiques, représentant ainsi l'organisation structurelle des phrases et des questions. De plus, cette approche est appliquée dans le contexte du web, où la variabilité de la structure des phrases des documents est très différente de celle de documents journalistiques. Nous expliquons d'abord le fonctionnement général du système de questions-réponses, avant de détailler la méthode d'extraction des réponses.

Le corpus de document est indexé par le moteur de recherche Lucene [Apache 2007]. La sélection des documents est effectué en créant une requête composée des mots clefs de la question. Les 100 premiers documents sont alors analysés syntaxiquement par XIP [Aït-Mokhtar, et al. 2002]. XIP fournit les dépendances syntaxiques des phrases des documents. Les entités nommées sont elles aussi annotées par XIP. Le même traitement est effectué sur la question. FIDJI va ensuite sélectionner les phrases contenant le plus de dépendances syntaxiques de la question traitée. Enfin, la réponse est extraite à partir des dépendances syntaxiques de la phrase, du type de la question et du type attendu de la réponse.

L'exemple ci-dessous explique comment se déroule l'extraction de la réponse. Pour la question, *Quel premier ministre s'est suicidé en 1993 ?*, 5 dépendances syntaxiques ont été identifiées par XIP :

– DATE(1993)
– PERSONNE(réponse),
– SUJET(se suicider, réponse),
– attribut(réponse, ministre),
– attribut(ministre, premier).

La question est de type *factuelle*, et le type de la réponse attendue est *personne*, avec comme type spécifique *premier ministre*. Dans cet exemple, la phrase avec le plus de dépendances syntaxiques est la suivante : *Pierre Bérégovoy s'est suicidé en 1993.*, avec comme dépendances DATE(1993),

PERSONNE(Pierre Bérégovoy), et SUJET(se suicider, Pierre Bérégovoy). *Pierre Bérégovoy* instancie l'élément *réponse* de la question, et le système identifie alors les dépendances de la question validées. Dans ce cas, les trois premières dépendances sont validées. Le système va alors chercher les dépendances manquantes dans une phrase de la collection de document (par exemple *le premier ministre Pierre Bérégovoy*).

Le système a obtenu de bons résultats sur l'évaluation 2009 Quaero pour les questions factuelles en se classant second, avec un MRR de *0.37* et une précision de *33.0*%, le meilleur résultat étant de *0.54* et *50.2*%. On peut aussi noter que les résultats obtenus sur les questions *pourquoi* et *comment* sont bons : le MRR obtenu est respectivement de *0.32* et *0.49*, le second meilleur système obtenant *0.15* et *0.19*. Ces résultats montrent qu'une approche utilisant des dépendances syntaxiques est applicable sur des documents autres que des sources journalistiques. Il est donc possible d'utiliser des dépendances syntaxiques pour un module de réordonnancement. Par contre, l'analyseur XIP [Aït-Mokhtar et al. 2002] n'est pas adapté pour traiter des documents de l'oral. Ainsi, si les dépendances syntaxiques semblent pertinentes pour le réordonnancement, il semble nécessaire d'avoir une approche adaptée à l'oral.

3.1.2 Utilisation de dépendances syntaxiques et de méthode par apprentissage pour des transcriptions orales : le système de l'UPC

Le système de l'UPC [Comas et al. 2010](Université Polytechnique de Catalogne) a été conçu pour travailler sur des transcriptions orales, et a par ailleurs participé aux différentes campagnes de l'évaluation QAst [Turmo et al. 2008 ; Turmo et al. 2009]. Etant donné les caractéristiques de l'oral, les auteurs ont choisi de s'appuyer sur un analyseur syntaxique et sémantique développé au sein de l'UPC [Lluis, et al. 2009]. Seules les relations syntaxiques fournies par cet analyseur sont utilisées. En effet, l'analyseur n'est pas totalement adapté à l'oral. Les auteurs ont ainsi fait le choix d'extraire les caractéristiques robustes : l'information sémantique délivrée par l'analyseur n'est pas utilisée, et seules certaines informations syntaxiques sont reprises (principalement les dépendances syntaxiques). Ce système traite deux langues, l'espagnol et l'anglais. L'intérêt du système de l'UPC par rapport à notre travail est multiple. Deux différentes méthodes de réordonnancement sont comparées sur un ensemble de questions de la campagne QAst 2009. L'une des deux méthodes s'appuie sur les dépendances syntaxiques fournies par l'analyseur de l'UPC. Par ailleurs, ces méthodes sont évaluées dans le cadre de l'oral, ce qui permet d'observer leur robustesse par rapport à des documents transcrits de l'oral.

Le système de l'UPC utilise une architecture classique pour un système de questions-réponses. Le type de la question est d'abord détecté à partir d'un classifieur Perceptron utilisant des traits lexicaux, sémantiques et syntaxiques. A partir du type de la question, le système interroge à l'aide d'un moteur de recherche la base de documents et récupère un ensemble de passages candidats. L'extraction des réponses s'appuie sur un ensemble d'heuristiques évaluant les passages hypothèses selon les caractéristiques suivantes : les chaînes de mots équivalentes, les ponctuations, le nombre de mots de la

question suivant le candidat réponse, les mots de la question trouvés dans la même phrase que le candidat réponse et dans son contexte, la distance la plus élevée en mots entre deux mots de la question, et enfin la distance entre le focus de la question et la réponse candidate. Ces heuristiques sont transformées en un score dont chaque composante a un poids assigné à partir d'un paramétrage manuel effectué en fonction de la collection de documents.

A partir de cette architecture, les auteurs ont ajouté un module de réordonnancement prenant en entrée les candidats réponses. Ce module s'appuie sur un classifieur dont l'objectif est de déterminer si un candidat réponse répond ou non à la question. Le classifieur utilise un modèle généré par SVM à partir d'un corpus de développement pour déterminer si une réponse candidate est positive ou négative. Les valeurs retournées par les heuristiques sont converties en caractéristiques binaires qui sont ensuite utilisées comme traits par le classifieur. En plus des valeurs des heuristiques, les auteurs fournissent aussi au classifieur les valeurs (elles aussi binarisées) suivantes : le score total obtenu lors de l'extraction des réponses, le rang de la réponse candidate, la redondance de cette réponse, le type de l'entité nommée, et enfin le nombre de mots-clefs dans la question.

La dernière approche réutilise le même classifieur en ajoutant de nouveaux traits. Ces traits sont définis en s'appuyant sur les dépendances syntaxiques produites par l'analyseur utilisé par l'UPC. L'idée est de prendre les chemins de dépendances entre chaque mot-clef de la question et le marqueur interrogatif, et de les comparer avec ceux existant dans le passage candidat, cette fois entre les mots-clefs présents et la réponse candidate. Ces chemins sont au préalable simplifiés en enlevant certaines dépendances fréquentes : modifieurs de nom, prépositions, adverbes ... Par ailleurs, les verbes contingents sont réunis. En comparant ces chemins l'idée est d'identifier les mots-clefs proches d'une réponse candidate mais qui ne sont pas en relation, et au contraire valider les mots-clefs éloignés mais reliés syntaxiquement. A partir de cette hypothèse, les auteurs introduisent les traits suivants : le nombre de mots-clefs en relation avec la réponse et le ratio par rapport au nombre total, les distances en nombre de relations entre les mots-clefs et la réponse candidate, la longueur du plus long chemin équivalent entre la question et le passage candidat pour chaque mots-clefs, le ratio par rapport à la taille du chemin de la question, le maximum, le minimum et la moyenne, le nombre de dépendances à insérer pour obtenir des chemins équivalents pour chaque mots-clefs ainsi que la somme pour chaque type, et enfin la somme des dépendances dans le plus long chemin équivalent de chaque mots-clefs. Là aussi, les valeurs sont binarisées.

L'exemple 3.2 illustre le fonctionnement de cette approche. La question *"Where was Tenzin Delek arrested ?"* est annoté avec des dépendances syntaxiques, ainsi que le passage *"The case of Tenzin Delek Rinpoche was raised with me by several of my constituents in Scotland"*. *Scotland* est une des réponses candidates proposées par les heuristiques. Le label *ROOT* correspond au verbe principal de la phrase et de la question, qui est ensuite transformé en VC (VC correspond à *Verbal Chunk*, un groupe verbal). Le chemin entre le candidat réponse et l'élément de la question *Tenzin Delek* est simplifié en enlevant certains labels fréquents, comme les modifieurs de nom par exemple. En comparant le chemin simplifié de la question avec celui du passage, on peut voir que ce dernier contient une relation *LGS* supplémentaire. Cette dépendance représente le sujet logique d'un verbe à la forme passive. Cela signifie que *Scotland* modifie une phrase nominale qui a une relation syntaxique

avec le verbe principal qui n'est pas un modifieur de lieu. De ce fait, *Scotland* n'est pas forcément un lieu associé à Tenzin Delek.

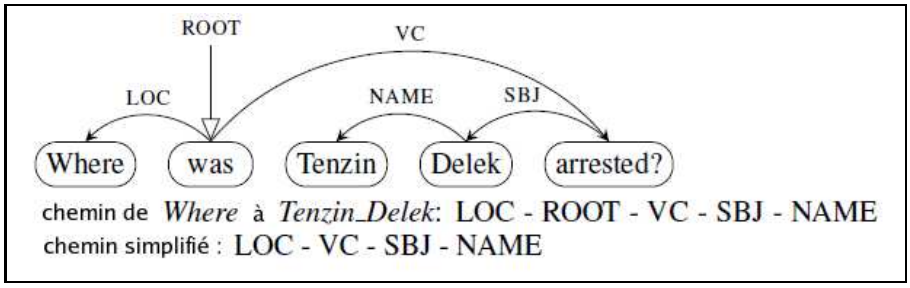

FIG. 3.2 – Dépendances existantes pour la question *"Where was Tenzin Delek arrested ?"* ; exemple tiré de [Comas et al. 2010]

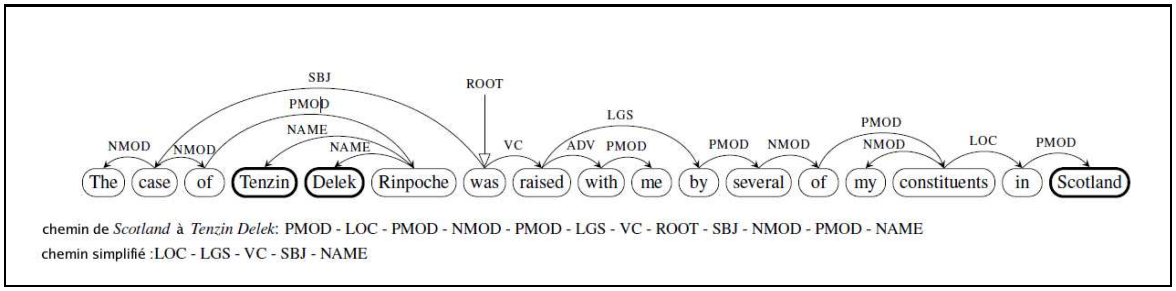

FIG. 3.3 – Exemple des dépendances existantes pour le passage *"The case of Tenzin Delek Rinpoche was raised with me by several of my constituents in Scotland"* ; exemple tiré de [Comas et al. 2010]

Ces deux approches de réordonnancement ont été évaluées sur les corpus de développement et de test de la campagne QAst 2009 [Turmo et al. 2009]. Le modèle a été entraîné tout d'abord sur le corpus de développement composé de 50 questions. Les résultats obtenus sont en progrès : le module d'extraction de réponses obtient un MRR de 0.301 et une précision de 20.0%, tandis que le réordonnanceur utilisant les traits basés sur les dépendances syntaxiques obtient un MRR de 0.369 et une précision de 29.33%. Le corpus de développement étant très petit, une deuxième expérimentation a été faite, où un modèle a été généré pour chaque question de test à partir d'un corpus composé des questions de développement et celles de test exceptée la question elle même. Les résultats obtenus sont bien évidemment meilleurs : le MRR est de 0.425 et la précision de 36.0%. L'approche employée par l'UPC est intéressante : elle montre qu'il est possible d'employer une analyse syntaxique dans le cadre de l'oral. Il est néanmoins nécessaire d'adapter l'analyse employée, pour pouvoir l'appliquer sur des transcriptions de l'oral. Par contre, si l'utilisation d'un classifieur semble efficace, il nécessite un corpus d'apprentissage important pour être vraiment efficace, comme le prouve la hausse des résultats entre les deux expérimentations. Un corpus de 50 questions est trop limité.

3.1.3 Utilisation de rôles sémantiques pour l'extraction des réponse : QASR

QASR [Stenchikova et al. 2006] est un système de questions-réponses en domaine ouvert cherchant les réponses à partir du web. Les auteurs utilisent Assert [Pradhan, et al. 2005], un annotateur de rôles sémantiques. Les rôles sémantiques correspondent à la relation existant entre un prédicat et un constituant syntaxique. Les rôles sémantiques typiques incluent par exemple l'agent, le patient ou encore l'instrument. Par exemple, dans *Johan ouvre la porte*, le prédicat est *ouvre*, et il est associé à deux rôles sémantiques : *Johan*, qui est l'agent, et *la porte*, qui est le patient. Les rôles sémantiques fournis par Assert sont utilisés dans l'extracteur de réponses pour trouver les bonnes réponses à des questions. Les auteurs se sont limités aux questions factuelles, et ce système a été développé pour l'anglais. QASR n'utilise pas de module de réordonnancement. Néanmoins, l'utilisation d'une analyse fondée sur des rôles sémantiques pour les différents modules du système nous semble intéressant dans notre contexte. Nous estimons notamment que l'emphase mise sur les prédicats verbaux et les relations avec leurs constituants est un élément important pour représenter le sens d'une phrase. Par ailleurs, QASR est appliqué sur un corpus du web, ce qui permet d'évaluer sa robustesse face à ce type de documents.

QASR utilise Google pour extraire du web les documents contenant potentiellement la bonne réponse. Pour ce faire, deux types de requêtes sont créées : une requête *exact*, où le marqueur interrogatif est enlevé et la structure de la question est transformée en une phrase affirmative, et une requête *inexact*, où Assert est utilisé pour identifer le prédicat de la question et ses rôles sémantiques (arguments). Par exemple, pour la question *When did Bell invent the telephone ?*, nous avons la requête *exact Bell invented the telephone* et la requête *inexact Bell AND invented AND the telephone*. Ces requêtes sont ensuite fournies à Google, et une liste de documents est retournée. Les documents sont segmentés en phrase en utilisant un outil développé dans le cadre du système de questions-réponses AnswerBus [Zheng 2002]. Les phrases candidates sont ensuite sélectionnées. Pour les requêtes de type *exact*, les phrases contenant la requête sont sélectionnées. Pour les requêtes de type *inexact*, les phrases contenant le prédicat de la requête sont sélectionnées.

Les marqueurs interrogatifs des questions sont ensuite identifiés puis analysés par le biais d'heuristiques et d'un classifieur de question pour déterminer le type de la question. Chaque question contient un prédicat, prédicat qui est aussi présent dans les phrases candidates. Chaque argument du prédicat dans ces phrases est considéré comme une réponse candidate à partir du moment où l'argument est du type attendu. Les candidats possibles sont extraits, et un score est attribué en fonction des occurences de chaque candidat dans le reste des phrases. Les auteurs se servent par ailleurs de la redondance inhérente au web pour donner un plus grand score aux réponses les plus fréquentes.

La question *"When did Bell invent the telephone ?"* illustre la sélection des candidats réponses. Le prédicat est *invented*. Une des phrases retournées est *"The telephone was invented by Bell in 1876.*". Ici, le prédicat *invented* a trois arguments : *the telephone*, *by Bell*, et *in 1876*. L'argument *in 1876* étant un rôle sémantique de type temps, il sera choisi comme réponse candidate par le système.

Une méthode de base a été adoptée pour des besoins de comparaison. Cette méthode n'est appliquée

que dans le cas de requêtes de type *exact*. Pour chaque phrase candidate, les candidats réponses sont extraits d'un côté ou de l'autre du contexte de la requête. Des heuristiques permettent selon le type de la question de déterminer de quel côté il faut extraire le candidat réponse. Ces réponses ont un score qui est ensuite attribué lui aussi selon la redondance dans l'ensemble des documents. Par exemple, pour la question *Who invented the silly paddy ?*, la requête exacte est *invented the silly paddy*. La réponse candidate sera composée de tous les mots du début de la phrase jusqu'à la requête, car le marqueur interrogatif est *Who*.

Ce système a été évalué sur un sous-corpus de l'évaluation TREC-9 [Voorhees 2000], composé de 190 questions, ce qui rend difficile la comparaison avec d'autres systèmes. Néanmoins, les résultats obtenus avec l'approche utilisant les rôles sémantiques sont meilleurs que ceux obtenus par l'approche standard : cette dernière obtient une précision de 19% et un MRR de 0.24, contre 24% et 0.29 pour l'approche utilisant les rôles sémantiques et les requêtes de type *exact*. Par ailleurs, en appliquant l'approche par rôle sémantique sur les requêtes de type *inexact* pour chaque question sans réponse, la précision monte à 30% et le MRR à 0.35. Les différences de performances entre l'approche standard et celle s'appuyant sur les rôles sémantiques montrent bien l'intérêt d'une telle analyse. Néanmoins, l'annotateur n'est utilisable que sur l'anglais. De plus, le fait de ne considérer que les arguments d'un prédicat comme candidats réponses semble limiter une telle approche, particulièrement dans le cas de phrases où l'information est dispersée.

3.1.4 Noyaux syntaxiques et sémantiques pour l'extraction de réponses dans le cadre du système YourQA

Les auteurs du système YourQA [Quarteroni & Manandhar 2009] de l'Université de York présentent une approche [Moschitti & Quarteroni 2010] basée principalement sur des noyaux syntaxiques et sémantiques d'arbres afin de classer les réponses potentielles à une question, puis de les réordonner. YourQA est un système de questions-réponses basé sur le web pouvant répondre à des questions factuelles et non factuelles. L'approche employée par les auteurs a plusieurs caractéristiques intéressantes pour notre travail. YourQA s'appuie sur un module de réordonnancement, et le système est appliqué sur le web. Par ailleurs, les auteurs utilisent différentes représentations de l'information, comme par exemple des arbres syntaxiques ou des prédicats sémantiques. Nous décrivons dans cette section le fonctionnement général du système de questions-réponses, puis les noyaux syntaxiques et sémantiques employés par le module de réordonnancement. Le fonctionnement du réordonnanceur étant en lui même très simple, nous concentrons notre présentation sur ces noyaux.

YourQA a trois phases de traitements : l'analyse de la question, la sélection des documents, et l'extraction de la réponse. Pour la première phase, YourQA détermine si la question est de type factuelle ou autre à partir d'une taxonomie de questions. Une fois le type attendu de réponse estimé, une requête est fournie au moteur de recherche, et les *n* premiers documents sont récupérés et segmentés en phrases. Enfin, chaque phrase est comparée à la requête en utilisant une mesure de similarité basée des critères syntaxiques, sémantiques et lexicaux, et les réponses identifiées sont extraites puis ordon-

nées. Nous ne détaillons pas ces mesures dans cette section, pour nous concentrer sur les approches à base de noyaux d'arbres présentées dans [Moschitti & Quarteroni 2010].

A partir de l'approche standard de YourQA, l'objectif des auteurs est d'étudier l'impact de noyaux pour utiliser des structures syntaxiques et sémantiques. L'idée est d'utiliser cette approche pour effectuer un apprentissage relationnel entre les questions et les réponses candidates. Pour rappel, les méthodes à base de noyaux font partie d'une large classe d'algorithmes d'apprentissage, dont les *Support Vector Machines* (SVMs) sont l'un des représentants les plus connus. SVM est un classifieur qui s'appuie sur une fonction de noyaux mesurant en quelque sorte la similarité entre différents objets à classer. Dans [Moschitti & Quarteroni 2010], les auteurs proposent des noyaux permettant de construire un classifieur d'arbres à partir d'un corpus d'apprentissage constitué d'exemples binaires (oui, non). Trois noyaux linguistiques sont proposés : les noyaux de chaînes de caractères et les noyaux d'arbres. Dans cette section nous nous intéressons à cette dernière catégorie.

Dans [Moschitti & Quarteroni 2010], les auteurs citent trois types de noyaux d'arbres : les noyaux d'arbres syntaxiques [Collins 2009], les noyaux d'arbres sémantiques superficiels [Moschitti, et al. 2007], et les noyaux d'arbres partiels [Moschitti 2006]. Nous n'allons détailler dans cette section que le fonctionnement des noyaux d'arbres partiels, ces derniers donnant les meilleurs résultats dans les expérimentations effectuées par les auteurs. Les noyaux d'arbres partiels sont une variante des noyaux d'arbres syntaxiques. Pour les noyaux d'arbres syntaxiques, les opérations de comparaisons s'effectuent sur des fragments d'arbres syntaxiques. Ces derniers sont formés à partir de n'importe quel sous-arbre de l'arbre syntaxique à condition qu'aucune règle grammaticale ne soit brisée. Par exemple, l'arbre *[VP [NP[N D]]]* a comme sous-arbre syntaxique *[NP [N D]]*, mais pas *[NP [N]]*. Pour les noyaux d'arbres partiels, les opérations de comparaison s'effectuent sur des fragments d'arbres partiels, qui correspondent à n'importe quel sous-arbre. Dans l'exemple précédent, *[NP [N]]* est un fragment d'arbre partiel.

A partir de ces noyaux d'arbres partiels, l'objectif des auteurs est de les utiliser pour identifier les relations sémantiques entre une question et un texte contenant une réponse candidate. La tâche de Semantic Role Labelling [Carreras & Màrquez 2005] décrite précédemment propose un formalisme de représentation des prédicats et de leurs arguments basé sur PropBank et FrameNet. Les auteurs ont construit leur propre annotateur de rôles sémantiques [Moschitti, et al. 2005], et les prédicats et les arguments identifiés sont convertis sous la forme d'arbres sémantiques superficiels. Par exemple, pour la phrase *"John likes apples"*, le prédicat identifié est *likes* et les arguments sont *John* et *apples*, avec comme rôles sémantiques *agent* (A0) et *thème* (A1). Le prédicat et ses arguments sont ensuite convertis en un arbre pour pouvoir être utilisés par des noyaux d'arbres partiels. Dans cet exemple, l'arbre aurait cette forme : *[[A0 [John]] [pred [likes]] [A1 [Apples]]]*.

Cette approche, que l'on nommera PTK (pour Partial Tree Kernel), a été évaluée sur deux corpus : TREC-QA, tiré de la dernière version du corpus AQUAINT [1] avec Lucene comme moteur de recherche, et WEB-QA, en utilisant Google directement sur le web. Les questions des deux corpus sont constituées à partir de questions de corpus de l'évaluation TREC. Outre les noyaux d'arbres

[1] trec.nist.gov/data/qa

partiels représentant les prédicats sémantiques, d'autres noyaux ont été mis en place, de manière à comparer les différentes approches ainsi que les combinaisons les plus efficaces : approche par sac de mots (BOW pour Bag Of Words) et comparaison des Parties du Discours (POS pour Part Of Speech) en utilisant des noyaux linéaires, comparaison de chaînes de caractères (WSK pour Word Sequence Kernel) et de chaînes de labels Parties du Discours (POS_{SK}) en utilisant des noyaux de séquence, comparaison d'arbres syntaxiques fournis par l'analyseur Charniak [Charniak 2000] en utilisant des noyaux d'arbres syntaxiques (STK pour Syntactic Tree Kernel), et des noyaux d'arbres sémantiques superficiels représentant eux aussi les prédicats sémantiques fournis par l'annotateur de rôles sémantiques de YourQA (SSTK pour Shallow Semantic Tree Kernel). Le classifieur obtient les meilleurs résultats en utilisant une combinaison de POS_{SK}, de STK et de PTK, aussi bien sur WEB-QA que TREC-QA.

Ce classifieur a donc été évalué dans le cadre du réordonnancement de réponses. YourQA retourne une liste de réponses candidates à une question à partir de la méthode standard, que le classifieur prend en entrée. Ce dernier évalue la paire réponse-question avec le passage d'où a été extraite la réponse. Si la réponse est classée comme positive, le classifieur passe à la réponse candidate suivante. Sinon, la réponse descend d'un rang, jusqu'à ce qu'une nouvelle réponse soit jugée comme étant négative. Cette approche augmente significativement les résultats obtenus, avec une augmentation du MRR de 0.303 à 0.342 sur TREC-QA, et de 0.562 à 0.811 sur WEB-QA. Ces résultats montrent donc qu'une telle approche est efficace pour traiter des documents issus du web. Néanmoins, ce système étant appliqué sur l'anglais, il fait appel à des ressources (FrameNet et PropBank) non disponibles. De plus, les auteurs utilisent des arbres syntaxiques qui sont difficilement applicables sur de l'oral. Enfin, l'utilisation d'un tel classifieur nécessite d'avoir un corpus d'apprentissage conséquent.

3.1.5 Implication textuelle par distance d'édition : le système EDITS

Le système du FBK, EDITS [Kouylekov & Negri 2010], s'appuie sur l'implication textuelle pour trouver la réponse à une question. Ce système n'utilise pas de module de réordonnancement. Néanmoins, nous estimons que la méthode utilisée pour extraire les réponses est très intéressante : l'implication textuelle permet de déterminer si le sens d'un extrait de texte implique celui d'un autre extrait. Ce type d'approche est applicable aux systèmes de questions-réponses [Hickl et al. 2006c ; Hickl et al. 2007] : l'idée est par exemple de déterminer si un passage contenant une réponse candidate implique le même sens que la question traitée. On peut aussi citer le système COGEX [Moldovan, et al. 2003], lui aussi du LCC, qui intègre un prouveur logique pour trouver la bonne réponse. Dans cette section, nous présentons d'abord plus en détail le domaine de l'implication textuelle. Nous présentons ensuite le fonctionnement général du système de questions-réponses, et son module de sélection et d'extraction de réponses.

L'implication textuelle (Textual Entailment) est un domaine du traitement des langues qui a pris beaucoup d'importance ces dernières années. L'objectif est de déterminer pour deux extraits de texte, respectivement appelés *Texte* (T) et *Hypothèse* (H), si H implique le sens de T, comme dans l'exemple suivant :

T : "Henri IV a été assasiné par Ravaillac."
H : "Henri IV est mort en 1610."

La tâche RTE [Bentivogli et al. 2009] (Recognizing Textual Entailment) a pour objectif d'évaluer les approches proposées sur un corpus de paires H/T où les systèmes doivent déterminer s'il y a implication textuelle. Différentes approches sont utilisées pour traiter l'implication textuelle : l'utilisation de méthodes par apprentissage, avec comme des traits syntaxiques et sémantiques [Hickl, et al. 2006a], des approches avec une analyse sémantique très profonde [Fowler, et al. 2006], basée sur des ressources linguistiques importantes, ou encore des techniques calculant un score de similarité à l'aide d'analyse lexicale ou syntaxique [Adams 2006]. Chaque approche a des avantages et inconvénients : les approches basées sur des analyses très profondes permettront de mieux identifier certains phénomènes linguistiques complexes, mais seront souvent limitées à une langue ou à un domaine particulier, au contraire de certaines approches avec une analyse plus superficielle.

Le système du FBK propose une approche basée sur un score de distance entre le texte et l'hypothèse. La possibilité d'avoir une implication textuelle est donc inversement proportionnelle au score obtenu. Le fonctionnement du système peut être décrit en 4 composants : l'algorithme de calcul de la distance, les fichiers de définition des coûts, l'algorithme d'optimisation, et les règles permettant de déterminer la probabilité d'une implication ou une contradiction entre des éléments du texte et de l'hypothèse.

Les auteurs proposent plusieurs algorithmes pour calculer la distance entre l'hypothèse et le texte. Deux algorithmes de distance d'édition sont proposés. L'idée est qu'un certain nombre d'opérations d'édition sont nécessaires pour transformer l'hypothèse (H) en texte (T). Trois types d'opérations sont proposés : la substitution, l'insertion et la suppression. Chaque opération a un coût associé, qui est décrit par les fichiers de définition des coûts. Les deux algorithmes sont les suivants : distance d'édition par composants, et distance d'édition d'arbre. La distance d'édition par composants est une variante de l'algorithme de distance de Levenshtein, où les opérations sont appliquées sur des composants (tokens). L'algorithme de distance d'arbre est une implémentation de celui décrit dans [Zhang & Shasha 1990] où les opérations sont cette fois appliquées sur des noeuds des arbres syntaxiques représentant l'hypothèse et le texte. Ces deux algorithmes s'appuient sur une analyse syntaxique choisie par l'utilisateur (par exemple TreeTagger [Schmid 1994]). Par ailleurs, cinq autres algorithmes de similarité lexicales/syntaxiques sont proposés. Ils sont adaptés de manière à être applicables avec les opérations d'édition et à fournir un score de distance.

Nous donnons un exemple d'illustration de ces opérations d'édition avec le texte et l'hypothèse suivants :

– *T : "Henri IV a été assassiné par Ravaillac."* ;
– *H : "Henri IV est mort en 1610."*.

Si nous prenons le cas de l'algorithme de distance d'édition par composants, on se retrouve avec 5 composants pour le texte (*Henri IV, a été, assassiné, par* et *Ravaillac*) et 5 composants pour l'hypothèse *Henri IV, est, mort, en* et *1610*. L'idée va être d'effectuer une suite d'opérations d'édition pour

transformer l'hypothèse en la question. Si *Henri IV* est présent dans le texte et l'hypothèse, ce n'est pas le cas pour les autres composants. Il faut ainsi appliquer deux opérations de suppression sur *en* et *1610*, et deux opérations d'insertion sur *par* et *Ravaillac*. Enfin, deux opérations de substitution sont appliquées, respectivement entre *est* et *a été*, et entre *mort* et *assassiné*. Chacune de ces opérations d'édition a un coût associé, qui est décrit dans le paragraphe suivant.

Les coûts des opérations sont fixés dans des fichiers de définition. Ces fichiers de définition de coûts, qu'ils soient ceux fournis par défaut ou ceux définis par l'utilisateur, ont des coûts fixés empiriquement. Les auteurs proposent donc un algorithme génétique [Mehdad 2009] pour optimiser les coûts définis. Les coûts sont convertis en paramètres, et l'algorithme itère sur les données d'apprentissage en utilisant différentes valeurs pour les paramètres jusqu'à ce qu'un ensemble optimal ait été trouvé. Enfin, un ensemble de règles est fourni au système. Ces règles permettent de connaître la probabilité d'une implication ou d'une contradiction entre des éléments du texte et de l'hypothèse. Ces règles peuvent être réutilisées dans les fichiers de définition des coûts. Elles ont été définies à partir de ressources lexicales utilisées régulièrement dans RTE : WordNet [Press 1998], VerbOcean [Chklovski & Pantel 2004], et les dictionnaires de similarité entre mots de Lin [Lin 2000]. Une règle possible pourrait ainsi définir la probabilité d'implication entre *mort* et *assassiné*.

EDITS a participé a plusieurs éditions de la campagne RTE. Il a notamment obtenu une précision de 60.17% à la cinquième édition [Bentivogli, et al. 2005], la moyenne des participants étant de 60.36%. EDITS a aussi été évalué sur la tâche RTE de l'évaluation EVALITA [Cabrio, et al. 2009] pour l'italien, où les résultats sont bons : le système se classe premier avec une précision de 71.0%. Par ailleurs, EDITS a été adapté pour être utilisé dans le cadre d'un système de questions-réponses [Negri, et al. 2008]. Les résultats obtenus sur un corpus dans un domaine spécifique (événements culturels) sont positifs : la bonne réponse est trouvée pour 83% des 400 questions. Ces résultats montrent l'intérêt de l'implication textuelle pour les systèmes de questions-réponses. Par ailleurs, l'utilisation d'opérations d'édition nous semble adaptable à n'importe quel type de documents. Il faut néanmoins garder à l'esprit que ces résultats sont obtenus dans un domaine non ouvert.

3.1.6 Conclusions préliminaires

Les approches présentées montrent qu'il n'y a pas vraiment de méthode standard à adopter lorsqu'il s'agit d'extraire les bonnes réponses à une question (ou de les réordonner). Tout va dépendre du type de problèmes que l'on veut traiter, et du contexte de travail dans lequel on veut s'inscrire. Par exemple, le système FIDJI utilise une approche fortement linguistique, qui donne de bons résultats sur un corpus tiré web. En revanche, l'analyse syntaxique utilisée n'est pas du tout adaptée lorsqu'il s'agit de traiter des données issues de l'oral.

On peut néanmoins constater que le modèle de représentation de l'information contenu dans les questions et documents a une influence significative sur la méthode retenue et joue un rôle prépondérant dans chaque approche présentée. Ainsi, un système comme celui de l'UPC, qui est confronté au problème de traiter des documents issus de l'oral, utilise un annotateur interne adapté à l'oral pour

annoter l'information syntaxique. La représentation des informations syntaxiques des documents et questions est primordiale dans la mise en oeuvre de leur système d'extraction et de réordonnancement de réponses.

Etant donné l'importance de la représentation des informations contenues dans un document ou une question, nous procédons à une étude de différentes analyses dans la section suivante. L'objectif est d'étudier les caractéristiques de ces analyses pour la suite de notre travail.

3.2 Modèles de représentation des questions et des documents

Différentes approches pour l'extraction et le réordonnancement de réponses à une question ont été présentées dans la section 3.1. Ces approches s'appuient sur une représentation des questions et des documents qui peut être très variée selon le système et les choix effectués : analyse syntaxique, analyse sémantique, représentation des données sous forme d'arbres ... La représentation choisie dépend aussi des ressources linguistiques disponibles dans la ou les langues traitées. Le choix d'un formalisme de représentation des questions et documents est donc critique à l'élaboration d'une méthode de réordonnancement des questions. L'idée générale de notre travail pour l'élaboration d'un nouveau formalisme est avant tout d'avoir une meilleure représentation de la structure des questions et des phrases au sein des documents. Il faut de plus que ce formalisme puisse être robuste quel que soit le type de données traité.

Pour représenter cette structure, une approche possible est d'ajouter des relations syntaxiques et sémantiques entre les mots ou groupes de mots. Les travaux se rapportant à l'annotation en rôle sémantique (SRL) [Carreras & Màrquez 2005], plus particulièrement les travaux concernant les systèmes de questions-réponses utilisant ces rôles sémantiques (par exemple [Stenchikova et al. 2006]), s'intéressent aux relations sémantiques. Deux observations peuvent être faites sur ces points : d'une part la majorité des travaux présentés concernent l'anglais ce qui implique l'utilisation de ressources pas forcément accessibles en français (ProbBanks, FrameNet ...), et d'autre part beaucoup concernent l'écrit et s'appuient sur une analyse syntaxique complexe dont nous ne disposons pas. Cependant, certains comme [Sakai, et al. 2004] ont montré que le SRL pouvait s'appuyer sur de simples segments.

Un autre aspect important est que nous traitons des données très variées : la robustesse face à des données journalistiques, issues du Web, ou orales, est une contrainte forte de notre travail. Or les analyseurs syntaxiques ne sont pas nécessairement bien adaptés à ces données. Une des conclusions de [Paroubek et al. 2008b] qui décrit la campagne d'évaluation Easy consacrée aux analyseurs syntaxiques du français est que ces analyseurs ne sont pas bien adaptés pour traiter de l'oral. Enfin, l'analyse de notre système Ritel fournit des informations sémantiques utiles mais peu structurantes. Les types de Ritel peuvent néanmoins servir d'indice pour procéder à une segmentation et permettre de typer ces segments. Une segmentation typée de questions et d'énoncés nous permettra par la suite d'améliorer notre représentation en y ajoutant une information structurelle. La définition du chunking peut être très variable selon l'utilisation que l'on veut en faire. On peut notamment citer la tâche

Chunking de ConLL2000 [Tjong, et al. 2000] et la campagne EASY [Paroubek et al. 2008a].

Une segmentation en composants typés des phrases des documents et questions semble donc prometteur dans la mise en place d'un nouveau formalisme de représentation des documents. Dans les sections suivantes, nous décrivons successivement des travaux concernant la segmentation de phrases et de questions en groupes de mots typés, puis concernant l'ajout des relations entre les groupes de mots.

3.2.1 Segmentation et annotation de groupes de mots

La tâche *Chunking* [Tjong et al. 2000] de CoNLL 2000 porte sur l'analyse de textes journalistiques en anglais. Le but de cette tâche est de diviser des phrases en groupes de mots (chunks). Ces ensembles sont créés de manière à ce que les mots contenus à l'intérieur soient liés syntaxiquement. Onze types de chunks sont utilisés : nominaux, verbaux, prépositions, adverbes, adjectifs, prépositions multiples, conjonctions, particules, listes et conjonctions et disjonctions de mots. Trois familles d'approches ont été proposées : à base de règles, apprentissage par mémorisation (instance-based learning), statistiques. En outre une méthode par combinaisons d'approches a été proposée [Kudoh & Matsumoto 2000] qui a obtenu les meilleurs résultats (f-mesure de 0,93). Par la suite, l'utilisation de champs conditionnels aléatoires [Lafferty, et al. 2001] a été proposée pour cette même tâche et les modèles appris ont permis d'obtenir un très bon résultat (0,93 de f-mesure), équivalent au meilleur de la campagne. Différents traits sont utilisés pour détecter les bornes des segments. Il s'agit le plus souvent de la ponctuation et des parties du discours. La taille du contexte utilisé varie de 2 à 3 mots précédents et suivants pour chaque mot observé. En moyenne, la f-mesure obtenue par les autres systèmes est de 0,91, le moins bon résulat étant de 0,85.

Pour le français de manière similaire, la campagne EASY [Paroubek et al. 2008a] propose une analyse en constituants. L'annotation en relation permet ensuite d'établir les liens entre les constituants. Six types de constituants sont utilisés : le groupe nominal, le groupe prépositionnel, le noyau verbal, le groupe adjectival, le groupe adverbial et le groupe verbal introduit par une préposition. Une quinzaine de systèmes ont été évalués sur différents corpus de données, la meilleure F-mesure obtenue sur tous les corpus confondus étant de 0,89. Plus spécifiquement, les meilleurs résultats obtenus sur les corpus nous intéressant directement sont : 0,92 pour les documents journalistiques (le Monde), 0,93 pour les questions, 0,92 sur le web (type wikipédia) et 0,79 sur l'oral. L'objectif d'EASY est de faire une analyse en constituants et en relation équivalente à une analyse syntaxique.

3.2.2 Relations entre groupes de mots

Les systèmes de questions-réponses présentés dans la section 3.1 s'appuient sur différentes analyses permettant de représenter l'information des questions ou des phrases des documents. Ces analyses ont chacune des caractéristiques différentes, mais permettent généralement de représenter des relations

entre groupes de mots, en utilisant par exemple des dépendances syntaxiques ou des relations sémantiques. Nous présentons dans les sections suivantes les représentation de l'information utilisées par ces analyses [Aït-Mokhtar et al. 2002 ; Pradhan et al. 2005 ; Lluis et al. 2009] ainsi qu'un formalisme proposé dans [Vergne 1999].

3.2.2.1 XIP, un analyseur de dépendances syntaxiques

Les relations typées entre mots ou groupes de mots sont souvent des composantes importantes de systèmes de questions-réponses. Ces relations peuvent aussi bien être d'ordre syntaxique que sémantique. Pour les relations syntaxiques, les systèmes de questions-réponses ont tendance à s'appuyer sur des analyseurs déjà existants, produisant par exemple des arbres de dépendances. On peut par exemple citer un analyseur comme Charniak [Charniak 2000] pour l'anglais. Le défaut de tels analyseurs est que s'ils sont bien adaptés pour traiter des textes issus de corpus journalistiques, leur application sur d'autres types de documents, particulièrement oraux, pose des problèmes.

Néanmoins, l'analyseur XIP [Aït-Mokhtar et al. 2002] a été utilisé avec succès sur un corpus tiré du web, dans le cadre du système de questions-réponses FIDJI [Tannier & Moriceau 2010], présenté dans la section 3.1.1. L'objectif de cet analyseur est d'extraire les dépendances syntaxiques de façon robuste. XIP propose un formalisme de règles que l'on peut classer en deux catégories : des règles pour construire des syntagmes noyaux, et des règles pour construire les dépendances entre ces noyaux. Les syntagmes noyaux peuvent être vus comme des segments (chunks) typés. L'utilisation de XIP dans FIDJI [Tannier & Moriceau 2010] a donné de très bons résultats sur le corpus web de QUAERO [Quintard et al. 2010].

La figure 3.4 illustre l'annotation effectuée par XIP sur la phrase "*The Chunking rules define and produce a chunk tree*". Les règles de chunking génèrent cet arbre, et ensuite un ensemble de règles va permettre de générer les dépendances de cet arbre. Dans ce cas, deux dépendances sont déduites, *SUBJ(define,rule)* et *VCOORD(define, produce),* qui correspondent respectivement à une relation sujet entre le verbe *define* et le nom *rule,* et une conjonction de coordination entre les deux verbes *define* et *produce*. A partir de ces deux dépendances, une nouvelle est déduite : *SUBJ(produce,rule),* entre le verbe *produce* et le mot *rule*.

Les dépendances syntaxiques identifiées par XIP permettent de représenter les relations entre les groupes de mots. C'est ce type d'information que nous voulons utiliser dans notre travail. Néanmoins, cet analyseur ne semble pas adapté pour traiter de l'oral, contrairement aux données du web. L'idée serait donc d'avoir une approche similaire, mais adaptée à l'oral.

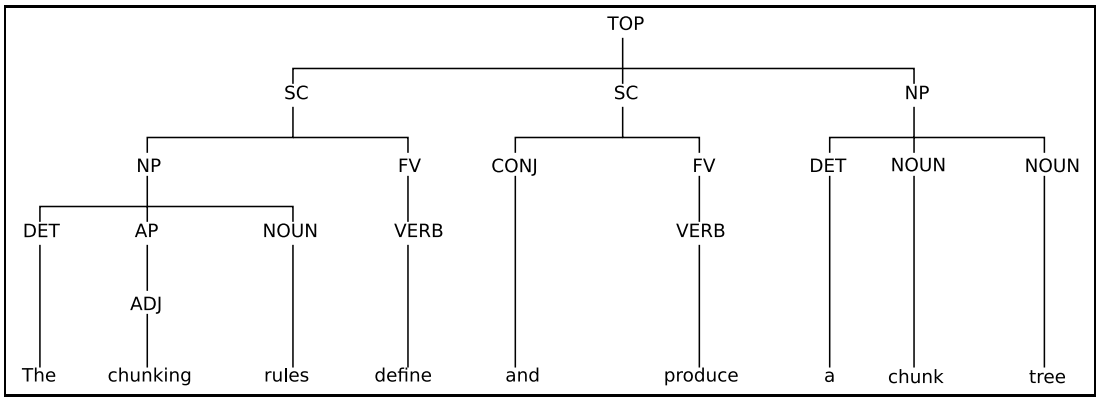

FIG. 3.4 – Exemple d'arbre de chunking généré par XIP [Aït-Mokhtar et al. 2002] pour la phrase *"The Chunking rules define and produce a chunk tree."*

3.2.2.2 Assert, un annotateur de rôles sémantiques

Pour l'ajout de relations sémantiques, beaucoup de systèmes utilisent leur propre analyseur. L'ajout de relations sémantiques a ainsi été évalué par le biais de la tâche Semantic Role Labelling des éditions 2004 et 2005 de CoNLL [Carreras & Màrquez 2005]. L'objectif de cette tâche est d'annoter les différents constituants de chaque verbe de chaque phrase. Le formalisme était imposé et provenait de PropBank [Palmer et al. 2005]. En plus des constituants d'un verbe, les participants devaient aussi annoter des arguments optionnels (disjonction, cause ...), toujours selon le formalisme de PropBank. Un exemple d'annotation de rôles sémantiques est décrit dans la figure 3.5 pour la phrase *He wouldn't accept anything of value from those he was writing about*. Le prédicat verbal **accept** et ses différents constituants y sont représentés.

[$_{A0}$He] [$_{AM-MOD}$would] [$_{AM-NEG}$n't] [$_{V}$**accept**] [$_{A1}$anything of value]
from [$_{A2}$those he was writing about]

V : verbe
A0 : Agent
A1 : Patient
A2 : Source
AM-MOD : modal
AM-NEG : négation

FIG. 3.5 – Exemple d'annotation en rôles sémantiques pour le prédicat **accept** selon le formalisme de PropBank [Palmer et al. 2005] pour la phrase *"He wouldn't accept anything of value from those he was writing about."*

Dernière obligation, les participants devaient utiliser des approches par apprentissage. La majorité des

participants utilisaient des analyseurs syntaxiques pour traiter dans un premier temps leurs phrases. Les classifieurs majoritairement appliqués sont Max-Ent et SVM, avec comme traits l'information extraites de PropBank et celles de l'analyse syntaxique effectuée : type d'un mot, structure d'une phrase, forme lemmatisée, catégorie des verbes, etc ... 19 systèmes ont participés à l'édition 2005 [Carreras & Màrquez 2005], et les résultats (F-mesure) vont de 0.65 à 0.78.

Le système de questions-réponses QASR [Stenchikova et al. 2006] s'appuie directement sur un module de détection de rôles sémantiques, Assert [Pradhan et al. 2005]. Tout comme les systèmes présentés dans l'édition 2005 de SRL, Assert utilise une approche statistique. Le classifieur utilisé est à base de SVMs. Les traits utilisés sont basés en partie sur les mêmes informations que celles utilisées dans la campagne SRL : lemme, annotation en Parties du Discours, contexte local, position par rapport aux prédicats, structure de l'arbre syntaxique. Là encore, PropBank est utilisé, ainsi que l'analyseur syntaxique Charniak [Charniak 2000]. Par ailleurs, les auteurs s'appuient sur l'annotateur Minipar [Lin 1998]. Ce parseur permet d'identifier les dépendances entre les mots d'une phrase en construisant un arbre de dépendance. Les auteurs utilisent comme analyseur le parseur CCG [Gildea & Hockenmaier 2003] (Combinatory Categorial Grammar). Enfin, Assert utilise un segmenteur basé sur une représentation avec des structures d'arbres [Hacioglu 2004]. Ces arbres contiennent principalement de l'information sémantique, et l'analyseur est entraîné sur PropBank. La combinaison de tous ces analyseurs donne de bons résultats. Assert a été évalué entre autre sur la tâche SRL de CoNLL 2005 [Carreras & Màrquez 2005], où il a été classé second avec une f-mesure de 0.773, contre 0.779 pour le premier. Assert est utilisé dans le cadre du systèmes de questions-réponses QASR, décrit dans la section 3.1.3.

L'annotation des rôles sémantiques des prédicats verbaux permet de représenter les relations sémantiques d'une phrase ou d'une question. Ce type d'annotation est intéressant pour notre travail : il permet ainsi d'identifier les mots en relation dans un passage, et ainsi affiner la recherche d'une bonne réponse. Néanmoins, Assert s'appuie sur un ensemble de ressources linguistiques non disponibles en français, ce qui en rend son utilisation pour notre travail difficile.

3.2.2.3 L'analyseur de dépendances syntaxiques de l'UPC

Le système de questions-réponses de l'UPC [Comas et al. 2010] est confronté aux mêmes problématiques que le système Ritel. Il a notamment participé aux différentes éditions de la campagne QAst (Question-Answering on Speech Transcripts), auxquelles Ritel a aussi participé. L'objectif de cette campagne est d'évaluer des systèmes de questions-réponses sur des transcriptions de documents audios. L'oral ayant des caractéristiques très particulières, les approches classiques ne fonctionnent pas toujours, particulièrement les analyseurs syntaxiques. De ce fait, le système de l'UPC a fait le choix de se baser sur son propre analyseur [Lluis et al. 2009], developpé au sein de l'UPC.

A l'origine, cet analyseur a été développé pour la tâche *Syntactic and Semantic Dependencies in Multiple Languages* CoNLL 2009 [Haji, et al. 2009], et avait pour objectif d'annoter les dépendances sémantiques et syntaxiques d'un corpus de documents. L'analyseur met en oeuvre une amélioration

de l'algorithme de Carreras [Carreras 2007], lui même basé sur l'algorithme d'Eisner [Eisner 1996]. L'architecture de l'analyseur est décomposé en quatre sous-modules : le pré-traitement et l'extraction de traits, le pré-tagage syntaxique, le parsing sémantico-syntaxique, et enfin la classification des prédicats.

Si l'analyseur obtient des résultats encourageants (mais pas encore au niveau des approches état de l'art [Lluis et al. 2009]), il a surtout été entraîné sur un corpus de documents tirés de textes journalistiques. L'analyseur n'a donc pas encore été adapté à la parole. Ainsi, l'utilisation de l'analyseur dans le cadre du système de questions-réponses de l'UPC s'est surtout focalisé sur l'extraction de traits robustes de manière à pouvoir appliquer l'analyseur à l'oral. Seules les dépendances syntaxiques sont utilisées par le système de questions-réponses. Aussi bien les questions que les documents sont annotés par l'analyseur, et chaque paire de mots est reliée par un chemin de dépendance. Le formalisme des dépendances est basé sur celui utilisé pour la tâche Syntactic and Semantic Dependencies in Multiple Languages de CoNLL 2009. Les dépendances sont organisées autour de pivots verbaux (étiquetés *ROOT*), et on retrouve des dépendances classiques comme *modifieurs de nom* (*NMOD*), *modifieur prépositionnel* (*PMOD*), *sujet SBJ* ... Ces dépendances sont ensuite utilisées dans le système de questions-réponses de l'UPC, décrit dans la section 3.1.2.

L'exemple présenté dans la figure 3.6 illustre l'analyse effectuée par le système de l'UPC. La phrase annotée est *"The case of Tenzin Delek Rinpoche was raised with me by several of my constituents in Scotland"*. Le pivot verbal de cette phrase est l'auxiliaire *was* associé au verbe *raised*. On peut ainsi observer la relation sujet *SBJ* entre le nom *case* et l'auxiliaire *was*.

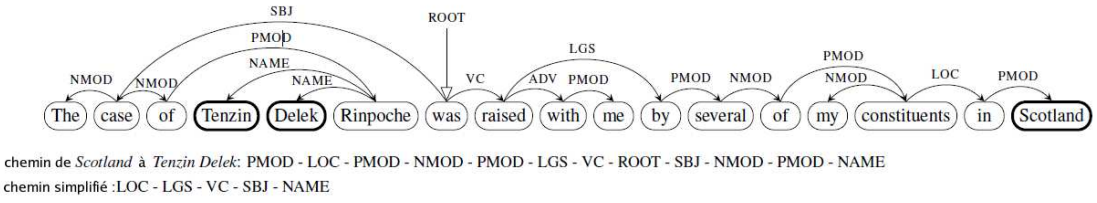

FIG. 3.6 – Exemple des dépendances existantes pour le passage *"The case of Tenzin Delek Rinpoche was raised with me by several of my constituents in Scotland"* ; exemple tiré de [Comas et al. 2010]

L'analyseur en lui même n'est pas utilisable pour notre travail, étant donné qu'il ne traite que l'anglais et l'espagnol. Il est néanmoins intéressant d'observer que le formalisme syntaxique proposé par les auteurs est adaptable à des textes oraux. Les dépendances permettent ainsi de représenter la structure des phrases, tout comme dans l'analyseur XIP. L'adaptation de l'analyseur effectuée par les auteurs pour un cadre oral montre cependant qu'il est nécessaire d'adopter un formalisme syntaxique relativement robuste.

3.2.2.4 Les Syntagmes Non Récursifs

Les travaux présentés dans [Vergne 1999] ne portent pas précisément sur les systèmes de questions-réponses, mais plus sur les analyseurs syntaxiques. Dans le premier chapitre, l'auteur présente ses idées et son approche générale de la représentation des phrases, et introduit plus particulièrement la notion de Syntagmes Non Récursifs (SNR). Les SNRs sont très proches des chunks communément trouvés dans la littérature. De plus, l'auteur modélise des relations de dépendance entre ces segments, qui, si elles ne sont pas typées, se rapprochent des dépendances syntaxiques que l'on peut retrouver dans certains analyseurs. L'analyse est appliquée et les SNRs et dépendances en sont les résultats. Nous nous appuyons dans la suite de cette section sur l'exemple de la figure 3.7, illustrant la représentation de la phrase *"A l'issue de la réunion de son cabinet, le président a déclaré que les combats qui ont débuté au mois de décembre ont provoqué la fuite de nombreaux réfugiés."*.

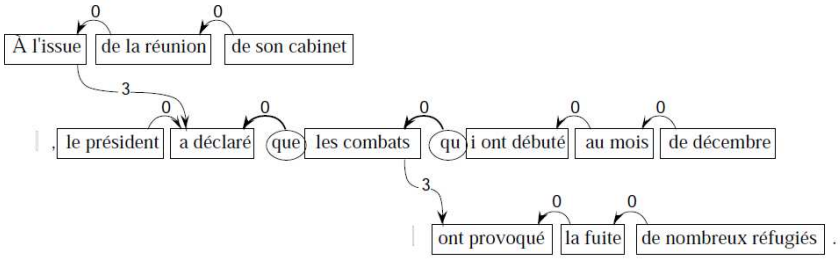

FIG. 3.7 – Exemple des SNRs et des dépendances identifiés dans la phrase *"A l'issue de la réunion de son cabinet, le président a déclaré que les combats qui ont débuté au mois de décembre ont provoqué la fuite de nombreaux réfugiés."* ; exemple tiré de [Vergne 1999]

Les SNRs présentés dans [Vergne 1999] sont par définition constitués d'un élément central, le plus souvent un nom ou un verbe (conjugué, infinitif, participe présent ou passé), entouré éventuellement d'autres éléments. Ainsi, un SNR nominal peut être constitué de conjonctions de coordination et/ou de subordination, prépositions, déterminants, etc ... De même, un SNR verbal peut être constitué d'adverbes, de négations, de pronoms, etc ... Dans la figure 3.7, *de la réunion* est un SNR nominal, et *a déclaré* un SNR verbal.

A partir de cette définition des SNRs, l'auteur propose une représentation particulière des propositions subordonnées. Le régissant de sa hiérarchie interne est son SNR verbal. Le régissant est par ailleurs associé à la conjonction de subordination. La figure 3.7 donne un exemple de ces propositions subordonnées. Pour la proposition *qui ont débuté au mois de décembre*, le régissant est le SNR verbal *ont débuté*, qui est associé à la conjonction de subordination *qui*.

L'auteur décrit aussi des relations de dépendances entre les différents SNRs. Pour des besoins de mesurer la longueur de ces dépendances, l'auteur introduit une métrique, qui est défini par les caractéristiques suivantes :

– l'unité de la métrique est le SNR ;
– la longueur de la dépendance entre deux SNRs est le nombre de SNR les séparant : deux SNRs voisins ont donc une longueur nulle ;
– la longueur d'un groupe de SNRs contigus est le nombre de SNRs qu'il contient ;
– la longueur de la dépendance entre une proposition subordonnée et son régissant est le nombre de SNRs qui séparent ce régissant du début de la proposition subordonnée.

La figure 3.7 donne un apperçu des longueurs des dépendances pouvant exister entre les SNRs. On a ainsi une longueur 0 pour la dépendance entre deux SNRs voisins, par exemple entre *de décembre* et *au mois*. De même, la longueur de la dépendance entre *A l'issue* et *a déclaré* est de 3, car ils sont séparés par trois SNRs : *de la réunion*, *de son cabinet* et *le président*.

Les Syntagmes Non Récursifs sont centrés autour de deux types, nominal et verbal. Cette formalisation est plus simple que celle que l'on peut trouver dans la tâche Chunking de CoNLL 2000 [Tjong et al. 2000] ou dans l'évaluation EASY [Paroubek et al. 2008b]. Nous estimons ainsi qu'elle est potentiellement plus adaptable à des textes oraux. Par ailleurs, les dépendances permettent de représenter une notion de distance entre deux SNRs. Cette formalisation permet de représenter la structure entre les groupes de mots d'une phrase ou d'une question.

3.3 Discussion

La section 3 présente un ensemble de systèmes de questions-réponses. Chacun de ces systèmes a une approche avec des spécificités pour l'extraction et/ou le réordonnancement de réponses. Ces spécificités dépendent en partie des choix effectués lors de la conception de ces systèmes. Ces choix dépendent principalement du contexte d'expérimentation et des objectifs : type de documents, type de questions, langue traitée, ressources linguistiques ... Ainsi, le fait de traiter l'anglais permet par exemple d'avoir accès à des ressources comme FrameNet ou PropBank, donnant accès à des informations potentiellement utiles pour les traitements. A contrario, le français ne dispose pas toujours de ressources équivalentes, ce qui a un impact conséquent sur les traitements mis en oeuvre. Dans la section 3.2, nous avons présenté les modèles de représentation de l'information contenue dans les questions et les documents. Ces représentations permettent aussi très souvent de représenter la structure et les relations entre les différents mots et groupes de mots.

Le système FIDJI [Tannier & Moriceau 2010] du LIMSI obtient de bons résultats sur la campagne d'évaluation Quaero. Cette campagne a par ailleurs la particularité d'être effectuée sur un corpus de documents du web. Cela implique que l'approche d'extraction de réponses utilisée est robuste par rapport à la variabilité que peut avoir la structure des phrases dans des documents du web : certaines pages peuvent être écrites dans une langue bien formée (articles de sites journalistiques), tandis que d'autres peuvent être dans une syntaxe plus relachée (blogs). Par ailleurs, le système fonctionne sur le français. Cependant, les auteurs emploient une analyse syntaxique, fournie par XIP, qui n'est pas adaptée pour traiter n'importe quel type de documents, particulièrement ceux tirés de l'oral. Notre

contexte de travail implique de traiter des documents tirés de l'oral. Si l'utilisation de dépendances syntaxiques semblent être une possibilité pour représenter la structure des questions et des phrases des documents, XIP n'est pas utilisable en l'état.

Le système de l'UPC [Comas et al. 2010] montre qu'il est possible d'utiliser une analyse syntaxique dans le cadre de documents oraux. Les résultats obtenus sont meilleurs que ceux de la baseline. Néanmoins, il a été nécessaire pour les auteurs d'adapter l'analyseur utilisé [Lluis et al. 2009]. A l'origine, cet analyseur a été développé dans le cadre d'une tâche CoNLL 2009 avec comme objectif d'annoter les dépendances syntaxiques et sémantiques. Le problème étant que cet analyseur n'est pas performant pour traiter des transcriptions de l'oral, ce qui explique que les auteurs ont adapté l'analyseur de manière à ne récupérer que les dépendances syntaxiques les plus fiables. L'idée est d'ensuite comparer les chemins de dépendances entre les mots-clefs et la réponse candidate du passage avec ceux de la question. Cette approche permet de comparer les structures et les relations existant entre les mots-clefs, et semble plutôt adaptée à l'oral, comme en attestent les bons résultats obtenus sur l'évaluation QAst 2009. Par contre, ce système ne fonctionne que sur l'espagnol et l'anglais. Ainsi, s'il n'est pas possible de réutiliser directement l'analyseur pour notre travail, il semble intéressant d'utiliser des dépendances syntaxiques typées pour les questions et documents traités. Par ailleurs, la comparaison des chemins de dépendances est une approche intéressante pour comparer la similarité syntaxique entre une question et un passage.

QASR [Stenchikova et al. 2006] est un système de questions-réponses pour la langue anglaise cherchant les réponses à des questions à partir du web. L'approche utilisée par les auteurs s'appuie fortement sur un annotateur de rôles sémantiques. Ce type d'approche est intéressant, car il est clair que des rôles sémantiques permettent de représenter l'information contenue dans des phrases. Par ailleurs, cette annotation est centrée sur la détection des prédicats et de ses arguments. Les arguments des prédicats sont considérés comme réponses candidates à une question. Si cette approche donne des résultats encourageants sur TREC-9, on peut néanmoins noter deux problèmes par rapport à notre contexte de travail. D'une part, ce type d'annotateur n'est utilisable que sur l'anglais, du fait de l'utilisation de FrameNet et ProbBanks. D'autre part, le fait de ne sélectionner que les arguments des prédicats comme réponses candidates semble trop limité, surtout dans le cas de textes avec des phrases n'ayant pas forcément une structure bien formée. Néanmoins, l'idée de centrer les traitements autour des prédicats verbaux d'une phrase est une approche intéressante. Du fait de l'impact structurel d'un verbe, il semble nécessaire que notre travail prenne en compte leur importance.

YourQA [Moschitti & Quarteroni 2010] est lui aussi un système de questions-réponses en domaine ouvert utilisant le web. La langue traitée est l'anglais. Ce système utilise principalement des combinaisons de noyaux pour réordonner les candidats retournés par l'extracteur de réponses. Les noyaux utilisent aussi bien des représentations d'arbres syntaxiques que de prédicats sémantiques transformés en arbre. Les résultats obtenus aussi bien sur TREC-9 que sur un corpus interne montre une augmentation sensible des performances. Néanmoins, par rapport à notre contexte de travail, cette approche présente quelques problèmes. D'une part, l'utilisation des arbres syntaxiques complets n'est pas possible sur des textes issus de l'oral. D'autre part, l'annotateur fournissant les prédicats sémantiques n'est utilisable que sur de l'anglais. En effet, cet annotateur est basé lui aussi sur FrameNet et

PropBanks, qui sont des ressources non disponibles en français. Par ailleurs, une méthode par apprentissage demande d'avoir un corpus conséquent, ce qui n'est pas forcément le cas pour le français.

EDITS [Kouylekov & Negri 2010] n'est pas un système de questions-réponses à proprement parler, mais une méthode de reconnaissance de l'implication textuelle entre deux passages. Les informations permettant de détecter l'implication textuelle sont utiles pour mieux évaluer les réponses-candidates aux questions, ce qui est intéressant pour notre travail. Cette approche a été adaptée à un système de questions-réponses. Les résultats sont encourageants, mais il faut garder à l'esprit que le tout n'a été évalué que sur un domaine fermé. L'approche utilise des opérations d'édition d'insertion, de suppression et de substitution pour calculer la similarité entre les deux passages. Ces opérations sont effectuées aussi bien sur des groupes de mots que des structures d'arbre. Nous estimons qu'une telle approche est adaptable à n'importe quel type de documents de part son fonctionnement. Le fait de se baser sur une distance d'édition où les opérations ont un coût calculé en fonction des composants sur lesquels sont appliqués les opérations peut justement permettre d'avoir des traitements robustes.

Comme nous pouvons le constater, les approches sont diverses, et ont chacune leurs avantages et inconvénients. De ce fait, il convient de rappeler les objectifs de ce travail. L'idée est de proposer une méthode de réordonnancement des réponses candidates robuste quelque soit le type de documents traités. De plus, cette méthode doit fonctionner sur le français. Cette approche est par ailleurs évaluée sur le système Ritel, ce qui amène quelques spécificités : prise en compte du formalisme de représentation des données déjà existant, entre autre. Ces différents points font que les approches citées précédemment ont des caractéristiques souvent difficilement applicables avec notre contexte de travail : on peut notamment pointer les analyseurs utilisés, mais aussi la langue ou les documents traités. Ainsi, certains analyseurs ne sont pas adaptés pour traiter de l'oral, comme XIP [Aït-Mokhtar et al. 2002]. De même, certaines approches s'appuient sur des ressources linguistiques non disponibles en français.

Néanmoins, on peut aussi remarquer que ces différentes approches partagent souvent des points communs, ce qui semble indiquer leur importance dans la conception d'un réordonnanceur. La présence d'une analyse syntaxique, même simplifiée comme dans le système de l'UPC, semble bénéfique pour représenter la structure et les dépendances entre les mots ou groupes de mots. De même, le rôle critique des verbes dans le sens et l'organisation des phrases, comme peut l'illustrer l'utilisation des prédicats sémantiques dans yourQA et QASR.

A partir de ces différentes observations, on peut élaborer certaines hypothèses qui servent de base au module de réordonnancement présenté dans les deux prochains chapitres. Ce module devra s'appuyer sur une représentation de la structure des questions et des documents en plus d'exploiter l'analyse fournie par Ritel. Cette représentation doit être robuste. Il n'existe pas à notre connaissance d'analyseur véritablement adapté à traiter n'importe quel type de documents. Au vu des observations précédentes, il est nécessaire d'avoir néanmoins une représentation des dépendances entre les groupes de mots. Les relations de dépendances permettent de représenter la structure d'une phrase, et semblent primordiales pour structurer l'information contenue dans un passage. Une segmentation typée des phrases en composants est donc nécessaire pour pouvoir mettre en place des dépendances entre ces

composants. A partir de cette représentation, il semble intéressant d'avoir une comparaison structurelle du passage d'une réponse candidate avec la question. Il est pertinent de s'appuyer sur un calcul de similarité prenant en compte ces dépendances. Enfin, le module doit lui-même être robuste. De ce fait, il semble prometteur de s'appuyer sur une distance d'édition, comme dans le système EDITS. A partir de ces observations, un module réordonnancement des candidats réponses peut être défini. Dans les deux prochains chapitres, nous présentons ce travail, que nous divisons en deux composants : le modèle de représentation des documents et question, la méthode de réordonnancement des réponses candidates.

Chapitre 4

Un modèle de représentation robuste des documents et questions

4.1 Présentation

Le présent chapitre a pour but de présenter la première contribution de ce document : un modèle de représentation robuste de la structure des documents et des questions. L'objectif de cette représentation est de représenter l'information structurelle comprise dans les questions et phrases des documents. La deuxième partie de notre travail, le module de réordonnancement présenté dans le chapitre 5, s'appuie sur cette représentation. Ce travail s'inscrit dans le contexte du système de questions-réponses Ritel. Ce système fournit déjà de l'information sémantique par le biais d'un analyseur sur lequel notre module de réordonnancement peut s'appuyer. Par contre, l'information syntaxique fournie actuellement par Ritel n'est pas suffisante pour mettre en place une méthode de réordonnancement, particulièrement pour représenter les relations entre les mots et groupes de mots des phrases. De ce fait, il est nécessaire de proposer un modèle de représentation supplémentaire de l'information structurelle contenue dans les phrases des documents et des questions. Ce modèle doit par ailleurs satisfaire deux contraintes : la robustesse par rapport aux types de documents traités, et l'utilisation du français qui implique le non accès à des ressources telles que PropBank [Palmer et al. 2005] ou FrameNet [Ruppenhofer et al. 2006].

Cette contribution s'appuie sur les conclusions tirées dans le chapitre 3. Les différents systèmes de questions-réponses présentés dans ce chapitre s'appuient sur des analyses syntaxiques, comme par exemple [Comas et al. 2010]. Néanmoins, il n'est pas possible d'utiliser une analyse syntaxique profonde comme dans [Tannier & Moriceau 2010] dans notre contexte de travail, particulièrement à cause du traitement de l'oral. Il est donc nécessaire de mettre en oeuvre un modèle de représentation fournissant de l'information structurelle et adapté à ce contexte de travail. Pour concevoir ce modèle, nous sommes inspirés de certains des travaux présentés dans la section 3.2 du chapitre 3. Nous esti-

mons tout d'abord que les Syntagmes Non Récursifs de [Vergne 1999], qui peuvent s'apparenter à des chunks, sont une approche intéressante pour segmenter les questions ou les phrases des documents en groupes de mots. De plus les dépendances syntaxiques semblent nécessaires pour représenter les relations entre les groupes de mots, comme dans [Comas et al. 2010]. Il a par ailleurs été montré qu'il était possible d'utiliser des dépendances syntaxiques dans le cadre de l'oral. Nous nous inspirons donc des dépendances présentées dans [Tannier & Moriceau 2010] et [Comas et al. 2010]. Enfin, nous prenons en compte dans la conception de notre modèle l'importance des prédicats verbaux, comme dans [Pradhan et al. 2005 ; Shen & Lapata 2007].

Dans les prochaines sections, nous présentons l'approche mise en place pour représenter nos documents et questions. On peut distinguer deux étapes :

– Segmentation en constituants typés des phrases des documents et des questions, qui permet de regrouper les mots en groupes typés. Notre objectif est surtout d'avoir une base de travail pour ajouter des relations entre groupes de mots. Cette segmentation est composée de deux types principaux et quatre sous-types. Ce qui a guidé ces distinctions en types et sous-types est l'utilisation que nous voulons en faire pour l'ajout des relations entre ces segments.

– Identification de relations typées entre les segments, qui consiste à déterminer les groupes de mots en relation, et ainsi représenter la structure des phrases des documents et des questions.

4.2 Segmentation en constituants typés

L'objectif de la segmentation est d'ajouter des relations entre des groupes de mots. La majorité des travaux sur l'ajout de relations concerne l'anglais avec des ressources qui ne sont pas toujours disponibles en français (PropBank, FrameNet ...). D'autre part, beaucoup s'appuient sur une analyse syntaxique complexe dont nous ne disposons pas. Certains, comme [Sakai et al. 2004] ont montré que l'ajout de relations pouvait s'appuyer sur une segmentation des phrases en groupes de mots typés. Les types de Ritel peuvent servir d'indice pour procéder à une segmentation et permettre de typer ces segments. La définition d'une segmentation peut être très variable selon l'utilisation que l'on veut en faire. On peut notamment citer la tâche Chunking de ConLL2000 [Tjong et al. 2000] et la campagne EASY [Paroubek et al. 2008a]. Comme [Pradhan et al. 2005 ; Shen & Lapata 2007], nous estimons que les verbes ont un rôle de pivot. Cette notion de pivot a un impact important sur la gestion de nos relations, et conditionne la définition de nos segments.

Pour rappel, nous donnons dans la figure 4.1 une illustration de l'analyse effectuée par Ritel. La phrase analysée est *"A 71 ans Nelson Mandela est sorti après 27 années de prison"*.

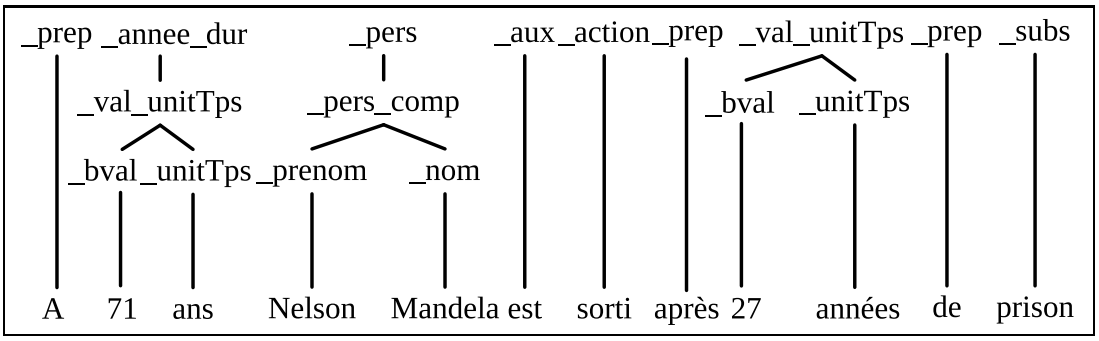

FIG. 4.1 – Annotation de la phrase *A 71 ans Nelson Mandela est sorti après 27 années de prison* par l'analyseur de Ritel .

4.2.1 Définition des segments

4.2.1.1 Formalisme EASY

La campagne EASY [Paroubek et al. 2008b] pour le français propose un formalisme de segmentation. EASY utilise six types de constituants : le groupe nominal, le groupe prépositionnel, le noyau verbal, le groupe adjectival, le groupe adverbial et le groupe verbal introduit par une préposition. Par rapport à cette représentation, nous avons choisi d'en adopter une relativement différente. Les résultats obtenus par les différents analyseurs sur les textes oraux dans le cadre de la campagne EASY nous ont en effet motivé à simplifier le formalisme proposé. En effet, les meilleurs systèmes obtiennent une f-mesure de 0.93 sur l'écrit, mais de seulement 0.79 sur l'oral. Une hypothèse est qu'en limitant le nombre de types de segments, la segmentation en constituants sera plus robuste par rapport aux documents issus de l'oral.

4.2.1.2 Formalisme adopté

Motivé par les Syntagmes Non Récursifs présentés dans [Vergne 1999], nous avons défini deux segments principaux : le segment verbal et le segment nominal. Le segment nominal est ainsi composé d'un nom et de plusieurs éléments, comme une préposition, un adjectif ou un déterminant. De même, le segment groupe verbal est composé d'un verbe et de plusieurs éléments : adverbe, négation, auxiliaire, préposition, etc ...

En plus de ces deux segments, nous ajoutons quatre autres types de segments, qui sont des sous-types

du segment nominal. Les segments de type temps et lieu représentent des segments de type nominal contenant respectivement des informations temporelles et des informations sur des lieux. L'objectif est de déterminer les relations de temps et de lieu existant entre les segments d'une phrase et d'une question. On se rapproche ainsi des rôles sémantiques de temps et de lieu, tels qu'ils sont définis dans [Carreras & Màrquez 2005]. Les segments de temps et de lieu sont relativement similaires à des groupes circonstanciels de temps et de lieu. L'objectif est de mesurer leur apport pour le réordonnancement et de considérer d'autres sous-types, comme par exemple un sous-segment représentant une personne.

Le troisième sous-type de segment nominal correspond au segment optionnel. Par optionnel, nous entendons les segments contenant des informations jugées non importantes **pour le moment**, principalement les figures de style. Il est cependant clair que ces informations sont nécessaires à moyen terme. L'idée est pour le moment d'expérimenter avec les segments déjà existants, afin d'évaluer notamment la robustesse face aux données de différents types de documents.

Enfin, un dernier sous-type de segment nominal est introduit, spécifique aux questions : le segment de type marqueur interrogatif. Ce type de sous-segments correspond au groupe de pronoms interrogatifs et de substantifs associés trouvé dans la majorité des questions factuelles (*"Combien de personnes"* par exemple). Ce type de segment est facilement identifiable grâce à l'analyse produite par Ritel. La détection de ce segment permet de le différencier du reste de l'information contenue dans la question. Ce segment est important pour les traitements à effectuer par la suite dans le module de réordonnancement.

– Segment verbal [SV] : ce type de segment est centré autour d'un verbe. Exemple : *"Le virus Ebola [SV] avait été identifié [/SV]"*

– Segment nominal [SN] : ce type de segment est centré autour d'un ou plusieurs noms. Exemple : *"[SN] le président de la société NCR [/SN]"*

– Segment temps [ST] : ce sous-type est comparable à un segment de type nominal, mais dont les mots qui le composent contiennent des informations temporels. Exemple : *"pour la première fois [ST] en 1976 [/ST] "*

– Segment lieu [SL] : ce sous-type est identique au segment temps sauf qu'il contient des informations sur des lieux. Exemple : *"Le président est revenu [SL] au Zaïre [/SL]"*

– Segment optionnel [SO] : un sous-type des segments de type nominal contenant des figures de style sans informations importantes. Exemple : *"[SO] De ce fait, [/SO] Arthur décida de rentrer chez lui."*

– Segment marqueur interrogatif [SMI] : ce sous-type des segments de type nominal correspond au groupe de pronoms interrogatifs se trouvant généralement au début d'une question. Exemple :*"[SMI] Combien de personnes [/SMI] sont nées en 2006 ?"*

La figure 4.2 donne un exemple de segmentation de la phrase *A 71 ans Nelson Mandela est sorti après 27 années de prison* :

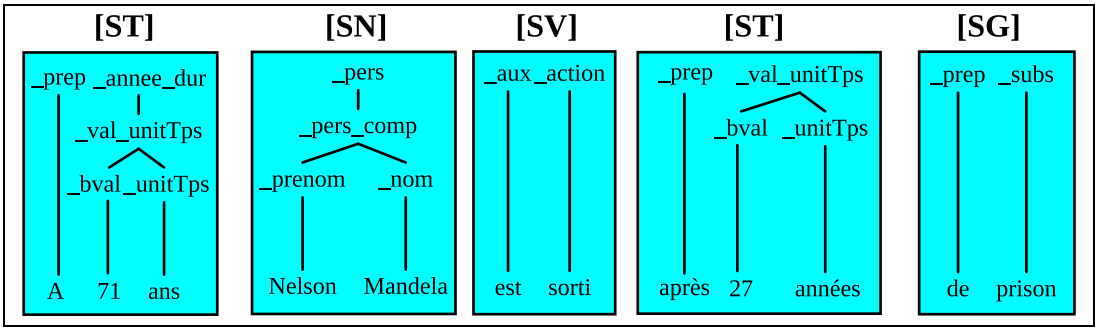

FIG. 4.2 – Annotation de la phrase *A 71 ans Nelson Mandela est sorti après 27 années de prison* par l'analyseur de Ritel (forêts d'arbres) et le segmenteur.

La figure présente l'annotation en forêts d'arbres faite par l'analyseur de Ritel. Les feuilles de chaque arbre correspondent à un mot de la phrase, par exemple *est* pour l'arbre <*aux*> *est* </*aux*>. Enfin, ces forêts d'arbres sont regroupées selon la segmentation en constituants typés effectuée. On a ainsi 5 segments :

– deux segments temps, *A 71 ans* et *après 27 années*
– deux segments nominaux, *Nelson Mandela* et *de prison*
– un segment verbal, *est sorti*

4.2.2 Annotation et typage des segments

Nous avons défini un formalisme de segmentation composé de deux types principaux, nominal et verbal. Le type nominal est de plus divisé en quatre sous-types : temps, lieu, optionnel et marqueur interrogatif. Une fois ce formalisme défini, il faut alors concevoir une approche pour appliquer ce formalisme sur les documents et les questions. Une annotation s'appuyant sur des règles écrites manuellement ne semble pas adaptée à notre contexte de travail : il est en effet difficile de couvrir l'ensemble des cas induits par notre formalisme, particulièrement pour le traitement de transcriptions orales.

Les champs aléatoires conditionnels ou Conditional Random Fields [Lafferty et al. 2001] (CRF) font partie de la famille des modèle probabilistes discriminants. Ils sont basés sur une approche conditionnelle pour étiqueter ou segmenter des séquences et sont particulièrement bien adaptées pour certains problèmes d'annotation en traitement de la langue naturelle [Sha & Pereira 2003]. Les CRFs ont été évalués sur la tâche Chunking de CoNLL 2000 [Tjong et al. 2000] et ont donné de très bons résultats :

la f-mesure obtenue est de 0.9438, ce qui est comparable aux meilleurs résultats obtenus sur la tâche (0.9439). Nous utilisons dans notre cas CRF++ [Kudoh 2007].

Nous nous appuyons donc sur les CRFs pour générer des modèles permettant d'annoter les documents et les questions. Deux modèles sont générés : un pour les questions et un pour les documents. Nous avons entraîné les modèles sur deux corpus que nous avons manuellement annoté en segments. Les différents corpus sont décrits dans la section 4.2.3.

Pour générer les modèles, il est évidemment nécessaire de s'appuyer sur différents traits. Du fait de notre contexte de travail, et particulièrement des nombreux types de documents que nous traitons, nous ne pouvons pas utiliser n'importe quels traits pour générer le modèle. Plusieurs des systèmes présentés dans la tâche de Chunking de CoNLL 2000 [Tjong et al. 2000] utilisent des annotations en Parties du Discours. Il semble donc pertinent de s'appuyer sur ce type d'analyse. L'analyseur en Parties du Discours de [Allauzen & Bonneau-Maynard 2008] est basé sur l'analyseur de Brill et a été entraîné sur le corpus Multitag [Adda, et al. 1998]. L'analyseur est testé sur un ensemble de phrases extraites du corpus Multitag. L'évaluation effectuée dans [Allauzen & Bonneau-Maynard 2008] montre que les résultats obtenus sont bons : 94,6% des mots sont bien annotés. Ces résultats sont proches de ceux obtenus par le TreeTagger français (95,7%). Le formalisme d'annotation en Parties du Discours utilisé est décrit dans [Allauzen & Bonneau-Maynard 2008]. Les labels utilisés sont divisés en une douzaine de catégories lexicales (nom, adjectif, verbe ...), chaque catégorie étant divisée en sous-catégories. Par exemple, pour un *nom*, trois sous-catégories sont définies : le *type* (commun, propre et cardinal), le *genre* (féminin ou masculin), et le *nombre* (singulier ou pluriel). Il en résulte un nombre d'étiquettes total de 1500. Ce type d'information est intéressant pour identifier les segments et ainsi générer notre modèle. Nous avons donc décider de nous appuyer sur ce formalisme et sur l'analyseur de [Allauzen & Bonneau-Maynard 2008].

Nous avons présenté l'analyseur de Ritel dans la section 2.2 du chapitre 2. Nous en rappelons les principales caractéristiques. L'analyse apporte principalement de l'information sémantique représentée sous la forme de forêts d'arbres. L'un des intérêts de cet analyseur est qu'il a été conçu pour être adapté au traitement de documents dont la syntaxe s'éloigne de celle de l'écrit classique. Les informations fournies par cet analyseur sont donc adaptées à notre contexte de travail, et complètent bien celles fournies par l'analyseur en Parties du Discours de [Allauzen & Bonneau-Maynard 2008]. L'analyse employée se compose d'environ 300 types différents, dont 20 d'ordre linguistique. Etant donné la structure d'arbre des analyses de Ritel, un même mot est associé à plusieurs types. Si l'on prend l'exemple de la figure 4.2, *Nelson* est associé aux types *prenom*, *pers_comp* et *pers*. Les types sémantiques étant généralement situés à la racine d'un arbre, nous avons décidé que chaque mot est associé au type Ritel associé situé à la racine de l'arbre. Dans le cas de *Nelson*, le type Ritel associé est donc *pers*.

Enfin, en plus des étiquettes associées à ces deux analyses, les traits sont composés des mots eux-mêmes et du contexte correspondant aux 2 ou 3 mots voisins. L'annotation manuelle des segments des corpus d'apprentissage respecte le format BIO. Le formalisme des étiquettes pour le type du segment se lit de la manière suivante : la première lettre peut soit être un *B* (begin), qui correspond au

début d'un nouveau segment, soit un *I* (inside) qui indique que le mot typé est dans le même segment que le mot précédent. Enfin, la lettre *O* (outside) indique une ponctuation.

mot	type PdD	type Ritel	type segment
Le	Da-ms-d	det	B-SN
duc	Ncms	titre	I-SN
Louis	Npms	pers	I-SN
entra	Vmis3s-	action	B-SV
vite	Rgp	adv	I-SV
en	Sp	prep	B-SN
conflit	Ncms	subs	I-SN
avec	Sp	prep	B-SN
le	Da-ms-d	det	I-SN
comte	Ncms	titre	I-SN
.	F	punct	O

TAB. 4.1 – Exemple de fichier annoté pour être traité par CRF++ pour la phrase *Le duc Louis entra vite en conflit avec le comte*. La première colonne correspond au mot, la deuxième à son étiquette en Parties du Discours et la troisième à son type Ritel. La quatrième colonne représente les bornes des segments selon le formalisme *BIO*.

La figure 4.1 illustre un exemple d'annotation pour la phrase *Le duc Louis entra vite en conflit avec le comte* selon le formalisme attendu par CRF++. L'exemple est divisé en 4 colonnes. La première colonne contient un mot par ligne, les deux colonnes suivantes contiennent les traits associés à ce mot, respectivement son type Partie du Discours, et le type issu de l'analyse de Ritel. Enfin la dernière colonne indique le type du segment dans lequel est contenu le mot. Ainsi, pour le mot *duc*, son type Partie du Discours est un *Ncms*, qui correspond à un nom commun masculin singulier, le type de l'analyseur de Ritel est un *titre*, et le type de segment est *I-SN*, qui signifie que *duc* fait partie du segment nominal *Le duc Louis*.

Nous présentons dans la section suivante les corpus d'apprentissage utilisés pour générer nos modèles, ainsi que plusieurs corpus de test pour évaluer notre segmentation.

4.2.3 Corpus d'apprentissage et de test

Deux corpus d'apprentissage ont été constitués, l'un pour les documents, et l'autre pour les questions. Le corpus d'apprentissage pour les documents comprend 180 000 mots et est constitué des articles du journal Le Monde de février 1993. Le corpus de questions est composé de 500 questions. Elles proviennent de divers évaluations : QA@CLEF04 [Magnini, et al. 2004], Qast08 [Turmo et al. 2008] et Quaero [Quintard 2009]. Afin de tester la robustesse du modèle obtenu, nous l'avons évalué sur différents corpus. Pour les documents, les trois corpus d'évaluation consistent en :

– un corpus de textes journalistiques (Le Monde juillet 1992)

– un corpus oral transcrit manuellement extrait de la collection utilisée dans la campagne d'évaluation QAST 2008 (émission d'informations radio et télédiffusées en français)
– un corpus composé de textes extraits de pages Web fourni par le projet Quaero

Les caractéristiques des corpus pour les documents sont présentées dans le tableau 4.2. Pour le corpus oral, les phrases sont en fait des segments se terminant par un "." tel que définit par l'application de méthodes de normalisation [Rosset, et al. 2008] présentées dans la section 2.1 du chapitre 2. On peut constater que le nombre moyen de mots par phrases est plus élevé dans le corpus oral.

Type du corpus	#mots	#phrases	#segments	#moy. de mots phrase
Corpus d'apprentissage	184090	22644	48652	8
Corpus de test journalistique	37020	5191	9813	7
Corpus de test oral	35612	3057	8889	11
Corpus de test web	30133	4180	8387	7

TAB. 4.2 – Caractéristiques des corpus d'apprentissage et de test pour les documents

Pour les questions, nous avons constitué deux corpus d'évaluation :

– le premier comprend 150 questions écrites provenant des campagnes d'évaluation de systèmes de questions-réponses QA@CLEF 2005, QAST 2008 et Quaero 2008 ;

– le second corpus comprend 150 énoncés extraits du corpus Ritel [Rosset et al. 2006]. Il ne s'agit là pas toujours de questions en tant que telles et ces énoncés sont donc très différents des questions écrites du premier corpus. L'énoncé *"J'aimerais savoir dans quel pays se trouve la ville de Rome."* donne un exemple des éléments se trouvant dans ce corpus. L'énoncé n'a pas la forme des questions traditionnellement rencontrées dans les campagnes d'évaluation (par exemple *"Qui est Jacques Chirac ?"*.

Ces deux corpus ont été annotés manuellement de la même façon que les corpus de documents. Les caractéristiques des deux corpus pour les questions se trouvent dans le tableau 4.3. Là aussi on peut remarquer que les énoncés oraux sont plus longs généralement que les questions écrites classiques que l'on retrouve dans les évaluations.

Nous présentons dans la section suivante les résultats obtenus par le segmenteur sur les différents corpus de test.

Type du corpus	#mots	#questions	#segments	#moy. de mots question
Corpus d'apprentissage	4159	501	1728	8
Corpus de test de questions écrites	1359	149	542	9
Corpus de test d'énoncés oraux	1724	150	722	11

TAB. 4.3 – Caractéristiques des corpus d'apprentissage et de test pour les questions

4.2.4 Résultats obtenus

Les résultats obtenus par le segmenteur sont présentés dans le tableau 4.4. Une évaluation plus détaillée ainsi qu'une analyse critique de ces résultats sont faites dans le chapitre 7. Ces mesures ont été effectuées sur l'ensemble des types du formalisme. On peut néanmoins constater que les résultats sont relativement similaires d'un type de documents à l'autre, excepté pour les énoncés oraux. Par contre, même si la comparaison est délicate, les résultats obtenus sur documents journalistiques sont moins bons que les meilleurs obtenus sur CoNLL 2000 (0.93) et EASY (0.92).

Type du corpus	précision(%)	rappel(%)	f-mesure
Documents journalistiques	83.2	82.9	0.83
Documents oraux	80.1	79.8	0.80
Documents du web	82.4	81.7	0.82
Questions écrites	82.6	81.4	0.82
énoncés oraux	59.4	58.5	0.59

TAB. 4.4 – Résultats globaux obtenus sur les cinq corpus de test

Nous présentons dans la section suivante nos conclusions sur cette segmentation ainsi que sur les résultats obtenus par cette évaluation.

4.2.5 Conclusions sur la segmentation

Nous avons présenté un formalisme de segmentation et son implémentation. Ce formalisme est conçu pour être adapté à tous types de documents. De ce fait, notre définition des segments reste volontairement simple. Elle est composée de deux segments principaux, qui sont aussi les plus fréquents : verbal et nominal. Nous définissons par ailleurs quatres sous-types de segments de type nominal : temps, lieu, optionnel et marqueur interrogatif.

L'identification des segments est effectuée en s'appuyant sur un modèle généré par des CRFs. Le segmenteur a été évalué sur trois corpus de documents de modalité différente : écrit journalistique, transcription orale, et web. Les résultats montrent que si les performances ne sont pas encore au niveau de celles obtenues sur CoNLL 2000 ou EASY, l'approche est suffisamment robuste pour ne

pas observer de baisse significative entre les différentes modalités. La segmentation des questions a aussi été évaluée sur deux corpus différents : questions écrites et énoncés oraux. Si les résultats sur les questions écrites sont équivalents à ceux obtenus sur les documents, les performances observées sur les énoncés oraux sont assez faibles, alors que les documents oraux donnent des résultats positifs.

Ces résultats nous semblent suffisants pour notre objectif : la représentation de la structure des questions et des phrases des documents. La deuxième partie de cette représentation est composée de relations typées entre segments, que nous présentons dans la section suivante. Ces relations sont fortement dépendantes de la segmentation. Nous estimons que les performances sont suffisamment bonnes pour mettre en place une identification des relations. Nous présentons une évaluation plus poussée ainsi qu'une analyse des résultats obtenus dans les chapitres 7 et 8.

4.3 Relations typées entre segments

La segmentation des phrases des documents et des questions en constituants typés permet d'avoir un regroupement en groupe de mots qui n'existait pas auparavant avec les analyses fournies par Ritel. Par ailleurs, le typage de ces segments permet d'avoir une représentation de la structure d'une phrase, avec notamment l'annotation des groupes verbaux. L'apport principal de la segmentation en constituants est de permettre l'ajout d'une représentation, et à partir de là, des relations entre les segments d'une phrase. Ces relations seront typées, et permettront de représenter les liens entre les groupes de mots, ce qui n'était pas possible avec le modèle de représentation initial utilisé par Ritel.

Le formalisme des relations qui va être introduit est pour le moment très simple et vise à traiter des structures de phrases courantes. L'objectif principal est de représenter les dépendances entre groupe de mots d'une phrase ou d'une question. La détection de cette relation doit être adaptée à n'importe quel type de documents. En utilisant les types des segments, on vise à déterminer les relations entre les différents groupes de mots. Le formalisme des relations a été défini après observations sur différents corpus issus de campagnes d'évaluation de systèmes de questions-réponses [Magnini et al. 2004 ; Vallin, et al. 2005 ; Turmo et al. 2008 ; Turmo et al. 2009]. Les relations ont été définies de manière à être appliquées sur différents types de corpus : journalistique, oral, web ... L'objectif secondaire de ces relations est de mettre au point un outil de travail pour observer les résultats d'une typologie préliminaire des relations, pour ensuite améliorer le modèle de représentation et ajouter de nouvelles relations.

4.3.1 Définition des relations

Notre définition des relations typées s'appuie sur les différents types de segments décrits dans la section 4.2.1. Notre objectif est d'avoir des relations simples de manière à être robuste à n'importe quel type de documents. Ainsi, nous n'avons qu'un faible nombre de relations. Si l'idée est de représen-

ter les dépendances syntaxiques comme cela peut être fait dans [Comas et al. 2010] ou [Tannier & Moriceau 2010], nos types de relations ne sont qu'au nombre de quatre, que nous décrivons dans les paragraphes suivants.

Nous donnons ci-dessous l'ensemble de ces relations avec un exemple pour illustrer chaque cas. Nous décrivons ensuite les raisons qui nous ont amené à définir ces types de relations.

– Relation nominale [RN] : relation que l'on retrouve entre deux segments nominaux voisins. Le segment nominal courant dépendra du segment nominal qui le précède immédiatement. Dans la phrase *"[SN] L'homme [/SN] [SV] habitait [/SV] [SN] dans une maison [/SN] [SN] à trois étages [/SN] ."*, il y a une relation nominale allant de *"dans une maison"* vers *"à trois étages"*.

– Relation verbale [RV] : les relations existant entre un segment verbal et ses constituants, en l'occurence majoritairement des segments groupes nominaux. Un segment verbal peut être en relation avec un constituant à sa gauche (le sujet), et plusieurs constituants à sa droite. Une relation est aussi ajoutée entre les constituants du segment verbal. Dans la phrase *"[SN] L'homme [/SN] [SV] habitait [/SV] [SN] dans une maison [/SN] [SN] à trois étages [/SN] ."*, le segment verbal *"habitait"* est en relation avec *"L'homme"* et *"dans une maison"*. Ces deux segments sont reliés avec une relation verbale.

– Relation temporelle [RT] : une relation temporelle est générée entre un segment temporel et tous les segments verbaux ou groupe nominaux d'une même phrase. Les segments temporels (en général des compléments circonstanciels) peuvent être déplacés n'importe où dans une phrase. Ainsi, dans la phrase *"[ST] En 1994 [/ST] [SN] Paul [/SN] [SV] habitait [/SV] [SN] dans un appartement [/SN]"*, il existe des relations temporelles entre *"En 1994"* et les autres segments.

– Relation spatiale [RS] : les relations spatiales sont similaires aux relations temporelles. Dans la phrase *"[SN] Paul [/SN] [SV] habitait [/SV] [SN] dans un appartement [/SN] [SL] à Paris [/SL]"*, il existe des relations spatiales entre *"à Paris"* et les autres segments.

Nous avons deux principaux types de relations, associés aux deux segments principaux, les segments nominaux et verbaux. Nous nous appuyons sur la définition des relations de dépendances présentée dans [Vergne 1999], où les Syntagmes Non Récursifs (SNR) de type nominal et verbal se rapprochent respectivement de nos segments nominaux et verbaux. Nous reprenons la notion de dépendance d'un segment nominal avec son voisin gauche. Si ce dernier est un segment de type nominal, alors on ajoute une relation nominale.

Pour le cas des segments verbaux, nous adoptons aussi la même approche que dans [Vergne 1999]. Ainsi, on ajoute une relation verbale entre le voisin de gauche et le segment verbal, et les voisins droits et le segment verbal. Contrairement aux dépendances entre SNRs, nous ne relions le segment verbal au segment voisin gauche que si ce dernier est un segment nominal. Si le segment voisin gauche est d'un autre type (généralement temps, lieu ou optionnel), alors on relie le premier segment situé à

gauche de type nominal. Par exemple, dans la phrase *"Nelson Mandela, à 71 ans, est sorti de prison "*, on aura une relation membres-verbe entre *Nelson Mandela* et *est sorti*. L'objectif est d'essayer de relier ce que nous estimons être le sujet avec le verbe associé. Pour les voisins à droite, l'idée est sensiblement la même. Plutôt que de ne relier qu'un seul segment comme avec les SNRs, nous avons décidé qu'au plus deux constituants pouvaient être reliés au verbe, à condition que l'un des segments soit un segment nominal et l'autre un segment spatial ou temporel. Sinon seul le segment voisin droit sera associé. Pour justifier ce choix, nous nous basons sur deux principes : d'une part un verbe a souvent plus d'un constituant associé à sa droite, comme montré dans la définition des rôles sémantiques de [Carreras & Màrquez 2005]. D'autre part, les segments de temps et de lieu sont considérés comme des groupes circonstanciels. On estime donc qu'on peut changer leur place dans une phrase sans en changer son sens. Dans la phrase *"Nelson Mandela est sorti après 27 années de prison."*, *après 27 années* et *de prison* ont une relation membres-verbe avec *est sorti*. Enfin, on ajoute aussi des relations membres-verbe entre les différents constituants d'un verbe. Dans *"Nelson Mandela est sorti de prison."*, *Nelson Mandela* et *de prison* sont des constituants du verbe *est sorti*, et une relation membres-verbes existent entre *Nelson Mandela* et *de prison*.

Enfin, les relations temporelles et spatiales sont associées aux segments de temps et de lieu, et ont un comportement un peu spécial. En effet, pour représenter le fait que ces segments sont comparables à des groupes circonstanciels, on fait le choix que ces segments sont en relation avec tous les segments d'une phrase. Nous nous appuyons sur la propriété des groupes circonstanciels qui font que l'on peut généralement les déplacer dans une phrase sans en changer le sens. Une approche plus fine serait de ne relier un segment temps ou lieu qu'avec les segments présents dans une même proposition. Le problème est que la détection des subordonnées n'est pas un problème aisé, d'autant plus pour l'oral où la structure des textes est très différente de l'écrit (ponctuation non fiable car fournie par un logiciel de reconnaissance de la parole par exemple). Nous avons donc fait un choix plus simple, mais potentiellement plus robuste. Nous traitons les probables ambiguités dans la méthode de réordonnancement présentée dans le chapitre 5.

En plus de leur type et des segments qu'elles relient, ces relations ont un sens de dépendance. Ce sens représente quel segment est dépendant d'un autre. Nous avons deux sens pour les relations :

– Relation entrante

– Relation sortante

Par exemple, dans une relation nominale, on dit que le segment dépend de son voisin gauche. La relation va donc du segment vers le voisin gauche. On dit que la relation est *entrante* pour le voisin gauche, et *sortante* pour le segment considéré. Dans le cas des relations verbales, les constituants (membres) sont dépendants du verbe. Dans ce cas, la relation est entrante pour le verbe et sortante pour les constituants. Dans le cas des relations existant entre les constituants d'un verbe, on considère qu'il n'y a pas de sens de dépendance. Enfin, pour les relations temporelles et spatiales, les segments d'une phrase sont dépendants de la relation en question. Elle est donc entrante pour le segment de temps ou de lieu, et sortante pour les autres segments de la phrase.

La figure 4.3 donne un exemple de représentation du passage *Après 27 années Nelson Mandela est libéré par le président Frederik Willem de Klerk*. Les relations temporelles ne sont pas représentées sur cette figure pour éviter de la surcharger. Les couleurs représentent le type du segment : bleu correspond aux segments temporels, jaune aux segments nominaux, et rouge aux segments verbaux.

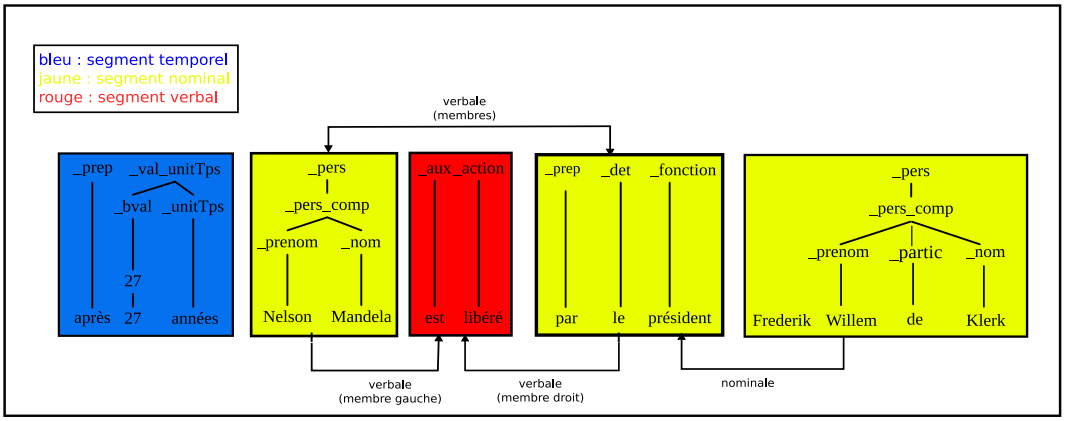

FIG. 4.3 – Exemple de relations entre segments typés correspondant au passage *"Après 27 années Nelson Mandela est libéré par le président Frederik Willem de Klerk ."*

L'identification des relations est effectuée à partir de règles. Ces règles ont été écrites manuellement à partir des observations sur les collections des campagnes d'évaluation de systèmes de questions-réponses pour générer ces relations.

4.3.2 Règles d'ajout des relations

On utilise des règles écrites manuellement à partir des observations sur les collections des campagnes d'évaluation de systèmes de questions-réponses. Ces règles sont pour l'instant implémentées directement dans le code du système. Un formalisme de règles avait été mis en place, mais il n'était pas adapté. A moyen terme, les règles seront réécrites selon un formalisme plus adapté, permettant ainsi de charger différents fichiers de règles de manière plus efficace plutôt que de devoir changer le code pour apporter des modifications.

Les règles sont écrites sous la forme de conditionnelles classiques. La figure 4.4 illustre la règle permettant d'identifier une relation nominale entre deux segments nominaux.

Le tableau **segments** représente l'ensemble des segments d'une question ou d'un passage. Ainsi, dans l'exemple de règle ci-dessus, si le segment nominal actuellement traité (**segments[i]**) possède à sa gauche un autre segment de type *nominal*, alors une relation **nominale** est créée. Les fonctions **type** et **estEnRelation** permettent respectivement de récupérer le type d'un segment et d'ajouter une relation entre deux segments. La fonction **estEnDependance** permet d'ajouter le sens de dépendance entre

```
si (segments[i].type() = "nominal" ET segments[i-1].type() = "nominal")
        segments[i].estEnRelation(segments[i-1],"Relation nominale")
        segments[i].estEnDependance(segments[i-1])
fsi
```

FIG. 4.4 – Exemple de règle pour l'ajout des relations

les deux segments : dans cet exemple, le segment courant (**segments[i]**) est dépendant du segment de gauche (**segments[i-1]**).

Ainsi, l'algorithme de détection des relations présenté dans la figure 4.5 est relativement simple : pour chaque segment, on cherche l'ensemble des règles s'appliquant au type du segment, et on cherche alors si les segments voisins permettent d'ajouter une relation.

```
Ajout des relations entre segment
pour chaque segment i de segments faire
        appliquerRegles(i.type(), segments)
fpour
```

FIG. 4.5 – Algorithme d'ajout des relations entre segment

La fonction **appliquerRegles** permet d'appliquer chacune des règles existantes pour identifier et ajouter les relations existant entre le segment i et les autres segments.

Si l'on reprend l'exemple de la figure 4.3, et que l'on applique la règle présentée dans la figure 4.4 sur le segment *Frederik William de Klerk*, la règle essaye d'abord d'accéder au type du segment, qui est égal à *nominal*. La première partie de la conditionnelle est validée, et on récupère **segments[i-1]**, qui correspond au segment précédent, c'est à dire *"par le président"*. Le type est récupéré, qui est aussi égal à *nominal*. La deuxième partie de la conditionnelle est validée. Une relation de groupe nominal est donc créée entre les deux segments.

4.4 Conclusions sur le modèle de représentation

Comme précisé précédemment, notamment dans le chapitre 2 détaillant le fonctionnement du système de questions-réponses Ritel, l'un des défauts majeurs de l'approche utilisée était le manque de prise en compte de la structure des phrases. L'ambiguité de certains cas où la réponse à une question n'est pas mentionnée de manière explicite, ou encore l'impact de la redondance expliquent les problèmes rencontrés par le système actuel.

L'état de l'art effectué sur le domaine des systèmes des questions-réponses montre qu'il est possible

d'utiliser des méthodes plus fines, malgré le contexte de travail (différents types de documents traités, ressources limitées dans d'autres langues que l'anglais ...). Ces méthodes se basent souvent sur des représentations syntaxiques des documents et des questions, de manière à avoir une représentation de la structure dans les questions.

Le modèle de représentation décrit dans cette section est conçu dans l'optique d'être applicable à n'importe quel type de documents. Il reste donc volontairement simple. Ce modèle est utilisé par la méthode de réordonnancement présentée dans le chapitre 5. La figure 4.4 illustre les différents modules utilisés pour mettre en place notre modèle de représentation, et l'appliquer sur les questions et documents traités.

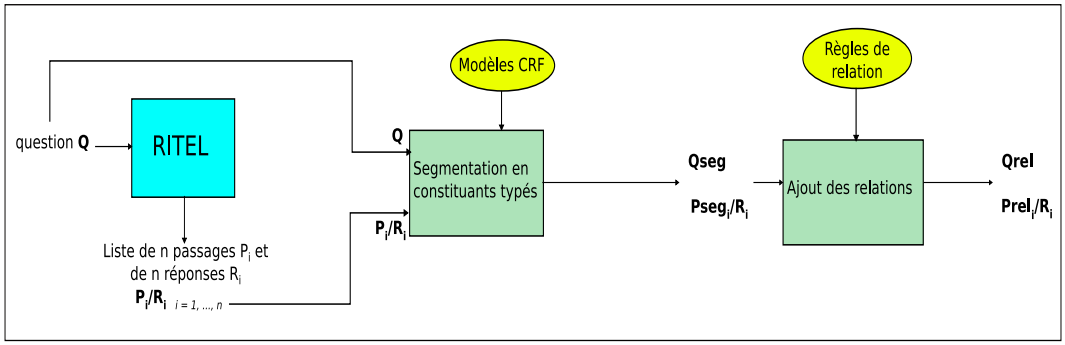

FIG. 4.6 – Schéma de la mise en oeuvre du modèle de représentation

Une question *Q* est traitée par le système de questions-réponses de Ritel, et une liste de réponses ainsi que les passages associés sont retournés. Cette question ainsi que les passages sont par ailleurs annotés par l'analyseur de Ritel. Le module de segmentation en constituants typés va alors traiter cette question et ces passages, en s'appuyant sur des modèles CRF. A la sortie de ce module, les segments ont été rajoutés sur la question et les passages. Enfin, le module d'ajout des relations, en s'appuyant sur des règles de détection, ajoute les relations décrites dans ce chapitre aux passages et à la question. C'est là-dessus que s'applique le module de réordonnancement, présenté dans le chapitre 5.

Chapitre 5

Une méthode de réordonnancement des candidats réponses

5.1 Introduction

Nous présentons dans ce chapitre notre méthode de réordonnancement des candidats réponses retournés par un système de questions-réponses. Le système fournit un ensemble d'hypothèses à une question, et le réordonnanceur doit calculer un nouveau score pour chaque réponse. Notre contexte de travail étant le système de questions-réponses Ritel, il nous faut prendre en compte les limites rencontrées par Ritel. Ces limites sont principalement liées à la non utilisation de la structure des phrases et des questions dans les méthodes d'extraction, ainsi que l'impact de la redondance dans les calculs. Par exemple, pour la question *"Combien d'années Nelson Mandela passa-t-il en prison ?"*, un des passages évalués pour répondre à cette question est « A 71 ans, Nelson Mandela est sorti de prison après 27 années. » . Deux réponses sont évaluées, *71 ans* et *27 années*, cette dernière étant la bonne réponse. La sélection des réponses effectuée par Ritel s'appuie sur la distance entre les éléments de la question et les réponses évaluées, ainsi que les autres occurences de ces réponses dans les différents passages traités. La distance moyenne par rapport aux éléments de la question est identique pour les deux réponses candidates. Pour peu que la réponse 71 ans apparaisse plus souvent dans le reste des documents, elle sera choisie comme étant le bon choix. Une mise en relation peut permettre dans ce cas de déterminer la bonne réponse : on identifie le rattachement entre *27 années* et *prison*. Pour répondre à ce manque, nous avons mis en place un modèle de représentation robuste, présenté dans le chapitre précédent, et fournissant notamment des relations entre les groupes de mots des questions et des phrases. L'idée est de s'appuyer sur cette représentation pour réordonnancer les réponses candidates à une question retournées par Ritel.

L'état de l'art sur les systèmes de questions-réponses a montré que très souvent des systèmes utilisaient un module de réordonnancement en fin de traitement. L'idée est d'avoir une première mé-

thode d'extraction des réponses, permettant d'obtenir un premier nombre de candidats, puis d'appliquer des méthodes plus fines sur cet ensemble de candidats. L'approche proposée par [Kouylekov & Negri 2010] nous a semblé particulièrement pertinente. Cette approche a été conçue dans le cadre de l'implication textuelle : l'objectif de cette tâche est de déterminer si le sens d'un texte implique celui d'un autre texte. Ce type d'approche est applicable aux systèmes de questions-réponses : l'idée est alors de comparer une question et un passage contenant une réponse candidate. L'approche proposée dans [Kouylekov & Negri 2010] s'appuie sur une distance d'édition. Les opérations de transformation (ou édition) permettent de quantifier la différence (aussi bien structurelle qu'au niveau du sens) existant entre deux textes. Dans [Kouylekov & Negri 2010], la distance d'édition est appliquée sur deux types de structures : les arbres syntaxiques, et les composants (ou groupe de mots). Nos segments typés présentés dans le chapitre 4 et regroupant les mots semblent adaptés à la distance d'édition sur composants. Nous nous sommes inspirés de cette approche pour notre module de réordonnancement. Par ailleurs, nous nous sommes aussi inspirés des méthodes basées sur des relations syntaxiques présentées dans le chapitre 3.

Notre idée est donc de représenter le problème du réordonnancement d'hypothèses comme le calcul d'un coût de transformation entre un passage et une question. Ce coût est calculé à partir d'opérations de transformation. Les opérations sont appliquées sur les segments typés issus de notre représentation. Le coût de transformation associé à une réponse permet ensuite de réordonner nos candidats : celui avec le coût le plus faible est choisi comme étant la bonne réponse à la question.

Nous allons d'abord présenter l'architecture principale du système dans la section 5.2. Cette présentation de l'architecture permet de donner une compréhension générale du réordonnanceur. Nous détaillons ensuite dans les sections suivantes les parties principales du système. Nous commençons par présenter les ressources utilisées par le réordonnanceur dans la section 5.3. Nous présentons ensuite les principaux modules de notre approche, que nous divisons en deux sections, les traitements préliminaires dans la section 5.4 et le calcul du coût de transformation dans la section 5.5.

5.2 Architecture du réordonnanceur

Pour une question donnée, le réordonnanceur traite les réponses candidates fournies par Ritel. Chaque réponse a été extraite d'un certain nombre de passages. Ainsi, pour chaque passage d'une réponse, un score de similarité est calculé entre le passage et la question par le réordonnanceur. Ce score est déterminé par le calcul d'un coût de transformation, s'appuyant sur une distance d'édition. Chaque passage d'une réponse est ainsi associé à un coût de transformation. Ce coût nous permet de représenter la similarité entre un passage et une question : un coût de transformation élevé implique que le sens d'un passage et sa structure sont éloignés de ceux de la question. Au contraire, un passage avec un coût de transformation faible est considéré comme étant proche de la question sémantiquement et structurellement. Une réponse étant associée à plusieurs passages, le score de similarité de la réponse est le coût de transformation du passage avec le coût le plus faible. Les réponses sont réordonnées suivant ce score de similarité.

L'exemple de la figure 5.1 illustre ce fonctionnement. La réponse *71 ans* est évaluée pour la question *"Quel âge avait Nelson Mandela à sa sortie de prison ?"*. Deux passages sont associés, *P1* et *P2*. Le score de similarité de cette réponse sera le coût de transformation du passage ayant la valeur la plus faible.

Question : *Quel âge avait Nelson Mandela à sa sortie de prison ?*
Réponse évaluée : *71 ans*
Passage 1 : *Le père de Nelson Mandela avait* **71 ans** *quand son père fût emprisonné.*
Passage 2 : A 71 ans, *Nelson Mandela est sorti après 27 ans de prison.*

FIG. 5.1 – Exemple des passages associés à la réponse **71 ans** de la question *Quel âge avait Nelson Mandela à sa sortie de prison ?*

Le calcul du coût de transformation doit mesurer la différence structurelle et sémantique entre une question et un passage. Les questions et passages fournis contiennent déjà de l'information sémantique : l'analyse effectuée par Ritel (voir section 2.2 du chapitre 2) apporte principalement de l'information sémantique sous forme de forêts d'arbres. Le résultat de cette analyse peut être utilisé pour mesurer la différence sémantique entre un passage et une question. Nous avons présenté dans le chapitre 4 un formalisme de représentation de l'information composé de segments typés et de relations entre ces segments. Il semble pertinent de s'appuyer sur ce formalisme pour mesurer la différence structurelle entre un passage et une question. De ce fait, nous divisons le fonctionnement de notre réordonnanceur en deux étapes principales :

– les traitements de structuration multi-niveaux, ou pré-traitements ;
– le calcul du coût de transformation entre une question et un passage.

Les pré-traitements ont pour objectif de procéder à une analyse préalable sur le passage et la question. L'information sémantique partagée par un passage et une question est identifiée. De plus, l'information structurelle apportée par les segments typés et les relations entre ces segments est ajoutée. La deuxième étape, le calcul du coût de transformation, s'appuie sur les informations apportées par les pré-traitements. Nous décrivons ici ces deux étapes principales, de manière à avoir une vue globale du fonctionnement du réordonnanceur. Ces différentes étapes sont ensuite présentées en détail (voir section 5.4 à 5.5).

5.2.1 Traitements de structuration multi-niveaux

L'objectif des traitements de structuration multi-niveaux est d'analyser un passage et une question et d'apporter des informations qui sont utilisées dans la deuxième étape de notre réordonnanceur, le calcul du coût de transformation. Ces traitements peuvent être divisés en deux parties principales : l'identification de l'information sémantique partagée par la question et le passage, et l'ajout d'information structurelle par le biais des segments typés et des relations entre segments. A ces deux parties

principales, on peut ajouter un troisième traitement, que l'on nommera la réduction du passage. L'objectif de ce traitement est d'éviter de traiter des passages trop longs. Dans la suite de cette section, nous donnons les idées principales des différents pré-traitements. Ces pré-traitements sont ensuite détaillés dans la section 5.4.

L'ajout de l'information structurelle est d'abord apporté par l'identification des segments typés entre la question et le passage. Ces segments sont composés principalement de deux types, nominal et verbal. Ce formalisme est aussi composé de quatre sous-types du segment nominal : temps, lieu, optionnel, et marqueur interrogatif. Nous avons présenté dans la section 4.2 du chapitre 4 ce formalisme de segmentation et son implémentation. Son apport principal est de regrouper les mots autour d'une entité principale (un mot ou un verbe), et ainsi de donner une première représentation de la structure d'une phrase ou d'une question. La première étape des pré-traitements pour ajouter de l'information structurelle est de typer les segments de la question et du passage traité.

Parallèlement à ce typage des segments, le réordonnanceur procède à l'identification de la similarité sémantique du passage et de la question. Cette identification s'appuie sur l'analyse produite par Ritel. Cette analyse est composée de plus de 300 types différents, dont la majorité apporte de l'information sémantique. Il existe néanmoins une vingtaine de types d'ordre linguistique, comme déterminant ou préposition. Nous nous appuyons sur cette classification pour identifier la similarité sémantique : nous comparons les mots de la question et du passage ayant un type sémantique. Nous estimons que ces mots permettent de représenter en partie le sens d'une phrase ou d'une question. Ainsi, plus une question et un passage partagent de mots de type sémantique en commun, plus nous considérons le passage et la question comme ayant un sens proche. Il est néanmoins possible que l'on ne retrouve pas la même forme d'un mot entre une question et un passage. Par exemple, dans la question *"Quel âge avait Nelson Mandela à sa libération de prison ?"*, le mot *libération* se retrouve dans le passage *"A 71 ans, Nelson est libéré de prison."*, mais sous une forme différente : *libéré*. De ce fait, nous nous appuyons sur plusieurs ressources linguistiques pour identifier les mots similaires entre une question et un passage. Ces ressources permettent d'identifier des similarités de lemmes, de synonymes et de transformations morpho-syntaxiques. Nous décrivons ces ressources dans la section 5.3.

Si les mots identifiés permettent d'identifier la similarité sémantique entre un passage et une question, nous estimons néanmoins que l'unité du mot n'est pas suffisante pour représenter cette similarité : un mot est généralement associé à une structure syntaxique, comme un groupe verbal ou un groupe nominal, qui ont une influence importante sur son sens. De ce fait, nous estimons que l'identification de la similarité sémantique doit s'appuyer sur les segments typés identifiés. Si le typage de ces segments est éloigné d'un typage plus traditionnel (par exemple composé de groupes prépositionnels), il permet néanmoins d'avoir un regroupement des mots. Ainsi, en nous appuyant sur les mots similaires entre la question et le passage et les segments typés, le réordonnanceur identifie des relations de similarité entre les segments. Nous appelons ces relations de similarités *ancres*. Ces ancres permettent de représenter la similarité sémantique entre deux segments, en s'appuyant sur les mots de type sémantique en commun. Ainsi, les segments *le gros chat* et *un gros chat* ont une ancre entre eux composée des deux mots en commun, *gros* et *chat*. Ces deux segments sont considérés comme très similaire. Les deux segments *le petit félin noir* et *le chat* ont aussi une ancre en commun, du fait de la

synonymie entre *chat* et *félin*, mais leur sens est bien plus éloigné car *petit* et *noir* ne sont pas commun aux deux segments, et que la similarité entre *chat félin* est de type synonyme. L'ancre contient l'ensemble de ces informations, qui sont ensuite utilisées dans le calcul du coût de transformation pour quantifier la similarité sémantique entre la question et le passage. Ces ancres ne prennent pas en compte le contexte structurelle des deux segments qu'elles relient : cet aspect est laissé au calcul du coût de transformation. Par ailleurs, il pourrait être intéressant d'avoir accès à des ressources linguistiques supplémentaires pour améliorer les informations contenues dans une ancre, comme par exemple lorsque deux mots s'opposent, comme *gros* et *maigre*.

En s'appuyant sur les ancres identifiées, le réordonnanceur procède à une réduction des passages traités. Il arrive fréquemment que Ritel retourne des passages très longs, composés de plus de 500 mots. C'est particulièrement le cas lorsque l'on traite des documents du web. L'information d'une question est donc dispersée dans l'ensemble du passage, et il est très difficile de traiter ces longs extraits pour le réordonnanceur. Plusieurs observations ont été faites sur ces longs passages, aussi bien pour ceux contenant la bonne réponse que ceux contenant une réponse erronée. Il a été observé que dans bien des cas, la dispersion des éléments de la question à travers plusieurs phrases impliquait que la réponse évaluée était erronée. Il arrive cependant que cela ne soit pas le cas, particulièrement dans le cas de documents dont la structure repose beaucoup sur l'utilisation des anaphores. L'implémentation d'un module de résolution des anaphores qui permettrait de traiter ces problèmes va bien au delà du cadre de cette thèse. Pour le moment, il a été décidé de ne pas prendre en compte les éléments de la question identifiés dans un passage trop éloignés de la réponse évaluée. Nous nous appuyons sur les ancres pour réduire ces passages, les ancres permettant de représenter les segments contenant l'information similaire entre un passage et une question.

Finalement, la dernière étape des pré-traitements est d'ajouter les relations entre segments typés. Cette identification participe à l'ajout de l'information structurelle entre la question et le passage. Le formalisme des relations et leur implémentation sont présentés dans la section 4.3 du chapitre 4. Ces relations sont composées de deux types principaux, nominal et verbal, et deux sous-types, temps et lieu. L'objectif de ces relations est de représenter les dépendances structurelles entre les différents segments d'une phrase ou d'une question. Ces dépendances restent relativement simples par rapport aux dépendances pouvant être trouvées dans des analyseurs syntaxiques comme XIP [Aït-Mokhtar et al. 2002]. Elles nous permettent néanmoins de compléter les segments typés, et ainsi d'avoir une représentation de la structure dans les phrases et les documents. Le calcul du coût de transformation peut alors s'appuyer sur ces relations pour mesurer la similarité syntaxique entre un passage et une question.

5.2.2 Calcul du coût de transformation

Une fois les traitements de structuration multi-niveaux terminés, un coût de transformation est calculé entre la question et le passage traités. Ce coût de transformation s'appuie sur les informations apportées par les pré-traitements : la similarité sémantique entre le passage et la question, représentée par les ancres entre les segments, et l'information structurelle, représentée par les segments typés, et les

relations entre ces segments. Le coût de transformation est calculé à partir d'une distance d'édition sur composants, les composants étant les segments typés dans notre travail. L'objectif de cette distance d'édition est de quantifier la différence sémantique et structurelle entre une question et un passage en appliquant des opérations d'édition, que l'on appelle dans notre travail opérations de transformation. Les opérations sont appliquées sur les segments du passage et de la question : un coût est calculé pour chaque opération appliquée. L'objectif de ce coût est de représenter l'impact structurel et sémantique des modifications appliquées au passage. La somme des coûts de chaque opération donne le coût de transformation d'un passage.

Dans une distance d'édition traditionnelle, il existe trois types différents d'opérations de transformation : la substitution, l'insertion et la suppression. Appliquées à une question et un passage segmentés, ces opérations permettent de quantifier les transformations nécessaires pour transformer le passage en la question. Ainsi, l'opération de suppression permet de supprimer les segments du passage contenant de l'information qui n'est pas en rapport avec l'information contenue dans la question. De même, l'opération d'insertion permet d'insérer les segments de la question contenant l'information que l'on ne retrouve pas dans le passage. Enfin, l'opération de substitution permet de remplacer les segments du passage ayant un sens proche de ceux de la question. L'exemple de la figure 5.2 permet d'illustrer une application possible de la distance d'édition dans notre contexte de travail. L'objectif est de transformer le passage "**[A 71 ans]**, *[Nelson Mandela] [est libéré] [après 27 ans] [de prison]*" en la question "*[Quel âge] [avait] [Nelson Mandela] [à sa sortie de prison] ?*", la réponse candidate évaluée étant **71 ans**. Il semble tout d'abord pertinent d'associer la réponse candidate au segment marqueur interrogatif *Quel âge* : ce dernier contient les informations indiquant la réponse cherchée. Par la suite, une première suite d'opérations possibles est de supprimer du passage les segments *est libéré* et *après 27 ans*, que l'on ne retrouve pas dans la question. Les segments *de prison* et *à sa sortie de prison* ayant un sens proche, une substitution est alors possible. Enfin, on insère le segment manquant *avait*.

Question : *[Quel âge] [avait] [Nelson Mandela] [à sa sortie de prison] ?*
Réponse évaluée : *71 ans*
Passage : [A 71 ans], *[Nelson Mandela] [est libéré] [après 27 ans] [de prison]*.
Opérations possibles : Suppression de *est libéré*, suppression de *après 27 ans*, substitution de *de prison* par *à sa sortie de prison*, insertion *avait*.

FIG. 5.2 – Exemple d'une suite de transformations possibles entre la question et le passage, selon la réponse candidate **71 ans**. Pour simplifier la lecture, on ne représente pas les types des segments, mais seulement leurs bornes. De même, nous n'indiquons que certaines des relations identifiées entre les segments.

Cet exemple illustre certains des problèmes inhérents à l'application d'une distance d'édition dans notre contexte de travail. Tout d'abord, la transformation se fait entre un passage et une question. Ces deux éléments n'ont évidemment pas la même construction syntaxique, ce qui rend délicat une comparaison. Une idée possible est de d'abord transformer la question sous une forme affirmative. Néanmoins, un tel travail va bien au delà de cette thèse. Une idée simple est alors dans un premier

temps de considérer comme dans l'exemple précédent que la réponse candidate est liée au segment de la question contenant les marqueurs interrogatifs. Néanmoins, d'autres problèmes sont illustrés par cet exemple. Ces différentes opérations ne prennent pas en compte les différences structurelles entre le passage et la question. Ces trois types d'opération permettent surtout de quantifier la différence sémantique entre le passage et la question. Ainsi, il est difficile avec une telle approche de valider **71 ans** comme étant une réponse correcte, et **27 ans** comme étant une réponse invalide. Une idée est alors de s'appuyer sur les relations identifiées entre les segments. Néanmoins, ces relations n'ont pas une portée très lointaine, et le formalisme est volontairement simple pour être adapté à tout type de documents. Si leur utilisation dans les opérations de substitution, insertion et suppression semble intéressante, il n'est pas suffisant.

Une possibilité est alors d'ajouter un nouveau type d'opération de transformation. Une opération de déplacement semble dans un premier temps intéressant. Cependant, un tel type d'opération implique de savoir gérer les transformations syntaxiques induites par l'opération. Par ailleurs, il faut aussi adapter l'approche à notre contexte de travail, et notamment aux différents types de documents à traiter. Une étude des différentes questions factuelles posées lors des campagnes d'évaluation montre que ces questions sont construites selon une structure assez simple. L'ensemble des éléments d'une question sont généralement dépendants, ou en tout cas en relation, avec le segment contenant le marqueur interrogatif. Une première étape possible est alors de déterminer si les segments du passages considérés comme similaires aux segments de la question sont en relation avec la réponse candidate. Une telle approche est possible en s'appuyant sur les relations identifiées entre les segments. Notre formalisme de relations étant simple, il est clair que les relations ne suffiront pas pour déterminer si un segment est en relation avec la réponse candidate. Ainsi, il est nécessaire de quantifier un coût pour mettre le segment en relation avec le candidat réponse. Nous appelons cette mise en relation *rattachement*. Nous introduisons donc dans ce travail un quatrième type d'opérations de transformation, l'opération de rattachement.

Par ailleurs, si nous considérons que tous les segments de la question sont en relation avec le segment marqueur interrogatif, il existe aussi des relations entre les segments de la question. Si des segments de la question sont en relation, il semble alors nécessaire de s'assurer que les segments du passage identifiés comme similaires soient eux aussi en relation. Si ce n'est pas le cas, une opération de rattachement entre les deux segments du passage serait appliquée. Une telle approche a été évaluée, mais n'a pas donné de bons résultats pour le moment. Un travail plus poussé semble donc nécessaire pour améliorer cet aspect de notre travail. Les rattachements présentés dans ce document ne sont donc effectués que entre un segment du passage contenant un élément de la question, et le candidat réponse.

Son objectif est de quantifier la différence structurelle entre un passage et une question, les opérations de substitution, d'insertion, et de suppressions ayant pour objectif de quantifier principalement la différence sémantique. Ces opérations s'appuient sur les segments typés et les relations entre segments, ainsi que les ancres identifiées par les pré-traitements. Le calcul du coût de transformation utilise donc ces quatres types d'opérations pour calculer le coût de transformation entre le passage d'une réponse candidate et la question. Ce coût est ensuite comparé aux coûts des autres passages associés à la réponse. Le coût le plus faible est considéré comme étant le score de la réponse candidate. Ce traitement

est effectué pour chaque candidat à une question, et les réponses sont réordonnées selon leur score. La figure 5.3 représente l'architecture du réordonnanceur en sortie du système de questions-réponses Ritel.

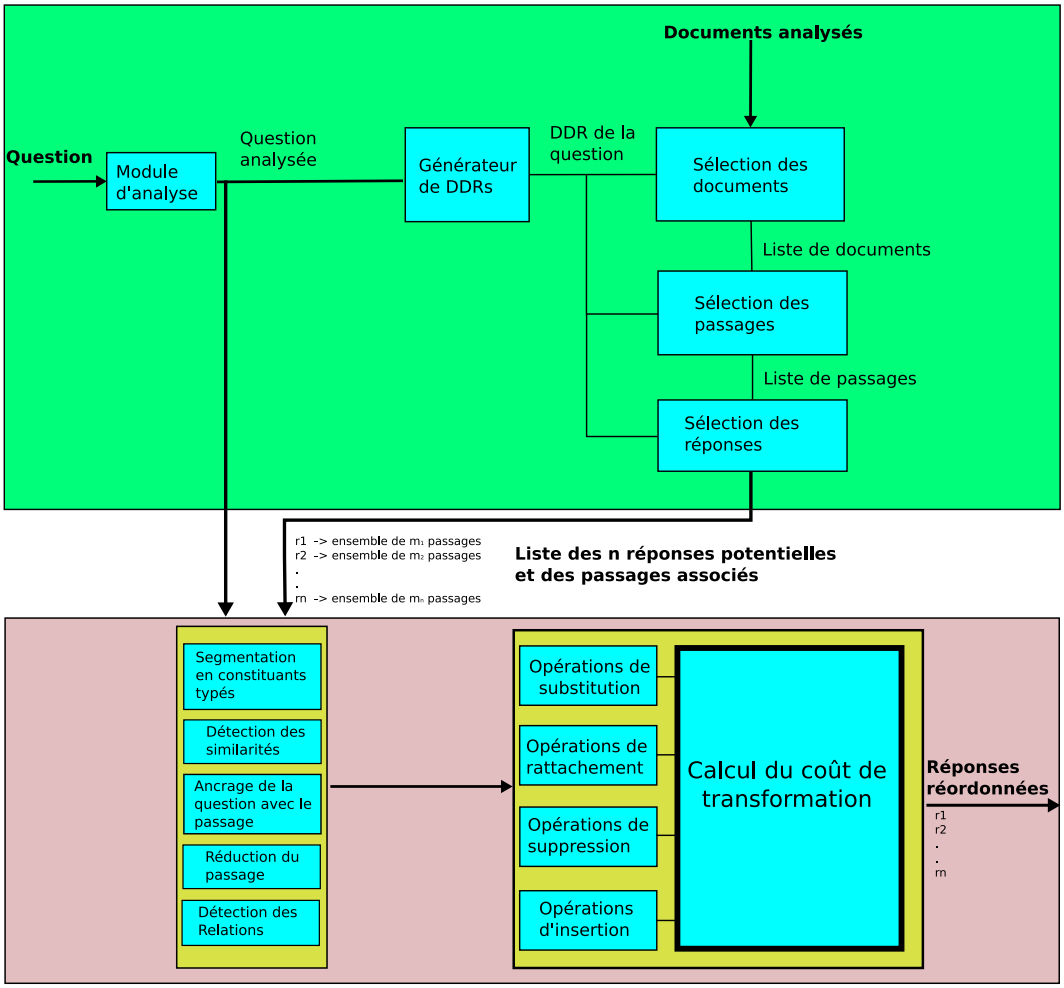

FIG. 5.3 – Architecture générale du réordonnanceur en sortie du système de questions-réponses de Ritel. Ritel est dans la partie haute du schéma, et le réordonnanceur dans la partie basse. La gauche du schéma de réordonnanceur correspond aux traitements de structuration multi-niveaux, et la droite au calcul du coût de transformation.

Dans la section 5.3, nous présentons les ressources linguistiques sur lesquelles s'appuie une partie de notre travail. Puis dans les sections 5.4 et 5.5, nous présentons les deux parties principales du réordonnanceur, les traitements de structuration multi-niveaux, et le calcul du coût de transformation.

5.3 Ressources linguistiques

Le réordonnanceur utilise les mêmes ressources lexicales que le système de questions-réponses de Ritel. L'objectif de ces ressources est de fournir une liste de transformations possibles pour un mot donné. Dans le contexte de notre travail, nous nous appuyons sur ces ressources pour identifier les similarités entre mots. Etant donné qu'un mot de la question peut se retrouver sous une forme différente mais néanmoins similaire dans le passage, ces ressources permettent d'identifier les formes différentes associées à un mot. Par exemple, le mot *libération* est une transformation morphologique de *libéré*. Trois types différents de dictionnaires sont utilisés :

– un dictionnaire contenant des paires mot/lemme (par exemple *adouci -> adoucir*). Il y a approximativement 546000 paires mot/lemme.

– un dictionnaire de synonymes, permettant d'identifier les synonymes possibles d'un mot (par exemple *abandon -> cession*). Il y environ 15000 mots associés à des synonymes.

– un ensemble de dictionnaires contenant des associations mot-dérivé, où *mot* est un substantif ou un verbe, et *dérivé* est un dérivé nominal, verbal ou adjectivial (par exemple *ajourner -> ajournement*). Il y a approximativement 11000 associations.

Pour le dictionnaire de synonymes, certains synonymes sont assez éloignés du sens que peut avoir le mot en général, ce qui peut conduire à de mauvaises transformations selon le contexte d'une phrase. Une évaluation de l'impact des synonymes est effectuée dans le chapitre 8.

5.4 Traitements de structuration multi-niveaux

Les traitements de structuration multi-niveaux (ou pré-traitements) correspondent à la première partie du module de réordonnancement. L'ensemble des analyses nécessaires à effectuer sur la question et les passages traités sont faits dans cette partie. Le calcul du coût de transformation s'appuie sur ces pré-traitements dans la seconde partie du module de réordonnancement. Le schéma 5.4 illustre les pré-traitements réalisés.

5.4.1 Description générale

L'objectif de cette partie est de procéder à une analyse préalable de chaque question et des passages associés aux réponses candidates. Ces pré-traitements sont donc effectués sur chaque passage d'une réponse candidate. Dans les traitements proposés pour la suite, nous avons comme objectif d'enrichir

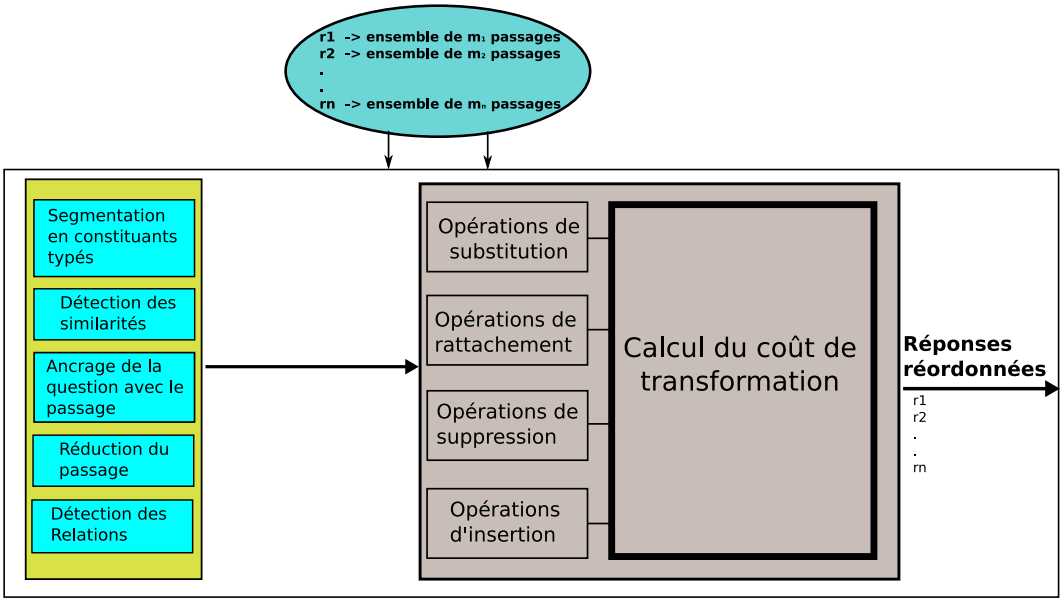

FIG. 5.4 – Schéma de fonctionnement du réordonnanceur. Les pré-traitements sont situés à gauche.

les annotations issues de Ritel, notamment en ce qui concerne la représentation de la structure de la question et des passages des réponses candidates. De plus, nous voulons aussi identifier les similarités sémantiques entre la question et chaque passage traité. Enfin, dans le cas où le passage traité est trop long, nous procédons à une réduction de sa taille. Nous décrivons dans cette section l'architecture générale des traitements de structuration multi-niveaux présentés dans la section 5.2.1. Ces traitements sont ensuite présentés de manière plus détaillée dans la section 5.4.2.

Les pré-traitements permettent de préparer le futur calcul du coût de transformation du passage évaluée. Ainsi, pour une question donnée et les passages associés aux réponses candidates, nous appliquons ces traitements qui sont réalisés en cinq modules :

– la segmentation et l'annotation de la question et des passages

– l'identification des similarités entre les mots de la question et des passages

– l'ancrage des segments de la question avec les segments des passages

– la réduction des passages

– la détection des relations entre segments

Le module de segmentation en constituants typés a déjà été décrit dans la section 4.2 du chapitre 4.

Cette segmentation s'appuie sur le formalisme défini. Deux types principaux de segments existent, nominal et verbal. Les segments nominaux sont composés par ailleurs de quatres sous-types. A partir de ce formalisme, nous avons généré deux modèles : un pour les documents et un pour les questions. Ce modèle est généré en utilisant des champs conditionnels aléatoires (CRFs) et un corpus d'apprentissage annoté manuellement.

Les similarités entre mots dépendent directement des ressources linguistiques que nous avons décrites dans la section 5.3. Elles permettent de représenter deux mots ayant un sens proche, en se servant des transformations possibles fournies par les ressources lexicales. Il y a quatre types de similarité. Nous listons ces types avec des exemples ci-dessous :

– identité, lorsque deux mots sont identiques.
– lemme, lorsque les formes lémmatisées des deux mots sont identiques : *mangeait -> manger*.
– morpho-syntaxique, lorsque qu'un mot est équivalent à l'un des dérivés morpho-syntaxiques d'un autre mot : *clamer -> clameur*.
– synonyme, lorsqu'un mot peut être considéré comme synonyme d'un autre mot : *abandonner -> abdiquer*.

Par ailleurs, seuls les mots ayant une importance sémantique sont considérés lors de la recherche des similarités. On utilise les annotations produites par Ritel [Galibert 2009] pour déterminer si un mot est important. Plus généralement, les entités nommées et les expressions à mots multiples sont considérées comme mots importants. Si cette hypothèse a pour défaut de ne pas prendre en compte certains mots important sémantiquement (comme les prépositions), elle est néanmoins adaptée à notre travail.

Une hiérarchie a été fixée selon le type de la similarité. L'ordre est le suivant : identité, lemme, morpho-syntaxique, synonyme. La similarité de type identité est considérée comme la plus forte, et synonyme la plus faible. Cette hiérarchie a son importance dans nos pré-traitements, et aussi plus tard pour le calcul du coût de transformation. Un poids est fixé pour chaque similarité, identité ayant le poids le plus faible.

Les mots similaires identifiés sont ensuite utilisés pour créer les ancres entre un passage et une question. Les ancres permettent de représenter deux segments ayant un sens similaire. S'il existe une ancre entre un segment de la question et un segment du passage, cela implique qu'il existe au moins une paire de mots ayant une similarité. Une ancre doit représenter la similarité sémantique entre les deux segments : elle correspond donc à l'ensemble des paires de mots similaires entre deux segments, ainsi qu'à chaque type de similarité (identité, synonyme ...). Il existe toujours au moins une ancre entre la question et le passage. Elle correspond à celle existant entre le segment de type marqueur interrogatif de la question, et le segment du passage contenant la réponse candidate évaluée.

Dès qu'un passage est ancré avec la question, il est traité phrase par phrase (on se base sur la ponctuation, même dans le cas de transcriptions écrites de données orales). Le passage est réduit en supprimant les phrases trop éloignées de la réponse candidate. La méthode d'extraction des passages

par Ritel fait que la recherche des éléments critiques les plus proches d'une réponse peut conduire à l'extraction de grands passages, malgré les limites de taille du système.

Enfin, les relations sont ajoutées entre les segments typés. La détection des relations a déjà été décrite dans la section 4.3 du chapitre 4. Le formalisme de ces relations est simple et dépendant du formalisme de segmentation. Il y a donc deux types principaux, nominal et verbal, et deux sous-types, temps et lieu. Nous utilisons des règles écrites manuellement pour identifier les relations.

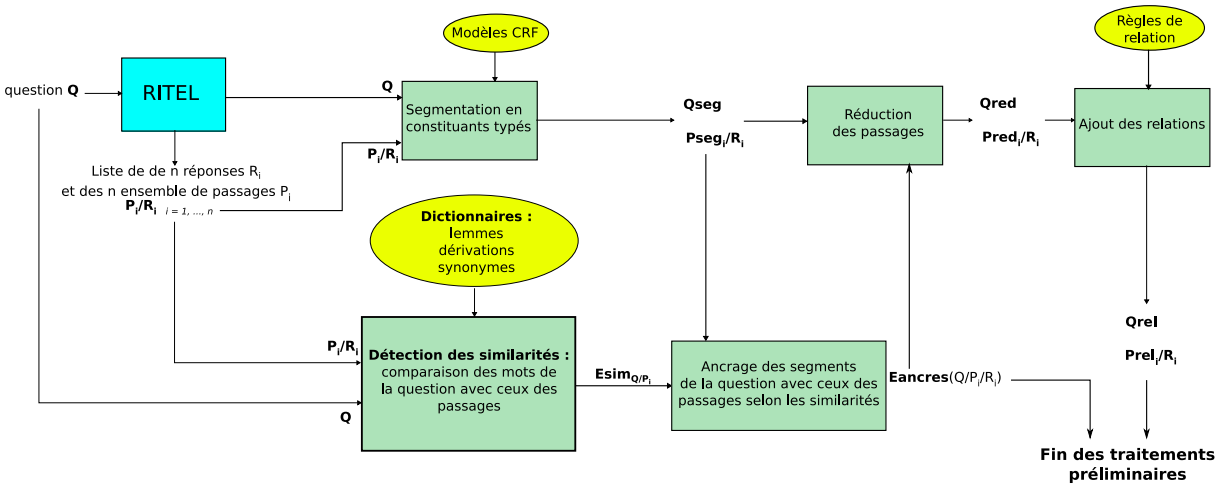

FIG. 5.5 – Architecture détaillée des traitements préliminaires.

La figure 5.5 décrit les traitements préliminaires effectués pour une question **Q**. Chaque réponse candidate fournie par Ritel est associée à un ensemble de passages d'où la réponse est extraite. Ces paires *passages/réponse* passent par une chaîne de traitements. La question est représentée dans la figure par Q et les paires *passages/réponse* par P_i/R_i, où R_i correspond à la i_{eme} réponse, et P_i à l'ensemble contenant les passages de la i_{ieme} réponse. Un ensemble de passages P_i est égal à une liste de n passages, soit $P_i = p_{i,1}, p_{i,2}, ..., p_{i,n}$. Selon ces conventions de notation, chaque question est traitée comme une paire $(Q, P_i/R_i)$.

A la fin de ces pré-traitements, l'information structurelle composée des segments typés et des relations entre segments a été ajoutée sur la paire $Q, P_i/R_i$, symbolisé par $Qrel, Prel_i/R_i$. De plus, les segments similaires ont été identifiés entre la question et les passages d'une réponse. Ces segments similaires sont représentés par l'ensemble des ancres $Eancres(Q/P_i, R_i)$. Ces deux éléments sont ensuite traités par le module de calcul du coût de transformation.

Dans les sections suivantes, nous décrivons les algorithmes principaux utilisés dans ces pré-traitements. Nous commençons par détailler les conventions de notation utilisées pour représenter les questions et les passages à différents niveaux (mots et segments). En nous appuyant sur ces conventions de notation, nous présentons les algorithmes principaux.

5.4.2 Fonctionnement algorithmique des traitements préliminaires

5.4.2.1 Conventions de notation des questions et des passages

Nous introduisons ici les conventions de notation utilisées pour représenter les questions et les passages dans les algorithmes. Nous nous appuyons sur la question Q *"Quel âge avait Nelson Mandela à sa sortie de prison ?"* et le passage $p_{1,1}$ *"A 71 ans, Nelson Mandela est libéré de prison. Il était resté enfermé pendant 27 années"*. Ce passage p_1 fait parti de l'ensemble des passages P_1 associé à la réponse candidate R_1 *71 ans*. Ces conventions sont représentées ci-dessous, dans la figure 5.6.

Q : question traitée
P_i : ensemble de taille J des passages contenant la i_{eme} réponse candidate R_i
R_i : i_{eme} réponse candidate à la question Q
$p_{i,j}$: j_{eme} passage de l'ensemble de passage P_i

FIG. 5.6 – Conventations de notation des questions et passages

Ces notations sont par ailleurs utilisées pour définir les différents états d'une question ou d'un passage. Ainsi nous écrivons $Qseg$ une question Q à laquelle les segments typés sont ajoutés. $Pseg_i$ correspond quant à lui à l'ensemble de passages segmentés de la réponse R_i, $pseg_{i,j}$ étant un passage segmenté. Similairement, nous avons $Qred$ et $Pred_i$ lorsque la réduction des passages est appliquée, et $Qrel$ et $Prel_i$ lorsque les relations sont ajoutées.

En nous appuyant sur ces notations, nous définissons les phrases, segments, et mots des questions et passages. Nous utilisons dans les prochaines sections ces trois éléments pour définir nos algorithmes. La s_{eme} phrase d'un passage $p_{i,j}$ est notée $PH_s^{p_{i,j}}$. Pour le passage $p_{1,1}$, la phrase $PH_2^{p_{1,1}}$ est donc *"Il était resté enfermé pendant 27 années."*.

Le x_{eme} segment d'une question Q est noté S_x^Q. Similairement, le y_{eme} segment d'un passage $p_{i,j}$ est noté $S_y^{p_{i,j}}$. Pour le passage $p_{1,1}$, le segment $S_2^{p_{1,1}}$ est *Nelson Mandela*.

Enfin, le k_{eme} mot d'une question Q est noté M_k^Q, et le l_{eme} mot d'un passage $M_l^{p^{i,j}}$. Pour la question Q, le mot M_5^Q est *Mandela*. Toutes ces notations sont représentées dans la figure 5.7.

5.4.2.2 Segmentation typée

Le module de segmentation des phrases et des questions en constituants typés est présenté dans la section 4.2 du chapitre 4. La segmentation est donc effectuée en s'appuyant sur le modèle correspondant, généré en utilisant des CRFs. Ce module prend donc en entrée une question Q ainsi que la réponse candidate R_i et l'ensemble des passages P_i. La question et les phrases des passages sont segmentées.

$PH_s^{p_{i,j}}$: s_{eme} phrase du passage j contenant la i_{eme} réponse candidate
S_x^Q : x_{eme} segment de la question Q
$S_y^{p_{i,j}}$: y_{eme} segment du passage contenant la i_{eme} réponse candidate
M_k^Q : k_{eme} mot de la question Q
$M_l^{p^{i,j}}$: l_{eme} mot du passage contenant la i_{eme} réponse candidate

FIG. 5.7 – Conventations de notation des phrases, segments et mots

L'exemple de la figure 5.8 illustre cette segmentation.

Question Q : *"Quel âge avait Nelson Mandela à sa libération de prison ?*
Question $Qseg$: *"[SMI] Quel âge [/SMI] [SV] avait [/SV] [SN] Nelson Mandela [/SN] [SN] à sa libération de prison [/SN] ?*

FIG. 5.8 – Exemple d'une question avant et après ajout des segments typés. SMI : segment marqueur interrogatif ; SV : segment verbal ; SN : segment nominal.

On représente par $Qseg$ et $Pseg_i$ la question et l'ensemble des passages segmentés. La figure 5.9 représente le module de segmentation en constituants typés.

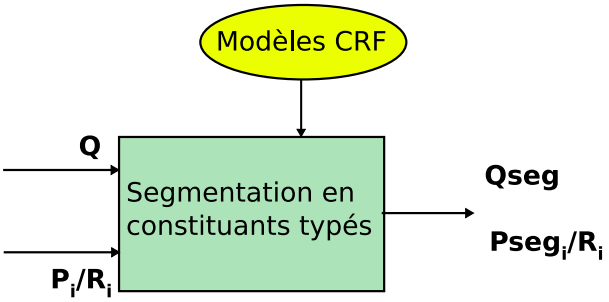

FIG. 5.9 – Module de segmentation des questions et des passages, avec en entrée la question Q et l'ensemble des passages P_i et la réponse R_i, et en sortie la question et les passages segmentés.

5.4.2.3 Détection des similarités

Le module de détection des similarités prend en entrée une question Q ainsi que la réponse candidate R_i et l'ensemble des passages P_i qui lui est associé. L'objectif est d'identifier les mots de la question et les mots de chaque passage partageant une même information, c'est à dire les mots de la question ayant un sens similaire avec ceux du passages. Quatre types de similarité sont définis : identité, lemme, morpho-syntaxique et synonymie. S'il existe une similarité entre deux mots, *t* définit le type de cette

similarité. Ainsi, dans l'exemple 5.10, le mot *libération* a un type de similarité *t morpho-syntaxique* avec le mot *libéré*.

Question Q : *"Quel âge avait Nelson Mandela à sa* **libération** *de prison ?"*
Passage $p_{i,j}$: *"A 71 ans, Nelson Mandela est* **libéré** *après 27 années de prison."*

FIG. 5.10 – Exemple d'une question et d'un passage utilisé pour illustrer le fonctionnement de la détection des similarités ; (libération, libéré) : similarité morpho-syntaxique.

Les similarités détectées entre la question et chaque passage traité sont stockées dans l'ensemble des similarités $Esim_{Q/P_i}$. Concrètement, cet ensemble est représenté par un tableau de taille J contenant dans chaque case les similarités entre le passage $p_{i,j}$ et la question Q. La figure 5.11 représente le module de détection des similarités.

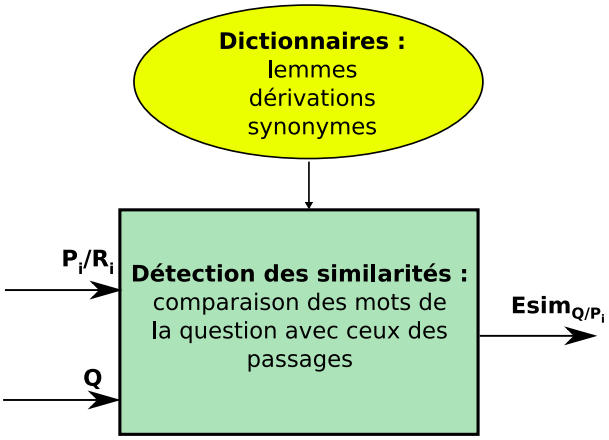

FIG. 5.11 – Module de détection des similarités, avec en entrée la question Q et l'ensemble des passages P_i et la réponse R_i, et en sortie un ensemble contenant les similarités des mots de la question avec les mots de chaque passage de P_i.

Le module de détection des similarités compare chaque mot de la question avec chaque mot de chaque passage de P_i. Nous représentons les similarités entre une question et un passage à partir d'un tableau à deux dimensions de taille K et L : K correspond au nombre de mots dans la question, et L au nombre de mots dans le passage. Ainsi, chaque ligne du tableau correspond à un mot de la question, et chaque colonne à un mot du passage. Chaque case contient alors le type de la similarité entre les deux mots correspondant de la question, s'il existe une similarité. Sinon, la case est vide. Ainsi le tableau des similarités entre les mots de la question Q et du passage $p_{i,j}$ est noté $similaritésMots_{Q/p_{i,j}}[K, L]$. Dans l'exemple 5.10, la question a 10 mots et le passage 12. Dans le tableau $similaritésMots_{Q/p_{i,j}}[K, L]$, K est donc égal à 10 et L à 12. La case d'indice *k=8* et *l=7* correspond à la similarité entre les mots *libération* et *libéré*, qui est de type *morpho-syntaxique*. Ainsi, chaque case de $Esim_{Q/P_i}$ contient un tableau de similarités $similaritésMots_{Q/p_{i,j}}[K, L]$. Ces conventions de notation sont représentées dans la figure 5.12.

$Esim_{Q/P_i}$: ensemble des similarités entre les mots de la question Q avec chaque passage de P_i
$similaritésMots_{Q/p_{i,j}}[K, L]$: tableau des similarités entre les mots de la question Q et du passage $p_{i,j}$.

FIG. 5.12 – Convention de notation des éléments utilisés dans le module de détection des similarités

L'algorithme présenté dans la figure 5.13 représente le fonctionnement du module de détection des similarités. Pour chaque passage de P_i, l'algorithme traite l'ensemble des mots importants sémantiquement (entités nommées, expressions à mots multiples ...) de la question et du passage $p_{i,j}$ tels que repérés par Ritel. Les mots identiques ou les transformations possibles sont identifiés. Le fonctionnement est très simple : l'ensemble des mots importants de la question et du passage sont parcourus. Ainsi, des déterminants comme *de* ne sont pas pris en compte lors de la recherche des similarités. Nous considérons qu'il est pertinent de chercher les similarités uniquement sur les mots ayant une importance sémantique.

Pour chaque comparaison entre un mot de la question et un mot du passage, on cherche si une similarité existe : s'ils sont identiques, le type de la similarité est *identique*. Sinon, on appelle la fonction **comparerTransformations**, pour trouver la similarité la plus proche entre les deux mots. S'il n'existe aucune similarité, alors le type est *rien*. Le type de la similarité entre chaque paire est stocké dans le tableau à deux dimensions similaritésMots$_{Q/P_i}[K, L]$. Ainsi, le contenu de la case *k,l* nous donne la similarité entre le k_{ieme} mot de la question et le l_{ieme} mot du passage.

Une fois toutes les comparaisons possibles effectuées, le tableau de similarité nouvellement construit est stocké dans $Esim_{Q/P_i}$, et le passage suivant $p_{i,j}$ est traité.

La fonction **comparerTransformations** permet de déterminer si une transformation entre un mot de la question M_k^Q et un mot du passage $M_l^{P^i}$ existe. Si c'est le cas, la fonction retourne le type de la similarité : lemme, morpho-syntaxique ou synonyme. S'il n'existe pas de transformation, le type prend la valeur *NIL*. Cette fonction s'appuie sur les ressources linguistiques pour identifier les transformations possibles.

5.4.2.4 Ancrages des segments

Le module d'ancrage des segments prend en entrée l'ensemble des similarités $Esim_{Q/P_i}$ entre la question Q et chaque passage de P_i. La réponse candidate ainsi que la question et l'ensemble des passages annotés avec les segments typés sont eux aussi fournis en entrée. On les note respectivement R_i, $Qseg$ et $Pseg_i$. Pour chaque passage traité, l'objectif est d'ancrer les segments de la question avec ceux du passage. On définit une ancre comme étant un lien unissant deux segments ayant un sens proche. Pour déterminer si un segment de la question a un sens proche d'un segment du passage,

Variables d'entrée :
– Q : question
– P_i : ensemble de passages contenant la réponse candidate
– R_i : réponse candidate
Variable de retour :
$Esim_{Q/P_i}$: ensemble des similarités entre les mots de la question Q avec les mots de chaque passage de P_i

Corps de la fonction :

pour chaque $p_{i,j}$ de P_i, j = 1, ... , J
  pour chaque M_k^Q de Q, k = 1, ... , K
    pour chaque $M_l^{p_{i,j}}$ de $p_{i,j}$, l = 1, ... , L
      si $M_k^Q = M_l^{p_{i,j}}$ **alors**
        t = identité
      sinon
        t = comparerTransformations(M_k^Q, $M_l^{p_{i,j}}$)
      fsi
      similaritésMots$_{Q/p_{i,j}}[k,l]$ = t
    finpour
  finpour
  $Esim_{Q/P_i}[j]$ = similaritésMots$_{Q/p_{i,j}}$
finpour
retourne $Esim_{Q/P_i}$

FIG. 5.13 – Algorithme de détection des similarités entre mots

on s'appuie sur l'ensemble des similarités $Esim_{Q/P_i}$. Si un segment de la question contient au moins un mot identifié comme étant similaire à un mot d'un segment du passage, alors ces deux segments sont considérés comme ayant un sens similaire. On ancre alors ces deux segments. Par exemple, l'ancre A entre le segment *le gros chat* et le segment *le gros félin* est décrite ci-dessous :

Ancre A :
Segment 1 : *le gros chat*
Segment 2 : *le gros félin*
Similarité 1 : (gros, gros), t : identité
Similarité 2 : (chat, félin), t : synonyme

L'objectif est d'identifier les groupes de mots avec un sens proche, et ainsi de déterminer si le sens d'un passage est proche du sens d'une question. Néanmoins, il peut y avoir plus d'un mot du passage avec un sens proche d'un mot de la question. Par exemple, le segment *[SN] un groupe varié [/SN]* a deux mots similaires avec le segment *[SN] un groupe diversifié [/SN]* : *varié* a une similarité de type

synonyme avec *diversifié*, et on retrouve le mot *groupe* une similarité de type identité. Notre objectif étant de représenter au mieux la similarité entre deux segments, une ancre contient l'ensemble des types de similarité.

Ce module d'ancrage des segments fournit en sortie un ensemble $Eancres(Q/P_i/R_i)$ contenant les ancres entre les segments de la question et les segments de chaque passage traité. Cet ensemble est traité comme un tableau de j cases. La figure 5.14 illustre ce module.

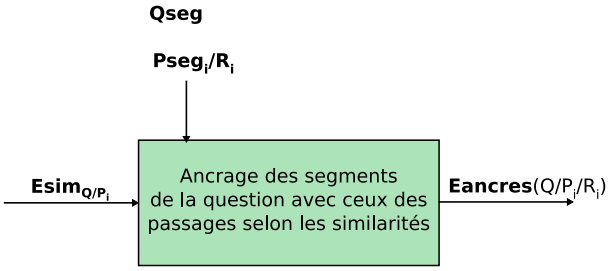

FIG. 5.14 – Module d'ancrage des segments, avec en entrée la question segmentée $Qseg$, l'ensemble des passages segmentés $Pseg_i$ et la réponse R_i, et en sortie un ensemble contenant les segments de la question ancrés avec les segments de chaque passage de $Pseg_i$.

Pour chaque passage traité, les segments ancrés sont stockés dans un tableau $ancresSegments_{Q/p_{i,j}}$ à deux dimensions de taille X et Y, respectivement le nombre de segments dans la question et dans le passage. Ce tableau contient les similarités entre chaque segment de la question et du passage. Chaque case contient un tableau de similarités $similaritésMotsSegments_{S_x^Q/S_y^{p_{i,j}}}[R,S]$ tel qu'il a été décrit dans la section 5.4.2.3. Chaque case correspond à l'ancre existant entre un segment de la question et du passage (si le tableau contenu dans la case n'est pas vide).

Pour construire ce tableau, le module génère un tableau $similaritésMotsSegments_{S_x^Q/S_y^{p_{i,j}}}$ pour chaque comparaison entre un segment de la question et un segment du passage. Ce tableau est à deux dimensions de taille R et S qui correspondent respectivement au nombre de mots dans le segment de la question et au nombre de mots dans le segment du passage. Concrètement, chaque ligne correspond à un mot de la question, et chaque colonne à un mot du passage. Une case contient alors le type de similarité existant entre les deux mots. S'il n'y a pas de similarité, alors la case contient le label *NIL*. Nous rappelons les différentes conventions de notation présentées dans cette section dans la figure 5.15.

L'exemple 5.16 illustre le fonctionnement de ce module d'ancrage. Si l'on compare le segment *[SN] à sa libération de prison [/SN]* avec le *[SV] est libéré [/SV]*, *libéré* et *libération* ont une similarité morpho-syntaxique. Ainsi, le tableau $similaritésMotsSegments_{S_4^Q/S_3^{p_{1,1}}}$ a 5 lignes et 2 colonnes. La case $similaritésMotsSegments_{S_x^Q/S_y^{p_{i,j}}}[5,2]$ contient alors la similarité entre *libéré* et *libération*. Il n'y a aucune autre similarité entre ces deux segments, les autres contiennent donc le label *NIL*. Une ancre est alors créée entre ces deux segments.

$Eancres(Q/P_i/R_i)$: ensemble des ancres entre la question Q et chaque passage de P_i
$ancresSegments_{Q/p_{i,j}}[X,Y]$: tableau des ancres entre la question Q et du passage $p_{i,j}$
$similaritésMotsSegments_{S_x^Q/S_y^{p_{i,j}}}[R,S]$: tableau des mots similaires entre le segment de la question S_x^Q et le segment du passage $S_y^{p_{i,j}}$

FIG. 5.15 – Convention de notation des éléments utilisés dans le module d'ancrage

Question $Qseg$: *"[SMI] Quel âge [/SMI] [SV] avait [/SV] [SN] Nelson Mandela [/SN] [SN] à sa libération de prison [/SN] ?"*
Passage $pseg_{i,j}$: *"[ST] A 71 ans [/ST] , [SN] Nelson Mandela [/SN] [SV] est libéré [/SV] [ST] après 27 années [/ST] [SN] de prison [/SN] ."*

FIG. 5.16 – Exemple d'une question et d'un passage utilisé pour illustrer le fonctionnement de l'ancrage des segments.

L'algorithme de la figure 5.17 illustre le fonctionnement du module d'ancrage. Il utilise le tableau des similarités créés par l'algorithme de détection des similarités entre mots. Le but est de créer des ancres entre chaque segment de la question et du passage partageant des mots similaires. Une ancre est donc caractérisée par un segment de la question, un segment du passage, et l'ensemble des mots similaires. De plus, une ancre entre le segment de la question de type marqueur interrogatif et le segment du passage contenant la réponse évaluée est automatiquement ajoutée. L'objectif du module d'ancrage étant d'identifier les groupes de mots ayant un sens similaire, on émet l'hypothèse que les marqueurs interrogatifs d'une question correspondent à la réponse candidate évaluée. Cet ancrage a par ailleurs une importance dans le module de réduction des passages, et le calcul du coût de transformation.

Le fonctionnement de l'algorithme est décrit rapidement dans la suite. On parcourt chaque segment du passage et de la question. Pour chaque paire de segment, on interroge le tableau similaritésMots$_{Q/p_{i,j}}[K,L]$ par le biais de la fonction **similaritésSegments** qui récupére l'ensemble des similarités entre les mots importants contenus dans ces deux segments. Cette fonction prend en paramètre les deux segments traités ainsi que le tableau des similarités correspondant au passage et à la question traités. A partir des mots identifiés par le module de détection des ancres, un tableau $similaritésMotsSegments_{S_x^Q/S_y^{p_{i,j}}}$ est créé.

Il peut arriver que l'on ait identifié plusieurs relations de similarité pour un mot de la question ou du passage. La fonction **enleverConflits** est appliquée. Elle prend en paramètre le tableau $similaritésMotsSegments_{S_x^Q/S_y^{P_i}}$ et permet de résoudre ce type de conflit : la similarité avec le poids le plus faible est choisie. La hiérarchie est la suivante : identité, lemme, morpho-syntaxique, synonyme.

Après avoir enlevé les conflits, si le tableau n'est pas vide, alors on créé une ancre entre ces deux segments. Cette ancre est caractérisée par les segments de la question et du passage ancrés, et l'ensemble

des similarités. On stocke donc cet ensemble dans le tableau à deux dimensions des ancres dans la case correspondant aux indices des deux segments. Ainsi, comme pour le tableau des similarités, le contenu de la case *k,l* nous donne l'ancre entre le k_{ieme} segment de la question et le l_{ieme} segment du passage, et donc l'ensemble des similarités entre ces deux segments.

Variables d'entrée :
- $Esim_{Q/P_i}$: ensemble des similarités entre les mots de la question Q avec les mots de chaque passage de P_i
  .
- $Qseg$: question segmentée
- R_i : réponse candidate
- $Pseg_i$: passage segmenté contenant la i_{eme} réponse candidate R_i

Variable de retour :

$Eancres_{Q/P_i/R_i}$: ensemble des ancres entre la question Q et les passages de P_i.

Corps de la fonction :

pour chaque $p_{i,j}$ de P_i, j = 1, ... , J
  pour chaque $S_y^{p_{i,j}}$ de $p_{i,j}$, y = 1, ... , Y
    pour chaque S_x^Q de Q, x = 1, ... , X
      similaritésMotsSegments$_{S_x^Q/S_y^{p_{i,j}}}$ = similaritésSegments($S_y^{p_{i,j}}$,S_x^Q,$Esim_{Q/P_i}[j]$)
      enleverConflits(similaritésMotsSegments$_{S_x^Q/S_y^{p_{i,j}}}$)
      si similaritésMotsSegments$_{S_x^Q/S_y^{p_{i,j}}}$ n'est pas vide **alors**
        $ancresSegments_{Q/p_{i,j}}[x,y]$ = similaritésMotsSegments$_{S_x^Q/S_y^{p_{i,j}}}$
      finsi
    finpour
  finpour
$Eancres_{Q/P_i/R_i}[j]$ = $ancresSegments_{Q/p_{i,j}}$
finpour
retourne $Eancres_{Q/P_i/R_i}$

FIG. 5.17 – Algorithme de création des ancres

5.4.2.5 Réduction du passage

Le module de réduction des passages prend en entrée une question segmentée $Qseg$, un ensemble de passages segmentés $Pseg_i$ et une réponse candidate R_i. Par ailleurs, il prend aussi la sortie du module d'ancrage des segments, à savoir l'ensemble $Eancres_{Q/P_i/R_i}$ des ancres entre la question $Qseg$ et les passages de $Pseg_i$. L'objectif principal de ce module est de réduire chacun des passages traités, c'est à dire d'enlever les phrases que l'on ne juge pas pertinentes par rapport à la réponse candidate évaluée. Des passages trop longs ont pour effet de pénaliser notre méthode de réordonnancement. De

plus, nous estimons qu'une phrase, même si elle contient des éléments critiques de la question, n'est pas en rapport avec la réponse candidate d'un passage si elle en est trop éloignée. Ce module retourne donc en sortie une question $Qred$, un ensemble de passages épurés $Pred_i$, et une réponse R_i. La figure 5.18 illustre ce module.

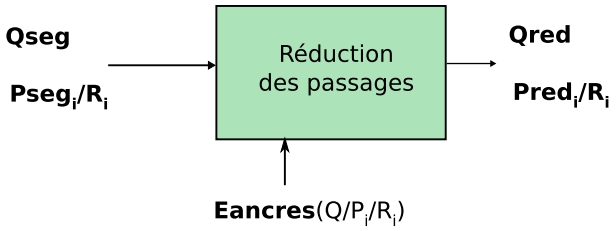

FIG. 5.18 – Module de réduction des passages, avec en entrée la question segmentée $Qseg$, l'ensemble des passages segmentés $Pseg_i$ et une réponse candidate R_i, et en sortie une question $Qred$, un ensemble de passages épurés $Pred_i$ et une réponse candidate R_i.

Concrètement, nous procédons à la réduction d'un passage en deux phases. Tout d'abord, nous traitons les phrases précédant la phrase contenant la réponse candidate. Si deux phrases de suite ne contiennent pas d'éléments similaires de la question alors elles sont supprimées ainsi que le reste des autres phrases précédentes. Sinon, on recommence le même traitement à partir de la dernière phrase contenant un élément critique. Une fois les phrases précédant la phrase réponse sont traitées, on utilise la même approche pour les phrases suivantes.

Ce type d'approche peut amener à perdre de l'information importante. Néanmoins, étant donné que nous n'avons pas de gestion d'anaphores, il est difficile de relier de l'information d'une phrase à une autre. Cette méthode de réduction est donc une première étape adaptée à notre module de réordonnancement. L'exemple 5.19 illustre le fonctionnement de ce module. La phrase contenant la réponse candidate *1990* à la question *"En quelle année fut libéré Nelson Mandela ?"* est la (4). Les éléments de la question sont *Nelson Mandela* et *libéré*. Ainsi, les phrases (1), (4) et (5) contiennent des éléments de la question. Les phrases (3) et (2) ne contiennent pas d'éléments critiques. De ce fait, ces deux phrases sont supprimées, ainsi que la phrase (1).

(Q) En quelle année fut libéré Nelson Mandela ?

(1) Nelson Mandela a été le premier président noir d'Afrique du Sud.
(2) Ce pays a été marqué par l'apartheid.
(3) Il a fallu attendre 1990 avant que la situation ne s'améliore.
(4) En 1990, Nelson Mandela est libéré de prison.
(5) Nelson Mandela y était enfermé depuis 27 ans.

FIG. 5.19 – Exemple d'un passage à réduire. La réponse candidate est *1990*, et la phrase réponse est donc la (4). Après réduction, les phrases (1), (2) et (3) sont supprimées.

La figure 5.20 illustre le fonctionnement du module de réduction. L'algorithme traite les phrases du passage (délimitées par un point). On va d'abord identifier la phrase contenant la réponse évaluée par le biais de la fonction **phraseReponse**. Cette fonction prend en entrée le passage $p_{i,j}$ traité et retourne l'indice de la phrase contenant la réponse candidate.

Puis, chaque phrase autour de la *phrase réponse* va être analysée pour déterminer si elle contient des éléments de la question. Les ancres indiquant les segments d'un passage contenant des éléments de la question, cet algorithme s'appuie sur les ancres détectées. Le traitement est d'abord effectué sur les phrases précédant la *phrase réponse*, puis sur les phrases suivantes. Pour déterminer si une phrase contient un élément de la question, on s'appuie sur la fonction **estAncree**, qui permet de déterminer si l'un des segments de la phrase traitée est ancrée avec la question. Cette fonction prend en entrée l'ensemble des ancres $Eancres_{Q/P_i/R_i}$ et une phrase $PH_s^{p_{i,j}}$.

Si deux phrases de suite ne contiennent aucun élément, la recherche s'arrête dans le sens actuellement traité (phrases précédant la réponse, ou après la réponse). Une fois les deux sens traités, on supprime alors les phrases du passage en utilisant la fonction **réduirePassage**. Cette fonction prend en paramètre le passage $p_{i,j}$ à réduire, ainsi que deux entiers *borneinf* et *bornesup* indiquant quelles phrases du passages doivent être supprimées. Une fois le passage traité, il est mis à jour (réduit) dans l'ensemble des passages P_i.

5.4.2.6 Identification des relations entre segments

Le module d'identification des relations entre segments est présenté dans la section 4.3 du chapitre 4. Le formalisme des relations est simple et dépendant des types des segments. Nous avons deux types principaux, nominal et verbal, et deux sous-types, temps et lieu. La détection des relations est effectuée par le biais de règles écrites manuellement. Ce module prend en entrée une question $Qred$ ainsi que la réponse candidate R_i et l'ensemble des passages $Pred_i$. Les relations sont ajoutées sur la question et les passages. L'exemple de la figure 5.21 montre les relations ajoutées à un passage. Les relations temporelles ne sont pas représentées sur cette figure pour éviter de la surcharger. Les couleurs représentent le type du segment : bleu correspond aux segments temporels, jaune aux segments nominaux, et rouge aux segments verbaux.

On représente par $Qrel$ et $Prel_i$ la question et l'ensemble des passages auxquels les relations ont été ajoutées. La figure 5.22 représente le module d'ajout de ces relations.

5.4.3 Conclusion sur les traitements de structuration multi-niveaux

Nous avons présenté l'ensemble des traitements préliminaires effectués par le réordonnanceur. En plus de la segmentation des phrases et questions et de l'ajout de relations entre ces segments, trois traitements supplémentaires sont appliqués dans cette phase préliminaire : l'identification des simila-

Variables d'entrée :
– $Eancres_{Q/P_i/R_i}$: ensemble des ancres entre la question et les passages de P_i
– Q : question segmentée
– P_i : ensemble de passages segmentés
– R_i : réponse candidate
Variable de retour :
Q, P_i : les passages de P_i ont été réduits.
Corps de la fonction :

pour chaque $p_{i,j}$ de P_i, j = 1, ... , J
  entier indrep = phraseReponse($p_{i,j}$)
  entier nbphrases_sansancres = 0
  entier borneinf = indrep - 1
  entier bornesup = indrep + 1
  pour chaque $PH_s^{p_{i,j}}$ de $p_{i,j}$, s = indrep - 1 à 0
    si estAncree($PH_s^{p_{i,j}}$,$Eancres_{Q/P_i/R_i}$) == vrai **alors** nbphrases_sansancres = 0
    sinon nbphrases_sansancres = nbphrases_sansancres + 1
    si nbphrases_sansancres == 2 **alors sortir de la boucle**
    borneinf = borneinf + 1
  finpour
  nbphrases_sansancres = 0
  pour chaque $PH_s^{p_{i,j}}$ de $p_{i,j}$, s = indrep + 1 à S
    si estAncree($PH_s^{p_{i,j}}$,$Eancres_{Q/P_i/R_i}$) == vrai **alors** nbphrases_sansancres = 0
    sinon nbphrases_sansancres = nbphrases_sansancres + 1
    si nbphrases_sansancres == 2 **alors sortir de la boucle**
    bornesup = bornesup + 1
  finpour
  $P_i[j]$ = réduirePassage($p_{i,j}$,borneinf,bornesup)
finpour
retourne Q, P_i, R_i

FIG. 5.20 – Algorithme de réduction du passage

rités, la création des ancres, et l'épuration du passage. Par soucis de compréhension nous donnons un exemple dans cette section de l'ensemble des traitements effectués par ces trois modules.

Pour illustrer notre propos, nous nous appuyons sur la question et le passage de la figure 5.23. La réponse candidate évaluée est *71 ans*.

Le module d'identification des similarités entre mots détecte deux phrases contenant des éléments de la question, la (2) et la (5). Dans la (2), on trouve une similarité *identité* sur *Nelson*, *Mandela* et

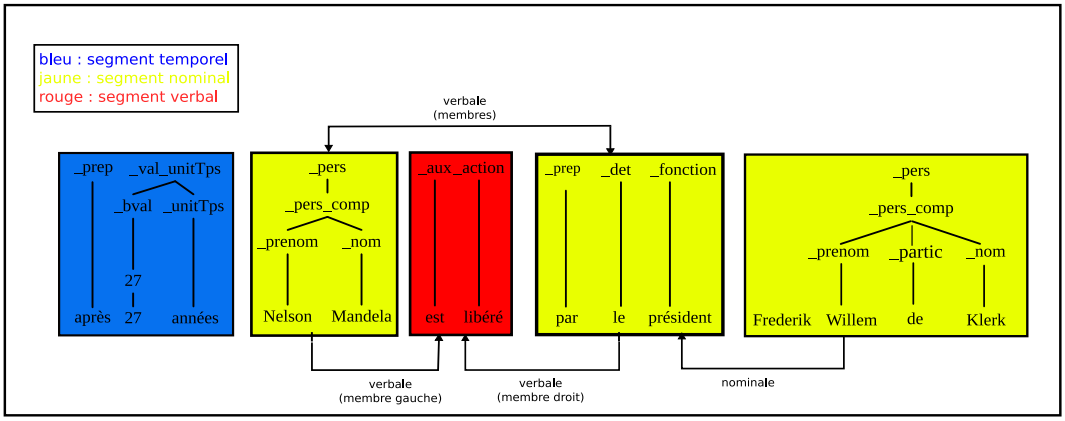

FIG. 5.21 – Exemple de relations entre segments typés correspondant au passage *"Après 27 années Nelson Mandela est libéré par le président Frederik Willem de Klerk ."*

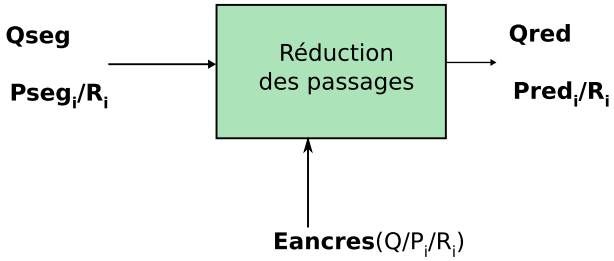

FIG. 5.22 – Module d'identification des relations des questions et passages, avec en entrée la question $Qred$ et l'ensemble des passages $Pred_i$ et la réponse R_i, et en sortie la question $Qrel$ et l'ensemble des passages $Prel_i$.

(Q) Quel âge avait Nelson Mandela à sa sortie de prison ?

(1) Une nouvelle importante pour l'Afrique du Sud.
(2) A **71 ans**, Nelson Mandela est sorti après 27 années de prison.
(3) L'actuel président d'Afrique du Sud en a fait l'annonce ce matin.
(4) Le pays tout entier est en liesse.
(5) Nelson Mandela étant libéré, la situation devrait rapidement évoluer.

FIG. 5.23 – Exemple de référence utilisé pour illustrer le fonctionnement des traitements préliminaires. La réponse candidate évaluée est *71 ans*

prison, et une similarité *morpho-syntaxique* entre *sortie* et *sorti*. Dans la (5), nous avons une similarité *identité* sur *Nelson* et *Mandela*, et une similarité *synonymie* entre *sortie* et *libéré*.

En s'appuyant sur ces similarités, le module d'ancrage des segments va créer les ancres appropriées. Les phrases (2) et (5) segmentées sont :

– *"[ST] A 71 ans [/ST] [SN] Nelson Mandela [/SN] [SV] est sorti [/SV] [ST] après 27 années [/ST] [SN] de prison [/ST]"* ;
– *"[SN] Nelson Mandela [/SN] [SV] étant libéré [/SV] [SN] la situation [/SN] [SV] devrait rapidement évoluer [/SV]"*.

La question segmentée a la forme *"[SMI] Quel âge [/SMI] [SV] avait [/SV] [SN] Nelson Mandela [/SN] [SN] à sa sortie de prison [/SN] ?"*. Entre la phrase (2) et la question, le système crée les ancres suivantes :

– Ancre entre le segment de la réponse *[ST] A 71 ans [/ST]* et le segment marqueur interrogatif *[SMI] Quel âge [/SMI]*
– Ancre entre les deux segments *[SN] Nelson Mandela [/SN]*, contenant deux similarités *identité*
– Ancre entre le segment *[SV] est sorti [/SV]* et le segment *[SN] à sa sortie de prison [/SN]*, contenant une similarité *morpho-syntaxique*
– Ancre entre le segment *[SN] de prison [/ST]* et le segment *[SN] à sa sortie de prison [/SN]*, contenant une similarité *identité*

Enfin, entre la phrase (5) et la question, le système créé les ancres suivantes :

– Ancre entre les deux segments *[SN] Nelson Mandela [/SN]*, contenant deux similarités *identité*

– Ancre entre le segment *[SV] étant libéré [/SV]* et le segment *[SN] à sa sortie de prison [/SN]*, contenant une similarité *synonymie*.

Une fois les ancrages effectués, le passage est traité par le module de réduction des passages. La première phase traite les phrases avant la phrase contenant la réponse, la (2) : la phrase (1) n'étant pas ancrée, on la supprime du passage. Puis, la seconde phase traite les phrases suivant la phrase réponse : la (3), la (4) et la (5). Les phrases (3) et (4) n'étant pas ancrées, on supprime les trois phrases. Le passage n'est donc composé que de la phrase (2).

La figure 5.24 représente les différentes ancres existant entre la question *"Quel âge avait Nelson Mandela à sa sortie de prison ?"* et le passage réduit *"A 71 ans, Nelson Mandela est sorti après 27 ans de prison.*". Nous ne représentons pas les relations entre les segments sur cette figure, pour ne pas surcharger le schéma. La question, le passage réduit et leurs ancres vont ensuite être traités par le module de calcul du coût de transformation.

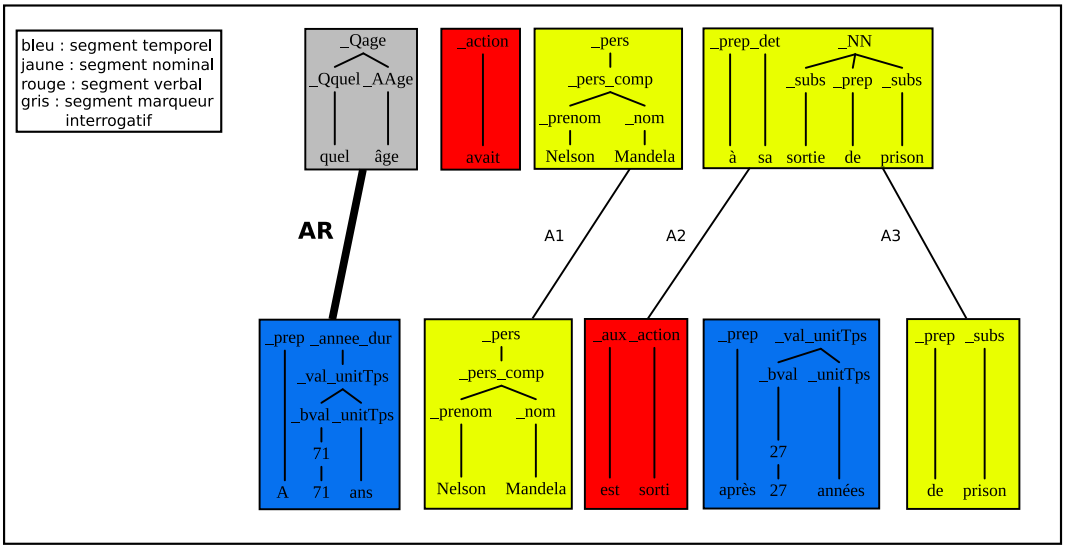

FIG. 5.24 – Ancrages entre la question *“Quel âge avait Nelson Mandela à sa sortie de prison ?”* et le passage *“A 71 ans, Nelson Mandela est sorti après 27 ans de prison.”*.
AR : ancre entre le segment du passage et le segment marqueur interrogatif.
A1 : ancre entre les segments *Nelson Mandela* du passage et de la question.
A2 : ancre entre les segments *est sorti* et *à sa sortie de prison*.
A3 : ancre entre les segments *de prison* et *à sa sortie de prison*.

5.5 Calcul du coût de transformation

Les traitements précédents visent à structurer les questions et les passages réduits via les segments typés et les relations entre ces segments, et à identifier l’information sémantique commune à la question et aux passages grâce aux ancres. Ceci sera maintenant exploité par le réordonnancement afin de calculer un nouveau score pour les candidats réponses.

L’objectif de cette partie est de calculer le coût de transformation entre une question et chaque passage d’une réponse candidate. Ce coût de transformation représente la similarité entre le passage et la question. Idéalement, la bonne réponse à une question reprend tous les termes de la question sauf le marqueur interrogatif qui est remplacé par la réponse. Ainsi plus le coût de transformation est faible, et plus le passage est considéré comme similaire à la question. Ce coût est calculé pour chaque passage d’une réponse, le score de la réponse étant le coût de transformation du passage avec la valeur la plus faible.

Pour une question et un passage, le calcul du coût de transformation dispose des informations suivantes : les segments typés, les relations entre les segments, et les ancres entre la question et le passage. Ces éléments fournissent de l’information structurelle et sémantique. A partir de ces élé-

ments, le calcul du coût de transformation est effectué en s'appuyant sur un ensemble d'opérations de transformation, inspirée du calcul de la distance d'édition présenté dans [Kouylekov & Negri 2010]. L'objectif de ces opérations est de quantifier la similarité sémantique et structurelle entre le passage et la question. Quatre types d'opérations sont utilisés : suppression, insertion, substitution et rattachement. Les trois premiers types d'opérations sont utilisés habituellement pour une distance d'édition. Les opérations de rattachement ont été ajoutées car nous estimions que les autres types n'étaient pas suffisants pour mesurer la différence structurelle entre un passage et une question. L'opération de rattachement permet de relier les segments du passage contenant des éléments de la question à la réponse candidate.

Pour chaque passage d'une réponse candidate, le coût de transformation du passage en la question est calculé à partir de ces opérations. Le système choisit le coût de transformation le plus faible des passages associés à une réponse candidate comme score de cette réponse candidate. Enfin, les réponses sont réordonnées en fonction de leur score, celle ayant obtenu le score le plus bas étant considéré comme la meilleure réponse. La figure 5.25 donne un schéma du réordonnanceur. Le calcul du coût de transformation se trouve dans la partie droite du schéma.

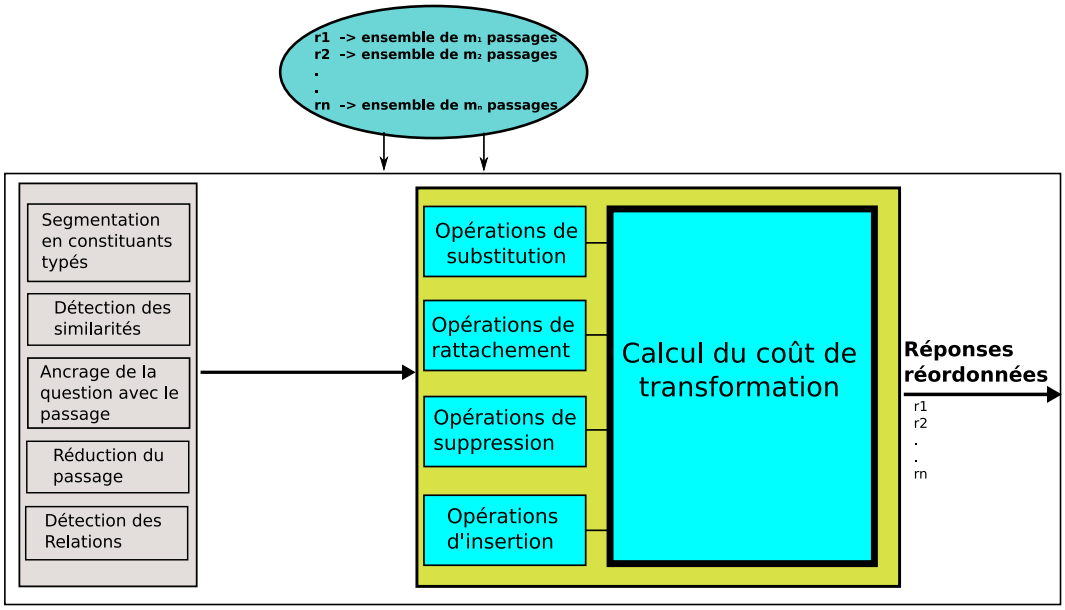

FIG. 5.25 – Schéma du réordonnanceur, le calcul du coût de transformation se situant à droite de la figure.

5.5.1 Description générale

L'objectif du calcul du coût de transformation est de mesurer la similarité entre un passage et une question. Ce passage est associé à une réponse candidate à la question. Chaque réponse candidate

est associée à un ensemble de passages. Le passage ayant le coût de transformation le plus faible est considéré comme étant le plus proche de la question, et donc celui possédant le sens le plus proche. Le coût de ce passsage est donc considéré comme étant le score de la réponse candidate.

Le coût de transformation est calculé en appliquant des opérations de transformation visant à transformer le passage en la question. Chacune des opérations appliquées a un coût qui est calculé selon différents critères propres à chaque opération. Ainsi, l'application d'une suite d'opérations de transformation aura un coût global qui lui sera associé. C'est le coût de transformation du passage. Plus ce coût est faible, plus le passage est considéré proche de la question. Le calcul de ce coût peut être divisé en deux parties. La première partie correspond à la génération des quatre types d'opérations de transformation, et la seconde partie au calcul du coût de transformation selon les opérations générées. Pour décrire ces deux phases, nous nous appuyons sur la question et le passage de l'exemple 5.26. Nous donnons aussi les ancres détectées entre la question et le passage.

Question : "*[SMI] Quel âge [/SMI] [SV] avait [/SV] [SN] Nelson Mandela [/SN] [SN] à sa libération de prison [/SN] ?*"
Passage : "*[ST] A 71 ans [/ST] , [SN] Nelson Mandela [/SN] [SV] est sorti [/SV] [ST] après 27 années [/ST] [SN] de prison [/SN] .*"
AR : ancre réponse entre le segment du passage correspondant à la réponse et le segment marqueur interrogatif.
A1 : ancre entre les segments *Nelson Mandela* du passage et de la question.
A2 : ancre entre les segments *est sorti* et *à sa sortie de prison*.
A3 : ancre entre les segments *de prison* et *à sa sortie de prison*.

FIG. 5.26 – Exemple d'une question et d'un passage segmentés pour illustrer le calcul du coût de transformation.

La génération des opérations est principalement basée sur les ancres identifiées dans les traitements préliminaires, ainsi que les segments de la question et du passage. Dans un premier temps, trois types d'opérations sont générés : substitution, suppression, et insertion. Les opérations de substitution sont déterminées en fonction des ancres. Leur objectif est de substituer le contenu des segments du passage par ceux de la question ayant des mots similaires. Le calcul du coût de cette opération dépend du nombre de mots similaires ainsi que du type de la similarité.

Les opérations de suppression ont pour objectif de supprimer tous les segments du passage qui n'ont pas été ancrés, et donc pas affectés par une opération de substitution. On supprime donc du passage les segments contenant de l'information non comprise dans la question. Les opérations d'insertion ont un objectif similaire, à savoir d'insérer l'information manquante de la question dans le passage. Le coût de ces deux types d'opération dépend à la fois des mots importants contenus dans les segments et des relations avec les autres segments.

Si on reprend l'exemple 5.26, trois opérations de substitution sont initialement générées : entre les deux segments *Nelson Mandela*, entre *est sorti* et *à sa sortie de prison*, et entre *de prison* et *à sa*

sortie de prison. Etant donné que le segment de la question *à sa sortie de prison* est relié à deux segments du passage, on génère une quatrième opération de substitution, une substitution fusion : les deux segments du passage concernés, *est sorti* et *de prison* sont substitués par *à sa sortie de prison*. Une fois les opérations de substitution générées, le système génère alors les opérations d'insertion et de suppression : l'opération d'insertion du segment de la question *avait*, et l'opération de suppression du segment du passage *après 27 années*.

Une fois ces trois types d'opérations générés, les opérations de rattachement sont créées. Ce type d'opération est directement dépendant des opérations de susbtitution. Dans les questions que nous traitons, nous estimons que les segments d'une question sont rattachés au segment marqueur interrogatif. De fait, les segments substitués du passage doivent eux aussi être en relation avec le segment réponse du passage. Ainsi, si le segment du passage traité par une opération de substitution n'est pas relié au segment réponse, une opération de rattachement est créée. L'objectif est alors de calculer le coût nécessaire pour relier les deux segments. Dans l'exemple 5.26, trois opérations de rattachement sont générés : une pour le segment *Nelson Mandela*, une pour le segment *est sorti* et une pour le segment *de prison*.

Une fois l'ensemble des opérations générées, on passe à l'étape de calcul du coût de transformation. Selon les opérations possibles, il existe potentiellement plusieurs suites d'opérations pour transformer le passage en la question. Par exemple, si nous avons une opération de substitution entre *le gros chat* et *le gros félin*, et une autre substitution entre *le gros chat* et *le chat noir*, l'application d'une de ces deux opérations annulera l'autre opération. L'idée est donc de se baser sur un algorithme de recherche pour trouver la suite d'opérations la moins coûteuse. Nous avons décidé d'utiliser l'algorithme de recherche en coût uniforme [Hall 2003]. Ce dernier va alors traiter les opérations générées, et trouver la suite d'opération la moins coûteuse, et permettre le calcul du coût de transformation entre la question et le passage.

Dans l'exemple 5.26, une suite d'opérations possible est d'abord d'appliquer l'opération de substitution entre les deux segments *Nelson Mandela*, puis l'opération de substitution fusion entre les segments du passage *est sorti* et *de prison*, et le segment de la question *à sa sortie de prison*. Les opérations de rattachement associées sont appliquées : rattachement de *Nelson Mandela*, *est sorti* et *de prison* à *A 71 ans*. On supprime enfin le segment du passage *après 27 années* et on insère le segment de la question *avait*. Chacune de ces opérations a un coût associé, qui est calculé par le système. La somme des coûts de chaque opération donne **un** coût de transformation du passage en la question. Ce coût est comparé à ceux obtenus par les autres suites de transformation possibles. Le coût de transformation le plus faible est défini comme étant le coût de transformation du passage. Chaque passage d'une réponse candidate a un coût de transformation : le plus faible est défini comme étant le score de la réponse candidate. L'architecture de ce module de calcul du coût de transformation est présentée dans la figure 5.27.

Dans les sections suivantes, nous décrivons tout d'abord chaque type d'opération de transformation utilisé. Puis nous détaillons l'étape de génération des opérations de transformation, puis l'étape de recherche de la suite d'opération de transformation la moins coûteuse par l'algorithme de recherche en

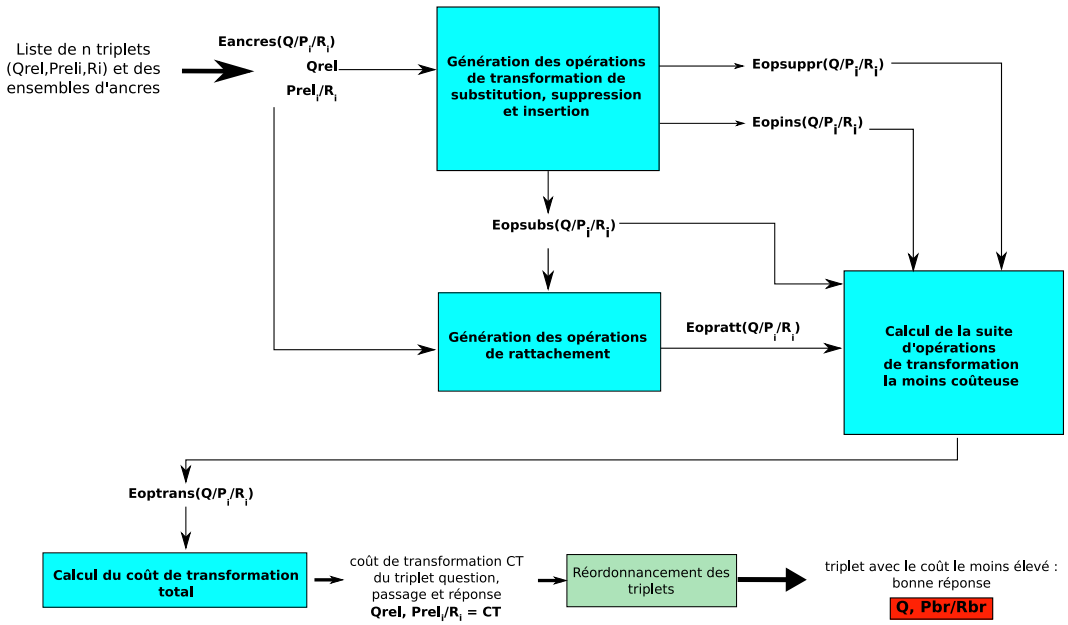

FIG. 5.27 – Architecture générale du module de calcul du coût de transformation.

coût uniforme. Pour cette dernière étape, nous présentons d'abord l'algorithme de recherche en coût uniforme, et son utilisation dans notre travail. Nous présentons ensuite une notion, que nous appelons *poids du segment*, qui permet de quantifier l'importance d'un segment. Enfin, nous présentons le calcul du coût de chaque type d'opération. Il est à noter que nous ferons intervenir plusieurs fois des variables de paramétrage dans nos calculs. Ces variables sont fixées dans un fichier de paramétrage, dont nous expliquons la conception dans la section 5.5.5.

5.5.2 Définition des opérations de transformation

Nous expliquons dans cette section les choix effectués dans la définition des quatres types d'opérations de transformation. Nous décrivons en particulier l'influence sémantique et structurelle de l'application d'une opération. Nous nous appuyons sur les motivations exposées dans la section 5.2.2 pour expliquer nos choix. Nous nous appuyons sur les définitions des opérations présentées dans cette section pour mettre en place les modules de génération des opérations de transformation et de recherche de la suite d'opérations la moins coûteuse, présentés respectivement dans les sections 5.5.3 et 5.5.4.

5.5.2.1 Opérations de substitution

L'opération de substitution a pour objectif de substituer le contenu des segments du passage similaires aux segments de la question. Le système s'appuie sur les ancres créées dans la section 5.4.2.4 pour identifier les segments similaires. Néanmoins, nous voulons que le coût de ces opérations reflètent la différence entre les deux segments. Plus la similarité d'un segment est faible, et plus le coût de substitution doit être élevée. Nous estimons que la similarité peut être évaluée selon deux critères.

Premièrement, nous prenons en compte le nombre de mots différents entre les deux segments, c'est à dire les mots sur lesquels le système n'a pas détecté de similarité. Nous estimons en effet que si deux segments ont un grand nombre de mots sémantiquement importants différents, il est probable que leur sens soit relativement éloigné l'un de l'autre, même s'il partage un mot similaire. Si on prend par exemple les segments *le ministre des affaires étrangères français* et *le ministre anglais*, ces deux segments ont un mot avec une similarité *identité*, *ministre*. Néanmoins, on identifie trois mots importants sémantiquement qui ne sont pas partagés par ces deux segments : *affaires, étrangères* et *anglais*. Nous nous appuyons sur l'analyse effectuée par Ritel pour identifier les mots sémantiquement importants.

Deuxièmement, le type des similarités détectées pour les mots ayant un sens proche nous semble lui aussi important pour déterminer le coût d'une substitution. Nous avons quatre types de similarité : *identité, lemme, morpho-syntaxique* et *synonymie*. Nous estimons que deux segments sont au moins partiellement identiques si l'ensemble des similarités sont de type *identité*. Au contraire, une similarité de type synonymie implique que les segments ont un sens potentiellement différent : les deux segments *un abandon* et *une cession* ont une relation synonymie, mais il est possible que le sens des deux segments ne soient pas le même. De ce fait, nous estimons que le type des similarités doit être pris en compte dans le calcul.

Néanmoins, avec la définition actuelle des opérations de substitution, nous ne représentons que des subsitutions entre un segment de la question et un segment du passage, que nous qualifions de *simple*. Or, il est fréquent d'avoir un segment de la question associé à plusieurs ancres. Cela implique que deux segments du passage contiennent des informations. Par exemple, le segment *à sa libération de prison* a une ancre avec le segment *est libéré* et une avec le segment *de prison*. De même un segment du passage peut avoir plusieurs ancres avec différents segment de la question. De ce fait, il semble donc pertinent d'avoir deux autres types d'opération de substitution. Nous qualifions ces deux nouveaux types de *fusion* (un segment de la question est ancrée avec plusieurs segments du passage) et scission (un segment du passage est ancré avec plusieurs segments de la question). Une évaluation a été faite pour mesurer la répartition des types d'opérations de substitution : il y a 88% d'opérations simples, 10% de fusion, et 2% de scission.

Le coût des opérations de substitution ne prend en compte que le niveau sémantique, c'est à dire l'information relative au sens des segments. Le niveau structurel sera pris en compte par les opérations de rattachement.

5.5.2.2 Opérations de rattachement

L'opération de rattachement est appliquée après une opération de substitution. Son objectif est d'estimer si les segments du passage affectés par une opération de substitution sont reliés au segment contenant la réponse candidate évaluée.

Nous considérons ainsi que chaque segment du passage ayant été substitué doit être en relation avec le segment contenant la réponse évaluée. Si les opérations de substitution permettent de quantifier la similarité sémantique entre des segments de la question et du passage, les opérations de rattachement ont pour objectif de quantifier la similarité structurelle. L'information structurelle des phrases est représentée d'une part par les segments, et d'autre part par les relations identifiées entre ces segments. Les opérations de rattachement s'appuient donc sur les segments et relations pour déterminer si un segment du passage est rattaché à la réponse candidate.

Si un segment a déjà une relation en commun avec le segment contenant la réponse, alors on estime que le segment est déjà rattaché à la réponse : le coût est donc nul. Sinon, le coût de l'opération est quantifié en fonction des types des segments et des relations de la phrase. Nous estimons que les relations de temps et de lieu ne doivent pas être prise en compte pour déterminer si le segment a déjà une relation commune avec le segment contenant la réponse. Nous avons fait l'hypothèse dans la section 4.3 du chapitre 4 qu'un segment de temps ou de lieu était en relation avec tous les segments d'une phrase. Nous motivions ce choix par le fait que les segments de temps et de lieu représentent des compléments circonstanciels. Or, il est possible de déplacer un complément circonstanciel au sein d'une proposition sans en changer le sens. Il est cependant difficile de détecter les différentes propositions d'une phrase, spécialement dans un contexte oral. Nous avons évalué notre réordonnanceur en prenant en compte les relations de temps et de lieu pour identifier si un segment est en relation avec le candidat réponse. Les résultats obtenus sont moins bons que lorsque les relations de temps et de lieu n'entrent pas en compte dans ce cas de figure. Un travail est nécessaire pour affiner le traitement des relations de temps et de lieu, particulièrement lors de leur identification.

Notre modélisation de l'information structurelle des phrases, particulièrement les relations, reste relativement simple de manière à être appliquée sur n'importe quel type de documents. Nous estimons que le rattachement peut être modélisé par un *chemin* de relations, similaire à ceux présentés dans [Comas et al. 2010] : si un segment A est relié à un segment B, qui lui même est relié à un segment C, alors le chemin entre A et C est *A -> B -> C*. Nous voulons donc que le coût de l'opération de rattachement quantifie ce chemin, en prenant en compte les types de relation. Ainsi, nous estimons que le coût du rattachement est calculé par une suite d'opérations intermédiaires, correspondant aux différents segments du chemin. Nous définissons ces calculs intermédiaires comme étant des *permutations*.

Néanmoins, nous émettons deux hypothèses, dépendant directement des choix faits dans la création du modèle de représentation. Tout d'abord, nous estimons que les segments verbaux ont un rôle de pivot dans la phrase. De ce fait, nous voulons que les rattachements dont le chemin de relations passe par un segment verbal soient plus contraignants, c'est à dire plus coûteux. D'autre part, les segments de temps et de lieu correspondent à des groupes circonstanciels. Du fait qu'un groupe circonstan-

ciel a un placement plus libre au sein de la structure d'une phrase, les rattachements de segments de temps ou de lieu sont moins coûteux. Nous nous appuyons sur des règles écrites manuellement pour déterminer le coût de chaque permutation. L'application d'une règle dépend du contexte de la permutation : le type des segments que l'on veut permuter, les relations en commun, et le sens de la dépendance s'il y a une relation en commun. L'application d'une règle a un poids associé. Le coût d'un rattachement correspond donc à la somme du poids des différentes permutations. Ces règles ont été écrites après des observations sur corpus. Les poids ont été fixé empiriquement, puis affiné par le biais d'expérimentations. Nous avons pour le moment une vingtaine de règles, et couvre des cas très généraux. Nous voudrions augmenter ce nombre pour couvrir plus de cas par la suite.

La phrase "*[ST] A 71 ans [/ST] , [SN] Nelson Mandela [/SN] [SV] est libéré [/SV] [SN] de prison [/SN] [SN] par Frederik Willem de Klerk [/SN].*" illustre les différents choix effectués pour les rattachements. Les relations suivantes sont identifiées :

– relation de groupe nominal entre *[SN] de prison [/SN]* et *[SN] par Frederik Willem de Klerk [/SN]*
– relations membres-verbe entre *[SV] est libéré [/SV]*, et *[SN] de prison [/SN]* et *[SN] Nelson Mandela [/SN]*.

Le segment *de prison* n'est pas relié à *A 71 ans*, on calcule donc un coût de rattachement qui correspond au chemin entre ces deux segments, qui est composé de *Nelson Mandela* et *est libéré*. Il y aura donc deux permutations pour rattacher *de prison* : une avec *est libéré* et une avec *Nelson Mandela*. Le coût de ces permutations dépend du type des segments, et des relations de la phrase. Par exemple, la permutation avec *est libéré* est plus coûteuse car c'est un segment verbal.

5.5.2.3 Opérations de suppression et d'insertion

Les opérations de suppression et d'insertion ont un comportement sémantique et syntaxique très proche. L'objectif des opérations de suppression est d'enlever les segments du passage ne contenant pas d'information similaire aux segments de la question. Les opérations d'insertion permettent d'ajouter dans le passage les segments de la question pour lesquels le réordonnanceur n'a pas identifié de segments similaires. Ces deux opérations ont évidemment un impact fort sur l'information et la structure d'une phrase : enlever et ajouter de l'information change complètement le sens d'une phrase. De ce fait, nous estimons que ces deux types d'opérations ne doivent être appliquées qu'après les opérations de substitution et de rattachement : il devient en effet plus difficile d'évaluer l'impact structurel dans le cas contraire.

Le coût d'une suppression ou d'une insertion doit donc prendre en compte ces deux aspects. Nous souhaitons que le coût de ces deux types d'opération dépende du nombre de mots importants sémantiquement du segment, et du nombre et du type des relations du segment. Par ailleurs, le fait qu'une relation soit sortante ou entrante doit avoir une importance particulière. En effet, comme nous avons vu dans la section 4.3 du chapitre 4, une relation a un sens de dépendance : ce sens représente quel segment est dépendant de l'autre. Par exemple, dans une relation nominale, le segment dépend de son

voisin gauche : on parle d'une relation sortante pour ce segment, et d'une relation entrante pour le voisin gauche. Ainsi, nous donnons un coût plus important aux segments ayant des relations sortantes.

Nous estimons aussi que l'insertion d'un nouveau segment doit avoir un coût plus important qu'une suppression. En effet, l'idée étant de déterminer si un passage est proche d'une question, on peut se poser la question si une insertion d'information a plus d'impact qu'une suppression. De plus, les passages étant généralement plus longs que les questions, une suppression avec un coût trop important pénaliserait les passages les plus grands. Nous avons évalué cette hypothèse avec différents paramètres. Les résultats étaient meilleurs lorsque le coût d'insertion était plus élevé.

Nous considérons que tous les segments de la question sont reliés au segment réponse. De ce fait, les opérations d'insertion sont générées pour les segments n'ayant pas de similarités avec des segments du passage. Par contre, dans un passage tous les segments ne sont pas forcément en relation avec la réponse, et n'apportent pas nécessairement de l'information en lien avec la réponse. Or, le calcul du coût de transformation du passage en la question a surtout pour objectif de quantifier la différence sémantique et syntaxique de l'extrait du passage contenant la réponse, et non le passage dans son intégralité. La réduction du passage présentée dans la section 5.4.2.5 permet de supprimer certaines phrases que nous ne jugeons pas ou peu en rapport avec la réponse candidate. Néanmoins, il peut rester des phrases du passage qui n'apportent pas véritablement d'information à la réponse évaluée. De même, dans une phrase longue, certains segments peuvent n'avoir aucun rapport avec la réponse candidate. Le passage et la question de la figure 5.28 donnent un exemple illustrant ces phénomènes.

Q : *"[SMI] Quel âge [/SMI] [SV] avait [/SV] [SN] Nelson Mandela [/SN] [SN] à sa libération de prison [/SN] ?"*
P : *"[ST] A 71 ans [/ST] , [SN] Nelson Mandela [/SN] [SV] est libéré [/SV] [ST] après 27 années [/ST] [SN] de prison [/SN] ."*
"[SN] Nelson Mandela [/SN] [SV] étant libéré [/SV] [SN] la situation [/SN] [SV] devrait rapidement évoluer [/SV]"

FIG. 5.28 – Exemple d'une question et d'un passage segmentés pour illustrer les suppressions et insertions.

Dans cet exemple, la deuxième phrase du passage contient les éléments de la question *Nelson Mandela* et *libéré*. Si une opération de substitution est appliquée sur un de ces deux éléments, il semble logique de considérer l'ensemble des éléments de cette phrase lors de l'application de la suppression des éléments. De même, la première phrase contenant le candidat réponse, les opérations de suppression s'appliquent sur cette phrase. Par contre, si aucune opération de substitution n'est appliquée sur la seconde phrase, nous estimons que la suppression des segments de cette phrase doit être nulle. Au sein d'une phrase, la détection des segments ayant un coût de suppression nul s'appuie sur la même idée. Si un segment n'est ni en relation avec le segment réponse, ni un des segments ayant été substitués, alors son coût de suppression est nul. Enfin, si un segment a fait parti d'un *chemin* lors d'un rattachement, il est aussi considéré comme ayant un coût de suppression non nul.

Enfin, le cas des suppressions des segments de temps et de lieu est plus particulier. Un segment de ces deux types est automatiquement considéré comme ayant un coût nul de suppression si aucun segment du même type n'est affecté par les transformations de substitution. Du fait que les segments de temps et de lieu sont des groupes circonstanciels, leur suppression d'une phrase entrainera principalement une perte d'information sémantique, mais pas structurelle. Nous estimons donc que leur suppression ne doit être coûteuse que si d'autres segments du même type sont présents dans la même phrase. Idéalement, nous voudrions que cette suppression ne soit coûteuse que si il y a une ambiguïté entre l'information de segments temps ou lieu (par exemple une date différente). Des essais ont été effectués pour mettre en place une détection de ces ambiguïtés, mais les résultats ne sont pas convaincants pour le moment.

5.5.3 Génération des opérations de transformation

Nous avons défini dans la section 5.5.2 les différentes opérations de transformation utilisées par notre système. Nous décrivons dans cette section la première phase du calcul du coût de transformation : la génération des opérations de transformation entre un segment et une question.

A ce stade du traitement, les coûts de chaque opération ne sont pas calculés : l'algorithme détermine juste les opérations possibles étant donné les ancres identifiées. Nous décrivons la méthode employée pour générer chaque type d'opération dans les sections suivantes. Le but de cette génération est de représenter l'ensemble des opérations possibles, permettant par la suite de calculer le coût de transformation minimum pour transformer le passage en la question.

Ce module prend en entrée une question Q segmentée et dont les relations entre segments ont été identifiées, un ensemble P_i dont tous les passages ont été traités comme la question, et une réponse R_i. Par ailleurs, le module prend aussi en entrée l'ensemble $Eancres_{Q/P_i/R_i}$ des ancres identifiées entre les passages et la question. Le module fournit en sortie quatre ensembles d'opérations : les opérations de substitution $Eopsubs_{Q/P_i/R_i}$, les opérations de rattachement $Eopratt_{Q/P_i/R_i}$, les opérations de suppression $Eopsuppr_{Q/P_i/R_i}$, et les opérations d'insertion $Eopins_{Q/P_i/R_i}$. La figure 5.29 illustre ce module.

La figure 5.30 illustre un ancrage entre la question *"Quel âge avait Nelson Mandela à sa libération de prison ?"* et le passage *"A 71 ans Nelson Mandela est libéré après 27 années de prison à Pollsmor par Frederik Willem de Klerk"*. Les segments ainsi que les relations sont aussi représentées. Nous nous appuyons dans les sections suivantes sur cet exemple pour illustrer la génération des opérations.

5.5.3.1 Opérations de substitution

Les ancres décrites dans la section 5.4.2.4 permettent de représenter les similarités au niveau informationnel entre la question et le passage. Elles vont être utilisées pour générer les opérations de

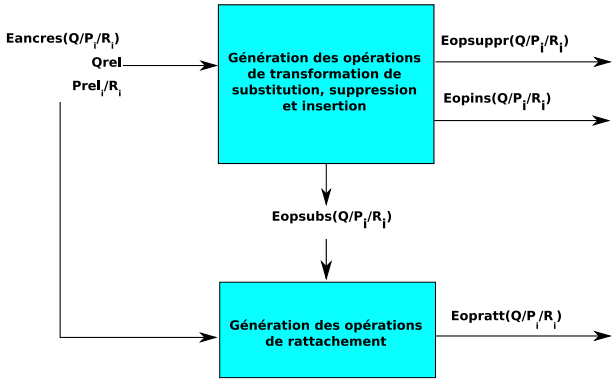

FIG. 5.29 – Architecture du module de génération des opérations de transformation. Q est une question segmentée ; R_i une réponse candidate ; P_i un ensemble de passages associés à la réponse ; $Eancres_{Q/P_i/R_i}$ l'ensemble des ancres entre Q et les passages de P_i.

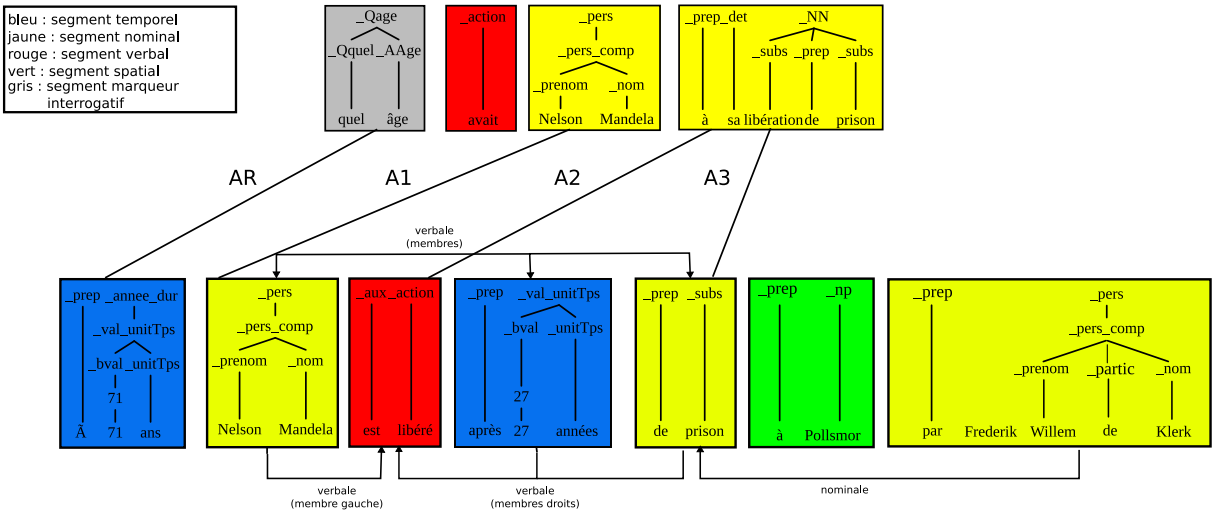

FIG. 5.30 – Exemple de référence pour illustrer la génération des opérations de transformation. A1, A2 et A3 correspondent aux ancres identifiées.

substitution. A ce stade, les coûts de substitution ne sont pas encore calculés.

Dans la figure 5.30, on peut voir que différents cas peuvent apparaître sur la répartition des ancres. Un segment de la question peut être relié à un seul segment du passage (*Nelson Mandela* et *Nelson Mandela*), mais aussi à plusieurs (*à sa libération de prison*, et *est libéré* et *de prison*). Enfin un segment du passage peut potentiellement être relié à deux segments de la question. A partir de ces cas, trois types d'opérations de substitution sont définies :

– substitution simple : un segment du passage est remplacé par un segment de la question (*le petit chat* par *le chat noir*) ;
– substitution fusion : plusieurs segments du passage sont remplacés par un segment de la question (*est libéré* et de *de prison* par *à sa libération de prison*)
– substitution scission : un segment du passage est remplacé par plusieurs segments de la question (*à sa libération de prison* par *est libéré* et *de prison*)

L'approche pour générer les opérations de substitution est la suivante. Pour chaque passage $p_{i,j}$ de P_i, on récupère le tableau des ancres $ancresSegments_{Q/p_{i,j}}$ de l'ensemble $Eancres_{Q/P_i/R_i}$. Ce tableau est parcouru colonne par colonne, ce qui correspond à regarder pour chaque segment du passage s'il existe une ancre avec les segments de la question. Si une case contient une ancre, alors une opération de substitution de type *simple* est créée.

De plus, toutes les ancres du segment du passage sont utilisées pour créer une ou plusieurs opérations de type *scission* : toutes les combinaisons possibles sont créées. Par exemple, si un segment du passage *SP* est ancré avec trois segments de la question *S1, S2* et *S3*, quatre opérations de type *scission* seront créées : une substitution de *SP* par *S1* et *S2*, une par *S1* et *S3*, une par *S2* et *S3*, et une par *S1, S2* et *S3*.

Par ailleurs, on crée aussi au fur et à mesure les opérations de type *fusion*. A chaque fois qu'une opération de substitution simple est créée, l'algorithme analyse le segment de la question. Si ce segment a plusieurs ancres, une ou plusieurs opérations de fusion peuvent donc être générées. Comme pour les opérations de type *scission*, on génère l'ensemble des combinaisons possibles.

Une opération de substitution $OPsubs_x$ est composée de deux attributs : son type et un ensemble d'ancres. Le type est un des trois types d'opérations de substitution : *simple*, *fusion* et *scission*. L'ensemble d'ancres contient un certain nombre d'ancres, qui dépendent avant tout du type de la substitution. Dans le cas d'une opération simple, l'ensemble ne contiendra qu'une seule ancre, mais pour les deux autres types, ce nombre est variable. Pour chaque passage $p_{i,j}$ de P_i, l'algorithme génère un ensemble d'opération de substitution $opsSubs_{Q/p_{i,j}}$ entre ce passage et la question Q. Ces ensembles d'opération de substitution sont stockés dans $Eopsubs_{Q/P_i/R_i}$.

Par ailleurs, l'algorithme de génération des opérations utilise deux sous-fonctions, **créerOpérationsScission** et **créerOpérationsFusion**, qui prennent en paramètres respectivement un ensemble d'ancres partant du même segment du passage, un segment de la question et l'ensemble des ancres trouvées par le système. Ces deux sous-fonctions permettent de générer toutes les combinaisons d'opérations pour la fusion et la scission. L'algorithme de génération des opérations de substitution est présenté dans la figure 5.31. Il est à noter que nous présentons dans cette figure le traitement fait pour chaque passage de P_i pour des raisons de lisibilité. De ce fait, l'algorithme prend en entrée l'ensemble d'ancres $ancresSegments_{Q/p_{i,j}}$ entre le passage $p_{i,j}$ et la question Q, et retourne un ensemble d'opération de substitution $opsSubs_{Q/p_{i,j}}$.

Dans l'exemple de référence de la figure 5.30, avec les différentes ancres entre les segments du passage et ceux de la question, l'algorithme créé les quatre opérations suivantes :

Variables d'entrée :
– $ancresSegments_{Q/p_{i,j}}$: ensemble des ancres entre la question et le passage de $p_{i,j}$
– Q : question segmentée avec relations entre les segments
– $p_{i,j}$: passage segmenté avec relations entre les segments
– R_i : réponse candidate évaluée
Variable de retour :
$opsSubs_{Q/p_{i,j}}$: ensemble d'opérations de substitution possibles entre la question et le passage $p_{i,j}$.

Corps de la fonction :

pour chaque $S_y^{p_{i,j}}$ de $p_{i,j}$, y = 1, ... , Y
  Eancres vancres_seg
  pour chaque S_x^Q de Q, x = 1, ... , X
    si $ancresSegments_{Q/p_{i,j}}[x, y]$ n'est pas vide **alors**
      OPsubs opsimple
      opsimple.type = simple ; opsimple.ancres.ajouter($ancresSegments_{Q/p_{i,j}}[x, y]$)
      $opsSubs_{Q/p_{i,j}}$.ajouter(opsimple)
      vancres_seg.ajouter($ancresSegments_{Q/p_{i,j}}[x, y]$)
      si nombreAncres(S_y^Q,$ancresSegments_{Q/p_{i,j}}$) > 1
        ET $ancresSegments_{Q/p_{i,j}}[x, y]$ == $ancresSegments_{Q/p_{i,j}}[x, 0]$ **alors**
        Eopsubs ops_fusion = créerOpérationsFusion(S_y^Q,$ancresSegments_{Q/p_{i,j}}$)
        $opsSubs_{Q/p_{i,j}}$.concaténer(ops_fusion)
      fsi
    fsi
  fpour
  si vancres_seg.taille()>1 **alors**
    Eopsubs ops_seg = créerOpérationsSegmentation(vancres_seg)
    $opsSubs_{Q/p_{i,j}}$.concaténer(ops_seg)
  fsi
fpour
retourne $opsSubs_{Q/p_{i,j}}$

FIG. 5.31 – Algorithme de génération des opérations de transformation

– Substitution simple entre *Nelson Mandela* et *Nelson Mandela*
– Substitution simple entre *est libéré* et *à sa libération de prison*
– Substitution simple entre *de prison* et *à sa libération de prison*
– Substitution fusion entre les segments du passage *est libéré* et *de prison* et le segment de la question *à sa libération de prison*

5.5.3.2 Opérations de suppression

Les opérations de suppression sont générées pour chaque segment du passage, même ceux qui sont ancrés. En effet, l'objectif des opérations de suppression est d'enlever les segments dont le contenu n'est pas similaire à celui de la question. Or, il peut arriver que des segments ayant été ancrés ne soient pas substitués. Par exemple, dans la figure 5.30 les deux segments du passage *est libéré* et *de prison* sont ancrés avec le segment de la question *à sa libération de prison*. Ces ancrages impliquent trois opérations de substitution : deux simples, et une fusion (voir la section 5.5.3.1). Si le système choisit d'appliquer l'opération de substitution simple entre *est libéré* et *à sa libération de prison*, cette opération invalide l'opération de substitution entre *de prison* et *à sa libération de prison* ainsi que l'opération de fusion correspondante. C'est la raison pour laquelle des opérations de suppression sont générées sur tous les segments du passage.

Une opération de suppression $OPsuppr_x$ ne contient que le segment du passage correspondant. Chaque opération de suppression d'un passage $p_{i,j}$ est stockée dans $opsSuppr_{Q/p_{i,j}}$. Enfin, les ensembles d'opération de suppression de chaque passage sont stockés dans $Eopsuppr_{Q/p_i/R_i}$.

L'algorithme générant les opérations de suppression est très simple : chaque segment du passage est traité et l'opération est créée et stockée, excepté pour le segment contenant la réponse candidate évaluée. Ainsi, dans l'exemple de la figure 5.30 il y a sept segments dans le passage, dont un est le segment contenant la réponse (*A 71 ans*). Le système génère donc six opérations de suppression.

5.5.3.3 Opérations d'insertion

La génération des opérations d'insertion suit la même logique que pour les opérations de suppression. On génère une opération d'insertion pour chaque segment que comporte la question, excepté le segment de type marqueur interrogatif. Une opération d'insertion $OPins_x$ contient le segment de la question correspondant. Chaque opération d'insertion entre une question Q et un passage $p_{i,j}$ est ajoutée dans $opsIns_{Q/p_{i,j}}$. Enfin, tous les ensembles d'opération d'insertion sont ajoutés dans $Eopins_{Q/P_i/R_i}$.

Dans l'exemple de la figure 5.30, la question a quatre segments, dont un est le segment marqueur interrogatif. Le système génère donc trois opérations d'insertion.

5.5.3.4 Opérations de rattachement

Lors d'une substitution le système vérifie si le segment substitué est relié au segment du passage contenant la réponse évaluée. L'identification s'appuie sur les relations identifiées lors des pré-traitements. Si il existe une relation entre ces le segment du passage et le candidat réponse, on considère le rat-

tachement comme déjà effectué, et aucune opération n'est créée. Si ce n'est pas le cas, le système applique une opération de rattachement sur le segment substitué pour quantifier le coût nécessaire pour avoir les deux segments en relation.

De ce fait, les opérations de rattachement sont générées pour chaque opération de substitution où le segment (où les segments dans le cas d'une substitution fusion) n'est pas relié au segment réponse. L'opération de rattachement créée contient le ou les segments à rattacher. Si il existe une relation temporelle ou spatiale entre les deux segments, cette relation n'est pas considérée comme *valide*, et l'opération de rattachement est quand même générée. Nous nous appuyons sur les choix présentés dans la section 5.5.2 pour justifier cette approche. Dans l'exemple de référence de la figure 5.30, le segment contenant la réponse candidate est *A 71 ans*. Aucune relation de type *groupe nominal* ou *membres-verbe* existe entre ce segment et les segments ancrés (et donc potentiellement substitués) : *Nelson Mandela*, *est libéré* et *de prison*. Les opérations de rattachement sont donc générés pour chacune des opérations de substitution contenant ces segments. Si l'on se réfère aux opérations de substitution générées dans la section 5.5.3.1, il y a trois opérations de rattachement :

– Rattachement de *Nelson Mandela*
– Rattachement de *est libéré*
– Rattachement de *de prison*

Pour les opérations de substitution de type *fusion*, qui font intervenir plusieurs segments du passage, plusieurs opérations de rattachement sont créées. Par exemple, pour l'opération de type *fusion* impliquant *est libéré* et *de prison*, deux opérations de rattachement sont associées.

Une opération de rattachement $OPratt_x$ contient donc le segment à rattacher, et est stocké dans l'ensemble des opérations de rattachement $opsRatt_{Q/p_{i,j}}$ du passage pi, j. Enfin, l'ensemble d'opération de chaque passage est stocké dans $EoprattQ/P_i/R_i$.

5.5.4 Algorithme de recherche de la suite d'opérations de transformation la moins coûteuse

Pour la section présente ainsi que ses sous-sections, nous nous appuyons de nouveau sur le même exemple utilisé lors de la génération des opérations de transformation. Nous redonnons cet exemple dans la figure 5.32 pour simplifier la lecture.

L'algorithme de recherche a pour objectif de trouver la suite d'opérations de transformation la moins coûteuse. Il peut en effet y avoir plusieurs suites d'opérations possibles pour transformer le passage en la question. Par ailleurs, l'application de chaque opération engendre un coût. Le coût est variable quelque soit le type d'opération appliqué. Pour une opération de substitution, le coût dépend des mots similaires et du type de similarité. Pour les opérations de suppression et d'insertion, le coût dépend du nombre de mots important sémantiquement au sein du segment, ainsi que des relations

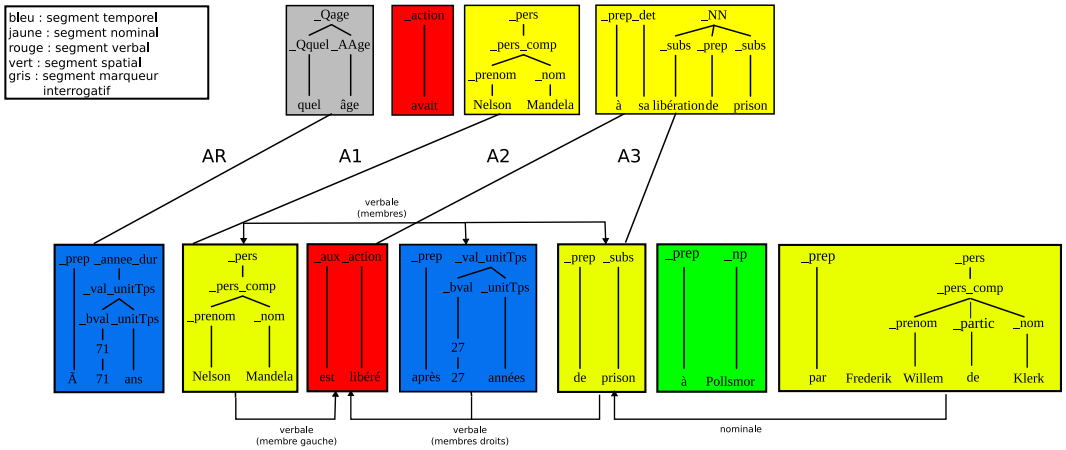

FIG. 5.32 – Exemple de référence pour illustrer l'algorithme de recherche de la suite d'opérations de transformation la moins coûteuse. A1, A2 et A3 correspondent aux ancres identifiées.

entrantes ou sortantes associées au segment. Enfin, le coût des opérations de rattachement dépend des permutations à effectuer. Chaque permutation a une règle associée, et chaque règle a un poids d'application. La somme des poids des permutations donne le coût du rattachement.

Les autres calculs effectués s'appuient aussi sur un certain nombre de poids. Ainsi, chaque type de similarité entre les mots des segments du passage et de la question est associé à un poids, représentant le degré de similarité. La hiérarchie des similarités est utilisée pour fixer ces poids : identité, lemme, morpho-syntaxique, synonyme. Le type identité a un poids nul, et le type synonyme le poids le plus élevé. Les relations typées entre segments ont aussi un poids dépendant du type de la relation. Ces poids ont été fixés selon l'importance structurelle d'une relation. Ainsi, nous estimons qu'une relation verbale est plus importante qu'une relation nominale, et que son poids est donc plus élevé. Enfin, nous intégrons ainsi à nos calculs plusieurs valeurs de paramétrage et de pondération.

Dans les sections suivantes, nous présentons l'algorithme utilisé, la recherche en coût uniforme, et son application à notre travail. Nous présentons ensuite les traitements utilisés pour déterminer les coûts de chaque opération. Nous commençons par introduire la notion de *poids du segment* qui est utilisée par la suite dans le calcul des opérations de transformation. Puis nous décrivons le processus de calcul du coût de chaque type d'opération. Enfin, différents paramètres sont utilisés dans le calcul des coûts des opérations. Nous décrivons comment ont été fixés ces paramètres.

5.5.4.1 Description de l'algorithme

Il existe potentiellement plusieurs enchaînements d'opérations possibles pour effectuer la transformation : l'application d'une opération de substitution simple empêchera par exemple l'application d'une

opération de substitution fusion. Un algorithme choisissant simplement l'opération de substitution avec le coût le plus faible ne sélectionnera jamais les opérations de fusion, ces dernières étant plus coûteuses que les opérations simples. Or, d'un point de vue global, la sélection d'une opération de fusion peut entrainer un coût de transformation plus faible. De ce fait, un algorithme de recherche est utilisé pour trouver la meilleure solution. L'idée initiale était d'utiliser l'algorithme A*, mais l'élaboration d'une fonction heuristique se révélait assez complexe. L'objectif de ce travail étant de s'intéresser plus particulièrement à l'impact des transformations sur un passage, il a donc été choisi dans un premier temps d'utiliser l'algorithme de recherche en coût uniforme. De plus, nous travaillons sur des graphes de recherche relativement petits. Les recherches ne sont donc pas trop coûteuses en temps de calcul.

L'algorithme de coût uniforme utilise une fonction g(n) qui calcule pour chaque noeud *n* le coût maximum pour atteindre la solution depuis l'état initial jusqu'à l'état final du graphe de recherche. Ainsi, à chaque étape de recherche, l'algorithme choisit dans l'ensemble des noeuds successeurs celui minimisant le coût de cette fonction g. La recherche s'arrête dès lors que l'algorithme a trouvé un noeud final dont le coût total pour l'atteindre est inférieur à toutes les autres solutions développées. L'algorithme de coût uniforme est **complet** et **optimal**. Par contre, il n'est pas le plus efficace pour trouver la bonne solution. Nous revenons sur ce défaut plus loin dans cette section.

Le fonctionnement général de cet algorithme se transpose de la manière suivante à notre problème. Le noeud de départ du graphe de recherche correspond à l'état initial du passage. On avance d'un noeud à un autre en appliquant une opération de substitution sur le passage. Si besoin est, une opération de rattachement est appliquée après l'opération de substitution. Lorsqu'il n'y a plus d'opérations de substitution possibles à appliquer, on considère que l'on a atteint la fin du graphe de recherche. Le système applique alors toutes les opérations de suppression et d'insertion nécéssaires pour terminer la transformation. Cette approche permet d'avoir des graphes de recherche relativement courts, mais aussi d'être plus logique syntaxiquement parlant au niveau des opérations appliquées : le fait de par exemple pouvoir supprimer des segments à tout moment ferait que les opérations de substitution auraient moins de sens par rapport à la prise en compte de la structure du passage.

L'algorithme nécessite une fonction permettant d'évaluer la distance maximale restante pour accéder à la solution pour pouvoir orienter la recherche de la meilleure suite d'opérations. Cette fonction devant toujours retourner une valeur supérieure ou égale à la distance réelle restante, on n'utilise que les opérations de suppression et d'insertion pour évaluer la distance maximale restante. En effet, la substitution d'un segment du passage par un segment de la question ayant toujours un coût inférieur à la suppression puis l'insertion, la distance maximale générée sera toujours plus grande que la distance réelle. De ce fait, le coût de la distance maximale pour atteindre le noeud final correspond à l'application des opérations de suppression sur tous les noeuds du passage non substitués, et des opérations d'insertions sur les noeuds de la question non substitués. Nous revenons plus loin sur le calcul des coût de chacune des opérations.

Le coût de transformation d'un passage est donc la solution la moins coûteuse retourné par l'algorithme de recherche. On peut formaliser ce coût **CTP** de la manière suivante :

$$CTP = UniformCost(Passage, Question)$$

De ce fait, pour *n* suites de transformations possibles, le coût de transformation du passage est celui de la suite de transformation ayant le coût le plus faible.

$$CTP^* = \min_n(CT_n)$$

Il existe quatre types d'opération de transformation : substitution, rattachement, suppression et insertion. Les opérations sont traitées dans un certain ordre, selon leur type. L'algorithme de recherche calculera des suites de paire d'opérations de substitution et de rattachement, avec chacune un coût, auquel sera rajouté le coût total de suppression puis d'insertion. Le coût de transformation d'un passage est donc la somme des coûts des opérations de substitution, de rattachement, d'insertion et de suppression.

$$CTP^* = \min_n(COP_{sub_n} + COP_{ratt_n} + COP_{suppr_n} + COP_{ins_n})$$

où $COP_{sub}, COP_{ratt}, COP_{suppr}$ et COP_{ins} sont respectivement le coût des opérations de substitution, rattachement, suppression et insertion de la solution *n*. Le calcul du coût de chaque type d'opération est défini dans les sections suivantes. Nous définissons d'abord la notion de *poids du segment* sur laquelle ces calculs s'appuient pour déterminer le coût des opérations.

5.5.4.2 Poids du segment

Avant de détailler le calcul du coût en fonction des types d'opérations, nous allons d'abord introduire une notion nécessaire à ces calculs : le poids d'un segment. Ce poids représente l'importance informationelle et aussi structurelle d'un segment dans le passage ou la question. Nous nous appuyons sur nos différentes annotations sémantiques et structurelles. Le poids d'un segment est calculé à partir de plusieurs paramètres. Il dépend des mots importants sémantiquement contenus dans le segment, ainsi que des relations avec les autres segments (passage ou question). De ce fait, le poids d'un segment dépend de deux valeurs, le poids des mots, et le poids des relations. Afin de pouvoir paramétrer l'importance relative du poids des relations par rapport au poids des mots, on pondère le poids des relations avec un paramètre λ_{rels}. Ainsi, $Pseg_i$ représente le poids du segment i, $Pmots_i$ le poids des mots du segment i et $Prels_i$ son poids des relations.

$$Pseg_i = Pmots_i + Prels_i * \lambda_{rels}$$

Le poids des mots d'un segment dépend des types associés à chaque mot par l'analyse de Ritel. On utilise la même classification que pour l'ancrage de la question et du passage : les types sémantiques sont considérés comme importants, comme par exemple une fonction politique (président). Ainsi, un poids est associé aux mots importants. Pour le moment, ce poids est identique pour tous les mots importants sémantiquement. Ce poids permet de pondérer l'importance des mots dans notre calcul. Une étude possible serait de d'associer des poids différents en fonction du type sémantique du mot.

Le poids d'un segment est donc le nombre de mots importants dans un segment multiplié par le poids. Le poids des mots importants est noté λ_{imp}.

$$Pmots_i = |(motsimportants)|_i * \lambda_{imp}$$

Le poids des relations d'un segment dépend de deux paramètres : le type des relations, et si elles sont entrantes ou sortantes. Pour ce dernier point, les relations entrantes sont considérées comme étant plus importantes car cela implique que des segments dépendent du segment que l'on cherche à supprimer, d'où le poids plus élevé. Chaque type de relation a un poids qui lui est associé. Le poids total des relations correspond donc à la somme du poids de chaque relation sortante et à la somme des poids de chaque relation entrante, le poids total des relations entrantes étant multiplié par une valeur de paramétrage. Si l'on considère le nombre de relations entrantes d'un segment i est K, et son nombre de relations sortantes L, nous notons $PrelationEntrante_{i,k}$ la $k_{ième}$ relation entrante du segment i , $PrelationSortante_{i,l}$ la $l_{ième}$ relation sortante du segment i. Le paramètre λ_{relE} permet de pondérer le poids des relations entrantes par rapport à celui des relations sortantes. La formule ci-dessous représente le calcul du poids des relations d'un segment.

$$Prels_i = \sum_{k=1}^{K}(PrelationEntrante_{i,k}) * \lambda_{relE} + \sum_{l=1}^{L}(PrelationSortante_{i,l})$$

Les relations temporelles et spatiales ne sont pas prises en compte dans le poids du segment. Etant donné que les segments spatiaux et temporels sont considérés comme des compléments circonstanciels, nous faisons l'approximation qu'ils peuvent être déplacés dans l'ensemble de la phrase sans en changer le sens.

L'exemple ci-dessous détaille le calcul effectué sur le segment *de prison*, tiré de la figure 5.32. Ce segment est le cinquième segment du passage.

Les différents poids fixés sont :

- poids relations verbale : 0,4
- poids relations nominale : 0,2
- λ_{imp} : 0,4
- λ_{rels} : 0,5
- λ_{relE} : 2

Le calcul est donc le suivant :

- il n'y a qu'un seul mot important, *prison*, le poids des mots est donc $Pmots_5$ = 1*0,4 = 0,4 ;
- il y a trois relations entrantes, une nominale, et deux verbales ; le poids des relations entrantes est donc $\sum_{k=1}^{K}(PrelationEntrante_{5,k})$ = 0,2 + 0,4 + 0,4 = 1,0 ;
- il y a trois relations sortantes, toutes les trois verbales ; le poids des relations sortantes est donc $\sum_{l=1}^{L}(PrelationSortante_{5,l})$ = 0,4 + 0,4 + 0,4 = 1,2 ;
- le poids des relations est donc $Prels_5$ = 1,0*2 + 1,2 = 3,2 ;

– finalement, le poids du segment est de $Pseg_5$ = 0,4 + 3,2*0,5 = 2,0

Le poids du segment est donc de 2,0.

Chaque segment a donc un poids associé. Ce poids des segments sera réutilisé pour chacun des types d'opération. Le poids de l'ensemble des segments de l'exemple de référence est décrit ci-dessous (on ne donne pas le poids du segment marqueur interrogatif) :

Segments du passage :

– S1 : *"A 71 ans"* = 0,8
– S2 : *"Nelson Mandela"* = 2,0
– S3 : *"est libéré"* = 1,6
– S4 : *"après 27 ans"* = 2,0
– S5 : *"de prison"* = 2,0
– S6 : *"à Pollsmor"* = 0,4
– S7 : *"par Frederik Willem de Klerk"* = 0,9

Segments de la question :

– SQ1 : *"avait"* = 0,8
– SQ2 : *"Nelson Mandela"* = 1,8
– SQ3 : *"à sa libération de prison"* = 0,9

Au cours des sections suivantes, nous expliquons le calcul du coût de chaque type d'opération. Pour illustrer ces calculs, nous utilisons l'exemple de référence de la figure 5.32, et nous le déroulons au fur et à mesure de la description l'algorithme de recherche. L'algorithme doit évaluer les coûts des substitutions et des rattachements associés : la meilleure transformation sera déterminée en fonction de ces coûts. Nous présentons les types d'opération dans l'ordre dans lequel l'algorithme les applique : d'abord les opérations de substitution et rattachement, et ensuite les opérations de suppression et insertion. Enfin, nous présentons le calcul du coût total.

5.5.4.3 Opérations de substitution

Le calcul du coût de transformation commence par les opérations de substitution. En fonction des ancres identifiées lors de l'étape des pré-traitements, un certain nombre d'opérations ont été générées (voir section 5.5.3). Leur coût n'a pas encore été calculé, mais leur type (simple, fusion, scission) ainsi que les mots impliqués par la substitution et les transformations appliquées ont été enregistrés. De ce fait, l'algorithme de recherche va déterminer les suites d'opérations de substitution possibles, et leur coût. A chaque fois qu'une opération de substitution est appliquée, elle sera suivie de l'opération de rattachement correspondante. Le coût d'une opération de substitution dépend du coût de remplacement des mots du segment du passage par les mots du segment de la question. Dès qu'une suite

d'opérations conduit à l'état final du graphe de recherche (lorsqu'il n'y a plus d'opérations possibles), le coût total des substitutions est calculé.

Le coût de substitution dépend du contenu de la relation d'ancre entre le (ou les) segment(s) du passage et le (ou les) segment(s) de la question. Le coût de substitution prend en compte le type des transformations entre les mots en commun, ainsi que le nombre de mots importants des deux segments pour lesquels on n'a pas trouvé d'équivalence. L'idée est donc de reprendre la valeur du poids de chaque segment, mais en ne gardant que le poids des mots, et pas le poids des relations.

Si un segment de la question et un segment du passage ont un contenu sans aucune similarité, leur substitution peut être représentée par la suppression des mots du segment du passage, et l'insertion des mots du segment de la question. Le coût d'une telle opération correspond alors au poids des mots des deux segments. On peut appeler cette valeur le coût maximum d'une substitution. Si les deux segments partagent un mot similaire, avec une similarité identique, il faut alors enlever le coût de ce mot dans la valeur du poids des mots des deux segments. Par contre, si le type de la similarité n'est pas identique, nous voulons pénaliser cette similarité. Cette pénalité correspond alors au poids associé à la similarité. Ce poids a été fixé en fonction de la hiérarchie des similarités : identité, lemme, morpho-syntaxique, synonyme. Le type identité ayant un poids nul, et synonyme le poids le plus élévé.

Du fait de ce raisonnement, le coût de substitution correspond à la somme du poids en mots du segment du passage et du segment de la question, auquel on soustrait le produit du nombre de mots similaires par le poids des mots important. Ce produit est multiplié par deux, car un mot commun apparaît à la fois dans la question et le passage. Enfin, pour pondérer selon le type des transformations, on ajoute au calcul la somme des poids de chaque transformation. On note $Csub_{i,j}$ le coût de substitution entre le segment i du passage et le segment j de la question. $Pmots_i$ et $Pmots_j$ sont respectivement le poids des mots du segment i du passage et le poids des mots du segment j de la question. $NbSim$ correspond au nombre de similarité entre les deux segments, et λ_{imp} au poids des mots importants. Enfin, $Psimilaritési, j$ correspond au poids total des similarités entre les deux segments, c'est à dire la somme des poids de chaque similarité identifiée. La formule est donc :

$$Csub_{i,j} = Pmots_i + Pmots_j \text{ - } 2 * (NbSim * \lambda_{imp}) + Psimilarités_{i,j}$$

Les formules pour les opérations de fusion et de scission sont relativement similaires, la différence étant qu'il y a soit plusieurs segments du passage, soit plusieurs segments de la question. De ce fait, on fait respectivement la somme du poids des mots des segments du passage ou la somme du poids des mots des segments de la question. On note $Csub_{(1,...,F),j}$ l'opération de fusion entre les segments du passage 1 à F compris dans la fusion et le segment de la question j. L'opération de scission est noté $Csub_{i,(1,...,S)}$, où le segment i du passage est substitué par les segments 1 à S de la question. Les deux formules sont présentées ci-dessous :

$$Fusion : Csub_{(1,...,F),j} = \sum_{f=1}^{F}(Pmots_f) + Pmots_j + ...$$

$$Scission : Csub_{i,(1,...,S)} = Pmots_i + \sum_{s=1}^{S}(Pmots_s) + ...$$

Si on reprend l'exemple de référence, quatre opérations de substitution sont générées. Prenons le cas de l'opération de substitution simple entre les deux segments *Nelson Mandela*. La logique, vu que les segments sont identiques, est que le coût de substitution soit nul. Le poids des mots de ces deux segments est identique, c'est à dire 0,8. Il y a deux équivalences, entre les mots *Nelson* et *Mandela*. Le poids des mots importants est le même que celui utilisé dans l'exemple du poids des segments, c'est à dire 0,4. Comme les mots sont équivalents, il n'y a pas de poids de transformation associé. Le coût de substitution du segment est donc le suivant :

$$Csub_{2,2} = 0,8 + 0,8 - 2 * (2 * 0,4) + 0 = 1,6 - 1,6 = 0$$

Prenons maintenant l'opération de fusion générée, qui concernait la substitution des segments du passage *est libéré* et *de prison* et le segment de la question *à sa libération de prison*. Le poids des mots de ces segments est respectivement 0,4, 0,4 et 0,8. Il y a deux équivalences, sur le mot *prison*, et entre les mots *libéré* et *libération*, qui fait donc appel à une transformation morpho-syntaxique. Ce type de transformation a un poids de 0,2. Le coût de la substitution fusion est la suivante :

$$Csub_{(3,5),3} = (0,4 + 0,4) + 0,8 - 2 * (2 * 0,4) + 0,2 = 1,6 - 1,6 + 0,2 = 0,2$$

Le coût de substitution des deux autres opérations est indiqué ci-dessous. Ces résultats seront réutilisés dans la suite de ce chapitre.

Substitution simple entre *est libéré* et *à sa libération de prison* : 0,6

Substitution simple entre *de prison* et *à sa libération de prison* : 0,4

A chaque fois qu'une opération de substitution est calculée et appliquée, l'opération de rattachement associée sera alors appliquée à son tour. Nous décrivons ces opérations de rattachement dans la section suivante.

5.5.4.4 Opérations de rattachement

Une fois le coût de substitution calculé, le système va procéder au rattachement du segment (ou des segments dans le cas d'une opération de fusion) au segment contenant la réponse évalué. Les rattachements sont fait pour le moment entre le segment substitué et le segment contenant la réponse. L'idée générale est que même si on retrouve des segments communs entre la question et le passage, les mots équivalents d'un passage ne sont pas forcément en rapport avec la réponse évaluée. Le coût de rattachement représente cette notion.

L'objectif est d'utiliser les relations entre les segments identifiées ainsi que les types des segments. Le système analyse d'abord si le segment à rattacher a une relation commune avec le segment contenant la réponse. Les relations temporelles et spatiales ne sont pas considérées comme relations valides dans ce cas (voir section 5.5.2). Si c'est le cas, on estime que le coût de rattachement est nul. Sinon

le coût de rattachement va être calculé en effectuant des permutations successives jusqu'à ce que le segment soit rattaché au segment réponse. Chaque permutation aura un coût qui sera calculé à partir des relations existantes et des types des segments, l'idée étant de représenter par ces coûts la vraisemblance qu'un segment soit en relation avec la réponse.

Le coût de chaque permutation est déterminé par le biais de règles. Ces règles prennent en compte le type des deux segments que l'on veut permuter, ainsi que des relations existant entre eux. Ainsi, un segment temporel ou spatial sera moins coûteux à permuter qu'un segment groupe nominal. De même, un groupe verbal, à cause de son importance aussi bien au niveau de la structure d'une phrase que de son sens, ne pourra être permuté que dans certains cas bien précis.

Il est à noter que selon certains contextes, un rattachement peut ne pas être possible, et dans ce cas l'opération de substitution associée devient impossible à appliquer. Cependant, l'opération n'est pas pour autant supprimée des opérations de substitution possibles dans le graphe de recherche. En effet, l'application d'autres opérations de substitution, et les opérations de rattachements qui vont avec, peuvent faire que l'opération devient de nouveau applicable. Ce genre de situation est assez fréquente avec les segments verbaux, qui du fait de leur importance dans la structure d'une phrase, ont des règles de permutation assez strictes.

Un dernier point important sur ces permutations est qu'elles ne changent pas la structure d'une phrase. On parle de permutations, mais il s'agit surtout d'un rattachement entre deux segments (le segment réponse et le segment affecté par l'opération de substitution) qui est effectué en plusieurs étapes. De ce fait, on peut voir ce calcul comme la construction de nouvelles relations entre certains segments, et non comme le déplacement de segments dans la phrase.

Ainsi, le coût de rattachement entre un segment i du passage et le segment r contenant la réponse candidate est noté $Crat_{i,r}$. Ce coût correspond à la suite des permutations nécessaires entre i et r, que l'on note $\sum_{p=i+1}^{p=r}(Cperm_{i,p})$. On peut lire cette notation comme étant la somme des poids de chaque permutation entre le segment i et le segment r. Chaque permutation est effectuée entre i et un segment p. Ce segment est un des segments se trouvant entre i et r. La formule complète est décrite ci-dessous :

$$Crat_{i,r} = \sum_{p=i+1}^{p=r}(Cperm_{i,p})$$

Dans le cas d'une opération de substitution fusion, un coût de rattachement entre les segments du passage est aussi calculé. Sur le même modèle que le rattachement entre un segment du passage et le segment réponse, le système va rattacher les segments de la fusion. L'approche est faite séquentiellement, en prenant d'abord le segment de la fusion le plus éloigné de la réponse, et en calculant le coût de rattachement avec le segment de la fusion le plus proche. Puis, s'il y a plus de deux segments concernés par la fusion, un deuxième coût de rattachement est calculé, cette fois-ci entre le deuxième segment de la fusion et le suivant. L'opération est répétée jusqu'à ce que le dernier segment de la fusion soit rattaché.

Prenons l'exemple de référence de la figure 5.32. Il y a quatre opérations de substitution, trois simples et une fusion. L'algorithme de recherche, après avoir calculé le coût de substitution de ces quatres opérations, va entrer dans l'état initial de la recherche, et va choisir la première opération à appliquer. Les coûts de rattachement vont être calculés pour ces quatre opérations.

- Rattachement de *Nelson Mandela* :
  - le segment n'est pas en relation directe avec le segment réponse.
  - première permutation : le segment est directement voisin de la réponse, fin de la permutation.
  - coût du rattachement = 0

- Rattachement de *est libéré* :
  - le segment n'est pas en relation directe avec le segment réponse.
  - première permutation : avec le segment *Nelson Mandela*
    - permutation entre un segment verbal et un segment nominal en relation verbale
    - permutation stricte car segment verbal
    - *Nelson Mandela* n'est pas en relation avec la réponse, permutation impossible
  - rattachement impossible pour le moment

- Rattachement de *de prison* :
  - le segment n'est pas en relation directe avec le segment réponse.
  - première permutation : avec le segment *après 27 années*
    - permutation entre un segment nominal et un segment temporel en relation temporelle
    - permutation gratuite car segment temporel
  - deuxième permutation : avec le segment *est libéré*
    - permutation entre un segment nominal et un segment verbal en relation verbale
    - permutation avec un coût moyen (segment verbal en relation)
    - coût permutation = 0,2
  - troisième permutation : avec le segment *Nelson Mandela*
    - permutation entre deux segments de type nominal en relation verbale
    - permutation avec un coût moyen (segment nominal en relation)
    - coût permutation = 0,2
  - quatrième permutation : le segment est directement voisin de la réponse, fin du rattachement.
  - coût du rattachement = 0,4

- Rattachement de *est libéré* et *de prison* (fusion) :
  - Rattachement de *de prison* avec *est libéré* :
    - les deux segments sont en relation directe
    - coût de rattachement nul.
  - Rattachement de *est libéré* avec le segment réponse :
    - Rattachement impossible (voir plus haut)

Parmis ces quatre calculs, deux sont impossibles à cette étape de la recherche. Si l'on se place au niveau de l'algorithme de recherche, cela signifie que seules deux opérations de substitution pourront être appliquées. De ce fait, à l'état initial du graphe de recherche, l'algorithme a le choix entre deux noeuds à développer, dont le coût correspond à ceux des opérations de substitution et rattachement associées. Ainsi, le coût $Cnoeud$ pour passer d'un noeud à un autre est formalisé ci-dessous :

$$Cnoeud = Csubi, j + Crati, r * \lambda_{ratt}$$

oú λ_{ratt} permet de pondérer les rattachements par rapport aux substitutions.

Dans cet exemple, nous considérons le poids des rattachements comme étant fixé à 1, de ce fait le calcul du coût total des opérations de substitution est le suivant :

Substitution de *Nelson Mandela* = 0 + 0 = 0

Substitution de *de prison* = 0,4 + 0,4 = 0,8

L'algorithme prend donc comme première solution la substitution de *Nelson Mandela*. Le nouveau noeud du graphe de recherche comporte désormais trois opérations, dont deux qui étaient impossibles précédemment. Cependant, le rattachement de *Nelson Mandela* peut permettre à ces deux opérations d'être désormais applicables.

- Rattachement de *de prison* :
  - Même coût que précédemment.
- Rattachement de *est libéré* :
  - le segment n'est pas en relation directe avec le segment réponse.
  - première permutation : avec le segment *Nelson Mandela*
    - permutation entre un segment verbal et un segment nominal en relation verbale
    - permutation stricte car segment verbal
    - *Nelson Mandela* est en relation avec la réponse.
    - permutation avec coût élevé : 0,4
  - seconde permutation : segment réponse, fin du rattachement
  - coût du rattachement = 0,4
- Rattachement de *est libéré* et *de prison* (fusion) :
  - Rattachement de *de prison* = 0
  - Rattachement de *est libéré* = 0,4 (voir opération précédente)

Le coût total pour passer d'un état à un autre est donc :

Substitution de *de prison* = 0,8

Susbtitution de *est libéré* = 0,6 + 0,4 = 1,0

Substitution fusion de *est libéré* et *de prison* = 0,2 + (0+0,4) = 0,6

Le coût total du noeud du graphe de recherche développé étant 0, l'opération suivante choisie est celle de fusion. Le nouveau noeud n'a alors plus que deux opérations possibles. Cependant, ces deux opérations impliquent un segment de la question qui a déjà été utilisé dans une opération de substitution (*à sa libération de prison*). De ce fait, elles deviennent impossibles à appliquer, et il n'y a plus d'opérations de substitution possibles pour ce noeud du graphe de recherche. C'est donc un état final avec un coût total de 0,6 de substitution. La transformation du passage n'est pas encore terminée, car il faut désormais appliquer les opérations de suppression et d'insertion, ce que nous voyons plus bas. Pour des raisons de démonstration, voici les deux autres suites d'opérations de substitution possibles ainsi que leur coût :

Substitution et rattachement des segments *Nelson Mandela* et *de prison* : 0,8

Substitution et rattachement des segments *Nelson Mandela* et *est libéré* : 1,0

5.5.4.5 Opérations de suppression

Les opérations de suppression ne commencent à être appliquées par le système qu'à partir du moment où l'ensemble des opérations de substitution applicables pour un état donné du graphe de recherche est vide. Le système détermine quels segments du passage sont identifiés comme ayant un coût de suppression non nul. Nous expliquons dans la section 5.5.2 sur quels choix est basée l'identification des segments.

L'approche est la suivante : l'algorithme traite séquentiellement chaque segment. Si le segment n'a pas été traité par une opération de substitution, le système détermine alors si la suppression est coûteuse. Pour identifier dans quels cas le segment se trouve, l'algorithme va à la fois analyser si le segment a au moins une relation commune avec le segment réponse ou l'un des segments substitués, et aussi si le segment a été affecté par un des rattachements effectués lors des substitutions. Si l'une des deux conditions est validée, alors le segment est considéré comme ayant un coût de suppression non nul.

Le cas de la suppression des segments de temps et de lieu est particulier. Un segment de ces deux types est automatiquement considéré comme ayant un coût de suppression nul si aucun segment du même type n'est affecté par les transformations de substitution. Sinon, on ajoute le poids du segment au coût total de suppression.

Une fois les segments coûteux identifiés par l'algorithme, le calcul est très simple : les segments non coûteux seront considérés comme *gratuits* à supprimer. Pour les autres, le coût correspondra simplement au poids du segment que l'on souhaite supprimer. Le coût total de suppression correspondra donc à la somme des poids des segments coûteux, pondéré par un paramètre. On note $Copsuppr$ le coût de suppression des segments d'un passage. Cette suppression correspond à la somme du poids de chaque segment supprimé, que l'on note $\sum_{i=1}^{X}(Pseg_i)$. Ce coût de suppression est pondéré par

une valeur λ_{suppr}. La formule est donnée ci-dessous :

$$Copsuppr = \sum_{i=1}^{X}(Pseg_i) * \lambda_{suppr}$$

Reprenons le passage de l'exemple de référence de la figure 5.32, avec comme première suite d'opérations de substitution :

Substitution simple de *Nelson Mandela* et substitution fusion de *est libéré* et *de prison* pour un coût de 1,1.

Dans ce cas, les trois segments suivants n'ont pas été affectés par les substitutions : *après 27 années*, *à Pollsmor* et *par Frederik Willem de Klerk*. Ces trois segments ont respectivement un poids de 2,0, 0,4 et 0,9. Le segment *après 27 années* sera considéré comme coûteux car un des segments affectés par les opérations de substitution est aussi de type temporel, en l'occurence *A 71 ans* . De même, le segment *par Frederik Willem de Klerk* est considéré comme coûteux à supprimer car il est relation avec *de prison*. Par contre, le dernier segment est gratuit à supprimer : *à Pollsmor* est le seul segment spatial. De ce fait, le coût total de suppression est de :

$Copsuppr$ = (2,9) * λ_{suppr} = 5,8

avec λ_{suppr} = 2.

Si on prend les deux autres suites d'opérations de substitution, le coût total va être différent. En effet, dans les deux cas, un segment supplémentaire n'a pas été affecté par les substitutions : soit *est libéré*, soit *de prison*. Le coût de suppression total dans ces deux cas est indiqué ci-dessous :

- Coût de suppression pour la suite d'opération :
  - Substitution simple de *Nelson Mandela* et substitution simple de *de prison*
  - le segment *après 27 années* a été affecté par des permutations, la suppression est donc coûteuse.
  - le segment *est libéré* a été affecté par des permutations, la suppression est donc coûteuse.
  - le segment *par Frederik Willem de Klerk* est en relation avec *de prison*, la suppression est donc coûteuse.
  - poids des segments : 2,0, 1,6, et 0,9, suppression totale : (2,0+1,6+0,9)*2 = 9,0
- Coût de suppression pour la suite d'opération :
  - Substitution simple de *Nelson Mandela* et substitution simple de *est libéré*
  - le segment *après 27 années* a été affecté par des permutations, la suppression est donc coûteuse.
  - le segment *de prison* a été affecté par des permutations, la suppression est donc coûteuse.
  - poids du segment : 2,0 et 2,0, suppression totale : (2,0+2,0)*2 = 8,0

5.5.4.6 Opérations d'insertion

Les opérations d'insertion sont appliquées après le processus de suppression dans le passage. Le système identifie les segments de la question qui n'ont pas été affectés par les opérations de substitution. Contrairement au processus de suppression, le système ne fait pas de traitements différent selon le type du segment. L'objectif est simplement d'insérer l'information manquante. De ce fait, le coût total d'insertion correspond simplement à la somme des poids des segments à insérer. On note $Copins$ le coût d'insertion total des segments de la question. Ce coût correspond à la somme des poids des segments j de la question, que l'on note $\sum_{j=1}^{X}(Pseg_j)$. Là aussi, ce coût total est pondéré par un paramètre, que l'on note λ_{ins}. La formule est présentée ci-dessous :

$Copins = \sum_{j=1}^{X}(Pseg_j) * \lambda_{ins}$ avec λ_{ins} = 4

Si on reprend l'exemple de référence, il n'y a qu'un seul segment à insérer quelque soit la suite d'opérations de substitution : *avait* , dont le poids est de 0,8. Le coût total de suppression est donc :

Copins = 0,8 * 4 = 3,2

5.5.4.7 Coût total

Les trois composantes du coût de transformation ont été calculées, et ce quelque soit la suite d'opérations de substitution. On a donc les coûts suivant :

Substitution simple de *Nelson Mandela* et substitution fusion de *est libéré* et *de prison*

- CTP = 0,6 + 5,8 + 3,2 = 9,6

Substitution simple de *Nelson Mandela* et substitution simple de *de prison*

- CTP = 0,8 + 9,0 + 3,2 = 13,0

Substitution simple de *Nelson Mandela* et substitution simple de *est libéré*

- CTP = 1,0 + 8,0 + 3,2 = 12,2

Le coût de transformation minimum CTP^* de ce passage est donc de 9,6. Ce coût sera ensuite comparé à ceux obtenus sur les autres passages de la réponse candidate. Le score de la réponse candidate est le coût de transformation le plus faible. Ce score est ensuite comparé à celui des autres réponses. Le candidat réponse avec le score le plus faible est choisi comme étant la bonne réponse à la question traitée.

Nous présentons dans la section suivante comment sont fixés les différents paramètres utilisés dans le calcul du coût de transformation.

5.5.5 Paramétrage du système

Nos calculs s'appuient sur des valeurs de pondérations ainsi que des poids fixés. Cette courte section a pour objectif de présenter les différentes valeurs de paramétrage utilisées, et comment elles ont été fixées. Les valeurs de paramétrages peuvent être classés en deux types :

– les valeurs de pondérations, utilisées pour pondérer un calcul par rapport à un autre.
– les valeurs de poids et de règles, utilisées pour unifier les différents coûts des règles et les poids.

Les valeurs de pondérations sont au nombre de 5 : facteur de suppression, facteur d'insertion, facteur de rattachement, facteur des relations et facteur des relations sortantes. Ces 5 valeurs permettent de pondérer un calcul par rapport à un autre. Les valeurs de poids et de règles sont au nombre de 3 : faible, moyen et important. Elles sont utilisées pour quantifier les poids des transformations pour les relations d'ancres, le poids des mots importants, le poids des relations et le poids des règles de permutation. Elles permettent ainsi d'avoir des coûts et des poids comparables. Ces valeurs ont d'abord été déterminées de manière empirique, après étude de corpus, avant d'être fixées par le biais d'expériences qui ont contribuées à régler ces valeurs.

5.5.6 Conclusions sur le module de calcul du coût de transformation

Cette section clôt la description du module de calcul du coût de transformation. L'objectif de cette thèse est de proposer une méthode robuste de réordonnancement des candidats réponses à une question. Notre contexte de travail implique de devoir traiter différents types de documents. Cette contrainte a un impact fort sur notre approche pour le réordonnancement. Il est en effet pas possible d'utiliser une analyse syntaxique pour représenter l'information structurelle contenue dans une question ou un passage. Or, différentes approches de réordonnancement étudiées s'appuyait au moins en partie sur de l'information syntaxique. Si les annotations fournies par l'analyseur de Ritel apportent de l'information sémantique, l'information syntaxique est quasiment absente. Nous avons donc décidé d'utiliser le formalisme présenté dans le chapitre 4 : les segments typés et les relations permettent de représenter l'information structurelle de manière simple mais robuste.

A partir de ce formalisme, l'objectif est d'identifier le passage d'une réponse candidate étant le plus proche de la question. Pour calculer ce score de similarité, nous utilisons une distance d'édition sur composants, que nous adaptons à notre problème : nous considérons les segments typés comme étant pertinents pour effectuer les opérations d'édition. Cette distance permet de représenter la similarité entre un passage : l'objectif est alors de calculer un coût de transformation, composée deux éléments principaux : la similarité structurelle et la similarité sémantique du passage à la question. Pour

quantifier cette similarité, nous utilisons quatre types d'opérations : la substitution, l'insertion et la suppression, auxquelles nous ajoutons le rattachement. L'objectif est ainsi de prendre en compte l'impact sémantique et syntaxique d'une transformation, tout en restant relativement robuste à n'importe quel type de données (journalistique, oral, web).

Les opérations de substitution, d'insertion et de suppression permettent principalement de quantifier la différence sémantique entre un passage et une question. Le coût de ces opérations est calculé en fonction des mots contenus dans les segments sur lesquels sont appliquées les opérations. L'opération de rattachement permet quant à elle de quantifier la différence structurelle. Pour le moment, nous estimons que chaque segment de la question est relié au segment marqueur interrogatif. Si ce n'est évidemment pas toujours le cas, cette hypothèse est validée par l'étude des questions traitées par le réordonnanceur. De ce fait, nous considérons que les segments du passage doivent être rattachés au segment de la réponse candidate. L'opération de rattachement permet de quantifier le coût nécessaire pour relier un segment du passage au segment réponse. Ces quatre opérations permettent ainsi de calculer la similarité sémantique et structurelle entre un passage et une question. Cette approche est évaluée dans le chapitre 7.

Cette approche semble adaptée à notre contexte de travail, mais un certain nombre de problèmes semblent encore présents. Si cette approche permet d'évaluer la similarité structurelle, notamment par le biais des opérations de rattachement, la méthodologie est surtout faite pour traiter une structure robuste mais relativement simple. Ainsi, nous n'avons pas de structure syntaxique précise. De même, la similarité sémantique est évaluée à un niveau local, sur un segment : il n'existe pas de relations sémantiques pour pouvoir étendre cette similarité. De plus, nous avons effectué des simplifications pour utiliser les segments de temps et de lieu, et les relations liées à ces deux segments. Notre objectif était de représenter des compléments circonstanciels et le fait de pouvoir les déplacer librement dans une proposition. Cependant, il est difficile de détecter les propositions, particulièrement sur l'oral, et nous permettons donc à ces segments d'être déplaçables dans une phrase entière. Cependant, cette particularité n'était pas totalement adaptée à notre travail, ce qui nous a amené à faire plusieurs simplifications. Nous ne sommes pour le moment pas totalement convaincu de l'utilisation de ce type de segments. Un travail est en cours pour améliorer l'utilisation de ces segments.

Discussion

Nous avons présenté dans cette partie les contributions de notre travail. Notre objectif était de proposer un module de réordonnancement de réponses candidates à une question répondant à plusieurs contraintes : être robuste à tous types de documents en entrée, et ne traiter que le français. Par ailleurs, ce travail a été réalisé dans le cadre du système de questions-réponses Ritel. Si le réordonnanceur a pour objectif d'être généralisable à d'autres systèmes, nous avons néanmoins du prendre en compte les avantages et inconvénients de ce système. Il a été montré notamment dans le chapitre 2 que si les résultats obtenus par Ritel étaient bons, l'approche utilisée pour sélectionner et extraire les réponses candidates montrent certaines limites. L'une des raisons expliquant ces résultats est la non prise en compte de la structure et des dépendances entre groupes de mots au sein des questions et des phrases des documents. L'analyse utilisée par Ritel est en effet fortement sémantique, mais fournie très peu d'information syntaxique.

Nous avons mené dans le chapitre 3 une étude des différentes approches utilisées pour le réordonnancement de réponses. Nous avons aussi étudiés certaines approches employées pour la sélection et l'extraction des réponses, car ces dernières utilisent des méthodes souvent très proches. Nous avons ainsi pu observer que toutes ces approches s'appuient sur une analyse qui est prépondérante dans les traitements utilisés. Par ailleurs, certains systèmes montrent qu'il est possible d'employer des analyses syntaxiques dans un cadre différent de l'écrit journalistique, comme le web [Tannier & Moriceau 2010]. Néanmoins, ces analyses se doivent d'être adaptées au cadre d'implication. Ainsi, le système de l'UPC [Comas et al. 2010], qui travaille sur des transcriptions écrites de l'oral, s'appuie sur un analyseur interne fournissant habituellement de l'information syntaxique et sémantique. Cet analyseur a du être adapté au contexte de l'oral. Ainsi, l'analyseur ne fournit que de l'information syntaxique.

A partir de ces observations, nous avons décidé de concevoir un modèle de représentation des questions et des phrases des documents. L'analyse de Ritel fournissant principalement de l'information sémantique, ce modèle doit fournir de l'information syntaxique, principalement la structure des phrases et des questions. Il doit aussi être robuste à tous types de documents (écrit, oral, web). Ce modèle de représentation est scindé en deux parties : la segmentation des questions et des phrases en composants typés, et les relations identifiées entre ces segments typés. Pour la segmentation, nous nous sommes principalement inspirés des Syntagmes Non Récursifs présentés dans [Vergne 1999].

[Paroubek et al. 2008b] propose un formalisme plus proche de ce que l'on a l'habitude de voir dans la littérature. Néanmoins, l'évaluation d'analyseurs syntaxiques utilisant ce formalisme a montré que les résultats obtenus n'étaient pas bons pour l'oral. Nous avons donc fait le choix d'utiliser un formalisme de segmentation plus simple, composé de deux types principaux, nominal et verbal, auxquels s'ajoutent quatre sous-types. A partir de cette segmentation, des relations sont ajoutées entre les différents segments. Ces relations permettent de représenter principalement les dépendances entre les différents segments. Là encore, notre représentation reste relativement simple, de manière à être appliquée dans un cadre robuste.

Le module de réordonnancement s'appuie sur cette représentation des questions et des documents. L'objectif du réordonnancement est de classer les réponses candidates fournies par Ritel selon un nouveau score. Nous avons représenté la tâche de réordonnancement comme un calcul de la similarité entre un passage contenant une réponse candidate et la question. Nous nous sommes inspirés des travaux présentés dans [Kouylekov & Negri 2010] pour calculer ce score de similarité. Nous quantifions donc cette similarité selon un coût de transformation entre un passage contenant la réponse candidate et la question. Ce coût est déterminé à partir d'un calcul d'une distance d'édition entre le passage et la question. Ritel retournant plusieurs passages pour une même réponse candidate, il a été décidé de tous les évaluer. Le coût de transformation le plus faible obtenu sur l'ensemble des passages est choisi comme score de la réponse candidate. Le candidat réponse ayant le score le plus faible est choisi comme bonne réponse à la question.

Le module de réordonnancement est divisé en deux phases : les pré-traitements, et le calcul du coût de transformation. Ces pré-traitements ont pour objectif de préparer le calcul du coût de transformation. Outre segmenter les questions et phrases du passage et ajouter les relations entre les segments, les pré-traitements ont pour objectif de détecter les similarités entre les mots du passage et ceux de la question. A partir de ces similarités, des ancres sont créées entre les segments de la question et ceux du passage. Ces ancres représentent les segments similaires entre la question et le passage. Le calcul du coût de transformation s'appuie sur ces ancres. L'approche employée utilise un calcul de la distance d'édition sur composants adapté à notre contexte de travail. La distance d'édition utilise des opérations de transformation, que nous appliquons sur les segments. Habituellement, trois types d'opération sont utilisées : l'insertion, la suppression et la substitution. Nous avons décidé d'ajouter un quatrième type, le rattachement. Ce type d'opération a pour objectif de déterminer si un segment du passage est relié à la réponse candidate. L'idée est de s'appuyer sur les relations et aini prendre en compte la structure d'une phrase dans nos calculs.

Si les traitements mis en place dans le module de réordonnancement restent relativement simples, notamment au niveau des règles utilisées pour les opérations de rattachement, nous estimons que notre approche est relativement bien adaptée au problème de la robustesse. Le modèle de représentation utilisé semble applicable dans n'importe quel contexte. De plus, le calcul du coût de transformation est basé sur un calcul de la distance d'édition. Nous estimons que ce type d'approche est particulièrement adaptée aux différents types de données.

Troisième partie

Evaluation et analyse

Introduction

L'un des objectifs à long terme d'un système de questions-réponses est de tendre vers un système de dialogue entre l'homme et la machine. Cependant, un dialogue amène un grand nombre de problématiques. De ce fait, définir des systèmes permettant de répondre aux questions posées par un utilisateur est une première étape vers un dialogue entre l'homme et la machine.

De tels systèmes demandent des évaluations spécifiques pour quantifier les progrès effectués dans le domaine. La tâche se doit donc d'être ni trop simple ni trop complexe, de manière à comprendre la pertinence des approches envisagées. Depuis la définition de la tâche de questions-réponses par [Voorhees & Tice 2000], les campagnes d'évaluation qui ont suivi ont essayé de maintenir l'équilibre entre intérêt applicatif, scientifique et difficulté.

Du fait de la complexité de la tâche, plusieurs points essentiels doivent entrer en compte lors de la mise en place d'une évaluation pour un système de questions-réponses :

– quels types de questions doivent être traitées ?
– comment créer ces questions ?
– comment évaluer la pertinence des réponses ?

Ces problématiques principales n'ont pas de réponses correctes ou incorrectes, mais dépendent plus de l'orientation que les organisateurs veulent donner à une campagne d'organisation. Ainsi, pour répondre à ces problématiques, les campagnes ont beaucoup évoluées au fur et à mesure des années. Par exemple, certaines campagnes ont intégré au cours des années de nouveaux types de questions que les systèmes devaient traiter. D'autres ont modifié le processus de création des questions.

Dans le chapitre 6, nous détaillons les caractéristiques principales d'une campagne d'évaluation, avant d'en présenter une sélection représentative. Pour chacune des campagnes auxquelles notre système a participé, nous en détaillons les caractéristiques spécifiques.

Une fois ces campagnes présentées, nous procédons dans le chapitre 7 à l'évaluation de notre système, qui sera divisée en deux parties : l'évaluation du segmenteur par rapport à un corpus de test annoté manuellement, et l'évaluation du réordonnanceur sur les données de plusieurs campagnes d'évalua-

tion.

Ensuite (chapitre 8), nous procédons à l’analyse de ces résultats, et notamment nous présentons une analyse modulaire des résultats obtenus par notre système. Enfin, nous finissons par une discussion globale.

Chapitre 6

Présentation des campagnes d'évaluation

6.1 Présentation générale

Nous allons tout d'abord présenter succintement les campagnes internationales et françaises. Nous développons plus précisement les caractéristiques de chacune de ces campagnes dans le chapitre suivant.

La toute première conférence internationale à avoir intégré une tâche d'évaluation des systèmes de questions-réponses se nomme TREC [Press 2005] (Text REtrieval Conference). Cette campagne américaine est organisée par le NIST, et la tâche sur les systèmes de questions-réponses est intégrée depuis 1999 [Voorhees & Tice 1999], et ce jusqu'en 2007 [Dang, et al. 2007]. A partir de 2008, la conférence s'est dirigée vers de l'extraction d'opinion, et ne rentre plus dans nos intérêts. Une évaluation équivalente existe en Europe : CLEF (Cross-Language Evaluation Forum). L'objectif de cette conférence est d'être l'équivalent de TREC mais pour les langues européennes. CLEF a été créée en 2000 [Peters & Braschler 2001], et une tâche pour les systèmes de questions-réponses existe depuis 2003 [Magnini, et al. 2003] (QA@CLEF). Cette tâche n'existe plus depuis 2009 (arrêt en 2008). Par ailleurs, une sous-tâche, QAst (Question-Answering on Speech Transcriptions) existe depuis 2007 [Turmo, et al. 2007], et nous intéresse particulièrement, le système RITEL y ayant participé. Cette évaluation a pour objectif d'évaluer les systèmes sur des transcriptions manuelles ou automatiques de données orales, contrairement aux documents textuels habituellement utilisés dans ce type de campagne. QAst a duré jusqu'en 2009. Enfin, la campagne NTCIR (NII Test Collections for IR Systems) est un équivalent asiatique de TREC et CLEF. NTCIR a été créée en 1999, et une tâche pour les systèmes de questions-réponses existent depuis 2002 [Fukumoto, et al. 2002].

Pour les évaluations nationales deux ont eu lieu jusqu'à présent (l'une d'elle est toujours en cours). La première, EQueR [Ayache, et al. 2006] (Evaluation des systèmes de Questions-Réponses), a été créée dans le cadre du projet EVALDA, qui n'a pas eu de suite pour le moment. Enfin, la dernière

évaluation, toujours en cours, se fait dans le cadre du projet franco-allemand Quaero [Quintard 2009]. Ce projet porte sur le contenu numérique, et de ce fait l'extraction d'informations.

Le tableau 6.1 tiré de la thèse d'Olivier Galibert [Galibert 2009] illustre les caractéristiques de chacune de ces campagnes.

Nous décrivons plus en détails les caractéristiques spécifiques de chaque campagne dans la suite de ce document. Nous avons évalué notre système sur les données de trois campagnes : QA@CLEF, QAst et Quaero. Par ailleurs, nous avons participé à deux d'entre elle (QAst et Quaero). De ce fait, nous orientons la suite de notre présentation sur ces trois campagnes. Nous présentons d'abord les types de questions traités et comment ces dernières sont créées. Puis nous présentons les types de documents, et nous concluons enfin sur les métriques utilisées pour évaluer les systèmes.

6.2 Définition des questions

6.2.1 Types des questions

Lors de la création d'une campagne d'évaluation, un des premiers points à traiter est le type de questions à traiter. Nous donnons ci-dessous les types de questions que l'on peut retrouver dans les différentes campagnes d'évaluation.

- Question *factuelle* : *Quel est le nom du Président de la France ?*
- Question *oui/non* : *Est-ce-que l'Irlande a gagné le tournoi des VI Nations en 2010 ?*
- Question *définition* : *Qu'est ce que l'ONU ?*
- Question *pourquoi* : *Pourquoi De Gaule a-t-il démissionné ?*
- Question *comment* : *Comment construire un système de questions-réponses ?*
- Question *liste* : *Citer trois groupes de grunge.*

Si les questions *factuelles* sont par exemple généralement traitées par l'ensemble des systèmes de questions-réponses, d'autres types, tels que les questions *pourquoi* ou les questions *oui/non*, ne sont pas toujours gérés. Certains systèmes se sont néanmoins spécialisés pour traiter ces types de questions plus *complexes*, comme le système FIDJI [Tannier & Moriceau 2010]. De ce fait, les créateurs d'une campagne doivent d'abord déterminer sur quels critères les participants vont être évalués. Par exemple, doit-on d'abord évaluer des systèmes sur des questions *factuelles*, qui sont les questions les mieux traitées par les systèmes, puis complexifier au fur et à mesure la campagne dans les itérations suivantes ? Ou au contraire proposer plusieurs types de questions dés les premières itérations d'une campagne ? Ce dernier point est particulièrement important car des études [Kato, et al. 2006] ont montré que la majorité des questions posées par les utilisateurs étaient de type *pourquoi*, *comment* ou *définition* (environ 30%). Nous allons à présent décrire les types de questions rencontrés dans les campagnes d'évaluation.

	TREC									QA@Clef Main Track							QAst			NTCIR					EQueR	Quaero		
	1999	2000	2001	2002	2003	2004	2005	2006	2007	2003	2004	2005	2006	2007	2008	2009	2007	2008	2009	2002	2004	2005	2007	2008	2005	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

Sources :

– TREC : [Voorhees & Tice 1999 ; Voorhees 2000 ; Voorhees & Tice 2001 ; Voorhees 2002 ; Voorhees 2003 ; Voorhees 2004 ; Voorhees & Dang 2005 ; Dang, et al. 2006 ; Dang et al. 2007].
– QA@Clef Main Track : [Magnini et al. 2003 ; Magnini et al. 2004 ; Vallin et al. 2005 ; Magnini, et al. 2006 ; Giampiccolo, et al. 2007 ; Forner, et al. 2008] et des messages de la mailing-list pour la définition de la tâche 2009.
– QA@Clef QAst : [Turmo et al. 2008 ; Turmo et al. 2009] et des messages de la mailing-list pour la définition de la tâche 2009.
– NTCIR : [Fukumoto et al. 2002 ; 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].
– EQueR : [Ayache et al. 2006].
– Quaero : [Quintard 2009].

TAB. 6.1 – Tableau résumant les caractéristiques des principales évaluations Question-Réponse

Comme on l'a vu précédemment, les questions *factuelles* sont les questions les plus fréquemment traitées par les systèmes de questions-réponses. La réponse attendue est en général courte, composée de très peu de mots, très souvent une entité nommée. Par exemple, pour la question *"Combien d'années Nelson Mandela a-t-il passé en prison ?"*, la réponse attendue est *"27 ans"*. Assez proche, on trouve les questions *Liste*, qui attendent plusieurs réponses de type *factuelle*. Les questions listes se divisent en deux types, les listes fermées, comme pour la question *"Donnez les 7 Merveilles du Monde"*, ou les listes ouvertes, où le nombre attendu de réponses n'est pas précisé : "*Nommez les plus grandes villes de France"*. Si évaluer la validité d'une réponse n'est pas une tâche difficile pour ce type de question, les listes ouvertes peuvent provoquer quelques ambiguïtées quant au nombre minimum d'éléments réponses attendus.

Les questions *définitions* sont elles aussi traitées plutôt fréquemment, mais posent bien plus de problèmes au niveau de l'évaluation. Par exemple, pour la question *"Qu'est ce que QUAERO ?"*, une réponse possible pourrait être "*un projet franco-allemand"*, ce qui est correct. Néanmoins, est ce que cette définition est suffisamment complète ? Par exemple, on aurait pu préciser dans la définition que Quaero est un projet "scientifique". Pour traiter cette problématique, plusieurs approches ont été testées. Dans le cadre de TREC, le NIST a proposé une processus de création des questions de définition où les informations des documents pouvant être potentiellement contenues dans la définition devaient être classées entre critiques et optionnelles. Le problème rencontré est qu'il existait une grande variabilité d'opinion entre les différents évaluateurs. Dans le cas de CLEF, il a été décidé d'avoir une approche simplifiée, où des réponses relativement courte était acceptée, à partir du moment où elles contiennent une des informations attendues dans la définition. Par exemple, pour la question *"Qui est Nicolas Sarkozy"*, la réponse *"le président de la France"* est considérée comme étant valide.

Les questions de type *Pourquoi* (*"Pourquoi Matrix 3 est-il ennuyeux ?"*) *Comment* (*"Comment se termine Lost Highway ?"*) peuvent aussi être rencontrés dans les évaluations. Ces questions ont elles aussi tendance à poser des problèmes d'évaluation. Tout comme pour les questions de type *définition*, il est difficile de déterminer quand une réponse est suffisamment complète. De même, il n'est pas aisé pour les systèmes d'extraire, voir de construire une réponse potentielle à ce type de questions, les éléments de la réponse pouvant être répartis dans un document. Certains évaluations ont choisi de simplifier le traitement de ces questions. Ainsi, dans l'évaluation QA@CLEF, une réponse courte était attendue. Dans le cas de Quaero, les évaluateurs attendent des systèmes un extrait de documents ou une réponse correspondant à une phrase extraite d'un document. Plus généralement, ces deux types de questions ne sont pas fréquemment traités par les systèmes de questions-réponses, du fait des différences problématiques. De même, l'évaluation de la validité des réponses reste un problème.

Enfin, un dernier type assez fréquent est les questions *Oui-Non*, comme par exemple *"Hollie Waters ont-ils sorti un nouveau CD en 2010 ?"*. Si les réponses à formuler en elles-mêmes sont simples (oui ou non), les traitements à effectuer sont plus complexes que pour des questions factuelles, ou du moins relativement différents. De plus, pour des besoins d'évaluation, les campagnes demandent aux systèmes de donner un passage justificatif.

Par ailleurs, on peut retrouver dans certaines évaluations, des questions *enchainées*, qui ont pour but de simuler un dialogue entre le système et l'utilisateur. Par exemple, la première question de la liste va porter sur un sujet donné, et les questions suivantes porteront sur le même sujet :
– *Quel est le titre du prochain tome du Trône de Fer ?*
– *Quel est l'auteur ?*
– *Quel age a-t-il ?*
– ...
Ce type de questions amène en plus des problèmatiques classiques de recherche de la réponse des besoins de traitement des anaphores (implicites ou explicites) entre les questions. Dans le cadre de ce document, nous nous intéresserons aux questions factuelles.

6.2.2 Création des questions

En plus de déterminer quels types de questions traiter dans une évaluation, il faut aussi déterminer de quelle manière ces questions seront créées. En effet, selon le processus de création des questions utilisé, les systèmes ne seront pas confrontés aux mêmes problématiques. Par exemple, si les questions sont créées en paraphrasant des passages de la collection de documents contenant la réponse cherchée, les éléments de la question seront potentiellement proches de la réponse. Par contre, s'il est demandé de construire des questions toujours en utilisant des extraits des documents, mais sans que la réponse soit contenue dans cet extrait, la répartition des éléments de la question par rapport à la réponse sera différente. De même, si les personnes créant les questions sont naïves par rapport au fonctionnement d'un système de questions-réponses, la forme des questions sera potentiellement plus proche d'une question posée dans un cadre pratique.

Les procédures de création des questions entre les itérations 2004 [Magnini et al. 2004] et 2005 [Vallin et al. 2005] de QA@CLEF sont relativement similaires. Pour chacune des langues traitées dans la campagne, 100 questions sont créées à partir des corpus de documents. Les créateurs vérifient qu'il existe au moins une réponse à ces questions. En plus des 100 questions de base créées pour chaque langue, on ajoute 100 questions supplémentaires tirées des corpus des autres langues. Ces questions sont évidemment traduites dans la langue nécessaire. Il y a ainsi pour chaque langue traitée 200 questions. En 2004, les organisateurs avaient introduit des questions de type *comment* (*"Comment est mort Jimi Hendrix ?"*). Ce type a été abandonné en 2005 à cause de la difficulté pour évaluer les réponses retournées. Des questions à temporalité restreinte ont été introduites en 2005, comme par exemple *"Qui fut à la tête de la Commission européenne de 1985 à 1995 ?"*.

Entre les itérations 2008 [Turmo et al. 2008] et 2009 [Turmo et al. 2009] de la campagne QAst, le processus de création des questions est sensiblement différent. En 2008, les questions étaient créées par un évaluateur à partir des documents. En 2009, l'idée était d'avoir des questions plus spontanées. On fournissait à des locuteurs des extraits de documents. Il leur était alors demandé de poser des questions oralement sur des informations en relation avec le passage mais dont la réponse n'était pas présente. Pour 2008, 50 questions de développement et 100 questions de test ont été générées. Pour 2009, deux corpus étaient proposés à chaque fois : l'un contenant les questions posées par les

locuteurs et transcrites manuellement, et l'autre contenant ces mêmes questions mais réécrites pour qu'elles aient une forme correspondant à de l'écrit standard. De ce fait, il y avait deux corpus de développement de 50 questions, et deux corpus de test de 100 questions.

Pour Quaero [Quintard 2009], la création des questions pour 2008, 2009 et 2010 s'appuie sur les requêtes utilisateurs fournis par le moteur de recherche Exalead [1]. Ces requêtes utilisateurs ont aussi été utilisées pour créer le corpus de documents. A partir des requêtes, les évaluateurs imaginent des questions. En 2008, les évaluateurs vérifiaient qu'une bonne réponse existait dans le corpus de documents. En 2009, aucune vérification n'était effectuée. Un corpus de développement de 160 questions ainsi qu'un corpus de test de 250 questions ont été créés en 2008. En 2009, seul un corpus de test de 507 questions a été créé. En 2010, un corpus de 500 questions (pour 175 questions factuelles) a été créé en utilisant la même méthodologie qu'en 2009. De plus, deux sous-corpus ont été créés selon la procédure de QAst en 2009 : les locuteurs posent des questions oralement sur des informations en relationa avec le passage. Ces questions sont ensuite transcrites manuellement, puis réécrites pour correspondre à de l'écrit standard. Ces deux sous corpus ont une taille de respectivement de 88 et 56 questions. Les questions de ces deux sous-corpus sont toutes de type *factuelle*.

6.3 Type des documents

Avec la définition des questions, un autre point important dans la création d'une évaluation est le choix du ou des types de documents du corpus employé. Le tableau ci-dessous est aussi tiré de la thèse d'Olivier Galibert [Galibert 2009] et donne une description des documents utilisés dans les différentes évaluations de notre approche.

	QAst 08-09	QA@Clef	Quaero
Type	Parole	Journaux	Web
Années	2004	1994-95	2008
Nombre de documents	12	200K	500K
Nombre de phrases	2,3K	3M	82M
Nombre de mots	87K	70M	840M
Nombre de caractères	460K	400M	4,2G
Phrases/document	200	17	170
Mots/phrase	37	25	10
Caractères/mot	5,3	5,4	5,3

TAB. 6.2 – Types et tailles de plusieurs collections de documents utilisés dans des évaluations QR en français. QAst 2008 et 2009 contiennent des transcriptions d'émissions de radio. QA@Clef contient les années 1994 et 1995 du journal *Le Monde* et de l'*Agence Télégraphique Suisse*. Quaero est une collection de pages du Web.

[1] http ://www.exalead.fr

Si l'on se réfère aux tableaux 6.1 et 6.2, on peut observer un certain nombre de différences. Tout d'abord, le type des documents est très variable. Les documents provenant de sources journalistiques sont les plus courants. Dans ce type de documents, la qualité de la langue est plutôt bonne, et de plus, seulement une thématique est généralement abordée dans un document. Par ailleurs, les mêmes événements étant abordés sur plusieurs jours, on note une certaine redondance dans l'information. Ces documents ont cependant plusieurs défauts. D'abord, l'information étant limitée à l'actualité, il est difficile de couvrir des questions demandant un savoir encyclopédique, comme "*Où est né Jacques Chirac ?*". Par ailleurs, ces documents ont un coût commercial assez élevé. Les documents proviennent de peu de sources, en plus d'être généralement assez anciens, ce qui peut poser problème pour varier les questions créées à partir d'un corpus d'une année à l'autre. Pour palier à ce problème, les organisateurs de QA@Clef ont rajouté l'encyclopédie Wikipedia au corpus de documents. Cela permet donc d'avoir une quantité de documents et d'informations très importantes, tout en gardant un bon niveau de langue (même si le type de rédaction est différent de celui utilisé dans des documents journalistiques). Cependant, la structure très particulière de Wikipedia peut rendre problématique les traitements à effectuer pour rechercher les réponses. Ainsi, les méthodes classiques utilisées sur les documents journalistiques doivent être adaptées pour fonctionner sur Wikipedia. Par ailleurs, la grande présence d'anaphore ainsi que le peu de redondance de l'information sont aussi des problèmes à traiter.

Il arrive aussi que les évaluations soient effectuées sur des documents tirés de domaines spécifiques. Par exemple, l'évalaution EQueR s'est faite sur des documents politiques ainsi ques des articles scientifiques du domaine médicale. L'intérêt de ces tâches est qu'elles permettent d'avoir des possibilités d'application en pratique très intéressante. Par contre, le fait de devoir traiter une langue de spécialité oblige à avoir recours à des spécialistes du domaine pour créer les questions de l'évaluation.

QAst a été organisée pour permettre aux participants de l'évaluation de tester leur système sur des documents transcrits de la parole. Les transcriptions sont manuelles et automatiques, et leur origine est diverse. En 2008, les documents provenaient de séminaires (un locuteur principal avec un discours semi-préparé) et de réunions de travail (plusieurs intervenants se coupant très souvent la parole) en anglais, de transcriptions d'émissions de radio en français, et des transcriptions de débats du parlement européen (plusieurs locuteurs mais avec un discours semi-préparé) en anglais et en espagnol. Les niveaux de langue et la structure du discours varient beaucoup d'un type de document à l'autre. En 2009, seules ont été gardées les tâches utilisant les transcriptions d'émissions de radio pour le français et les débats du parlement européen pour l'anglais et l'espagnol. Cependant, le fait de travailler sur la parole implique aussi d'avoir un très faible nombre de documents (voir tableau). Ceci est dû au coût engendré par la transcription manuelle des documents ainsi que de la difficulté de disposer de sorties de transcripteurs automatiques.

La toute dernière source de documents employée est le web. S'il arrive que des parties plus spécifiques du web soient employées (Wikipedia, blogs ...), les dernières évaluations ont vu l'utilisation de corpus assez conséquent construits à partir de moteurs de recherches (Google par exemple). Différentes méthodes ont pu être employées pour construire ces corpus. On peut notamment citer l'utilisation de requêtes par mots-clefs entrées par des utilisateurs pour la construction du corpus Quaero. Contraire-

ment aux blogs ou à Wikipedia, les corpus construits de cette manière ne sont pas du tout uniformes aussi bien au niveau de la structure que du contenu. De ce fait, un certain nombre de problématiques se posent pour des systèmes habitués à traiter des documents journalistiques. Ainsi la qualité de la langue est très variable d'une page à une autre : une page peut contenir un article rédigé comme un document journalistique, alors qu'une autre aura un niveau syntaxique pauvre. De même, la structuration d'une page peut rendre l'extraction de l'information assez compliquée, et il peut y avoir une perte de paragraphes importante lors du filtrage des documents. De même, certaines pages publicitaires ou pornographiques peuvent contenir des listes de mots dans l'unique but d'attirer les moteurs de recherches. Malheureusement, cela a un impact sur les systèmes de questions-réponses, particulièrement lors de la phase de sélection de documents : les systèmes retournent ces documents parmi ceux pouvant potentiellement la bonne réponse, ajoutant un bruit conséquent pour la recherche de la bonne réponse. Enfin, la redondance de l'information est très faible dans ces types de corpus : la taille élevée du corpus est paradoxalement trop faible par rapport au nombre de thèmes abordés utilisés pour construire le corpus.

6.4 Métriques utilisées

Différentes métriques sont utilisées dans l'évaluation des systèmes de questions-réponses. Etant donné que nous nous intéressons principalement aux questions factuelles dans ce document, nous décrivons seulement les mesures communément employées pour évaluer les réponses à ce type de questions.

Une des métriques principales est la *précision*, ou *accuracy* en anglais. La précision (ou *top-1*) correspond au ratio entre nombre de réponses correctes retournées par le système et le nombre total de questions. Il arrive fréquemment que les systèmes retournent plus d'une réponse. Dans ce cas, seul la première réponse de chaque question est évaluée. La formule est donc la suivante ($\#RC_i$ correspond au nombre de réponses correctes retournées au rang *i* :

$$précision = \frac{\#RC_1}{\#questions}$$

Cette mesure permet d'évaluer la probabilité pour un système de questions-réponses de retourner la bonne réponse. Cependant, le problème d'une telle mesure est qu'elle ne permet pas de déterminer le ratio de questions où le système a réussi à trouver la bonne réponse, sans pour autant la retourner en première position. Il est intéressant de connaître la densité des bonnes réponses parmi l'ensemble des réponses retournées à chaque réponse. Cela permet notamment d'évaluer le progrès que peut effectuer un système lors de l'étape de scoring des réponses candidates à une question. La mesure correspondante se nomme *top-n*, que l'on peut aussi comparer au *rappel* et qui se calcule de la même manière que la précision mais en acceptant les réponses de rang 1 à *n* :

$$top - n = \frac{\#RC_{1,n}}{\#questions}$$

Cependant, un système de questions-réponses, s'il n'arrive pas à trouver la bonne réponse à une question, essaye néanmoins de la remonter dans les premiers rangs. Une autre mesure est donc nécessaire pour évaluer la qualité du ranking général des bonnes réponses. Le *Mean Reciprocal Rank* (*MRR* ou *Rang Réciproque Moyen*) permet d'évaluer la qualité du ranking des réponses. Le calcul est le suivant : la réponse correcte la mieux classée à une question est pondérée par rapport à l'inverse de son rang. Ainsi, si la bonne réponse est au deuxième rang, on ajoute 1/2, en troisième rang 1/3, etc ... S'il n'y a aucune réponse correcte, le résultat est nul. La valeur finale correspond à la moyenne des résultats obtenus pour chaque question. La formule est la suivante :

$$MRR = \frac{\sum \frac{1}{RC_{1,n}}}{\#questions}$$

Ces trois mesures ont été utilisées pour évaluer notre contribution.

Chapitre 7

Evaluation

7.1 Présentation

Dans ce chapitre, nous procédons tout d'abord (section 7.2) à une évaluation du segmenteur, à partir des corpus de test annotés manuellement et présentés dans le chapitre 4. La section suivante (section 7.3) est consacrée à l'évaluation du réordonnanceur à partir des données de certaines des campagnes d'évaluation des systèmes de questions-réponses présentées dans le chapitre précédent. Il est à noter que les analyses des résultats seront plus développés dans le chapitre suivant, avec des expérimentations et des mesures supplémentaires.

7.2 Evaluation du segmenteur

Dans la section 4, nous avons présenté un segmenteur de questions et de phrases de documents en constituants typés. Cette segmentation s'appuie sur notre propre formalisme et a pour objectif d'être robuste à tous types de documents. Cette segmentation est composée de deux segments principaux, le segment nominal (SN) et le segment verbal (SV). Ce formalisme initial est complété par 4 sous-types du segment nominal : segments de temps et de lieu (ST et SL), segment marqueur interrogatif (SMI), et segment optionnel (SO).

L'évaluation du segmenteur a été effectuée sur les corpus de test présentés dans le chapitre 4. Pour rappel, il y a 5 corpus de test : 3 pour les documents, tirés de corpus journalistique, oraux et web, et 2 pour les questions, écrites et orales. Nous redonnons les caractéristiques de ces corpus dans les tableaux 7.1 et 7.2.

Nous avons utilisé trois métriques pour cette évaluation : la précision et le rappel, dont les définitions

Type du corpus	#mots	#phrases	#segments	#moyenne de mots par phrase
Corpus d'apprentissage	184090	6023	48652	8
Corpus de test journalistique	37020	1172	9813	7
Corpus de test oral	35612	3057	1062	11
Corpus de test web	30133	980	8387	7

TAB. 7.1 – Caractéristiques des corpus d'apprentissage et de test pour les documents

Type du corpus	#mots	#questions	#segments	#moy. de mots question
Corpus d'apprentissage	4159	501	1728	8
Corpus de test de questions écrites	1359	149	542	9
Corpus de test d'énoncés oraux	1724	150	722	11

TAB. 7.2 – Caractéristiques des corpus d'apprentissage et de test pour les questions

sont légérement différentes de celles utilisées dans les évaluations de systèmes de questions-réponses, et la f-mesure.

Pour la tâche de segmentation, la précision correspond au ratio du nombre de segments correctement étiquettés par rapport au nombre de segments annotés.

$$précision = \frac{\#segmentscorrects}{\#segmentsannotés}$$

Le rappel quant à lui correspond au ratio du nombre de segments à identifier par rapport au nombre de segments trouvés par le segmenteur.

$$rappel = \frac{\#segmentstotal}{segmentsidentifiés}$$

La f-mesure est un score prenant en compte la précision et le rappel.

$$f_\beta = \frac{(1+\beta^2)*(précision*rappel)}{(\beta^2*précision+rappel)}$$

On utilise un β de 1 pour la f-mesure : le rappel et la précision sont ainsi pondérés de manière égale.

Nous donnons les mesures globales pour chaque corpus (tableau 7.3). Les résultats obtenus sur les documents sont encourageants (tableau 7.3). Il est intéressant de noter que les résultats globaux obtenus sur les corpus de test oral et du web ne baissent que respectivement de 3% et 1% par rapport aux résultats sur le corpus de textes journalistiques.

Type du corpus	précision	rappel	f-mesure
Documents journalistiques	83.2%	82.9%	0.83
Documents oraux	80.1%	79.8%	0.80
Documents du web	82.4%	81.7%	0.82
Questions écrites	82.6%	81.4%	0.82
énoncés oraux	59.4%	58.5%	0.59

TAB. 7.3 – Résultats globaux obtenus sur les cinq corpus de test

La chute était prévisible car le classifieur est entrainé sur un corpus d'apprentissage tiré de documents journalistiques, et la syntaxe de l'oral est différente de celle de l'écrit. Par exemple, on note beaucoup de répétitions dans l'oral, mais aussi des phrases bien plus longues, étant donné que la ponctuation est ajoutée lors des transcriptions. Le problème des pages internet est différent. Malgré un filtrage, on n'est jamais sûr de ce que l'on traite : tableaux, titres, liens etc ... De plus les textes ont eux aussi une syntaxe différente de l'écrit telle que rencontrée dans les textes journalistiques.

D'autres campagnes d'évaluation de segmentation ont existé. Nous nous sommes notamment intéressés à la tâche Chunking de CoNLL [Tjong et al. 2000] et à la campagne d'évaluation EASY [Paroubek et al. 2008b]. Les systèmes de la tâche Chunking étaient évalués sur l'anglais selon un formalisme de segmentation composé de 11 types. Le corpus d'apprentissage est composé de 211727 mots et le corpus de test de 47377 mots. Les documents sont tirés du corpus Wall Street Journal [Bies, et al. 1995]. Il n'y a pas de corpus de questions. EASY avait pour objectif d'évaluer les performances d'analyseurs syntaxiques du français sur deux composants : la segmentation en constituants typés, et les dépendances syntaxiques entre ces constituants. Nous ne nous intéressons qu'à la segmentation effectuée, composée de 6 types. De plus, les corpus à annoter proviennent de plusieurs sources : textes journalistique, textes littéraires, débats parlementaires, et transcriptions orales d'émissions de radio. Par ailleurs, les analyseurs devaient aussi traiter des sous-corpus de questions. Concrètement, le corpus est composé de 770 000 mots.

Si l'on compare les résultats du segmenteur avec ceux obtenus à la tâche Chunking, nos scores sont éloignés de la f-mesure moyenne, qui est d'environ 91%. Néanmoins, les résultats obtenus sur cette tâche ne sont pas totalement comparables à ceux obtenus par notre segmenteur. En effet, la tâche Chunking portait sur de l'anglais, et le formalisme utilisé est assez éloigné de celui utilisé par notre segmenteur. Si on compare avec les meilleurs résultats obtenus sur EASY (f-mesure de 0.92), on observe là aussi une baisse des résulats. Néanmoins, si EASY évalue les systèmes sur le français, le formalisme de segmentation utilisé n'est pas comparable. Par ailleurs, il faut aussi signaler que les résultats obtenus sur les transcriptions orales (f-mesure de 0.80) sont meilleurs que ceux obtenus par EASY (f-mesure de 0.79).

Les résultats obtenus pour la segmentation des questions de campagnes d'évaluation sont encourageants. Les résultats obtenus sur le corpus d'énoncés oraux ne permettent par contre pas d'envisager immédiatement l'utilisation de cette segmentation sur ce type de questions. La raison principale de

la chute observée sur ces questions est que le corpus d'apprentissage utilisée ne contenait principalement que des questions avec une structure classique comme on peut en trouver dans des campagnes d'évaluation. Contrairement aux documents, la différence entre les deux types de questions est bien plus prononcée. Outre les hésitations, un utilisateur a tendance à former ses questions à l'orale en utilisant des "règles de politesse". Ainsi, les questions sont souvent précédées par "je voudrais savoir ..." ou "pourriez vous me dire ..." . La structure est plus généralement très différente.

	Documents journalistiques				Questions écrites			
Type du segment	#Seg	P(%)	R(%)	f-m	#Seg	P(%)	R(%)	f-m
SN	6209	79.8	80.3	0.80	215	73.6	74.6	74.1
SV	2256	91.9	90.5	0.91	154	94.1	93.5	0.93
ST	183	87.4	72.1	0.79	12	100	50	0.66
SL	138	92.5	67.0	0.77	15	91.7	73.3	0.81
SO	56	38.3	61.6	0.47	0	-	-	-
SMI	0	-	-	-	148	0.94	95.8	0.95

TAB. 7.4 – Résultats obtenus pour chaque type de segments sur le corpus de test des documents journalistiques et sur les questions écrites. #Seg : nombre de segments ; R : rappel ; P : précision ; f-m : f-mesure

Nous avons aussi évalué les mesures détaillées pour chaque type de segment sur le corpus de test de documents journalistiques et le corpus de questions écrites (tableau 7.4). Pour les documents, les segments les plus importants, à savoir SV, ST et SL sont bien annotés. Pour les questions écrites, les résultats obtenus sont eux aussi encourageants, et ce quelque soit le type du segment, à l'exception des segments nominaux et des segments de temps, dont le score moyen est probablement du à la taille du corpus d'apprentissage.

Type du segment	#erreurs	SN	SV	ST	SL	SO	SMI
SN (d)	1270	85.45	9.8	1.22	1.94	0.93	-
SV (d)	298	38.93	60.06	0.03	0.03	-	-
ST (d)	15	13.33	6.66	79.99	-	-	-
SL (d)	7	71.42	-	-	28.57	-	-
SN (q)	50	82.83	7.46	2.23	2.98	-	3.73
SV (q)	14	50.00	42.85	-	-	-	7.14
ST (q)	-	-	-	-	-	-	-
SL (q)	1	100	-	-	-	-	-
SMI (q)	14	49.99	7.14	-	-	-	42.85

TAB. 7.5 – Matrice de confusion. d : documents journalistiques ; q : questions écrites

Enfin, nous évaluons aussi les cas où certains types de segments critiques pour la gestion des relations ont été mis de manière erronée (tableau 7.5). On évalue dans ce cas les segments SN, SV, ST, SL et aussi SMI, mais seulement sur le corpus de questions pour ce dernier. Nous évaluons ces types en

particulier car nous estimons qu'ils ont une importance dans le fonctionnement du réordonnanceur. Ainsi, le segment verbal a un rôle de pivot dans les phrases que l'on retrouve dans les traitements présentés dans le chapitre 5. Là encore, nous donnons les résultats seulement pour le corpus de test de documents journalistiques et le corpus de questions écrites. On peut noter qu'il arrive que le segment verbal soit mis de manière erronée à la place des segments nominaux, ce qui peut être un problème vis à vis du rôle de pivot de ce type de segment (tableau 7.5). Par contre, il est assez rare qu'un segment nominal soit pris pour un segment verbal : la majorité des erreurs pour le segment nominal proviennent de problèmes de frontières.

Nous avons aussi généré des modèles différents en fonction des traits utilisés en entrée pour l'apprentissage, de manière à évaluer leur importance. On obtient sur le corpus des documents journalistiques une baisse de 0.5% en utilisant juste l'analyseur en parties du discours et une baisse de 1% en utilisant uniquement l'analyseur de Ritel. On peut donc remarquer que si les meilleurs résultats sont obtenus avec les types de Ritel et les parties du discours, utiliser seulement l'un des deux traits ne change pas drastiquement les résultats.

Les résultats obtenus par ce segmenteur sont discutés plus en détail dans la section 7.4.

7.3 Evaluation du réordonnanceur

Dans cette section, nous allons procéder à l'évaluation du réordonnanceur. L'obtention des résultats se déroulent de la manière suivante : les questions sont traitées par le système de questions-réponses de Ritel, et 10 candidats réponses sont retournées et ordonnées selon le score fixé par Ritel. Pour chacune de ces réponses, un ensemble de passages où la réponse a été identifiée est fournie. Le réordonnanceur va pour chaque réponse traiter l'ensemble des passages, et calculer un coût de transformation entre chaque passage et la question. Le passage avec le coût le plus faible est sélectionné et le score de la réponse hypothèse considérée correspond au score de similarité du passage. Ce traitement est fait pour chaque réponse, et l'ensemble des réponses est réordonné dans l'ordre croissant selon les scores de similarité. Pour chaque campagne d'évaluation, les résultats obtenus par le réordonnanceur sont comparés avec ceux obtenus par le système Ritel. Les résultats de Ritel correspondent à ceux obtenus avec la dernière version du système, et non ceux obtenus lors de la participation du systèmes aux différentes évaluations. Les résultats obtenus lors de la participation à la campagne d'évaluation sont présentés dans le chapitre 2 (section 2.3.3).

Les trois corpus évalués ont chacun des caractéristiques différentes, ce qui permet d'avoir une idée de la robustesse du réordonnanceur : textes journalistiques pour QA@CLEF, transcriptions de données orales pour QAst, et documents web pour Quaero. Nous présentons les résultats obtenus, et nous en faisons une première analyse dans la section 7.4. Les analyses des résultats sont plus développés dans le chapitre 8.

7.3.1 La campagne d'évaluation QA@CLEF

A partir de 2003, la campagne européenne CLEF a intégré une tâche de questions-réponses [Magnini et al. 2003 ; Magnini et al. 2004 ; Vallin et al. 2005 ; Magnini et al. 2006 ; Giampiccolo et al. 2007 ; Forner et al. 2008]. Auparavant, les systèmes de questions-réponses avaient été principalement évalués par les campagnes d'évaluation de TREC [Voorhees & Tice 1999]. Si la tâche a évolué au cours des différentes itérations, en ajoutant par exemple de nouveaux types de questions, TREC ne s'est jamais intéressé au multilinguisme : les systèmes n'étaient évalués que sur l'anglais. La tâche de questions-réponses de CLEF a justement comme objectif d'évaluer les systèmes de questions-réponses sur différentes langues.

La toute première itération de la tâche a eu lieu en 2003, où 9 sous-tâches étaient proposées. Pour les sous-tâches en monolingue, trois langues ont été utilisées (Néerlandais, Italien et Espagnol), et cinq pour les sous-tâches bilingues (Néerlandais, Italien, Espagnol, Allemand et Français), où les participants devaient chercher les réponses aux questions dans des documents en anglais. A partir de 2004, plusieurs langues ont été ajoutées, dont le français.

Les systèmes étaient évalués sur des corpus de 200 questions. En 2004, 3 types de questions étaient proposés : les questions *factuelles*, les questions *définitions*, et les questions *comment*. En 2005, les questions comment ont été remplacées par les questions temporelles restreintes. Nous donnons des exemples de ces différents types de questions ci-dessous.

– Factuelle : En quelle année Thomas Mann a-t-il obtenu le prix Nobel ?
– Définition : Qui est Goodwill Zwelithini ?
– Comment : Comment se transmet le virus Ebola ?
– Temporelle restreinte : Qui fut à la tête de la Commission européenne de 1985 à 1995 ?

Le corpus de documents pour le français provient de trois sources : le journal Le Monde de 1994, et les dépêches de l'ATS de 1994 et 1995. Nous donnons dans le tableau 7.6 les caractéristiques de ce corpus de documents.

	Le Monde 94	ATS 94	ATS 95
Nombre de documents	44K	43K	42K
Taille (Méga-octets)	158	86	88
Taille moyenne des documents (octets)	1994	1683	1715

TAB. 7.6 – Caractéristiques de la collection de documents pour la campagne d'évaluation QA@CLEF.

Le réordonnanceur a été évalué sur les itérations 2004 et 2005 de la tâche questions-réponses de CLEF. Les mesures utilisées sont la précision, le MRR, et le top-n, avec n fixé à 10. Les résultats obtenus par Ritel et le réordonnanceur sont présentés dans le tableau 7.7. Le δ indique la différence sur le top-1 entre les résultats obtenus par Ritel par rapport au réordonnanceur.

		Ritel			Réordonnanceur			
Corpus	#q.	Préc.	MRR	top-n	Préc.	MRR	top-n	Δ
2004	200	50.8	0.563	68.4	46.8	0.523	68.4	-4
2005	200	40.0	0.448	55.0	38.5	0.426	55.0	-1.5

TAB. 7.7 – Résultats obtenus par Ritel et le réordonnanceur sur les corpus de test des évaluations 2004 et 2005 de QA@CLEF.

Les résultats obtenus sur QA@CLEF sont assez mitigés. On observe une baisse pour les deux corpus assez significative, qui laisse supposer que notre approche n'est pas très adaptée à du texte écrit avec une syntaxe traditionnelle. Nous discutons plus en détails ces résultats dans la section 7.4.

7.3.2 La campagne d'évaluation QAst

QAst, qui signifie *Question Answering on Speech Transcriptions*, a été créée en 2007 [Turmo et al. 2007] au sein de CLEF et s'intéresse à la problématique des systèmes de questions-réponses dans un cadre oral. En plus de cette particularité, l'évaluation est multilingue, et les systèmes sont évalués sur jusqu'à trois langues : le français, l'anglais, et l'espagnol. Enfin, les documents proviennent de différentes sources, ce qui implique des caractéristiques différentes. Quatre types de documents ont été étudiés, avec au moins deux types de transcriptions, manuelles ou automatiques. Nous n'évaluons le réordonnanceur que sur la sortie manuelle.

Le premier type traité correspond à des transcriptions de séminaires. Une personne parle seule avec un discours semi-préparé pendant que des personnes peuvent intervenir, même si les interruptions restent rares. Ces documents proviennent du corpus *CHIL* [CHIL 2007] et sont constitués de 25 séminaires. Ces séminaires sont en anglais, et ne seront donc pas traités par le réordonnanceur. Par ailleurs, ce type de documents n'a pas été utilisé dans l'édition 2009 de QAst.

Le second type traité correspond à des transcriptions de réunion de travail. Contrairement aux séminaires, plusieurs personnes parlent en même temps, en se coupant la parole. Le corpus *AMI* [AMI 2005] est utilisé et contient 168 réunions. Les réunions sont là aussi en anglais, et ce corpus n'a pas été utilisé dans l'édition 2009 de QAst.

Les documents de type *séminaire* et *réunion* étaient les seuls disponibles pour l'édition 2007. Ces documents étant en anglais uniquement, nous n'avons pas appliqué pas le réordonnanceur sur l'édition 2007 de QAst. A partir de 2008, deux nouveaux types de documents ont été rajoutés. Le premier correspond à des transcriptions d'émissions journalistiques de radio, en français. *ESTER* [Galliano, et al. 2006] est le corpus utilisé et contient 10 heures d'émissions, enregistrées de plusieurs sources : France Inter, Radio France International, Radio Classique, France Culture, Radio Télévision du Maroc. Il existe trois transcriptions automatiques pour ce corpus.

Le dernier genre utilisé à partir de 2008 correspond à des transcriptions de sessions du Parlement Européen en anglais et en espagnol. Chaque parlementaire prend la parole à tour de rôle avec un discours préparé. Le président de la session s'assure du déroulement de la scéance en donnant la parole aux parlementaires. Le corpus *TC-STAR* [TC-Star 2004-2008] est utilisé et contient 3 heures pour chaque langue. Là encore, il existe trois transcriptions automatiques différentes.

Au cours des trois éditions de QAst, quatres types de documents différents ainsi que trois langues différentes ont été traités. L'ensemble de ces sous-tâches n'était pas toujours présente. Nous résumons ces sous-tâches ci-dessous :

– Séminaires : discours semi-préparé en anglais, une transcription automatique (2007-2008)
– Réunions : discours coupé en anglais, une transcription automatique (2007-2008)
– Radio : extraits d'émissions journalistiques en français, trois transcriptions automatiques (2008-2009)
– Parlement : discours préparé en anglais et espagnol, trois transcriptions automatiques (2008-2009)

Le réordonnanceur n'a été appliqué que sur les données en français issues de transcriptions manuelles pour 2008. Pour 2009, la procédure de création des questions étant différente (voir chapitre 6, section 6.2.2), le réordonnanceur a été appliqué sur deux corpus en français : les questions écrites et les questions orales transcrites manuellement. Nous donnons dans le tableau 7.8 les caractéristiques du corpus pour le français. Ce corpus a été divisé en deux sous-corpus : un pour le développement et un pour le test.

	Développement	Test
Nombre de documents	6	12
Nombre de mots	35M	87M
Durée de la parole	2h15	5h40

TAB. 7.8 – Taille de la collection de documents pour l'évaluation QAst.

Il existe quelques différences entre les éditions 2008 et 2009 de QAst. Tout d'abord, les questions traitées ne sont que de type factuelle et définition simple. Les questions factuelles sont réparties sur sept types différents, présentés dans le tableau 7.9 : *personne, lieu, organisation, langue, méthode/algorithme, mesure, date/heure, couleur, forme, matériau et définition*. Environ 10% des questions étaient de type NIL (pas de réponses dans les documents) en 2008, contre environ 20% en 2009.

L'édition 2009 a par ailleurs deux particularités par rapport à l'édition 2008. Tout d'abord, le processus de création des questions a été modifié. En 2008, les questions étaient créées par un évaluateur à partir d'extraits du corpus de documents. En 2009, l'idée était d'avoir des questions plus spontanées. On fournissait à des utilisateurs des extraits de documents, et les utilisateurs devaient poser leur question à l'oral en utilisant des informations contenues dans le passage mais dont la réponse n'était pas présentée. Cette particularité en implique une deuxième : il existe pour chaque tâche de 2009 deux corpus de questions, l'un correspondant aux questions des utilisateurs transcrites manuellement,

Type	Exemple
personne	Qui est le nouvel entraîneur du Raja ?
lieu	Dans quelle ville un attentat a-t-il eu lieu ?
organisation	Quelle entreprise veut-elle réduire ses effectifs ?
langue	En quelle langue la chaine Al-Jazira est-elle diffusée ?
mesure	A combien s'élève l'amende de l'incendiaire de Daewoo ?
date/heure	Quand a été assassiné Hans Krasa ?
couleur	Quelle est la couleur des cagoules des combattants ?
définition	Qui est Claudine Sada ?

TAB. 7.9 – Classification des questions utilisées dans QAst.

et l'autre avec une réécriture de ces questions sous une forme "ecrite standard". Par ailleurs, les 50 questions de développement de 2008 ayant été considérée insuffisante, nous avons décidé de demander à des locuteurs natifs du français (de même pour les autres langues) de proposer des reformulation de ces questions. Six corpus de questions sont ainsi évalués :

– dev08+reform, corpus de développement de 2008 ajouté à un corpus de reformulation de ces questions
– test08, corpus de test de 2008
– dev09, corpus de développement de 2009 des questions orales réécrites
– odev09, corpus de développement de 2009 avec questions orales transcrites manuellement
– test09, corpus de test de 2009 des questions orales réécrites
– otest09, corpus de test de 2009 avec questions orales transcrites manuellement

Il y a chaque fois 100 questions pour les corpus de test, et 50 questions pour les corpus de développement, excepté dev08+reform qui contient les 50 questions initiales de 2008 plus 319 questions reformulées.

Les résultats obtenus par le réordonnanceur sur les éditions 2008 et 2009 sont contenus dans le tableau 7.10 avec les résultats obtenus par Ritel. Les mesures utilisées sont la précision, le MRR, et le top-n, avec n fixé à 10. Sur la tâche française de QAst, aussi bien en 2008 qu'en 2009, Ritel était le seul système à participer. Le δ indique la différence entre les résultats obtenus par Ritel par rapport au réordonnanceur.

Comme pour les résultats obtenus sur la campagne QA@CLEF, les résultats obtenus sur QAst sont assez mitigés. On peut observer une baisse des résultats sur l'ensemble des corpus. Il faut néanmoins indiquer que ces baisses sont relativement légères, particulièrement sur les corpus de développement et de test de 2009. Ainsi, les corpus de développement ne contenant que 50 questions, une baisse de -4 ne correspond qu'à une seule question non trouvée par rapport à Ritel. Ainsi, si les résultats ne sont pas positifs, on peut néanmoins supposer que l'approche est plus adapté aux transcriptions orales qu'à des documents journalistiques.

		Ritel			Réordonnanceur			
Corpus	#q.	Préc.	MRR	top-n	Préc.	MRR	top-n	Δ
dev08+reform	369	64.2	0.667	72.6	59.5	0.637	72.6	-4.7
test08	100	50.0	0.534	62.0	45.0	0.488	62.0	-5
dev09	50	34.0	0.382	61.5	30.0	0.343	61.5	-4
odev09	50	32.0	0.365	61.0	30.0	0.325	61.0	-2
test09	100	28.0	0.390	60.0	26.0	0.340	60.0	-2
otest09	100	28.0	0.390	59.0	26.0	0.323	59.0	-2

TAB. 7.10 – Résultats obtenus par Ritel et le réordonnanceur sur les corpus de développement et de test de QAst 2008 et 2009.

7.3.3 La campagne d'évaluation Quaero

Le projet européen Quaero [http ://www.quaero.org/ 2008] inclut une tâche consacrée aux systèmes de questions-réponses [Quintard et al. 2010]. Il y a eu pour le moment trois évaluations avec des corpus de questions différents. La première évaluation avait pour objectif de servir de *baseline*. Cela permet ainsi d'évaluer les progrès effectués par les participants d'années en années. Par la suite, deux autres évaluations ont été effectuées, en gardant les mêmes caractéristiques à chaque fois de manière à évaluer le progrès effectué avec le biais le plus faible (les questions changeant d'une année à une autre). Cette tâche est effectuée sur le français et l'anglais. Comme précédemment, le réordonnanceur n'a été appliqué que sur le français.

Le corpus de documents a été construit par Exalead, une société française participant au projet Quaero, et possédant un moteur de recherche [1]. Le corpus est constitué de documents du Web. Des requêtes utilisateurs entrées dans le moteur de recherche ont été collectées. Les 1000 premiers documents pour chaque requête retournés par le moteur de recherche ont été utilisés pour constituer le corpus. Un seul corpus de deux millions de documents est constitué, mais seul une sous-partie, de 500000 documents, a été utilisée pour les campagnes d'évaluation 2008, 2009 et 2010. Il a été choisi de ne travailler sur les trois premières évaluations que sur le corpus de cinq cent mille documents. Les caractéristiques du corpus de 500 000 documents en français sont contenues dans le tableau 7.11.

Les questions ont été définies de manière à se rapprocher de ce que l'être humain pose en général comme type de questions. Cependant, l'idée était aussi d'avoir des questions évaluables pour la tâche. Une grande part des questions posées par l'humain sont de type *pourquoi*, comment et définition (34% [Kato et al. 2006]). Par ailleurs, [Toney, et al. 2008] a aussi montré qu'environ 10% des questions posées étaient de type oui/non. En se basant sur ses observations, il a été choisi de travailler sur 6 types de question : factuelles, définitions, oui/non, pourquoi, comment et listes fermées. Nous donnons ci-dessous un exemple pour chacune de ces types de question.

[1] http ://www.exalead.fr

	Quaero
Type	Web
Langue	Français
Nombre de documents	500K
Nombre de phrases	82M
Nombre de mots	840M
Nombre de caractères	4,2G
Phrases/document	170
Mots/phrase	10
Caractères/mot	5,3

TAB. 7.11 – Types et tailles de la collection de documents pour les différentes itérations de l'évaluation Quaero.

– Factuelle : Quand a eu lieu le règne d'Elizabeth ?
– Définition : Qu'est ce que l'anaphase ?
– Oui/Non : Est-ce que Chopin était français ?
– Pourquoi : Pourquoi le Concorde ne vole-t-il plus ?
– Comment : Comment est morte Cléopatre ?
– Liste fermée : Qui sont les 4 frères Dalton ?

Une *adjudication* a été effectuée après chaque évaluation pour rectifier les résultats lors de désaccords entre les évaluateurs et les participants sur la validité ou non d'une réponse à une question. Enfin, les systèmes doivent fournir une justification à la réponse. L'idée est d'avoir un extrait de texte d'au maximum 250 caractères permettant de convraincre un utilisateur humain de la validité de la réponse.

Le processus de création des questions a été différent entre 2008 et 2009 [Quintard et al. 2010]. En 2008, en se basant sur les requêtes collectées, les évaluateurs sélectionnaient un des documents retournés par une requête utilisateur, et devaient écrire une question par rapport au contenu du document. Par exemple pour la requête *"wood manufacturing"*, le passage suivant est trouvé : *"Drew Graham, a wood manufacturing technician from Nova Scotia required a website to showcase his work for potential employers"*. La question *Who is Drew Graham ?* est construite à partir de ce passage, avec comme réponse *wood manufacturing technician*. En 2009, les questions ont été créées sans regarder les documents. Les requêtes ont été utilisées pour générer les questions lorsque cela était possible. En tout, il y a eu 167 questions factuelles créées en 2008, et 295 en 2009 (pour un total respectivement de 250 et 507). A partir de 2010, deux sous corpus de questions factuelles ont été ajoutés pour le français. L'objectif était de créer des questions plus spontanées, comme dans QAst 2009. Ainsi, on demandait aux utilisateurs de poser des questions à l'oral sans regarder les documents. Les questions des utilisateurs sont ensuite transcrites et nettoyées : les hésitations sont enlevées ainsi que les répétitions. Un corpus de 88 questions factuelle a ainsi été créé. A partir de ces questions, un deuxième sous corpus est conçu en réécrivant manuellement les questions orales. Ce deuxième sous corpus est composé de 56 questions. Enfin, un corpus principal a été construit en utilisant la même méthodologie

qu'en 2009. Il est composé de 175 questions factuelles.

Le réordonnanceur n'a été appliqué comme d'habitude que sur les questions factuelles, les questions restantes étant traitées par Ritel. Pour 2008 et 2009, outre le système Ritel, deux autres systèmes ont participé : FIDJI [Tannier & Moriceau 2010] et QRISTAL [Laurent et al. 2006] de la société française Synapse. A partir de 2010, le système QAVAL [Grappy & Grau 2010] du groupe LIMSI-ILES a participé. Le tableau ci-dessous présente les résultats obtenus par le système Ritel et comparé à ceux obtenus avec le réordonnanceur. Les mesures utilisées sont la précision, le MRR, et le top-n, avec n fixé à 10. Le $delta$ indique la différence entre les résultats obtenus par Ritel et ceux obtenus par le réordonnanceur. Les résultats obtenus par Ritel et le réordonnanceur sont présentés dans le tableau 7.12.

		Ritel			Réordonnanceur			
Corpus	#q.	Préc.	MRR	top-n	Préc.	MRR	top-n	Δ
2008	167	32.6	0.385	46.2	33.6	0.394	46.2	**+1.1**
2009	295	32.4	0.378	46.8	27.5	0.295	46.8	-4.9
2010 normal	175	34.3	0.431	58.3	30.9	0.393	58.3	-3.4
2010 spontanée	88	34.6	0.416	55.1	27.2	0.314	55.1	-7.4
2010 réécrite	56	32.1	0.416	55.4	33.9	0.373	55.4	**+1.8**

TAB. 7.12 – Résultats obtenus par Ritel et le réordonnanceur sur les corpus de test des évaluations 2008, 2009 et 2010 de Quaero.

Les résultats permettent d'observer une hausse relativement importante des scores obtenus si on réordonne les réponses proposées par Ritel pour le corpus de 2008. Par contre, les résultats sont en déça en 2009. Pour 2010, on observe aussi une baisse sur les questions traditionnelles. Une chute assez importante est aussi observée sur les questions orales, alors que les résultats pour les questions réécrites sont en hausse. On peut supposer que la réécriture a donc un impact sur les performances du réordonnanceur.

7.4 Discussion sur les résultats

Les résultats obtenus par le segmenteur sont relativement bons. Si le tout reste inférieur aux meilleurs résultats obtenus par les analyseurs évalués sur EASY, on peut néanmoins noter que les résultats obtenus sont robustes par rapport aux différents types de documents, malgré l'apprentissage effectué sur un corpus journalistique. Sur EASY, les analyseurs enregistrent une baisse très importante sur l'oral (de 0.91 sur l'écrit à 0.79). Par ailleurs, notre corpus d'apprentissage souffre de certains défauts, ce qui laisse supposer que les performances de notre segmenteur peuvent être améliorées. Ainsi, il existe des inconsistances dans le corpus, conséquence d'un formalisme de segments pas encore totalement défini au début de l'annotation. De même, la taille du corpus d'apprentissage (180 000 mots) semble trop faible : un aggrandissement du corpus semble nécessaire. A titre de comparaison, le corpus uti-

lisé pour EASY est composé de 770000 mots. Néanmoins, nous estimons que les résultats obtenus sont encourageants, et que le segmenteur est suffisamment performant pour être utilisé par le module de réordonnancement. Pour prouver cette hypothèse, nous présentons une analyse dans le chapitre 8 permettant d'évaluer l'impact du segmenteur sur les résultats du module de réordonnancement.

Pour le module de réordonnancement, les résultats obtenus sont plus moyens. Ainsi, les performances observées sur les corpus 2004 et 2005 de QA@CLEF sont assez mitigées. L'application du réordonnanceur entraine une dégradation des résultats assez importante. Les documents du corpus de QA@CLEF proviennent de sources journalistiques, et ont donc une syntaxe traditionnelle, contrairement aux documents issus de l'oral. Notre approche a pour objectif d'être relativement robuste à tous types de documents. De ce fait, le modèle de représentation des questions et documents reste relativement simple. De même, les opérations de rattachement utilisées dans le réordonnanceur sont volontairement simples, pour être appliquées sur n'importe quel type de documents. Cette simplification peut être une explication pour les résultats mitigés obtenus.

Sur les campagnes QAst et Quaero, les résultats obtenus sont plus encourageants. Pour QAst, si on observe une baisse des résultats sur tous les corpus, la régression est néanmoins assez faible, surtout en 2009. Pour Quaero, on peut observer une hausse sur les questions de 2008 et une baisse sur les questions de 2009. Pour 2010, les résultats sont assez contrastés. Sur le corpus de questions traditionnelles, on observe là encore une baisse des résultats. Il y a aussi une baisse assez importante sur les questions orales spontannées. Ce résultat est à corréler avec celui obtenu sur le corpus de questions réécrites, où une hausse est cette fois observé. Si cette hausse est assez faible, elle laisse néanmoins supposer que la forme des questions orales a eu un impact sur les performances du réordonnanceur, ce qui n'était pas le cas pour QAst 2009, où les résultats entre les deux types de questions étaient comparables. Par rapport à nos hypothèses sur les résultats obtenus sur QA@CLEF, on peut en déduire que notre approche est potentiellement plus efficace sur des textes à la syntaxe différente des textes journalistiques. On peut par ailleurs supposer que si le réordonnanceur permet d'obtenir de meilleurs résultats sur certains corpus de questions, il est probablement meilleur sur certains types de questions, ou du moins sur certaines questions ayant des caractéristiques spécifiques (nombre d'éléments, longueur, type de réponse attendue ...). Ces hypothèses sont développées dans le chapitre 8.

Chapitre 8

Analyse critique des résultats

8.1 Présentation générale

Nous avons présenté dans le chapitre 7 l'évaluation de notre segmenteur ainsi que du module de réordonnancement. Les résultats obtenus par le segmenteur sur les trois corpus de test évalués (avec des textes respectivement tirés de sources journalistiques, d'émissions de radio, et du web) sont encourageants. On peut notamment observer que ce segmenteur est robuste à ces trois types de données. Par contre, si les résultats sont plutôt bons, on peut néanmoins observer une baisse par rapport aux meilleurs résultats obtenus sur les textes journalistiques lors de la campagne EASY [Paroubek et al. 2008b], où les meilleurs systèmes atteignent une f-mesure de 0.92 (par comparaison à 0.83 pour notre segmenteur). Notre hypothèse est, que si ce segmenteur peut encore être amélioré, ses performances sont suffisamment bonnes pour que le module de réordonnancement s'appuie sur les segments créés. Pour évaluer l'impact des erreurs du réordonnanceur, nous procédons à une analyse modulaire du segmenteur dans la section 8.2.1.

Les résultats du module de réordonnancement sont plus mitigés, notamment sur les corpus journalistiques de la campagne QA@CLEF. Pour QAst, si les résultats sont aussi relativement mitigés, la baisse est moins prononcée, surtout au cours de l'édition 2009. On peut par contre noter une hausse des résultats par rapport à ceux obtenus par Ritel sur certains corpus de questions de la campagne Quaero. On peut en déduire que notre approche a du potentiel, mais qu'il est nécessaire d'en améliorer certains aspects pour obtenir de meilleurs résultats. L'un de ces aspects concernent les opérations de rattachement. Notre approche est inspirée en partie des travaux sur la distance d'édition présentés par [Kouylekov & Negri 2010]. La distance d'édition est calculée traditionnellement à partir de trois types d'opérations : insertion, suppression et substitution. Nous avons décidé d'ajouter un quatrième type, les opérations de rattachement. Nous allons évaluer l'impact de ce type d'opération dans la section 8.2.2.

Le module de réordonnancement s'appuie sur différentes ressources linguistiques présentées dans la section 5.3. Ces ressources sont particulièrement importantes pour identifier les similarités entre les mots d'un passage et les mots d'une question (voir section 5.4.2.3). Nous utilisons notamment un dictionnaire de synonymes. Certains de ces synonymes peuvent selon le contexte d'une phrase avoir un sens éloigné du mot auquel ils sont associés. Ce phénomène peut conduire à traiter des similarités éronnées. Nous mesurons ainsi l'impact des synonymes dans la section 8.2.3.

Si les résultats montrent une hausse par rapport à ceux obtenus par Ritel sur un des corpus de questions de Quaero, on peut néanmoins en déduire que le module de réordonnancement n'est pas encore assez performant pour être appliqué sur tous les types de questions factuelles. L'analyse présentée dans la section 8.3 a pour objectif d'évaluer les perfomances du module selon certaines caractéristiques des questions. Nous évaluons deux types de caractéristiques : le sujet d'une question (temps, lieu, nombre, personne, etc ...), et le nombre d'éléments qu'elle contient.

Enfin, nous avons pu constater l'importance des différentes campagnes d'évaluation pour les systèmes de questions-réponses. Ces campagnes permettent d'observer les progrès effectués sur le domaine d'une année à une autre. Néanmoins, les comparaisons de résultats peuvent être légèrement biaisés. Si l'on compare par exemple les résultats obtenus entre QA@CLEF et QAst, la comparaison est difficile à faire du fait que l'origine des documents est différente (respectivement sources journalistes et transcriptions d'emissions de radio). De ce fait, on a tendance à quantifier l'évolution d'un système sur plusieurs itérations d'une même campagne : par exemple les années 2008, 2009 et 2010 de l'évaluation Quaero. Néanmoins, même en gardant le même contexte d'évaluation, il peut arriver que certaines caractéristiques ne soient pas prises en compte par les évaluateurs. Cela conduit à un biais lors de la comparaison de deux itérations d'une même campagne. Nous proposons une mesure simple dans la section 8.4 pour évaluer l'impact de l'évolution des caractéristiques des campagnes d'évaluation sur les systèmes de questions-réponses.

8.2 Analyse modulaire

L'objectif de cette section est de procéder à une analyse plus poussée de l'impact de nos différentes contributions. Dans la section 8.2.1, nous évaluons l'impact des erreurs faites par le segmenteur sur le module de réordonnancement. Puis dans la section 8.2.2, nous évaluons l'impact des opérations de rattachement sur les résultats obtenus par le réordonnanceur.

8.2.1 Impact du segmenteur

Le chapitre 7 présentait les résultats obtenus par le module de segmentation. Le meilleur résultat obtenu (f-mesure de 0.83 sur les documents journalistiques) implique qu'un certain nombre de segments sont mals annotés. Le module de réordonnancement s'appuyant sur ces segments, il est évident que

ces erreurs ont un impact sur les performances du module de réordonnancement. Notre objectif ici est de mesurer l'impact des erreurs du segmenteur sur le réordonnanceur.

Nous avons annoté manuellement un sous-corpus de questions et de passages selon le formalisme de segmentation présenté dans la section 4. Chaque passage est associé à un ensemble de questions tirées aléatoirement des différentes campagnes d'évaluation présentées dans 7. Nous avons sélectionné une trentaine de questions pour chaque campagne d'évaluation : 33 pour CLEF (journalistique), 34 pour QAst (oral), et 33 pour Quaero (web). Ces questions ont été tirées à chaque fois sur l'ensemble des itérations des campagnes à notre disposition (par exemple 2008 et 2009 pour QAst). De plus, nous n'avons sélectionné que des questions pour lesquelles le réordonnanceur ne retournait pas la bonne réponse en première position. Les questions ainsi sélectionnées sont annotées manuellement.

Pour chacune des questions, nous annotons deux passages : celui contenant la bonne réponse à une question, et celui retourné en première position habituellement par le réordonanceur. Ainsi, pour chaque question, nous évaluons le réordonnanceur sur deux passages. Lorsque cet ensemble de questions et de passages associés est traité par le segmenteur, le bon passage (et donc la bonne réponse) ne remonte pas en premier rang. L'objectif est d'observer si en annotant ces passages et questions manuellement, le réordonnanceur fait remonter certain bons passages en premier rang. Ces résultats sont présentés dans le tableau 8.1.

Corpus	#q.	#br.
QA@CLEF	33	0
QAst	34	3
Quaero	33	1

TAB. 8.1 – Nombre de bonnes réponses trouvées en annotant les passages et les questions manuellement ; le réordonnanceur ne trouve habituellement pas la bonne réponse pour ces questions ; q. signifie questions ; br. signifie bonnes réponses.

On peut observer que l'annotation manuelle des questions et des passages permet effectivement de trouver la bonne réponse sur un certain nombre de questions. Si on on observe aucun changement sur le sous-corpus QA@CLEF, les deux autres sous corpus enregistrent des modifications : 3 bonnes réponses sont trouvées pour QAst, et 1 pour Quaero. Ces différences entre chaque corpus sont peu surprenantes. Notre modèle est entrainé sur des articles journalistiques utilisés pour l'évaluation QA@CLEF. De ce fait, notre segmenteur est plus à même de traiter les textes journalistiques, même si notre formalisme de segmentation reste relativement robuste. Pour rappel, on obtient une f-mesure de 0.83 sur le corpus de test journalistique du segmenteur, contre 0.80 pour l'oral, et 0.82 pour le web. Les passages annotés pour QAst ont une syntaxe très différente de celle des passages annotés pour QA@CLEF. C'est aussi le cas pour Quaero, mais à une échelle moins grande : les documents issus du web sont de qualité diverse.

On peut émettre l'hypothèse que cette annotation manuelle a permis d'améliorer les résultats pour le sous-corpus QAst. La syntaxe des énoncés oraux étant assez éloignée des articles journalistiques, il est

possible que la correction des erreurs engendrées par le segmenteur permette d'améliorer les résultats. Pour le cas de la bonne réponse remontée pour Quaero, l'erreur est due à une erreur de typage. Les deux passages annotés de la question où la bonne réponse est trouvée ont une syntaxe proche de l'écrit. C'est la correction de l'annotation erronée d'un segment nominal en un segment verbal qui explique que la bonne réponse soit trouvée. Ces résultats peuvent être mettre en lien avec la matrice de confusion du chapitre 7 de la section 7.2 : on peut ainsi observer que les segments nominaux sont peu confondus avec des segments verbaux (9.8% des erreurs). Néanmoins, le mauvais typage a dans ce cas là suffit à avoir un impact, du fait de l'importance des verbes dans le réordonnanceur.

Si on peut donc observer une différence relativement faible entre l'annotation automatique et l'annotation manuelle, il faut néanmoins rappeler que le nombre de questions sur lesquelles sont effectuées cette expérience est peu élevé. De même, nous n'évaluons à chaque fois que deux passages, contre dix habituellement. Une expérience similaire mais avec plus de questions et de passages évalués devrait permettre d'avoir des résultats plus sures. On peut néanmoins émettre l'hypothèse que l'annotation automatique entraine des erreurs d'annotation, et que l'amélioration du modèle de segmentation doit permettre une amélioration des résultats.

8.2.2 Impact des relations et des opérations de rattachement

Le module de réordonnancement a pour objectif de réordonner les réponses candidates retournées par un système de questions-réponses. Ce module s'appuie notamment sur un score de similarité calculé entre un passage et une question. La méthode de calcul du score est fondée sur une distance d'édition entre les segments typés identifiés par le segmenteur. Généralement, cette distance d'édition est calculée à partir de trois types d'opérations d'édition : l'insertion, la suppression, et la substitution. Nous avons ajouté un quatrième type d'opération, dite de rattachement.

Ce nouveau type d'opération est l'un des points principaux de notre approche. Nous avons en effet constaté dans le chapitre 2 que l'un des désavantages du système Ritel, qui nous sert de contexte expérimental, est la non prise en compte de la structure des questions et des phrases dans les documents. Ces opérations de rattachement ont pour objectif de prendre en compte cette structure, en s'appuyant notamment sur le modèle de représentation présenté dans le chapitre 4. Les opérations de rattachement sont appliquées après une opération de substitution. Le but est de déterminer si le segment sur lequel est appliqué une opération est relié au candidat réponse évalué. Un score de rattachment est alors fourni, qui est calculé à partir d'un ensemble de règles écrites manuellement (voir le chapitre 5).

Si les résultats obtenus sont relativement mitigés sur certaines campagnes (notamment CLEF 2004 et 2005), nous estimons que cette approche a un potentiel intéressant, comme le montre les résultats obtenus sur une campagne comme Quaero. Une de nos hypothèses quant à l'origine de ces résultats est que les règles utilisées par les opérations de rattachement ne sont pas encore suffisamment nombreuses pour capturer l'information structurelle comprise dans une phrase. En effet, notre objectif est d'avoir une approche suffisamment robuste pour traiter tous types de documents, notamment des transcriptions de l'oral, dont la syntaxe est très différente de celle trouvée dans les documents journa-

listiques. De ce fait, nous avons volontairement simplifié certains traitements car nous voulions avant tout avoir un terrain d'expérimentation.

Par ailleurs, l'information structurelle d'une phrase est aussi représentée par les relations entre segments décrites dans le chapitre 4. Ces relations sont utilisées d'une part par les opérations de rattachement, mais aussi par les opérations d'insertion et de suppression. L'idée étant que ces relations représentent les dépendances entre les différents groupes de mots, un poids est attribué lors de l'insertion ou de la suppression d'un segment, pour quantifier l'importance structurelle d'une opération.

L'expérience présentée dans cette section a pour objectif d'observer les apports des relations et des opérations de rattachement sur le module de réordonnancement. Nous avons ainsi évalué trois versions du module sur les corpus de questions des différentes campagnes d'évaluation présentées dans le chapitre 7 :

– un réordonnancement effectué sans l'apport des relations dans le calcul des opérations d'insertion, de suppression, et de rattachement ;
– un réordonnancement effectué sans les opérations de rattachement ;
– un réordonnancement effectué sans relations ni opérations de rattachement.

Ces résultats sont présentés dans le tableau 8.2.

	QA@CLEF		QAst		Quaero	
Type réordonnanceur	Préc. (%)	MRR	Préc. (%)	MRR	Préc. (%)	MRR
réord-sans-rel-ratt	39.2	0.42	42.1	0.493	27.9	0.316
réord-sans-rel	39.7	0.42	42.4	0.494	28.2	0.319
réord-sans-ratt	40.0	0.43	42.9	0.496	28.2	0.321
réord	41.5	0.47	44.1	0.502	29.2	0.34

TAB. 8.2 – Evaluation de l'impact des relations et des opérations de rattachement ; réord-sans-* correspondent aux différentes versions du réordonnanceur : dans l'ordre, sans relations et rattachements, sans relations et sans rattachement ; réord correspond au réordonnanceur dans sa version normale.

On peut ainsi observer l'impact des relations et des opérations de rattachement. Une première analyse permet de constater qu'aussi bien les relations que les rattachements ont un impact sur les résultats. En effet, les meilleures performances sont obtenues lorsque le réordonnancement est appliqué avec les relations et les opérations de rattachement. Par contre, on peut constater que l'impact des relations et des rattachements sur les résultats du réordonnanceur est pour le moment trop faible : on obtient au plus une hausse de 2.3% (pour QA@CLEF). On peut aussi noter que l'impact le plus fort est sur le corpus sur lequel le réordonnanceur obtient les moins bons résultats. Il est néanmoins difficile de tirer une conclusion sur ce résultat.

Ces résultats laissent sous entendre que si les apports proposés ont un effet positif, il est encore nécessaire de les améliorer. Aussi bien nos relations que les règles utilisées dans nos opérations de rattache-

ment ont pour objectif de proposer un terrain d'expérimentation pour évaluer si ce type d'approche est pertinent. Les résultats montrent que les relations demandent à être améliorées, par exemple en ayant des types supplémentaires. On peut aussi observer que les rêgles des rattachements sont pour l'instant trop limitées pour avoir l'impact espéré. Une étude approfondie permettrait de déterminer les règles à ajouter. Ces règles sont en plus dépendantes des relations identifiées, ce qui prouve aussi le besoin de nouvelles relations.

8.2.3 Impact des synonymes

Le réordonnancement des candidats réponses s'appuie sur les similarités entre mots détectées entre la question et le passage d'un candidat réponse. Ces similarités sont de différents types : *identité*, *lemme*, *morpho-syntaxique*, et *synonymie*. La détection fait ainsi appel à un ensemble de dictionnaires, dont un de synonymes. Ce dictionnaire permet d'identifier les synonymes possibles pour un mot. Le problème est que nous ne gérons pour le moment pas le contexte d'application des différents synonymes.

Ainsi, selon le sens d'un mot, un synonyme associé peut avoir un sens proche ou très éloigné. Par exemple, pour le mot *abattu*, un synonyme possible est *démoralisé*. Ce synonyme est correct dans le cas où la phrase a comme contexte le moral d'une personne, comme dans *Johan était abattu à la suite de sa défaite au dernier match*. Par contre, son emploi est faux si *abattu* était utilisé dans un contexte physique, comme pour *L'arbre a été abattu*. De tels cas de figure peuvent ainsi conduire à identifier de mauvaises similarités, et donc avoir un fort impact sur le fonctionnement du réordonnancement.

Nous évoluons ainsi dans cette section les performances du réordonnanceur en fonction de l'utilisation ou non du dictionnaire des synonymes pour l'identification des similarités. Le tableau 8.3 présente les résultats obtenus par le réordonnanceur avec et sans l'utilisation du dictionnaire de synonymes sur les trois campagnes d'évaluation.

	QA@CLEF		QAst		Quaero	
Type réordonnanceur	Préc. (%)	MRR	Préc. (%)	MRR	Préc. (%)	MRR
réord-sans-syno	42.1	0.49	45.6	0.51	30.3	0.36
réord	41.5	0.47	44.1	0.50	29.2	0.34

TAB. 8.3 – Evaluation de l'impact du dictionnaire des synonymes sur le réordonnancement ; réord-sans-syno correspond à l'application du réordonnanceur sans l'utilisation des synonymes pour l'identification des similarités entre mots ; réord-avec-syno correspond au réordonnanceur dans sa version normale, c'est à dire en utilisant les synonymes.

Les résultats montrent une significative baisse des performances lorsque l'on utilise le dictionnaire des synonymes. S'il semble pertinent que l'utilisation des synonymes peut apporter lors de la détection des similarités entre mots, il est clair que l'utilisation que nous faisons du dictionnaire n'est

pas adaptée. Une amélioration de l'utilisation de ce dictionnaire, par exemple en gérant le contexte d'application des synonymes, est un travail envisagé pour le futur.

8.3 Analyse selon les caractéristiques des questions

Le réordonnanceur n'obtient pas de résultats clairement positifs. On observe toutefois des améliorations sur le corpus de questions 2008 de Quaero. De plus, la baisse des résultats sur QAst restent relativement basse. On peut ainsi émettre l'hypothèse que le réordonnanceur effectue des améliorations locales. Pour répondre à cette question, nous nous sommes focalisés sur une évaluation des caractéristiques des questions : la classe sémantique (type attendu de réponse) et le nombre d'éléments dans les questions.

8.3.1 Classes des questions

Une question factuelle est un type de question que l'on retrouve communément dans les campagnes d'évaluations comme QAst, Quaero ou QA@CLEF. Une question factuelle attend comme réponse un groupe de mots, généralement une entité nommée. Par exemple, pour la question *"Qui est le président de la France ?"*, la réponse attendue est *Nicolas Sarkozy*.

Les sujets possibles pour une question factuelle pouvant être variés, les évaluateurs des différentes campagnes d'évaluation ont très tôt classés ces questions. On retrouve plus ou moins les mêmes classes de questions d'une campagne à une autre (QAst et Quaero par exemple). Pour notre expérience, nous avons décidé d'utiliser les classes de questions apparaissant le plus fréquemment dans les campagnes d'évaluation, et contenant généralement un nombre élevé de questions. Nous rappelons cette classification dans le tableau 8.4.

Type	**Exemple**
personne	Qui est le nouvel entraîneur du Raja ?
lieu	Dans quelle ville un attentat a-t-il eu lieu ?
mesure	A combien s'élève l'amende de l'incendiaire de Daewoo ?
date/heure	Quand a été assassiné Hans Krasa ?

TAB. 8.4 – Classification des questions factuelles utilisée pour l'expérimentation

Nous avons donc générer les résultats obtenus dans chaque campagne selon cette classification des questions factuelles. Ces résultats sont présentés dans le tableau 8.5.

Les résultats présentés permettent de faire une première analyse sur les performances du réordonnanceur en fonction de la classe des questions. On peut en effet observer que le réordonnanceur n'entraine

	QA@CLEF			QAst			Quaero		
Classe question	#q	Réord	Ritel	#q	Réord	Ritel	#q	Réord	Ritel
personne	49	40.8	42.9	151	19.2	27.2	132	33.3	37.9
lieu	51	69.3	78.4	116	34.5	49.1	115	39.3	44.6
mesure	42	38.1	45.2	104	53.8	62.5	134	28.4	28.4
date/heure	50	48.0	50.0	107	36.4	44.9	127	**35.3**	31.4

TAB. 8.5 – Résultats obtenus sur les campagnes d'évaluation QA@CLEF (2004-2005), QAst(2008-2009) et Quaero(2008-2009) en fonction de la classe de la question ; précision obtenue pour le réordonnanceur (Réord) et Ritel ; #q : nombre de questions

généralement pas de hausse des résultats sur une catégorie de questions particulières. On peut néanmoins noter que les performances sur les questions date/heure sont plutôt bonnes, particulièrement sur Quaero. Par contre, les résultats obtenus sur les questions lieu sont mitigés. Etant donné que notre réordonnanceur utilise des relations de temps et de lieu, il est en effet intéressant de constater que l'impact de ces relations. Ainsi, si les questions de lieu ne semblent pas être bien traitées par le réordonnanceur, les questions de temps semblent mieux gérées, surtout sur Quaero. On peut néanmoins en déduire que les traitements spécifiques aux questions de temps et de lieu sont pour le moment trop simplistes pour avoir un véritable impact. Il est donc nécessaire de développer des mécanismes dans le traitement des relations de temps et de lieu, surtout pour les règles de rattachement.

8.3.2 Nombre d'éléments de la question

Le système Ritel, que nous avons présenté dans le chapitre 2 et qui sert de contexte expérimental à notre travail procède à une analyse de la question. Cette question génère un Descripteur De Recherche (DDR). Ce DDR contient les informations nécessaires au système de questions-réponses pour chercher la réponse à la question. On y trouve entre autre les éléments critiques et secondaires.

Les éléments d'une question sont utilisés par Ritel pour trouver une réponse candidate à une question. Le module de réordonnancement décrit dans le chapitre 5 s'appuie sur les relations entre les différents segments. Le module quantifie la similarité entre une question et un passage en calculant un score de transformation. Le module identifie les segments de la question partageant de l'information avec les segments du passage. Ces segments contiennent donc des éléments (critiques ou secondaires) de la question. Une fois cette identification terminée, des relations sont ajoutées entre les différents segments. En s'appuyant sur ces relations et ces segments, le réordonnanceur calcule le score de transformation, à partir d'opérations de transformation.

Ritel ne tient pas en compte de l'information structurelle des questions et des passages dans ses calculs. Le réordonnanceur s'appuie par contre sur le formalisme de représentation composé des segments typés et des relations entre les segments. Les opérations de rattachement s'appuient fortement

sur ces relations. Notre hypothèse est que les relations vont avoir un impact important dans le cas de questions où il est important d'avoir de l'information structurelle. Nous voulons confirmer cette hypothèse en observant les résultats obtenus selon la taille des questions : plus une question est longue, plus les relations et les opérations de rattachement vont être utilisés.

L'objectif de cette section est d'évaluer les résultats obtenus par le réordonnanceur sur chaque campagne en fonction du nombre d'éléments de chaque question. Ainsi, en s'appuyant sur la détection des éléments effectuée par Ritel, on classe les questions selon six classes, correspondant au nombre d'éléments contenus dans la question : de 1 élément jusqu'à 6 éléments.

Le tableau 8.6 présente les résultats obtenus sur les trois campagnes d'évaluation (QA@CLEF, Qast, Quaero) en fonction du nombre d'éléments contenus dans les questions.

	QA@CLEF			QAst			Quaero		
Nombre d'éléments	#q	Réord.	Ritel	#q	Réord.	Ritel	#q	Réord.	Ritel
1	88	58.0	63.6	109	48.6	67.0	71	30.3	39.4
2	89	34.8	51.7	173	42.2	61.8	201	**34.2**	32.3
3	83	43.4	43.4	215	67.4	77.9	177	23.4	27.6
4	49	28.6	42.9	113	65.5	77.9	95	22.7	22.7
5	22	**68.1**	50.0	58	**65.5**	60.3	34	**33.3**	25.0
6	11	**81.8**	63.6	21	**85.7**	76.0	13	**66.7**	33.3

TAB. 8.6 – Résultats obtenus sur les campagnes d'évaluation QA@CLEF (2004-2005), QAst(2008-2009) et Quaero(2008-2009) en fonction du nombre d'éléments dans la question ; #q : nombre de questions.

Contrairement aux résultats présentés dans le tableau 8.5, une première analyse des résultats montrés dans le tableau 8.6 permet de distinguer deux classes de questions bien traitées par le réordonnanceur : les questions à 5 et 6 éléments. On observe une hausse des résultats sur les trois campagnes. Ces résultats semblent prouver que le réordonnanceur est plus efficace lorsqu'un grand nombre de relations doit être traité, ce qui est le cas lorsque les questions ont un grand nombre d'éléments.

On peut aussi noter que l'on obtient pour Quaero de bons résultats sur les questions à 2 éléments. Néanmoins, si l'on met en parallèle ce résultat avec ceux obtenus pour les autres campagnes, il n'est pas possible de dire que le réordonnanceur est efficace pour cette classe de questions. Ce bon résultat est plus une conséquence de résultats moyens de Ritel dans le cadre de Quaero qu'une véritable efficacité du réordonnanceur. Enfin, ces bons résultats sur les questions avec un nombre élevé d'éléments laissent supposer que cette approche peut être efficace pour des questions plus complexes.

Du fait de ces bons résultats, le réordonnanceur a été évalué officiellement à l'édition 2010 de Quaero. Les questions factuelles comprenant 2, 5 ou 6 éléments sont traitées par le réordonnanceur, tandis que les autres questions sont traitées par l'approche habituelle de Ritel. Les résultats obtenus sont présentés dans le tableau 8.7.

		Ritel		Ritel+Réord.		Δ
type questions	#q.	Préc. (%)	MRR	Préc. (%)	MRR	
toute	309	34.0	0.424	37.2	0.452	**+3.2**
normale	175	34.3	0.431	37.7	0.463	**+3.4**
orale	88	34.6	0.416	37.2	0.440	**+2.6**
réécrite	56	32.1	0.416	35.7	0.434	**+3.6**

TAB. 8.7 – Résultats officiels obtenus à l'édition 2010 de Quaero pour les questions factuelles.
Ritel : résultats obtenus par le système Ritel.
Ritel+Réord. : résultats obtenus en appliquant le réordonnanceur sur les questions à 2, 5 et 6 éléments.
toute : toutes les questions factuelles de 2010 ; normale : questions créées selon la méthodologie de 2009 ; orale : questions spontanées posées à l'oral ; réécrite : questions spontanées réécrites.

On observe ainsi une hausse significative des résultats. Ces résultats sont très positifs et prouvent notre hypothèse de départ : un grand nombre d'élements dans une question implique que le réordonnancement fait plus appel aux relations et aux opérations de rattachement. Ainsi, si le réordonnanceur n'est pas encore applicable à un niveau global, l'approche a néanmoins un fort potentiel. Par ailleurs, on peut observer que l'approche est robuste aux différentes formes des questions. Ce résultat est particulièrement intéressant pour les questions orales : lors de l'application du réordonnanceur sur toutes les questions orales, une baisse assez significative avait été observée par rapport aux résultats obtenus par Ritel (la précision chutait de 7.4%). La hausse observée dans le tableau 8.7 montre donc que notre approche est robuste à n'importe quelle forme de questions longues (traditionnelle, orale, ou réécrite). Ces résultats sont évidemment très positifs.

8.4 Impact des caractéristiques des campagnes sur les systèmes de questions-réponses

8.4.1 Présentation

Lors de la création d'un corpus de questions pour une campagne d'évaluation de systèmes de questions-réponses, les organisateurs doivent définir un certain nombre de directives : types de documents traités, création des questions, types de questions traités, etc ... Le nombre de caractéristiques à prendre en compte est assez grand, et il peut ainsi être difficile de comparer l'évaluation d'un système de questions-réponses entre deux campagnes différentes. Si l'on prend l'exemple des campagnes QA@CLEF et QAst, le type des documents traités est différent : articles journalistiques pour QA@CLEF, et transcriptions d'émissions de radio pour QAst. Ainsi, la comparaison des résultats obtenus par un même système est forcément biaisée : les documents issus de l'audio n'ont par exemple pas la même syntaxe que ceux issus de sources journalistiques et posent des difficultés différentes aux systèmes.

Il est alors logique d'évaluer les progrès effectués par un système sur différentes itérations d'une même campagne. Le contexte d'évaluation étant le même, il est ainsi plus simple d'observer l'évolution d'un système de questions-réponses. Néanmoins, d'une année à l'autre, les auteurs d'une campagne d'évaluation ont tendance à ajouter de nouvelles caractéristiques. On peut par exemple citer l'ajout des questions factuelles temporelles restreintes dans QA@CLEF 2005 [Vallin et al. 2005]. L'intérêt est ainsi de confronter les systèmes de questions-réponses à de nouvelles problématiques. Néanmoins, il arrive que certains ajouts d'une année à l'autre aient pour effet de changer les critères d'évaluation d'une campagne.

Pour l'itération 2009 de QAst [Turmo et al. 2009], les auteurs ont introduit un nouveau modèle de création des questions par rapport à 2008. L'objectif des organisateurs était de créer des questions plus spontannées (voir section 6.2.2 du chapitre 6). Ces questions sont sous deux formes : les questions orales transcrites manuellement, et les questions orales réécrites sous forme de "questions écrites". On n'observe pas de différences significatives pour les systèmes sur ces deux formes de questions [Turmo et al. 2009]. Par contre, on observe une importante régression des résultats par rapport à 2008. Cette régression peut être observée pour le système Ritel sur les trois langues évaluées dans QAst 2009 dans le tableau 8.8. La même version du système est utilisée sur les deux corpus.

	Français	
	Acc(%)	Δ
QAst 2008	50	-22
QAst 2009	28	
	Anglais	
	Acc(%)	Δ
QAst 2008	52	-25
QAst 2009	27	
	Espagnol	
	Acc(%)	Δ
QAst 2008	56	-20
QAst 2009	36	

TAB. 8.8 – Variation entre les résultats obtenus sur QAst 2008 et 2009. Le Δ mesure la différence entre les résultats des deux années.

Par ailleurs, cette chute a aussi été observée pour les autres systèmes participant sur l'anglais [Comas & Turmo 2009 ; Bernard, et al. 2010]. Pour l'espagnol, le seul autre système à avoir été évalué obtient de meilleurs résultats qu'en 2008. Cependant, les résultats de 2008 proviennent d'une différente version que celle du système utilisé en 2009. La comparaison est donc biaisée.

Il ressort donc de ces résultats que le changement dans la méthode de création des corpus de questions a une grande importance sur les caractéristiques d'évaluation de la campagne. A cause de ce fort impact, il est difficile d'évaluer l'évolution des systèmes de questions-réponses entre les itérations 2008 et 2009 de QAst. Par contre, cela prouve qu'il est intéressant de mesurer l'évolution d'une

campagne d'évaluation, pour ainsi comprendre sur quelles caractéristiques les systèmes de questions-réponses sont évalués.

La distance entre les éléments critiques d'une question et une réponse candidate joue très souvent un rôle important dans le fonctionnement d'un système. Si l'on prend le système Ritel [Rosset et al. 2006], la distance entre les éléments et une réponse entre en compte dans le scoring d'un candidat réponse. De même, le système de l'UPC [Comas & Turmo 2009] utilisé pour QAst 2009 définie les passages comme étant des segments où deux mots-clés consécutifs sont séparés par au plus *w* mots.Ainsi, cette distance est une caractéristique importante d'une campagne d'évaluation. Pouvoir mesurer cette distance peut permettre d'analyser les difficultés rencontrées sur une évaluation. Nous proposons dans la section suivante une mesure simple permettant d'évaluer la distance moyenne entre les éléments de la question et la réponse.

8.4.2 Distance moyenne entre les éléments de la question et la réponse

Dans certaines campagnes d'évaluation (QAst par exemple), seuls les réponses correctes et les documents où elles peuvent être extraites sont fournis par les évaluateurs. De ce fait, nous ne connaissons pas les passages utilisés pour créer les questions. Ces passages contiennent les éléments de la question (ou des transformations de ces éléments) utilisés pour créer la question. De plus, nous savons dans quel document la réponse peut être trouvée, mais il existe souvent plusieurs occurences de cette même réponse dans le document. Comme nous ne savons pas où sont situés les éléments utilisés pour créer la question, nous avons besoin d'une approche évaluant la répartition globale des éléments de la question par rapport à chaque occurence de la question dans le document.

L'objectif de cette mesure est de fournir une distance "globale" entre les éléments de la question et la réponse, pour ainsi évaluer la répartition des informations dans les documents. Cette distance est une distance *physique* qui est calculée en comptant le nombre de mots entre chaque élément de la question et la réponse.

Pour chaque question liée à une collection de documents nous calculons la distance globale entre les éléments de la question et chaque portion du document contenant une occurence de la bonne réponse. Seuls les éléments définis comme critiques sont utilisés dans le calcul. Les éléments considérés comme étant important sont des entités nommées (classiques, étendues et non-spécifiques) et des expressions à mots multiples. La mesure globale de la question est donc la moyenne des distances de chaque élément. Cette distance globale est calculée pour chaque occurence d'une réponse, puis le système choisit celle avec la valeur la plus faible comme distance globale de la question. De ce fait, la mesure globale d'un corpus est la moyenne des distances globales de chaque question.

Les deux exemples ci-dessous expliquent comment la mesure globale est calculée pour deux questions. Dans le premier exemple, la réponse correcte à la question *Quelle organisation belge a été déclarée criminelle ?* est *Vlaams Blok*. Les distances ont été calculées entre cette réponse et chaque élément critique de la question : *belge*, *organisation* et *criminelle*. Les distances correspondantes en

mots sont 7, 2 et 3. La mesure globale est donc de 4.

Quelle **organisation belge** *a été déclarée* **criminelle** *?*

La Court suprême **belge** *a confirmé un précédent arrêt déclarant que Vlaams Blok est une* **organisation criminelle**.

Le deuxième exemple est basé sur un passage de texte plus long. La bonne réponse à la question *Quel chef politique de Palestine est mort récemment ?* est *Arafat*. Les éléments critiques sont *mort*, *Palestine*, *politique* et *chef*. Les distances correspondantes sont 0, 11, 36 et 35, et la mesure globale est donc de 20.

Quel **chef politique** *de* **Palestine** *est* **mort** *récemment ?*

La **mort** *d'Arafat implique que nous allons avoir à présent une nouvelle élection en* **Palestine**. *L'Union Européenne a déclaré à Israel que le dialogue entre les deux pays est important. Il est nécessaire d'avoir un nouveau* **chef politique** *aussi vite que possible.*

8.4.3 Evaluation de la mesure

Cette mesure est appliquée sur les corpus des campagnes QAst et Quaero, à chaque fois pour les éditions 2008 et 2009, et aussi pour l'édition 2010 dans le cas de Quaero. Nous n'avons pas appliqué cette mesure sur la campagne QA@CLEF, car nous ne disposions pas des documents dans lesquels sont contenus les bonnes réponses pour l'itération de 2005, ce qui empêche d'effectuer la comparaison avec 2004.

Le tableau 8.9 montre les résultats obtenus avec cette mesure sur QAst et Quaero. On donne la moyenne, qui correspond à la distance globale obtenue pour un ensemble de questions, et l'écart type.

	QAst		Quaero	
	Moy.	Ecart type	Moy.	Ecart type
2008	45	100	33	163
2009	143	431	37	289
2010	-	-	29	99

TAB. 8.9 – Evolution de la distance moyenne entre QAst 2008 et QAst 2009 et entre Quaero 2008, Quaero 2009 et Quaero 2010.

On peut observer sur QAst dans le tableau 8.9 une hausse de la distance moyenne entre 2008 et 2009 (+98). Il semble donc que le processus de création des questions de 2009 a un impact sur la distance entre les éléments d'une question et la réponse. Néanmoins, il convient de corréler ces résultats avec

ceux observés sur l'anglais et l'espagnol, présentés dans [Bernard et al. 2010]. Si on observe aussi une hausse pour l'anglais (+39), la distance chute considérablement pour l'espagnol, passant de 381 à 22. De ce fait, s'il semble possible que le nouveau processus de création des questions ait un impact sur la distance, il ne l'augmente pas forcément.

Pour mieux observer la répartition des valeurs des distances obtenus pour les questions, nous présentons la distribution de ces valeurs dans la figure 8.1. Nous partageons ces valeurs en 9 catégories, allant des questions avec une distance de zéro jusqu'aux questions avec une distance supérieur à 500. L'axe des X représente ces catégories et l'axe des Y le nombre de questions de chaque catégorie. Cette distribution permet d'observer l'évolution du corpus de questions entre 2008 et 2009.

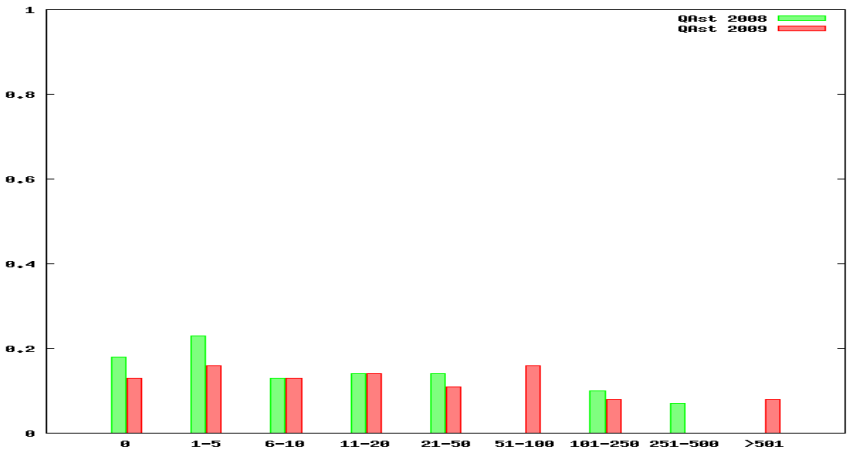

FIG. 8.1 – Distribution des distances sur QAst 2008 et QAst 2009

Cette distribution montre qu'il y a une grande dispersion des valeurs des distances des questions entre 2008 et 2009, ce qui expliquent la hausse observée. Ainsi, en 2009 il y a par exemple près de 4 questions avec une valeur supérieure à 500, tandis qu'en 2008 il y a plus de questions avec une distance faible. On peut ainsi observer l'évolution du corpus entre 2008 et 2009 pour la campagne QAst.

Pour la campagne Quaero, les résultats présentés dans le tableau 8.9 ne montrent pas une hausse importante de la distance. Le processus de création a changé, mais cela ne semble pas avoir entrainé de différences fondamentales entre 2008 et 2009 au niveau de la distance entre les éléments de la question et la réponse. De même pour 2010, où l'ajout des questions spontanées sous deux formes (transcriptions manuelles et réécrites) n'a pas entrainé d'écarts importants : l'écart maximal observé est entre 2009 et 2010 (+6).

Comme pour la campagne QAst, nous illustrons la distribution des valeurs des distances obtenues pour les questions de Quaero dans la figure 8.2.

On peut observer que la distribution des valeurs est relativement identique entre 2008, 2009 et 2010 :

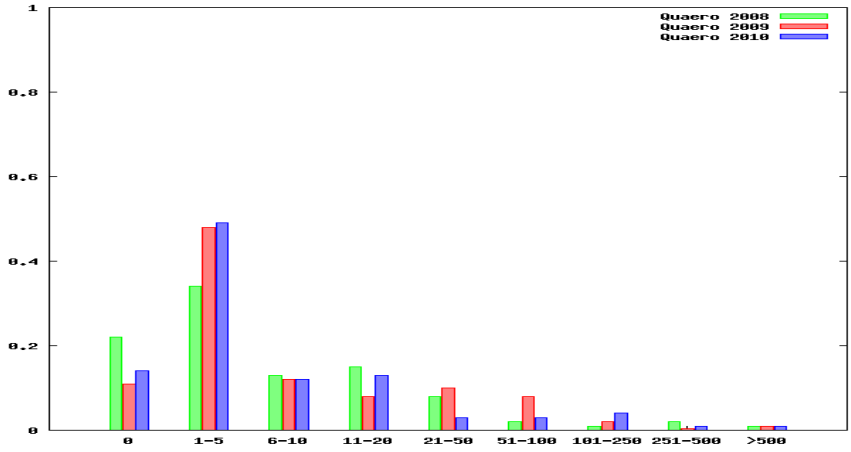

FIG. 8.2 – Distribution des distances sur Quaero 2008, Quaero 2009 et Quaero 2010.

excepté pour les questions avec une distance globale comprise entre 1 et 5, la distribution est relativement similaire. Il nous semble aussi intéressant d'observer si la forme d'une question a un impact sur la distance pour 2010. Nous donnons donc dans le tableau 8.10 les moyennes et écarts types obtenus sur chaque corpus de 2010 : questions normales, spontanées et réécrites.

Type question	Moy.	Ecart type
normale	25	74
spontanée	13	31
réécrite	65	190

TAB. 8.10 – Comparaison de la distance moyenne entre les différents corpus de questions de Quaero 2010.

On peut ainsi observer une hausse importante de la moyenne et de l'écart type pour les questions réécrites. Il semble donc que la réécriture des questions a entrainé une différence importante de certaines questions par rapport aux questions spontanées correspondantes. Pour mieux observer cette hausse, nous donnons la distribution des valeurs des distances dans la figure 8.3.

La hausse observée dans le tableau est effectivement illustrée par cette distribution. Ainsi, si les distributions sont relativement proches entre les questions normales et les questions spontanées, on peut voir que dans la classe des questions avec une valeur supérieure à 500, on trouve un nombre assez important de questions réécrites. Cette observation est d'autant plus intéressante qu'il n'y a aucune question spontanée dans cette classe. Une étude est cours pour comparer les questions réécritées avec une valeur supérieure à 500 aux questions spontanées correspondantes.

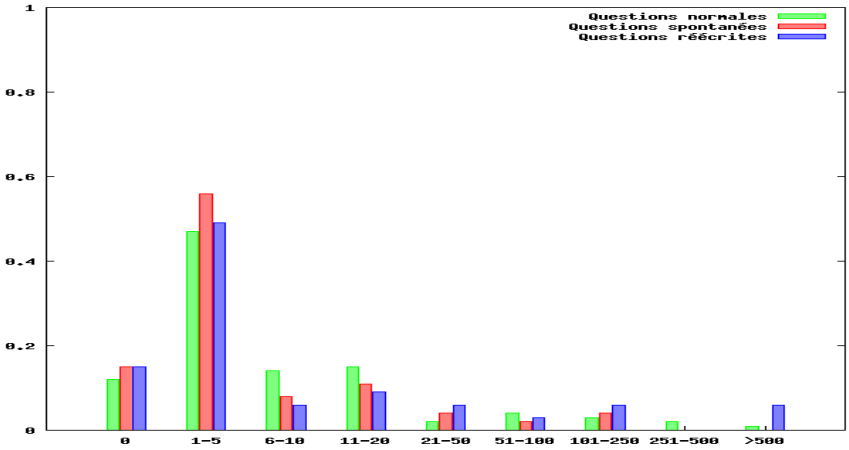

FIG. 8.3 – Distribution des distances sur les différents corpus de questions de Quaero 2010.

8.4.4 Impact de la mesure sur les systèmes de questions-réponses

La distance entre les éléments d'une question et un candidat réponse est souvent une composante fondamentale de l'extraction des passages par un système de questions-réponses. Les documents sont segmentés en passages, pour ainsi préparer l'extraction de la réponse. Dans la section 2.3.2.2 du chapitre 2, l'extraction des passages de Ritel est ainsi basée sur un certain nombre de paramètres indiquant la taille maximale d'un passage. Or, ces paramètres sont déterminés par le biais d'un corpus de développement. De ce fait, si les données d'une campagne sont différentes d'une année à une autre, les paramètres utilisés par Ritel ne seront pas adaptés pour traiter les questions. Ce mécanisme peut expliquer les difficultés rencontrés par le système. On peut citer d'autres approches dépendant aussi de paramètres de taille fixés sur un corpus d'apprentissage ou de développement pour les passages. Le système de l'UPC de 2009 [Comas & Turmo 2009], qui a lui aussi participé, définit les passages comme étant des segments où les mots-clefs ne sont pas séparés de plus de w mots, w étant fixé empiriquement.

De ce fait, si un système a fixé une taille trop grande pour l'extraction des passages, le bruit augmente, et il y a beaucoup plus de candidats réponses à évaluer, ce qui rend plus difficile l'extraction d'une bonne réponse. De même, si la taille est trop petite, alors le silence est important : il y a moins de passages avec une réponse proche des mots clefs de la question.

Cette distance a aussi un impact pour le réordonnanceur. Nous avons présenté dans le chapitre 5 une méthode de réduction des passages traités, dans le but d'éviter de traiter des passages trop long. Cependant, l'approche utilisée peut ne pas être adaptée dans le cas où les éléments d'une question sont généralement éloignés de la réponse. Ainsi, il peut être intéressant d'utiliser cette mesure pour déterminer une taille relativement optimale pour les passages, à la condition de disposer d'un corpus de développement de taille suffisante (représentabilité), et correspondant effectivement à la tâche. Les campagnes d'évaluation ont tendance à considérer que les données de l'année précédente servent de

développement/apprentissage pour l'année en cours. Cependant, cette option n'est pas raisonnable lorsqu'on change trop certains paramètres, comme nous l'avons montré avec cette mesure.

Discussion

Nous avons présenté les évaluations des deux composants principales de notre travail, le segmenteur et le réordonnanceur. Le segmenteur a été évalué à partir de trois corpus de test annotés manuellement, correspondant à trois différents types de texte : journalistique, oral, et web. Le réordonnanceur a été évalué à partir de différentes campagnes d'évaluation de systèmes de questions-réponses. Les campagnes d'évaluation fournissent aux systèmes de questions-réponses un cadre d'évaluation permettant d'observer les progrès effectués par les systèmes. De plus, les campagnes ont des caractéristiques différentes, permettant d'évaluer le comportement des systèmes sur différents critères. Ainsi, nous avons évalué le réordonnanceur sur trois campagnes différents, QA@CLEF, QAst et Quaero, utilisant respectivement des documents journalistiques, oraux, et du web.

L'évaluation du segmenteur a mis en évidence l'intérêt du formalisme employé quant à sa robustesse quelque soit le type de texte traité. Si on observe une chute relativement faible des résultats entre l'écrit et l'oral (f-mesure de 0.83 pour l'écrit et 0.80 pour l'oral), la perte reste relativement faible, surtout si on compare avec les résultats de la campagne d'évaluation des analyseurs syntaxes EASY, où le meilleur système obtient une f-mesure de 0.79 sur l'oral, 0.92 sur l'écrit. Cependant, même si le formalisme utilisé pour la campagne EASY est différent de celui employé par le segmenteur, la f-mesure de l'écrit reste bien plus basse que celle des meilleurs systèmes d'EASY (0.83 contre 0.92). Le formalisme employé par le segmenteur est fondés sur deux segments principaux : le segment nominal et le segment verbal. Il existe par ailleurs quatre sous-segments nominaux. Les deux segments principaux sont primordiaux dans les algorithmes utilisés par le réordonnanceur. Or, il arrive que des segments verbaux soient mis de manière eronnés à la place de segments nominaux. Les segments verbaux ayant un rôle de pivot dans les traitements effectués par le réordonnanceur, ces erreurs ont forcément un impact sur le réordonnanceur. Une analyse a été effectuée en annotant manuellement un sous-ensemble de questions et passages pour mesurer l'impact des erreurs de segmentation. Cette analyse montre que si ces erreurs ont un impact sur les résultats du réordonnanceur, la perte engendrée reste relativement faible sur le sous-corpus de questions évalué : sur la centaine de questions où le réordonnanceur ne trouvait habituellement pas la bonne réponse, seulement quatres bonnes réponses ont finalement été trouvées. Le comportement de notre segmenteur semble donc encourageant, même si certains points peuvent être améliorés.

L'évaluation du réordonnanceur a donné lieu à des résultats plus mitigés, surtout en les comparant à

ceux obtenus par le système Ritel. Sur les trois campagnes d'évaluation étudiées, les résultats globaux ne sont jamais supérieurs à ceux obtenus par Ritel. L'objectif du réordonnanceur étant d'améliorer les résultats obtenus par un système de questions-réponses, cet aspect de notre travail est un échec. Par contre, on peut constater une certaine robustesse par rapport aux types de données traités : les baisses observées sont au maximum d'environ 4%. On peut en déduire que l'approche utilisée par le réordonnanceur est stable quelque soit le type de données, ce qui était un de nos objectifs.

On peut noter une amélioration des résultats sur certains sous-corpus des campagnes d'évaluation, notamment Quaero 2008, ce qui laisse sous entendre que le réordonnanceur a un impact à un niveau local. Une analyse des résultats obtenus selon les classes de question a montré que les questions contenant un nombre élevé d'éléments étaient bien traitées par le réordonnanceur. Cette hausse des résultats est corrélée à l'identification des relations entre segments : des opérations de transformation sont appliquées sur les segments du passage contenant des éléments de la question. Or, les relations sont utilisées dans l'ensemble des opérations, particulièrement sur les opérations de rattachement. Ces résultats sont très positifs : l'approche est particulièrement adaptée pour traiter les questions longues. Ce fonctionnement du réordonnanceur peut expliquer la hausse des résultats sur les questions contenant plus d'éléments. Du fait de ce comportement, le réordonnanceur a été évalué officiellement sur l'édition 2010 de Quaero : en ne l'appliquant que sur certaines questions, nous avons pu ainsi observer une hausse d'environ 3% par rapport aux résultats obtenus par Ritel. De plus, cette hausse peut être observée aussi bien sur les questions traditionnelles que celles posées à l'oral. Ces résultats sont très prometteurs et montrent les possibilités de l'approche employée ainsi que sa robustesse.

On peut donc constater que le réordonnanceur a un impact positif à un niveau local, ce qui laisse sous entendre que l'approche proposée est prometteuse. Néanmoins, les résultats restent encore trop faibles par rapport à nos objectifs initiaux, et une analyse modulaire du système a été effectuée pour comprendre les problèmes. Nous avons déjà estimé que les erreurs de segmentation avait un impact relativement faible sur le comportement du réordonnanceur. Le reste de l'analyse modulaire avait pour objectif d'évaluer l'impact des deux autres contributions majeures de notre travail : les relations entre segments, et les opérations de rattachement. Les relations font parties avec la segmentation du modèle de représentation robuste adopté pour représenter la structure des phrases et questions. Le réordonnanceur s'appuie sur un calcul de coût de transformation utilisant une distance d'édition : en plus des opérations d'édition traditionnelles (suppression, insertion et substitution), nous rajoutons les opérations de rattachement. Ce type d'opération a pour objectif d'identifier les rattachements entre les éléments de la question et un candidat réponse, et ainsi prendre en compte la structure de la phrase dans le calcul de la distance d'édition. L'analyse modulaire a montré que les relations et les opérations de rattachement pouvaient apporter jusqu'à plus de 2% de précision supplémentaire, prouvant l'intérêt de ces deux composants. Néanmoins, l'apport reste relativement faible, laissant supposer qu'un travail supplémentaire doit être effectué. Un des besoins les plus évidents concerne les règles utilisées par les opérations de rattachement. Ces règles ont été écrites manuellement mais sont encore trop peu pour véritablement couvrir l'impact linguistique estimé par une opération de rattachement. Un ajout conséquent de règles pourrait ainsi permettre d'améliorer considérablement les résultats. Nous avons aussi montré que l'utilisation du dictionnaire de synonymes sur lequel le réordonnanceur s'appuie pour détecter certaines similarités entre les mots d'une question et d'un passage provoque

une dégradation des résultats. Si cette ressources contient des informations intéressantes, son emploi n'est pour le moment pas adapté à notre travail. Une gestion du contexte d'application des synonymes pourraient permettre d'intégrer de manière plus satisfaisante ce dictionnaire.

Enfin, nous avons présenté une mesure d'évaluation des campagnes d'évaluation de systèmes de questions-réponses. Nous avons déjà expliqué que les campagnes utilisées pour évaluer notre module de réordonnanceur avaient des caractéristiques différentes, principalement le type de documents traité : journalistique, oral, web. Ces caractéristiques sont en général connues des participants. Néanmoins, nous estimons que les choix effectués par les organisateurs d'une campagne peuvent avoir des effets non prévus : par exemple, pour l'édition 2009 de la campagne QAst, la nouvelle méthode de création des questions a eu un impact considérable sur les résultats des systèmes. L'idée était d'avoir des questions plus spontannées, mais a aussi eu pour effet de modifier la distance moyenne entre les éléments des questions et les bonnes réponses. Sur les trois langues traitées, la distance moyenne a considérablement augmenté pour l'anglais et le français, et énormément chuté pour l'espagnol. La mesure proposée a pour objectif d'évaluer cette distance moyenne, et ainsi fournir aussi bien aux organistateurs d'une campagne qu'aux participants un outil pour analyser les caractéristiques d'une campagne. La distance moyenne peut avoir un impact considérable pour les systèmes de questions-réponses, notamment au niveau de l'extraction des passages. Par ailleurs, le réordonnanceur emploie une méthode de réduction des passages qui peut potentiellement perdre de l'information sur certaines campagnes si la réponse est généralement trop éloignée des éléments de la question. C'est pourquoi nous estimons que cette mesure est importante dans le cadre des systèmes de questions-réponses.

Quatrième partie

Conclusions et perspectives

Chapitre 9

Conclusions

Le travail présenté dans ce document s'inscrit dans le cadre des systèmes de questions-réponses. L'objectif des systèmes de questions-réponses est de répondre à certaines limites des systèmes de Recherche d'Informations (RI). Les systèmes de RI permettent à un utilisateur de chercher des informations générales sur un sujet. L'utilisateur formule sa requête sous la forme d'une requête composée de mots-clefs, et le système retourne un ensemble de documents contenant potentiellement l'information recherchée. De tels systèmes ont cependant un certain nombre de limites. Ainsi, l'utilisateur est obligé de chercher dans les documents l'information désirée, et exprimer sa requête à l'aide de mots-clefs ne permet pas de définir précisément l'information cherchée, en plus d'être moins naturel pour un humain. Les systèmes de questions-réponses ont pour objectif de répondre à ces limites.

Un système de questions-réponses permet ainsi à un utilisateur de poser sa question en langue, et fournit une réponse précise à sa requête. Un système retournera à la question *"Quel âge avait Nelson Mandela à sa sortie de prison ?"* la réponse *72 ans*. Cette réponse est extraite d'une base de documents. De tels systèmes donnent de bons résultats en domaine fermé, comme par exemple pour la réservation de billets de train. Le travail présenté dans ce document s'inscrit dans un contexte ouvert : les questions pouvant être posées sur n'importe quel sujet, la structure des questions ainsi que la forme et le type de la réponse (âge d'une personne, date d'un événement ...) sont bien plus variés. Travailler en domaine ouvert soulève ainsi des problématiques supplémentaires. Par exemple, en domaine fermé les informations sont extraites de base de données, tandis qu'en domaine ouvert les systèmes travaillent généralement sur des documents textuels. Du fait du nombre élevé de problématiques, une grande partie des systèmes de questions-réponses ont été conçus pour traiter dans un premier temps des questions factuelles, où une réponse courte est attendue, comme pour *"Quel âge avait Nelson Mandela à sa sortie de prison ?"*. Par ailleurs, les systèmes traitent majoritairement des documents issus de textes journalistiques, même si certains ont été conçus pour ou adaptés à d'autres types de données, comme l'oral ou le web.

Il n'y a pas vraiment d'approche type pour les systèmes de questions-réponses. Néanmoins, on a

tendance à retrouver une architecture type. Les systèmes sont organisés en plusieurs modules : analyse de la question, sélection des documents, sélection des passages, et sélection et extraction de la réponse. L'objectif de ce type d'organisation est d'isoler les réponses candidates dans des textes de plus en plus courts, et ainsi appliquer des traitements de plus en plus complexes. Ainsi, certains systèmes utilise un module supplémentaire, dit de réordonnancement de réponses, placé en bout de chaine de traitement. L'objectif de ce module est de calculer un nouveau score sur un ensemble de candidats réponses, et de les réordonner en fonction de ce score. Les modules de réordonnancement utilisent généralement des approches plus complexes que les traitements précédents. Notre travail avait pour objectif de proposer un module de réordonnancement de réponses, avec un certain nombre de contraintes.

L'un de nos principaux objectifs est de proposer un module de réordonnancement robuste à tous types de données. Les documents journalistiques sont traités majoritairement par les systèmes de questions-réponses. Ces documents sont généralement bien écrits et ont une syntaxe classique, permettant par exemple d'appliquer des analyses syntaxiques et sémantiques complexes. Néanmoins, il a été montré que ces approches ne sont pas toujours applicables sur d'autres types de documents, comme dans des transcriptions de données oral, ou des documents du web. Avoir un module de réordonnancement devant traiter n'importe quel type de documents soulève ainsi de nouvelles problématiques que nous avons du prendre en compte. De même, notre travail a été effectué sur le français, ce qui implique entre autre d'avoir accès à des ressources linguistiques plus limitées qu'en anglais. Enfin, nous avons décidé de nous limiter aux questions factuelles : si une grande partie des questions posées par les utilisateurs sont d'un type différent, elles sont aussi plus complexes à traiter. Travailler sur des questions factuelles nous a permis ainsi de vérifier nos hypothèses sur des questions plus "simples" à traiter.

Si le réordonnanceur a comme objectif d'être généralisable à n'importe quel système de questions-réponses, nous nous sommes néanmoins placés dans un contexte expérimental particulier : le système de questions-réponses Ritel. Ritel est un système de dialogue ayant la recherche d'information comme une de ses fonctionnalités. Les auteurs ont choisi de s'orienter vers les systèmes de questions-réponses, et ont ainsi dû faire face aux contraintes de l'oral. Il existe trois familles d'approches pour les systèmes de questions-réponses. Les systèmes fortement linguistiques, basés sur des analyses profondes et utilisant des bases de connaissances importantes. A l'opposé, on retrouve des systèmes fortement statistiques n'utilisant quasiment pas de connaissances. Enfin, la famille la plus nombreuse est celle des systèmes intermédiaires, qui utilisent aussi des connaissances linguistiques, mais en nombre plus limité. Ritel fait parti de cette dernière catégorie.

Pour s'adapter aux contraintes de l'oral, les concepteurs de Ritel ont conçu un moteur d'analyse de la langue générique s'appliquant aux questions et documents traités par le système. Ce moteur fournit principalement de l'information sémantique. Le système de questions-réponses s'appuie sur cette analyse au travers des différents modules de traitement. La phase d'extraction de la réponse utilise en particulier une approche fondée sur deux composants : la répartition des éléments de la question par rapport à un candidat réponse, et la redondance de ce candidat dans l'ensemble de la collection de documents. Les résultats obtenus aux différentes campagnes d'évaluation sont bons. Toutefois, l'approche utilisée par le module d'extraction des réponses montre certaines limites. Ainsi, l'utilisation de la redondance entraîne certaines ambiguïtés. De plus, le manque de représentation de

l'information structurelle des phrases et des questions, comme l'absence de groupements de mots (groupe nominal, groupe verbal ...) entraînent certaines limitations. Le module de réordonnancement présenté dans ce document était donc évalué en fin de chaine de traitement du système Ritel.

Nous nous sommes donc intéressés aux différentes approches de réordonnancement de candidats réponses. Le réordonnancement n'étant pas un module que l'on trouve dans tous les systèmes de questions-réponses, nous avons aussi étudié différentes approches utilisées pour l'extraction de réponses : ces deux modules ont en effet pour point commun d'appliquer tous deux des méthodes relativement complexes. Les approches étudiées ont toutes des caractéristiques spécifiques : type de documents traités, langue utilisée, méthode par apprentissage ... Il n'existe ainsi pas d'approche standard pour ce type de problèmes. On peut néanmoins noter que le modèle de représentation des questions et documents utilisé a une part prépondérante dans chaque approche. Par exemple, le système de l'UPC [Comas et al. 2010] utilise un analyseur développé en interne fournissant habituellement de l'information sémantique et syntaxique, mais qui a dû être adapté aux données orales : dans le cadre de la tâche de questions-réponses, l'analyseur ne fournit alors qu'une analyse syntaxique simplifiée. Du fait de l'importance de ce modèle de représentation, nous avons procédé à une étude des analyses applicables dans le contexte des systèmes de questions-réponses. Les systèmes utilisent généralement une analyse syntaxique et/ou sémantique. Parmis les caractéristiques communes, on peut ainsi observer que les analyses s'appuient souvent sur une segmentation (chunking) des phrases et des questions en groupes de mots typés. De même, des dépendances syntaxiques sont souvent identifiées entre ces groupes de mots, permettant ainsi de représenter l'organisation structurelle d'une phrase ou d'une question. Enfin, les verbes ont généralement un rôle de pivot, soulignant leur importance au sein des phrases. A partir de ces deux études, nous avons décidé de diviser notre travail en deux composants : la mise au point d'un modèle de représentation des documents et des questions robuste à tous types de données, et une méthode de calcul d'un nouveau score s'appuyant sur ce modèle de représentation.

Nous avons proposé un modèle de représentation divisé en deux parties : la segmentation des phrases et des questions en groupe de mots typés, et l'identification de relations entre ces segments. Etant donné que l'analyse de Ritel fournit déjà de l'information sémantique, l'objectif était d'avoir une représentation de la structure des phrases et des questions adaptée à tous types de documents. Ainsi, nous avons décidé d'avoir un formalisme de segmentation relativement simple contrairement à ce qui peut être proposé dans la littérature : nous avons deux segments principaux, le segment nominal et le segment verbal. Ces segments sont centrés respectivement autour d'un nom ou d'un verbe, et regroupent les mots en rapport : un déterminant, un adjectif ou une préposition par exemple pour un segment nominal. Cette approche se rapproche fortement des Syntagmes Non Récursifs présentés dans [Vergne 1999]. Nous utilisons aussi quatre sous-types de segments nominaux : les segments de temps et de lieu, le segment marqueur interrogatif, et le segment optionnel. Les segments temps et lieu correspondent à des groupes circonstanciels, tandis que le segment marqueur interrogatif est exclusif aux questions et regroupe les pronoms interrogatifs et les substantifs liés d'une question. Enfin, le segment optionnel contient les figures de style sans informations importantes. La création de ce formalisme a été guidée par deux éléments : le besoin d'avoir un formalisme robuste aux types de documents, et l'utilisation que nous voulons en faire pour l'ajout des relations entre segments. Un corpus d'apprentissage a été annoté selon ce formalisme. A partir de ce corpus, un modèle est

entraîné en utilisant des champs aléatoires conditionnels (CRF). Ce modèle est ensuite utilisé pour annoter automatiquement les questions et les documents.

Les relations entre segments ont pour objectif de représenter les relations entre les différents groupes de mots des questions et des phrases des documents. Si ces relations sont assez éloignées de dépendances syntaxiques traditionnelles, l'idée reste néanmoins assez similaire : représenter l'organisation structurelle des questions et des phrases des documents par le biais de relations typées. Contrairement aux dépendances d'un analyseur tel que XIP [Aït-Mokhtar et al. 2002], le formalisme de nos relations est très simple. En nous appuyant sur le formalisme de segmentation, nous avons définis deux principaux types de relations : groupe nominal, et membres-verbe. Nous ajoutons par ailleurs deux autres types de relation, temporelle et spatiale, qui représentent les relations associées aux segments de temps et de lieu. Ces relations sont identifiées à partir de règles définies manuellement. Comme pour la segmentation, notre formalisme de relations reste relativement simple, de manière à être adaptable sur n'importe quel type de documents.

En nous appuyant sur ce modèle de représentation, nous avons défini un module de réordonnancement. Pour chaque question, l'objectif du module est de réordonner les réponses candidates fournies par Ritel selon un nouveau score. Nous définissons cette tâche comme un calcul de similarité entre un passage contenant une réponse candidate et la question traitée. Le calcul du score de similarité s'appuie sur une distance d'édition, inspiré des travaux présentés dans [Kouylekov & Negri 2010]. La similarité est donc quantifiée en un coût de transformation entre un passage et une question. Plus le coût est faible, plus la similarité entre un passage et une question est élévée. Les réponses sont réordonnées selon ce score. La réponse avec le coût le plus faible est considérée comme étant la meilleure réponse. La distance d'édition est appliquée sur les segments du passage et de la question. Traditionnellement, le calcul d'une distance d'édition s'appuie sur trois types d'opération de transformation : la suppression, l'insertion, et la substitution. Nous avons estimé que ces trois types d'opérations ne permettaient pas de quantifier suffisamment les différences structurelles entre une question et un passage. Ainsi, nous avons introduit un quatrième type d'opération, que nous nommons *rattachement*. Ce type d'opération s'appuie sur les relations identifiées entre les segments. Si un segment substitué du passage n'est pas en relation avec la réponse candidate, alors une opération de rattachement est effectuée, calculant un coût pour rattacher le segment à la réponse candidate.

Le segmenteur a été évalué sur trois corpus différents, correspondant aux différentes modalités de documents que nous voulons traiter : l'écrit journalistique, l'oral, et le web. Les résultats obtenus sont très encourageants, et montrent une certaine robustesse de notre formalisme de segmentation par rapport aux différents types de documents. On observe en effet une baisse très faible des résultats entre l'écrit journalistique (meilleur résultat), et l'oral et le web. On peut observer que certains types de segments sont moins bien détectés. Une évaluation supplémentaire a néanmoins montré que l'impact des erreurs sur le réordonnanceur était assez faible.

Les résultats obtenus par le module de réordonnancement sont plus mitigés. Le réordonnanceur permet une amélioration significative des résultats localement, dans des contextes particuliers d'utilisation. Néanmoins, globalement les résultats ne permettent pas de voir cet apport. Le réordonnanceur

a été évalué sur trois différentes campagnes d'évaluation, chacune correspondant à trois différents types de documents : l'écrit journalistique (type le Monde), l'oral (émissions de radio), et le web. Les résultats sont en déça de ceux obtenus par Ritel, même si la perte est assez faible. Par ailleurs, on n'observe pas de différences importantes entre les différentes modalités. Néanmoins, des évaluations complémentaires ont montré que notre approche était efficace pour traiter des questions composées d'un nombre élevé d'éléments. Les résultats sur cette catégorie de questions sont en effet en hausse par rapport à ceux obtenus par Ritel. Enfin, nous avons évalué l'impact de plusieurs composantes de notre système, principalement les relations entre les segments et les opérations de rattachement. Les résultats ont permis d'observer un impact positif, qui reste néanmoins relativement faible. Un travail supplémentaire est donc nécessaire sur les relations ainsi que les opérations de rattachement.

Ainsi, il est important de noter que les objectifs de notre travail sont atteints en partie. Les objectifs de robustesse sont atteints, aussi bien pour le modèle de représentation que pour le réordonnanceur. La segmentation donne par ailleurs de bons résultats. Si le réordonnanceur demande encore à être amélioré pour apporter une hausse des résultats plus générale pour Ritel, son comportement sur les questions avec un nombre élevé d'éléments est très intéressant. Ces résultats prouvent le potentiel de notre méthode : plus une question comporte d'éléments, et plus le réordonnancement fera appel à deux des composantes de notre travail, les relations et les opérations de rattachement. Enfin, l'approche proposée est adaptable à d'autres systèmes de questions-réponses.

Chapitre 10

Perspectives

Si les conclusions de ce travail sont plutôt positives, et que les objectifs sont remplis en partie, on peut néanmoins observer qu'un certain nombre de points peuvent être améliorés.

Si les résultats de la segmentation sont très positifs, un travail supplémentaire sur le corpus d'apprentissage pourrait être bénéfique. L'augmentation de sa taille peut potentiellement avoir un effet bénéfique sur les résultats. Par ailleurs, il existe certaines inconsistances dans l'annotation qui demandent à être corrigées. Nous avons étudié l'apport de types sémantiques pour les segments par le biais des segments de temps et de lieu. Notre travail a montré la nécessité d'en prévoir d'autres. En effet, si les segments de temps et de lieu permettent d'apporter de l'information supplémentaire, ils sont néamoins insuffisants pour traiter tous les types de questions, même en se limitant aux questions *factuelles*. Des segments de type *personne* ou *quantité* semblent par exemple être un bon point de départ pour améliorer notre formalisme. Ces interrogations nous amènent à un des points négatifs de notre approche pour effectuer la segmentation : chaque modification effectuée sur le formalisme demande de réannoter le corpus d'apprentissage. De ce fait, nous voudrions expérimenter une nouvelle approche : nous n'annoterions que les types de segments principaux dans le corpus d'apprentissage, à savoir les segments nominaux et verbaux. Tous les sous-types sémantiques seraient ensuite ajoutés par le biais de règles écrites manuellement basées sur les types de l'analyseur de Ritel. Si cette approche donne de bons résultats pour les segments de temps et de lieu, alors nous pourrons envisager l'étendre à de nouveaux types.

L'ajout de nouveaux types de segments implique aussi d'avoir de nouvelles relations. Un travail sera ainsi nécessaire pour déterminer la forme des nouvelles relations à ajouter, et leur application dans un cadre nécessitant une certaine robustesse. Par ailleurs, nous ne sommes pas totalement convaincus des apports des relations de temps et de lieu. Une modification de leur comportement semble nécessaire, en particulier pour la prise en compte des bornes de proposition. Ces bornes pourraient être utilisés dans différentes parties du réordonnanceur, notamment les opérations de rattachement.

Les résultats du module de réordonnancement sont mitigés lorsqu'il est appliqué sur n'importe quel type de question. L'une des causes principales supposées est le faible nombre des règles de rattachement. Les analyses ont montré que leur impact était encore trop faible. De ce fait, un travail supplémentaire pour augmenter le nombre de règles semble pertinent pour améliorer les résultats : plus de phénomènes linguistiques seraient ainsi couverts. Une étude approfondie des corpus seraient alors nécessaire.

Nous travaillons aussi actuellement à adapter notre approche à un classifieur SVM. Nous considérons le nombre de questions suffisant pour effectuer cette expérience au moins sur Quaero. L'idée serait de transformer les résultats de nos analyses ainsi que les traitements du réordonnanceur en traits et ainsi entraîner un classifieur SVM.

Enfin, si le travail présenté dans ce document avait pour objectif de traiter les questions factuelles, il pourrait être très intéressant d'étendre notre approche à d'autres types de questions. Le calcul de la similarité entre un passage et une question est inspirée de [Kouylekov & Negri 2010], qui est une méthode d'implication textuelle. De telles approches sont très intéressantes pour répondre aux questions de type *oui/non*. Une adaptation de notre système pour de telles questions pourraient donner de bons résultats.

Publications

G. Bernard, M. Adda-Decker et S. Rosset (2010). 'Etudes des caractéristiques des collections de documents pour les évaluations de système de questions-réponses : mesures de la difficulté'. In *Actes de Journées d'étude sur la parole (JEP 2010)*, Mons, Belgique.

G. Bernard, S. Rosset, M. Adda-Decker et O. Galibert (2010). 'A Question-answer Measure to Investigate QA System Progress'. In *Proceedings of the International Conference on Language Ressources and Evaluation (LREC 2010)*, Valletta, Malta.

G. Bernard, S. Rosset, O. Galibert, E. Bilinski et G. Adda (2009). 'The LIMSI participation to the QAst 2009 track : Experimenting on Answer Scoring'. In *Proceedings of the 10th Workshop of the Cross-Language Evaluation Forum (CLEF 2009)*, Corfu, Greece.

S. Rosset, O. Galibert, G. Bernard, E. Bilinski et G. Adda (2008). 'The LIMSI Multilingual, Multitask QAst System'. In *Proceedings of the 9th Workshop of the Cross-Language Evaluation Forum (CLEF 2008)*, Aarhus, Denmark.

Guillaume Bernard (2008). 'Méthode de réordonnancement de réponses par transformation d'arbres : présentation et analyse des résultats'. In *Actes de Traitement automatique des langues naturelles (TALN 2008)*, Avignon, France.

Guillaume Bernard (2008). 'Réordonnancement de réponses par transformation d'arbres pour un système de question-réponse oral interactif'. In *Proceedings of the 5th French Information Retrieval Conference (CORIA 2008)*, Trégastel, France.

Bibliographie

R. Adams (2006). ‘Textual Entailment Through Extended Lexical Overlap’. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

G. Adda, M. Adda-Decker, J. Gauvain et L. Lamel (1997). ‘Text Normalization and Speech Recognition in French’. In *Proceedings of Eurospeech’97*, vol. 5, pp. 2711–2714, Rhodes, Greece.

G. Adda, J. Mariani, J. Lecomte, P. Paroubek et M. Rajman (1998). ‘The GRACE French Part Of Speech Tagging Evaluation Task’. In *Proceedings of the First International Language Ressources and Evaluation (LREC’98)*.

A. Allauzen et H. Bonneau-Maynard (2008). ‘Training and Evaluation of POS Taggers on the French MULTITAG Corpus’. In *Proceedings of the Sixth International Language Ressources and Evaluation (LREC’08)*, Marrakech, Morocco.

AMI (2005). ‘The AMI meeting corpus’. http ://www.amiproject.org.

Apache (2007). ‘Apache Lucene, An overview’. http ://lucene.apache.org/java/docs/.

C. Ayache, B. Grau et A. Vilnat (2006). ‘EQueR : the French Evaluation campaign of Question-Answering Systems’. In *LREC 2006*, Genoa, Italy.

S. Aït-Mokhtar, J. Chanod et C. Roux (2002). ‘Robustness beyond shallowness : Incremental deep parsing’. In *Natural Language Engineering*, pp. 121–144.

C. Baker, C. Fillmore et J. Lowe (1998). ‘The Berkeley FrameNet project’. In *Proceedings of the COLING-ACL*, pp. 86–99, Montreal, Canada.

T. C. Belle, A. Moffat, I. Witten et J. Zobel (1994). http ://www.ncsi.iisc.ernet.in/raja/netlis/wise/mg/mainmg.html.

L. Bentivogli, I. Dagan, H. Dang et D. Giampiccolo (2009). ‘The Fifth PASCAL Recognizing Textual Entailment Challenge’. In *Text Analysis Conference 2009*, Gaithersburg, USA.

L. Bentivogli, B. Magnini, I. Dagan, H. Dang et D. Giampiccolo (2005). ‘The Fifth PASCAL Recognizing Textual Entailment Challenge’. In *Proceedings of the Fifth PASCAL Recognizing Textual Entailment Challenge*, Gaithersburg, Maryland USA.

A. Berger, R. Caruana, D. Cohn, D. Freitag et V. Mittal (2000). ‘Bridging the lexical chasm : statistical approaches to answer-finding’. In *Proceedings of the 23rd annual internation ACM SIGIR conference on research and develomment in information retrieval*, Athens, Greece.

G. Bernard, S. Rosset, M. Adda-Decker et O. Galibert (2010). ‘A question–answer distance measure to investigate QA system progress’. In *Proceedings of the seventh conference on International Language Ressources and Evaluation (LREC’10)*, Malte, Valetta.

J.-B. Berthelin, G. de Chalendar, F. Elkateb-Gara, O. Ferret, B. Grau, M. Hurault-Plantet, G. Illouz, L. Monceaux et I. Robba (2003). ‘Getting reliable answers by exploiting results from several sources of information’. In *2nd CoLogNET-ElsNET Symposium : Questions and Answers : Theoretical and Applied Perspectives*, Amsterdam.

A. Bies, M. Ferguson, K. Katz et R. MacIntyre (1995). ‘Bracket Guidelines for Tree-bank II Style Penn Treebank Project’.

E. Cabrio, Y. Mehdad, M. Negri, M. Kouylekov et B. Magnini (2009). ‘Recognizing Textual Entailment for Italien EDITS @ EVALITA 2009’. In *Proc. of EVALITA 2009*, Reggio Emilia.

X. Carreras (2007). ‘Experiments with a Higher-Order Projective Dependency Parser’. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, Prague, June.

X. Carreras et L. Màrquez (2005). ‘Introduction to the CoNLL-2005 Shared Task : Semantic Role Labeling’. In *Proceedings of CoNLL-2005*.

E. Charniak (2000). ‘A maximum-inspired-entropy parser’. In *Proceedings of NAACL*, pp. 132–139, Seattle, Washington.

CHIL (2007). ‘The European project CHIL’. http ://chil.server.de.

T. Chklovski et P. Pantel (2004). ‘VERBOCEAN : Mining the Web for Fine-Grained Semantic Verb Relations’. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, Jeju Island, South Korea.

M. Collins (2009). *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University Of Pennsylvania.

P. Comas et J. Turmo (2009). ‘Robust Question Answering for Speech Transcripts : UPC Experience in QAst 2009’. In *Working Notes of CLEF 2009*.

P. R. Comas, J. Turmo et L. Màrquez (2010). ‘Using Dependency Parsing and Machine Learning for Factoid Question Answering on Spoken Documents’. In *Proceedings of the 13th International Conference on Spoken Language Processing (INTERSPEECH 2010)*, Makuhari, Japan.

H. T. Dang, J. Lin et D. Kelly (2006). ‘Overview of the TREC 2006 Question Answering Track’. In *Text Retrieval Conference TREC-15*, pp. 99–116, Gaithersburg, MD, USA.

H. T. Dang, J. Lin et D. Kelly (2007). ‘Overview of the TREC 2007 Question Answering Track’. In *Text Retrieval Conference TREC-15*, Gaithersburg, MD, USA.

J. Eisner (1996). ‘Three New Probabilistic Models for Dependency Parsing : An Exploration’. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, Copenhagen.

C. Fellbaum (1998). *WordNet – An Electronic Lexical Database*.

P. Forner, A. Peñas, I. Alegria, C. Forascu, N. Moreau, P. Osenova, P. Prokopidis, P. Rocha, B. Sacaleanu, R. Sutcliffe et E. T. K. Sang (2008). ‘Overview of the CLEF 2008 Multilingual Question Answering Track’. In *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark.

A. Fowler, B. Hauser, D. Hodges, I. Niles, A. Novischi et J. Stephan (2006). 'Applying COGEX to Recognize Textual Entailment'. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

J. Fukumoto, T. Kato et F. Masui (2002). 'Question Answering Challenge (QAC-1) : Question answering evaluation at NTCIR Workshop 3'. In *Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering*, Tokyo, Japan.

J. Fukumoto, T. Kato et F. Masui (2004). 'Question Answering Challenge for Five Ranked Answers and List Answers - Overview of NTCIR4 QAC2 Subtask 1 and 2'. In *Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization*, Tokyo, Japan.

J. Fukumoto, T. Kato, F. Masui et T. Mori (2007). 'An Overview of the 4th Question Answering Challenge (QAC-4) at NTCIR Workshop 6'. In *Proceedings of the Sixth NTCIR Workshop Meeting on Evaluation of Information Access Technologies : Information Retrieval, Question Answering, and Cross-Lingual Information Access*, Tokyo, Japan.

O. Galibert (2009). *Approches et méthodologies pour la réponse automatique à des questions adaptées à un cadre interactif en domaine ouvert*. Thèse de doctorat, Université Paris Sud, Orsay.

S. Galliano, E. Geoffrois, G. Gravier, J. Bonastre, D. Mostefa et K. Choukri (2006). 'Corpus description of the ESTER Evaluation Campaign for the Rich Transcription of French Broadcast News'. In *Proceedings of LREC'06*, Genoa.

D. Giampiccolo, P. Forner, A. Peñas, C. Ayache, D. Cristea, V. Jijkoun, P. Osenova, P. Rocha, B. Sacaleanu et R. Sutcliffe (2007). 'Overview of the CLEF 2007 Multilingual Question Answering Track'. In *Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary.

D. Gildea et J. Hockenmaier (2003). 'Identifying Semantic Roles Using Combinatory Categorial Grammar'. In *Proceedings of the EMNLP*, Sapporo, Japan.

L. Gillard, L. Sitbon, P. Bellot et M. El-Beze (2006). 'Dernières évolutions de SQuaLIA, le système de Questions/Réponses du LIA'. *Traitement Automatique des Langues* **46**(3/2005).

A. Grappy et B. Grau (2010). 'Validation du type de la réponse dans un système de questions réponses'. In *? ? ?*

B. Grau, G. Illouz, L. Monceaux, P. Paroubek, O. Pons, I. Robba et A. Vilnat (2005). 'FRASQUES, le système du groupe LIR, LIMSI'. In *Atelier EQueR de TALN 05*.

K. Hacioglu (2004). 'A Lightweight Semantic Chunking Model Based On Tagging'. In *Proceedings of HLT/NAACL*, Boston, MA.

J. Haji, M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Stepanek, M. Surdeanu, N. Xue et Y. Zhang (2009). 'The CoNLL-2009 Shared Task ; Syntactic and Semantic Dependencies in Multiple Language Learning (CoNLL 2009) : Shared Task'. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009) : Shared Task*, Boulder, Colorado.

S. Harabagiu et A. Hickl (2006). 'Methods for Using Textual Entailment in Open-Domain Question Answering'. In *Proceedings of COLING-ACL*.

S. M. Harabagiu, G. A. Miller et D. I. Moldovan (1999). ‘Wordnet 2 - a morphologically and semantically enhanced resource’. In *SIGLEX Workshop On Standardizing Lexical Resources*, pp. 1–8.

A. Hick et J. Bensley (2007). ‘A Discourse Commitment-based Framework for Recognizing Textual Entailment’. In *Proceedings of the ACL 2007 Workshop on Paraphrasing and Textual Entailment*.

A. Hickl, J. Bensley, J. Williams, K. Roberts, B. Rink et Y. Shi (2006a). ‘Recognizing Textual Entailment with LCC’s Groundhog System’. In *The Second PASCAL Recognizing Textual Entailment Challenge*, Venice, Italy.

A. Hickl, J. Williams, J. Bensley, K. Roberts, B. Rink et Y. Shi (2006b). ‘Recognizing Textual Entailment with LCC’s Groundhog system’. Venice, Italy.

A. Hickl, J. Williams, J. Bensley, K. Roberts, Y. Shi et B. Rink (2006c). ‘Question Answering with LCC’s CHAUCER at TREC 2006’. In *The 15th TREC Conference (TREC 2006)*.

A. Hickl, J. Williams, J. Bensley, K. Roberts, Y. Shi, B. Rink et T. Jungen (2007). ‘Question Answering with LCC’s CHAUCER-2 at TREC 2007’. In *TREC*.

http ://www.quaero.org/ (2008). ‘Le programme Quaero’.

A. Ittycheriah et S. Roukos (2002). ‘IBM’s statistical Question-Answering system - TREC-11’. In *Proceedings of the TREC 2002 Conference*.

C. Jacquemin (2004). ‘A symbolic and surgical acquisition of terms through variation’. pp. 425–438. Springer-Verlag.

T. Kato, J. Fukumoto et F. Masui (2004). ‘Question Answering Challenge for Information Access Dialogue - Overview of NTCIR4 QAC2 Subtask 3’. In *Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization*, Tokyo, Japan.

T. Kato, J. Fukumoto et F. Masui (2005). ‘An Overview of NTCIR-5 QAC3’. In *Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies : Information Retrieval, Question Answering and Cross-Lingual Information Access*, Tokyo, Japan.

T. Kato, J. Fukumoto, F. Masui et N. Kando (2006). ‘WoZ Simulation of Interactive Question Answering’. In *NAACL Workshop on Interactive Question Answering*, New York, USA.

M. Kouylekov et M. Negri (2010). ‘An Open-Source Package for Recognizing Textual Entailment’. In *Proceedings of the ACL 2010 System Demonstrations*, Uppsala, Sweden.

T. Kudoh (2007). ‘CRF++’. http ://crfpp.sourceforge.net.

T. Kudoh et Y. Matsumoto (2000). ‘Use Of Support Vector Learning for Chunk Identification’. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal.

J. Kürsten, H. Kundisch et M. Eibl (2008). ‘QA Extension for Xtrieval : Contribution to the QAst track’. In *Working Notes of CLEF 2008 Workshop*, Aarhus, Denmark.

J. Lafferty, A. McCallum et F. Pereira (2001). ‘Conditional Random Fields : Probabilistic models for segmenting and labeling sequence data’. In *Proceedings of ICML*, Williamstown, USA.

L. Lamel, S. Rosset, J. Gauvain, S. Bennacef, M. Garnier-Rizet et B. Prouts (2000). ‘The LIMSI ARISE System’. *Speech Communication* **31**(4) :339–354.

D. Laurent, P. Séguéla et S. Nègre (2006). 'Cross Lingual Question Answering using QRISTAL for CLEF 2006'. In *Working Notes for the CLEF 2006 Workshop*, Alicante, Spain.

A. Ligozat (2005). 'Apport de l'analyse syntaxique des phrases dans un système de questions-réponses'. In *TAL. Volume 46*.

A.-L. Ligozat (2006). *Exploitation et fusion de connaissances locales pour la recherche d'informations précises*. Ph.D. thesis, Université Paris-Sud 11, Orsay, France.

D. Lin (1998). 'Dependency-based evaluation of MINIPAR'. In *In Workshop of the Evaluation of Parsing Systems*, Granada, Spain.

D. Lin (2000). 'Word Similarities Dictionnaries'. http ://webdocs.cs.ualberta.ca/ lindek/downloads.htm.

X. Lluis, S. Bott et L. Marquez (2009). 'A Second-Order Joint Eisner Model for Syntactic and Semantic Dependency Parsing'. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009) : Shared Task*, Boulder,Colorado.

B. Magnini, D. Giampiccolo, P. Forner, C. Ayache, P. Osenova, A. Peñas, V. Jijkoun, B. Sacaleanu, P. Rocha et R. Sutcliffe (2006). 'Overview of the CLEF 2006 Multilingual Question Answering Track'. In *Working Notes for the CLEF 2006 Workshop*, Alicante, Spain.

B. Magnini, S. Romagnoli, A. Vallin, J. Herrera, A. Peñas, V. Peinado, F. Verdejo et M. de Rijke (2003). 'The Multiple Language Question Answering Track at CLEF 2003'. In *Working Notes for the CLEF 2003 Workshop*, Trondheim, Norway.

B. Magnini, A. Vallin, C. Ayache, G. Erbach, A. Peñas, M. de Rijke, P. Rocha, K. Simov et R. Sutcliffe (2004). 'Overview of the CLEF 2004 Multilingual Question Answering Track'. In *Working Notes for the CLEF 2004 Workshop*, Bath, UK.

Y. Mehdad (2009). 'Automatic Cost Estimation for Tree Edit Distances Using Particle Swarm Optimization'. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, Suntec, Singapore.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young et R. Grishman (2004). 'The NomBank Project : An Interim Report'. In *HLT-NAACL 2004 Workshop : Frontiers in Corpus Annotation*, Boston, USA.

T. Mitamura, E. Nyberg, H. Shima, T. Kato, T. Mori, C.-Y. Lin, R. Song, C.-J. Lin, T. Sakai, D. Ji et N. Kando (2008). 'Overview of the NTCIR-7 ACLIA Tasks : Advanced Cross-Lingual Information Access'. In *Proceedings of the Seventh NTCIR Workshop Meeting on Evaluation of Information Access Technologies : Information Retrieval, Question Answering, and Cross-Lingual Information Access*, Tokyo, Japan.

D. Moldovan, C. Clark, S. Harabagiu et S. Maiorano (2003). 'COGEX : A Logic Prover for Question Answering'. In *In Proceedings of HLT-NAACL 2003*, Edmonton, Canada.

D. Molla, S. Cassidy et M. van Zaanen (2007). 'AnswerFinder at QAst 2007 : Named Entity Recognition for QA on Speech Transcripts'. In *Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary.

A. Moschitti (2006). 'Efficient convolution kernels for dependency and constituent syntactic trees'. In *Proceedings of ECML*.

A. Moschitti, B. Coppola, A. Giuglea et R. Basili (2005). 'Hierarchical semantic role labeling'. In *Proceedings of the CoNLL 2005 shared task*, Ann Arbor, Michigan.

A. Moschitti et S. Quarteroni (2010). 'Linguistic kernels for answer re-ranking in question answering systems'. In *Information Processing and Management*.

A. Moschitti, S. Quarteroni, R. Basili et S. Manandhar (2007). 'Exploiting syntactic and shallow semantic kernels for question/answer classification'. In *Proceedings of ACL*, Prague, Czech Republic.

M. Negri, B. Magnini et M. Kouylekov (2008). 'Detecting Expected Answer Relations through Textual Entailment'. In *Proceedings of CICling 2008*, Haifa, Israel.

M. Palmer, D. Gildea et P. Kingsbury (2005). 'The Proposition Bank : A Corpus Annotated with Semantic Roles'. In *Computational Linguistics Journal*.

M. Pardiño, J. Gómez, H. Llorens, R. Muñoz-Terol, B. Navarro-Colorado, E. Saquete, P. Martínez-Barco, P. Moreda et M. Palomar (2008). 'Adapting IBQAS to work with Text Transcriptions in QAst Task : IBQAst'. In *Working Notes of CLEF 2008 Workshop*, Aarhus, Denmark.

P. Paroubek, I. Robba et A. Vilnat (2008a). *L'évaluation technologique dans le domaine du traitement automatique de la langue : l'expérience du programme Technolangue*, chap. EASY : la campagne d'évaluation des analyseurs syntaxiques. Editions Hermès, Paris.

P. Paroubek, I. Robba, A. Vilnat et C. Ayache (2008b). 'EASY, Evaluation of Parsers of French : what are the results ?'. In *Proceedings of the Sixth International Language Ressources and Evaluation (LREC'08)*, Marrakech, Morocco.

C. Peters et M. Braschler (2001). 'European Research Letter : cross-language system evaluation : the CLEF campaigns'. *J. Am. Soc. Inf. Sci. Technol.* **52**(12) :1067–1072.

L. Plamondon et L. Kosseim (2003). 'Le web et la question-réponse : transformer une question en réponse'. In *Journées francophones de la toile (JFT 2003)*, pp. 225–234, Tours, France.

S. Pradhan, W. Ward, K. Hacioglu et J. Martin (2005). 'Semantic Role Labelling Using Different Syntactic Views'. In *In ACL 2005*, pp. 581–588.

S. Quarteroni et S. Manandhar (2009). 'Designing an interactive open domain question answering system'. In *Natural Language Engineering*.

S. Quarteroni et A. Moschitti (2010). 'A Comprehensive Ressource to Evaluate Complex Open Domain Question Answering'. In *Proceedings of the seventh conference on International Language Ressources and Evaluation (LREC'10)*, Valletta,Malta.

L. Quintard (2009). 'Overview of the QUAERO 2008 monolingual question-answering track'. http ://www.lne.eu/en/r_and_d/quaero.asp.

L. Quintard, O. Galibert, G. Adda, B. Grau, D. Laurent, V. Moriceau, S. Rosset, X. Tannier et A. Vilnat (2010). 'Question Answering on Web Data : The QA Evaluation in Quaero'. In *Proceedings of the Seventh conference on International Language Ressources and Evaluation (LREC'10)*, Valletta, Malta.

A. Reyes-Barragan, L. Villasenor-Pineda et M. M. y Gomez (2009). 'INAOE at QAST 2009 : Evaluating the usefulness of a phonetic codification of transcriptions'. In *Working Notes for the CLEF 2009 Workshop*.

S. Rosset (2008). ‘Systèmes de dialogue (oral) homme-machine : du domaine limité au domaine ouvert’. Mémoire d’Habilitation à Diriger les Recherches.

S. Rosset, O. Galibert, G. Bernard, E. Bilinski et G. Adda (2008). ‘The LIMSI participation to the QAst track’. In *Working Notes of CLEF 2008 Workshop*, Aarhus, Denmark.

S. Rosset, O. Galibert, G. Illouz et A. Max (2006). ‘Interaction et recherche d’informations : le projet RITEL’. *Traitement Automatique des Langues* **46**(3/2005).

J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson et J. Scheffczyk (2006). ‘FrameNet II : Extended Theory and Practice’. http ://framenet.icsi.berkeley.edu/book/book.html.

S. J. Russel et P. Norvig (2003). *Artificial Intelligence : A Modern Approach*. Upper Saddle River, New Jersey.

B. Sagot, K. Fort et F. Venant (2009). ‘Extension et couplage de ressources syntaxiques et sémantiques sur les adverbes’. In *Linguisticae Investigationes*, vol. 32(2), pp. 305–315.

T. Sakai, Y. Saito, Y. Ichimura, M. Koyama, T. Kokubu et T. Manabe (2004). ‘Askmi : A Japanese Question Answering System based on Semantic Role Analysis’. In *Proceedings of RIAO 2004*, Avignon, France.

E. F. T. K. Sang (2000). ‘Noun Phrase Recognition by System Combination’. In *Proceedings of NAACL-2000*.

Y. Sasaki (2005). ‘Question Answering as Question-Biased Term Extraction : A New Approach toward Multilingual QA’. In *Proceedings of the 43rd Annual Meeting of the ACL*, Ann Arbor.

Y. Sasaki, H.-H. Chen, K. hua Chen et C.-J. Lin (2005). ‘Overview of the NTCIR-5 Cross-Lingual Question Answering Task (CLQA1)’. In *Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies : Information Retrieval, Question Answering and Cross-Lingual Information Access*, Tokyo, Japan.

Y. Sasaki, H. Isozaki, J. Suzuki, K. Kokuryou, T. Hirao, H. Kazawa et E. Maeda (2004). ‘SAIQA-II : A Trainable Japanese QA System with SVM’. In *IPSJ Journal*, vol. vol. 45, pp. 635–646.

Y. Sasaki, C.-J. Lin, K. hua Chen et H.-H. Chen (2007). ‘Overview of the NTCIR-6 Cross-Lingual Question Answering (CLQA) Task’. In *Proceedings of the Sixth NTCIR Workshop Meeting on Evaluation of Information Access Technologies : Information Retrieval, Question Answering, and Cross-Lingual Information Access*, Tokyo, Japan.

H. Schmid (1994). ‘Probabilistic Part-of-Speech Tagging Using Decision Trees’. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

S. Sekine, K. Sudo, Y. Shinyama, C. Nobata, K. Uchimoto et H. Isahara (2002). ‘NYU/CRL QA system, QAC question analysis and CRL QA data’. In *Working Notes of NTCIR Workshop 3*.

F. Sha et F. Pereira (2003). ‘Shallow Parsing with conditional random fields’. Edmonton, Canada.

D. Shen et M. Lapata (2007). ‘Using Semantic Roles to Improve Question Answering’. pp. 12–21.

S. Stenchikova, D. Hakkani-Tur et G. Tur (2006). ‘QASR : Question Answering Using Semantic Roles for Speech Interface’. In *INTERSPEECH 2006 - ICSLP*, Pittsburgh, Pennsylvania.

X. Tannier et V. Moriceau (2010). ‘FIDJI : Web Question-Answering at Quaero 2009’. In *Proceedings of the seventh conference on International Language Ressources and Evaluation (LREC’10)*, Valletta,Malta.

M. Tatu et D. Moldovan (2006). ‘A logic-based semantic approach to recognizing textual entailment’. In *Proceedings of the COLING/ACL on Main conference poster sessions*.

TC-Star (2004-2008). http ://www.tc-star.org.

E. F. Tjong, K. Sang et S. Buchholz (2000). ‘Introduction to the CoNLL-2000 Shared Task : Chunking’. Lisbon, Portugal.

D. Toney, S. Rosset, A. Max, O. Galibert et E. Bilinski (2008). ‘An Evaluation of Spoken and Textual Interaction in the RITEL Interactive Question Answering System’. In E. L. R. A. (ELRA) (ed.), *Proceedings of the Sixth International Language Resources and Evaluation (LREC’08)*, Marrakech, Morocco.

J. Turmo, P. Comas, C. Ayache, D. Mostefa, S. Rosset et L. Lamel (2007). ‘Overview of the QAST 2007’. In *Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary.

J. Turmo, P. Comas, S. Rosset, O. Galibert, N. Moreau, D. Mostefa, P. Rosso et D. Buscaldi (2009). ‘Overview of QAst 2009’. In *Working Notes for the CLEF 2009 Workshop*, Corfu, Grèce.

J. Turmo, P. Comas, S. Rosset, L. Lamel, N. Moreau et D. Mostefa (2008). ‘Overview of QAst 2008’. In *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark.

A. Vallin, D. Giampiccolo, L. Aunimo, C. Ayache, P. Osenova, A. Peñas, M. de Rijke, B. Sacaleanu, D. Santos et R. Sutcliffe (2005). ‘Overview of the CLEF 2005 Multilingual Question Answering Track’. In *Working Notes for the CLEF 2005 Workshop*, Vienna, Austria.

J. Vergne (1999). ‘Etude et modélisation de la syntaxe des langues à l’aide de l’ordinateur - Analyse syntaxique automatique non combinatoire - Synthèse et Résultats’. Caen, France.

E. M. Voorhees (2000). ‘Overview of the TREC-9 Question Answering Track’. In *Text Retrieval Conference TREC-9*, pp. 71–80, Gaithersburg, MD, USA.

E. M. Voorhees (2002). ‘Overview of the TREC 2002 Question Answering Track’. In *Text Retrieval Conference TREC-11*, Gaithersburg, MD, USA.

E. M. Voorhees (2003). ‘Overview of the TREC 2003 Question Answering Track’. In *Text Retrieval Conference TREC-12*, pp. 54–68, Gaithersburg, MD, USA.

E. M. Voorhees (2004). ‘Overview of the TREC 2004 Question Answering Track’. In *Text Retrieval Conference TREC-13*, Gaithersburg, MD, USA.

E. M. Voorhees et H. T. Dang (2005). ‘Overview of the TREC 2005 Question Answering Track’. In *Text Retrieval Conference TREC-14*, Gaithersburg, MD, USA.

E. M. Voorhees et D. K. Harman (2005). *TREC : Experiment and Evaluation in Information Retrieval*. Digital Libraries and Electronic Publishing.

E. M. Voorhees et D. M. Tice (1999). ‘The TREC-8 Question Answering Track Report’. In *Text Retrieval Conference TREC-8*, pp. 77–82, Gaithersburg, MD, USA.

E. M. Voorhees et D. M. Tice (2000). ‘Implementing a Question Answering Evaluation’. In *In Proceedings of LREC’2000 Workshop on Using Evaluation within HLT Programs : Results and Trends*.

E. M. Voorhees et D. M. Tice (2001). ‘Overview of the TREC 2001 Question Answering Track’. In *Text Retrieval Conference TREC-10*, pp. 42–51, Gaithersburg, MD, USA.

P. Vossen (1998). ‘EuroWordNet A Multilingual Database with Lexical Semantic Networks’.

M. Walker, A. Rudnicky, J. Aberdeen, E. Bratt, J. Garofolo, H. Hastie, A. Le, B. Pellom, A. Potamianos, R. Passonneau, R. Prasad, S. Roukos, G. Sanders, S. Seneff et D. Stallard (2002). ‘DARPA Communicator Evaluation : Progress from 2000 to 2001’. In *ICSLP’02*, Denver, EU.

E. Whittaker, J. Novak, M. Heie et S. Furui (2007). ‘CLEF2007 Question Answering Experiments at Tokyo Institute of Technology’. In *Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary.

K. Zhang et D. Shasha (1990). ‘Fast Algorithm for the Unit Cost Editing Distance Between Trees’. In *Journal Of Algorithms - vol. 11*.

Z. Zheng (2002). ‘AnswerBus Question Answering System’.