

Repérage et typage d'expressions temporelles pour l'annotation sémantique automatique de pages Web - Application au e-tourisme

Stéphanie Weiser

▶ **To cite this version:**

Stéphanie Weiser. Repérage et typage d'expressions temporelles pour l'annotation sémantique automatique de pages Web - Application au e-tourisme. Linguistique. Université de Nanterre - Paris X, 2010. Français. NNT : . tel-00530785

HAL Id: tel-00530785
https://theses.hal.science/tel-00530785

Submitted on 29 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Paris Ouest Nanterre La Défense
École doctorale 139 – Connaissance, langage, modélisation
Laboratoire MoDyCo – UMR 7114

THÈSE DE DOCTORAT

DISCIPLINE : SCIENCES DU LANGAGE – TRAITEMENT AUTOMATIQUE DES LANGUES

Repérage et typage d'expressions temporelles pour l'annotation sémantique automatique de pages Web

Application au e-tourisme

PRÉSENTÉE PAR STÉPHANIE WEISER

SOUS LA DIRECTION DE JEAN-LUC MINEL ET PHILIPPE LAUBLET

Juin 2010

Jury :

Florence Amardeilh
Maître de Conférences associé, Université Paris Ouest Nanterre la Défense.

Delphine Battistelli
Maître de Conférences (HDR), Université Paris IV-Sorbonne.

Cédrick Fairon
Professeur des Universités, Université Catholique de Louvain-la-Neuve, rapporteur.

Éric Laporte
Professeur des Universités, Université Paris-Est Marne-la-Vallée, rapporteur.

Philippe Laublet
Maître de Conférences, Université Paris IV-Sorbonne, co-directeur.

Jean-Luc Minel
Professeur des Universités, Université Paris Ouest Nanterre la Défense, directeur.

REMERCIEMENTS

Le plus souvent, c'est d'une rencontre que naît l'idée d'un sujet de thèse. En ce qui me concerne, c'est d'une annonce de sujet qu'est née une rencontre, celle qui m'a permis de faire cette thèse, à MoDyCo, encadrée par Jean-Luc Minel et Philippe Laublet, dans de si bonnes conditions. Je tiens ainsi à les remercier sincèrement : Jean-Luc, pour sa disponibilité, ses conseils et la confiance qu'il m'a accordée ; Philippe, pour ses critiques toujours constructives, sa réactivité et ses relectures rigoureuses.

Je remercie Cédrick Fairon, qui, après m'avoir mis sur la voie de l'extraction d'information en m'accueillant en tant que stagiaire au Cental, a accepté d'être rapporteur de cette thèse. Je suis également reconnaissante à Éric Laporte d'avoir, lui-aussi, accepté d'être rapporteur, et à Delphine Battistelli d'avoir accepté de faire partie de mon jury. Merci à Florence Amardeilh qui, après m'avoir souvent conseillée, a également accepté de faire partie de mon jury.

Je tiens à remercier toutes les personnes grâce à qui cette thèse s'est bien déroulée. Pour le contexte scientifique, merci à tous les participants du projet Eiffel pour les échanges enrichissants, en particulier à l'équipe de Mondeca (Florence Amardeilh, Bernard Vatant, Olivier Carloni, Charles Teissèdre, Martin Coste), et à Jérôme Mainka, d'Antidot, pour sa disponibilité. Merci aussi à l'équipe de l'Inco, et en particulier Dina Wonserver, de m'avoir accueillie en Uruguay. Je remercie aussi les participants du projet TramedWeb, notamment Isabelle Garron et Javier Couto pour les nombreuses discussions.

Pour le cadre de travail, merci aux membres du laboratoire MoDyCo : les doctorants et jeunes chercheurs qui m'ont permis de m'intégrer quand je suis arrivée, et en particulier Céline Vaguer ; et l'équipe (plus ou moins) actuelle qui contribue à rendre ce cadre agréable, en particulier Xavier, Christophe, Romain, Christian, Sophie, Julie, Géraldine, Caroline, Dina.

Je suis très reconnaissante aux personnes qui ont effectivement contribué à cette thèse : Karma Hojeij, Michel Weiser et Sofia Kerrar pour leur relecture ; Florence Amardeilh pour les règles OPAL et les nombreuses explications ; Florence Weiser pour les schémas et traitements d'images, Romain Loth pour ses scripts et conseils et Julie Glikman d'avoir évalué Adetoa.

Au cours de cette thèse j'ai rencontré ou gardé des liens avec d'autres linguistes et talistes, doctorants ou non : merci à l'équipe de Paris VII, en particulier André Bittar, à celles de Marne-la-Vallée, de Louvain-la-Neuve et aux linguistes de soirées.

Enfin, je remercie chaleureusement mes parents qui m'ont toujours encouragée dans mes études. Merci aussi à mes sœurs et mes amis, qui, si certains commencent tout juste, maintenant que c'est terminé, à comprendre ce que j'ai bien pu faire pendant ces années, ont toujours été présents.

Je ne saurais comment remercier Arnaud pour son immense patience, son soutien, sa disponibilité quotidienne et l'aide qu'il a su m'apporter à tous les niveaux : intellectuel, psychologique, technique... Merci aussi à Uranium pour ses câlins.

Cette thèse n'aurait pu avoir lieu sans le financement de l'ANR pour le projet Eiffel.

RÉSUMÉ

Cette thèse présente Adetoa, système dédié au repérage et à l'annotation sémantique automatique d'expressions temporelles dans des pages Web pour une application de e-tourisme. La réalisation de ce système de TAL s'appuie sur une étude linguistique détaillée menée à partir d'une réflexion générale sur l'expression de la temporalité dans ce type de textes. Cette étude, réalisée sur des cas réels, a permis de mettre en évidence la complexité des formes linguistiques ayant une double spécificité : elles se trouvent sur des pages Web et sont propres au domaine du tourisme. Présentée dans les premiers chapitres, elle est à la base d'Adetoa qui s'intègre dans la plateforme du projet Eiffel pour laquelle les modèles ontologiques et les langages du Web Sémantique pour la représentation des connaissances et la recherche sémantique d'information sont utilisés.

Sur un plan linguistique, les contenus ont des particularités propres : les informations temporelles apparaissent rarement dans un texte rédigé, la syntaxe du français n'est pas toujours respectée et il y a peu de prédications. Leur placement dans les pages Web, organisées de manière fort variée, ne présente aucune régularité, rendant difficile voire parfois impossible l'automatisation de leur analyse, comme l'a montré l'analyse sémiotique des pages (par opposition par exemple avec les guides touristiques papiers).

Le développement d'Adetoa s'est nourri de ces études théoriques. L'analyse linguistique a permis de construire un ensemble important de transducteurs (avec Unitex) pour les tâches de repérage et d'annotation des expressions temporelles, ce qui constitue une ressource pouvant être généralisée. De plus, d'autres informations du domaine touristique sont repérées : les objets du tourisme et les adresses. Des transducteurs de liage permettent de grouper toutes les informations concernant une même offre touristique.

Un schéma d'annotation a été mis au point. Il est lié à une ontologie du tourisme, mais n'en est pas un calque direct car sa finalité est de rester au plus près des expressions linguistiques de manière à les caractériser finement. Pour l'intégration d'Adetoa au sein de la chaîne de traitement d'Eiffel, des règles de transformations permettant de faire le pont entre les expressions annotées et les données à stocker dans la base de connaissance ont été élaborées. Dans le cadre de cette thèse, parallèlement à ces développements, l'ontologie du projet a été adaptée de manière à ce que les données annotées puissent prendre place dans la base de connaissance qui lui correspond.

L'évaluation d'Adetoa, présentée dans le dernier chapitre, a montré des résultats satisfaisants aussi bien d'un point de vue théorique que pour cette application industrielle.

Mots-clés : Extraction automatique d'information, ontologie, schéma d'annotation, expressions temporelles, e-tourisme, transducteurs.

ABSTRACT

Extraction and mark-up of temporal expressions for automatic semantic annotation of Web pages

Application to E-tourism

This thesis presents Adetoa, a system designed to automatically locate temporal expressions in Web pages and tag them with semantic annotations, in the field of e-tourism. The implementation of this NLP system is based on a detailed linguistic study which is grounded in more theoretical considerations about temporality in this type of text. This study, carried out on real instances, revealed that the linguistic expressions are complex and display specific features attributable to two main parameters : the text type (Web pages) and the domain (tourism). The early chapters of the thesis detail this study, which subsequently led to the development of Adetoa, now part of the Eiffel project platform. In this platform, ontological models and Semantic Web languages are used for knowledge representation and semantic information retrieval.

On a linguistic level, the data have their own particular characteristics: temporal expressions rarely occur in passages of continuous text, French syntax is not always followed and there are only a few predications. The position of the temporal expressions on the Web pages, which are organised in various ways, does not show any regularity. An automatic analysis of their structure is therefore difficult or even sometimes impossible, in contrast to paper tourist guides, as is illustrated by a semiotic study.

The implementation of Adetoa is based on these theoretical studies. The linguistic analysis has led to the development of a large number of transducers (under Unitex) for the tasks of temporal expressions extraction and mark-up. The transducers can be regarded as a generally applicable resource. Other tourist information is also extracted, such as tourist objects and addresses. Linking transducers have been developed to group all the information concerning one tourist destination.

An annotation scheme has been devised. It is based on a tourism ontology but is not a direct replica, thus enabling the expressions to be accurately characterized on a linguistic level. To integrate Adetoa in the processing chain of the Eiffel project, transformation rules have been written, allowing the annotated expressions to map the knowledge base. In addition to these developments, this doctoral research has also involved making adaptations to the project ontology, so that the information can more easily be included in the corresponding knowledge base.

The evaluation of Adetoa, which is detailed in the last chapter, showed satisfying results, both on a theoretical level and for industrial purposes.

Keywords: Automatic information extraction, ontology, annotation scheme, temporal expressions, e-tourism, transducers.

Table des Matières

Remerciements....2
Résumé....3
Abstract....4
Index des Figures....9
Index des Tableaux....12
Introduction....13

Première Partie
Présentation Générale des Données et Objets d'Étude....18

Chapitre 1. Objets et Problématiques....19
Introduction....19
1. Les expressions temporelles....19
1.1. Quelques généralités sur la formulation du temps....19
1.2. Les expressions temporelles liées au domaine du tourisme....21
1.3. Étude des expressions temporelles dans les pages Web touristiques – mise en relation avec l'affiche....22
2. Pages Web touristiques étudiées....26
2.1. Comparaison avec des guides papier....27
2.2. Comment nomme-t-on dans une page Web ?....33
2.3. La théorie sémiotique et son application aux pages Web....33
2.4. Éléments d'une page Web....35
Conclusion....38

Chapitre 2. Étude Linguistique....40
Introduction....40
1. Caractérisation et classification des expressions....40
1.1. Expressions temporelles....40
1.1.1. « Office de tourisme / mairie / magasin »....41
1.1.2. « Hébergements »....42
1.1.3. « Restaurants »....42
1.1.4. « Manifestations touristiques »....43
1.1.5. Pages mixtes....43
1.1.6. Bilan....44
1.2. Expressions spatiales....45
1.3. Objets touristiques....46
2. Inventaire des problèmes d'interprétation de certaines expressions temporelles....48
2.1. Étude linguistique....48
2.2. Solutions envisagées pour Eiffel....51
2.2.1. Les dates seules....52
2.2.2. Les zeugmes....52
2.2.3. La distribution....52
2.2.4. Les exceptions....52
2.2.5. Le flou....52
Conclusion....53

Deuxième Partie
Manipulation des Données et Représentation....54

Chapitre 3. Traitement Automatique de Pages Web Touristiques....55
Introduction....55
1. État de l'art autour de l'extraction d'information....55
1.1. Extraction automatique d'information – généralités....56
1.1.1. Extraction et entités nommées....56
1.1.2. Extraction automatique d'informations temporelles....56
1.1.3. Principales méthodes d'extraction d'information....57
1.2. Extraction automatique d'information – dans des pages Web....57
2. Pages Web touristiques....59
2.1. Spécificités du corpus Eiffel....59
2.1.1. Crawl des pages Web....60
2.1.2. Tous types de pages touristiques, sans régularité....60
2.1.3. Spécificité technique : passage au XML....61
2.2. Exemples de pages traitées par Adetoa....61
2.2.1. Pages présentant un objet....62
2.2.2. Pages présentant plusieurs objets....65
2.3. Difficultés liées à la conception des pages....67
2.3.1. Difficultés structurelles....67
2.3.2. Difficultés liées au contenu des pages....74
Conclusion....84

Chapitre 4. Ontologie et Schéma d'Annotation....88
Introduction....88
1. État de l'art autour des ontologies et de l'annotation sémantique....89
1.1. Ontologies – ontologie et autres RTO....89
1.2. Rôle d'une ontologie pour l'annotation sémantique....92
2. L'ontologie Eiffel : le pivot du projet....93
2.1. Structure générale de l'ontologie d'Eiffel....93
2.2. Représentation du temps dans l'ontologie Eiffel....94
3. Schéma d'annotation....95
3.1. Choix du format....97
3.2. Mise au point du schéma....97
3.2.1. Annotation des expressions temporelles....98
3.2.2. Annotation des expressions spatiales....104
3.2.3. Annotation des objets....104
3.3. Retour sur l'ontologie....107
3.3.1. Ontologie – volet temporel V1....107
3.3.2. Ontologie – volet temporel V2....108
3.3.3. Propositions de modèles....109
Conclusion....110

Troisième Partie
Implémentation et Évaluation....112

Chapitre 5. Implémentation d'Adetoa....113
Introduction....113
1. Unitex....115
1.1. Principales caractéristiques utiles pour Eiffel et Adetoa....115
1.1.1. Éditeur visuel – expressivité....115
1.1.2. Ressources linguistiques....115
1.1.3. Logiciel Libre....115
1.1.4. Multi-plateforme....116
1.2. Fonctionnement et principes généraux....116

1.2.1. Grammaires....116
1.2.2. Exemples de graphes....117
2. Infrastructure JAVA....119
2.1. Extraction des données textuelles....121
2.2. Deux appels aux programmes Unitex....121
2.3. Module de transformations....122
2.4. Format des données....122
3. Développement des ressources....122
3.1. Transducteurs....123
3.1.1. Combinatoire....123
3.1.2. Typage – annotation....126
3.1.3. Ne pas repérer....129
3.2. Dictionnaire....131
3.3. Transformations sur les informations temporelles....131
3.3.1. Transformations dans les transducteurs....132
3.3.2. Transformations en JAVA avec XPath....132
Conclusion....137

Chapitre 6. Exploitation des Données Annotées dans une Chaîne de Traitement 138
Introduction....138
1. Mapping....139
1.1. Quel libellé choisir pour les différentes ressources touristiques ?....140
1.1.1. Ressources touristiques ayant un nom propre singulier....141
1.1.2. Ressources touristiques n'ayant pas de nom propre....143
1.2. Règles d'acquisition des connaissances....144
1.2.1. Création des instances d'offres touristiques à partir des annotations....144
1.2.2. Création des instances de périodes....147
1.2.3. Création des associations reliant les périodes aux offres touristiques....147
2. Consolidation....147
2.1. Fusionner – dédoublonner....148
2.2. Contrôle-qualité....148
2.3. Inférence....148
2.3.1. Enrichissement des données à l'aide de connaissances du domaine....149
2.3.2. Inférences à l'aide d'une modélisation possibiliste....152
3. Stockage....153
3.1. Saturation ?....153
3.1.1. Intervalles....154
3.1.2. Groupes de jours....154
3.1.3. Solutions adoptées dans Eiffel....155
3.2. Sérialisation....155
4. Interface....155
4.1. Critères de recherche....156
4.2. Degré de certitude et donnée textuelle....156
Conclusion....156

Chapitre 7. Expérimentation et Évaluation....158
Introduction....158
1. Principales méthodes d'évaluation....159
1.1. Rappel et précision....159
1.2. Autres mesures....160
2. Protocole d'évaluation....162
2.1. Mise au point du corpus....163
2.2. Quoi évaluer ?....163
2.3. Méthodologie d'évaluation....164

2.3.1. Comment évaluer le repérage ?....165
2.3.2. Comment évaluer l'annotation ?....168
2.3.3. Comment évaluer le module de transformations ?....172
2.3.4. Comment évaluer le liage ?....174
3. Résultats....176
3.1. L'évaluation du repérage....176
3.1.1. Évaluation expression par expression....176
3.1.2. Évaluation par page Web....180
3.1.3. Relevé d'erreurs....181
3.2. L'évaluation de l'annotation....184
3.2.1. Quelques chiffres....184
3.2.2. Relevé d'erreurs....185
3.3. L'évaluation du liage....186
3.3.1. Quelques chiffres....186
3.3.2. Relevé d'erreurs....187
3.4. L'évaluation des transformations....189
4. Évaluation complémentaire – RMM2....191
Conclusion....194
Conclusion....195
Bibliographie....198
Annexes....205
A. Documentation technique....205
1. Paramètres du module Adetoa....205
2. Utilisation en ligne de commande....205
3. Utilisation comme un objet dans un programme JAVA....206
4. Encodage des caractères....206
5. Installation....206
6. Sortie....206
B. Structure d'Adetoa....207
1. Lanceur....207
2. Analyseur_structurel....207
3. Unitex....207
4. Transformeur....207
C. Description de l'ensemble des programmes d'Unitex utilisés....208
1. Le pré-traitement....208
2. L'application des transducteurs....208
D. Ensemble des transducteurs....210
1. Transducteur global....210
2. Transducteurs pour les expressions temporelles....212
3. Transducteur pour les expressions de localisation....219
4. Transducteurs pour les objets touristiques....220
5. Transducteurs pour le liage....225
E. Règles OPAL....227
1. Création des instances d'offres touristiques à partir des annotations....227
2. Création des instances de périodes....229
3. Création des associations reliant les périodes aux offres touristiques, suivant si ces périodes sont ouvertes ou fermées....234
F. Guide pour l'évaluation (fourni à l'évaluateur)....235
1. Lre repérage....235
2. Le liage....237
3. L'annotation....237
4. Les transformations....239

Index des Figures

Figure 1 : Projet Eiffel – vue globale [Eiffel 2009]....14
Figure 2 : Page Web d'une mairie....23
Figure 3 : Page Web d'une auberge....24
Figure 4 : Page Web d'un producteur fermier....26
Figure 5 : Page Web d'un guide de promenades....27
Figure 6 : Extrait du Guide du Routard Paris balades....28
Figure 7 : Page Web du restaurant Foody's....29
Figure 8 : Page Web du restaurant Scoop....29
Figure 9 : Page Web d'une ferme-auberge....31
Figure 10 : Schéma du signe selon Saussure....34
Figure 11 : Schéma du signe selon Peirce....35
Figure 12 : Page Web dans laquelle les blocs significatifs ont été marqués....36
Figure 13 : Autre disposition pour le même contenu - première possibilité....37
Figure 14 : Autre disposition pour le même contenu - deuxième possibilité....37
Figure 15 : Autre disposition pour le même contenu - troisième possibilité – mais avec perte de cohérence....38
Figure 16 : Horaires d'ouverture présentés dans un tableau....42
Figure 17 : Page Web du restaurant « Le jardin des pâtes »....47
Figure 18 : Page d'une mairie....62
Figure 19 : Page Web d'un producteur fermier....63
Figure 20 : Page Web d'une ferme-auberge....64
Figure 21 : Page Web d'une mairie - 2....65
Figure 22 : Page présentant deux hôtels....66
Figure 23 : Page présentant plusieurs objets....66
Figure 24 : Tableau contenant des horaires....68
Figure 25 : Page Web d'une crêperie....72
Figure 26 : Code source simplifié de la page de la ferme-auberge....73
Figure 27 : Exemple de page-agenda....75
Figure 28 : Extrait d'une page-agenda (1)....76
Figure 29 : Extrait d'une page de ville avec agenda....77
Figure 30 : Extrait d'une page-agenda (2)....78
Figure 31 : Page d'un centre de loisirs mentionnant d'autres objets....81
Figure 32 : Page présentant plusieurs objets....82
Figure 33 : Page d'un hôtel-bar-resto....84
Figure 34 : Arbre de décision montrant les pages traitables par Adetoa....85
Figure 35 : Page « simple »....86
Figure 36 : Page « complexe »....87
Figure 37 : Ensembles des données....88

Figure 38 : Éléments influençant l'annotation....96
Figure 39 : Élaboration du schéma d'annotation et retour sur l'ontologie....97
Figure 40 : DTD correspondant au schéma d'annotation....106
Figure 41 : Propriétés d'Infrastructure dans l'ontologie V1....107
Figure 42 : Propriétés temporelles des offres touristiques dans l'ontologie V2....108
Figure 43 : Proposition de modèle (Bernard Vatant)....109
Figure 44 : Proposition de modèle (Charles Teissèdre)....110
Figure 45 : Situation d'Adetoa....114
Figure 46 : Adetoa – utilisation des technologies JAVA-XML....114
Figure 47 : Graphe « GN »....117
Figure 48 : Transducteur « GN »....117
Figure 49 : Transducteur « Objets »....118
Figure 50 : Graphe « jours »....119
Figure 51 : Transducteur avec variables....119
Figure 52 : Structure d'Adetoa....120
Figure 53 : Transducteur principal de liage....121
Figure 54 : Transducteur « heure »....125
Figure 55 : Transducteur « localisation »....125
Figure 56 : Transducteur « adresse »....126
Figure 57 : Transducteur « accueil »....127
Figure 58 : Transducteur « période-jours »....127
Figure 59 : Graphe « période »....129
Figure 60 : Chaîne d'annotation sémantique – Content-Augmentation Manager [Coste 2009] 138
Figure 61 : Règle OPAL pour les services locaux....146
Figure 62 : Calcul du rappel....159
Figure 63 : Calcul de la précision....159
Figure 64 : Calcul de la F-mesure – cas général....160
Figure 65 : Calcul de la F-mesure - F1....160
Figure 66 : Diagramme représentant les résultats, par expression, de l'évaluation du repérage 177
Figure 67 : Diagramme représentant les résultats, par page, de l'évaluation du repérage....181
Figure 68 : Diagramme représentant, par expression, l'évaluation de l'annotation....184
Figure 69 : Diagramme représentant, par page, l'évaluation de l'annotation 185
Figure 70 : Diagramme représentant l'évaluation du liage selon les balises....187
Figure 71 : Exemples de liage....189
Figure 72 : Transducteur « rdv »....209
Figure 73 : Transducteur « global »....210
Figure 74 : Transducteur « perio-detail »....211
Figure 75 : Transducteur « accueil »....211
Figure 76 : Transducteur « date »....212
Figure 77 : Transducteur « per-marq »....212
Figure 78 : Transducteur « periode-jours »....212
Figure 79 : Transducteur « horaire-seul »....212
Figure 80 : Transducteur « heures »....212
Figure 81 : Transducteur « periode »....213

Figure 82 : Transducteur « marq-ouv-infra »....213
Figure 83 : Transducteur « marq-ouv-acti »....213
Figure 84 : Transducteur « date-annot »....213
Figure 85 : Transducteur « perio-simple »....214
Figure 86 : Transducteur « per-marq »....214
Figure 87 : Transducteur « exc-horaire »....214
Figure 88 : Transducteur « rdv »....214
Figure 89 : Transducteur « heure-detail »....214
Figure 90 : Transducteur « ferm »....215
Figure 91 : Transducteur « 2424 »....215
Figure 92 : Transducteur « 77 »....215
Figure 93 : Transducteur « heure »....216
Figure 94 : Transducteur « jours »....216
Figure 95 : Transducteur « heure-per »....216
Figure 96 : Transducteur « mois »....217
Figure 97 : Transducteur « exception »....217
Figure 98 : Transducteur « jours-repet »....217
Figure 99 : Transducteur « horaire »....218
Figure 100 : Transducteur « horaire-ferm »....218
Figure 101 : Transducteur « sous-heure »....218
Figure 102 : Transducteur « localisation »....219
Figure 103 : Transducteur « adresse »....219
Figure 104 : Transducteur « cp »....219
Figure 105 : Transducteur « rue »....219
Figure 106 : Transducteur « objets »....220
Figure 107 : Transducteur « hebergements »....220
Figure 108 : Transducteur « resto »....221
Figure 109 : Transducteur « hebreg »....221
Figure 110 : Transducteur « serv »....221
Figure 111 : Transducteur « evenement »....221
Figure 112 : Transducteur « equip »....222
Figure 113 : Transducteur « gastronomie »....222
Figure 114 : Transducteur « prestation »....223
Figure 115 : Transducteur « promenade »....223
Figure 116 : Transducteur « sport »....223
Figure 117 : Transducteur « visite »....223
Figure 118 : Transducteur « artisan »....224
Figure 119 : Transducteur « nepasreperer »....224
Figure 120 : Transducteur « global-app »....225
Figure 121 : Transducteur « groupe-ut »....225
Figure 122 : Transducteur « ut »....225
Figure 123 : Transducteur « objet »....226
Figure 124 : Transducteur « loc »....226
Figure 125 : Transducteur « no-annot »....226

INDEX DES TABLEAUX

Tableau 1 : Horaires en tableau vertical....69
Tableau 2 : Classification des expressions repérées....168
Tableau 3 : Classification des expressions non repérées....168
Tableau 4 : Classification des annotations....172
Tableau 5 : Graduation des résultats....178
Tableau 6 : Taux de rappel et précision obtenus par Adetoa pour la tâche de repérage....178
Tableau 7 : F-mesures obtenues par Adetoa pour la tâche de repérage avec $\alpha = 1$....178
Tableau 8 : Résultats obtenus par d'autres systèmes d'extraction d'information....179
Tableau 9 : Taux de fallout....180
Tableau 10 : Résultats de l'évaluation sur les 513 expressions du corpus RMM2....193
Tableau 11 : Résultats de l'évaluation sur corpus partiel....193
Tableau 12 : R1 - Hébergement....227
Tableau 13 : R2 - Service Local 1....227
Tableau 14 : R3 - Service local 2....228
Tableau 15 : R4 - Service local 3....228
Tableau 16 : R5 – Concert....229
Tableau 17 : R6 – Période d'ouverture....229
Tableau 18 : R7 – Période de fermeture....230
Tableau 19 : R8 – Date début....230
Tableau 20 : R9 – Date fin....231
Tableau 21 : R10 – Jour....231
Tableau 22 : R11 – Heure début....232
Tableau 23 : R12 – Heure fin....232
Tableau 24 : R13 – Description....233
Tableau 25 : R14 – Exception....233
Tableau 26 : R15 – Période d'ouverture offre....234
Tableau 27 : R16 – Période de fermeture offre....234

INTRODUCTION

« Quand est-ce qu'on se fait une exposition ?

– Je suis libre jeudi à partir de 17 heures, qu'est-ce qu'on peut aller voir ?

– Attends, je regarde sur Internet. »

Ce court dialogue de la vie de tous les jours illustre parfaitement une pratique qui s'est largement répandue ces dernières années et qui consiste à se servir d'Internet comme d'une source d'information. Les internautes ont en effet de plus en plus recours au Web pour chercher toutes sortes d'informations, qu'il s'agisse d'informations encyclopédiques, d'informations pratiques, d'actualités, etc. Le domaine du tourisme, notamment, est abondamment représenté sur le Web ; les internautes effectuent des recherches aussi bien pour planifier un voyage avec diverses activités que pour prévoir une sortie particulière, vérifier une adresse ou des horaires d'ouverture.

Toutefois, l'une des problématiques inhérentes à la nature du Web réside dans la diversité et surtout dans la profusion d'informations qu'il contient, sans que celles-ci soient structurées ou organisées. Les internautes sont alors face à une multitude d'informations qu'ils peuvent avoir du mal à exploiter. Les moteurs de recherche sont là pour faciliter leur recherche et leur fournir des résultats conformes à leurs attentes. Malgré tout, les moteurs de recherche généralistes (Google, Yahoo!, etc.) ne permettent pas une recherche par domaine ou type d'information mais seulement une recherche par mots-clés – de manière plus ou moins évoluée, avec la possibilité, ou non, de reconnaître des variantes lexicales par exemple – et ne sont ainsi pas en mesure de répondre à toutes les attentes des internautes. Certains moteurs de recherche spécialisés (Légifrance, Sudoc, etc.), n'effectuent la recherche que sur un nombre restreint de pages Web d'un domaine. Cependant, ils se basent également sur un système de mots-clés.

La nécessité d'outils agrégeant, pour un domaine donné, les informations disséminées sur le Web s'est alors fait ressentir pour pallier cette abondance d'informations et leur disparité ; des plateformes dédiées ont donc vu le jour. De telles plateformes associent plusieurs outils et font appel à différentes technologies pour répondre au mieux aux besoins des internautes dans le domaine qu'elles couvrent. En premier lieu, elles contiennent un outil de crawling permettant de récupérer, à différents endroits du Web, les pages liées à leur domaine, à l'aide de filtres. En deuxième lieu, ces pages doivent être analysées et annotées pour en extraire les

informations utiles. Pour effectuer cette annotation, dite alors annotation sémantique, la plateforme doit faire appel à une ontologie du domaine. Le développement, ou du moins l'adaptation, d'une telle ontologie fait aussi partie de la mise au point de la plateforme. Une base de connaissance est ensuite créée pour stocker les instances et relations issues des contenus annotés. En troisième lieu, pour rendre ces données accessibles à l'utilisateur final de la plateforme, d'autres modules sont à prévoir : une interface lui permettant de définir facilement ses critères de recherche (recherche par type d'offre touristique, selon un calendrier, suivant des propositions ou en langage naturel, etc.), un module d'interprétation des requêtes se chargeant ensuite de formaliser la demande de l'internaute de façon à effectuer une recherche dans la base de connaissance. Un module de recherche dans la base de connaissance, associé éventuellement à un moteur de raisonnement et d'inférence, permet ensuite de récupérer les données qui peuvent intéresser l'internaute.

C'est dans ce contexte qu'a été lancé le projet Eiffel[1], auquel j'ai participé et dans le cadre duquel a été développée une plateforme touristique :

> « Eiffel innove par la mise en œuvre d'une plateforme logicielle permettant, autour d'une ontologie tourisme :
>
> - de sélectionner, classer, qualifier des contenus distribués sur le web ou d'autres sources ;
>
> - de traiter des requêtes mettant en œuvre le contenu de l'ontologie du territoire et les contenus web indexés et d'offrir des fonctions de navigation dans la base de connaissance ;
>
> - d'assister l'utilisateur dans la construction d'un voyage, séjour à partir des ressources sélectionnées ;
>
> - d'analyser les usages et les comportements utilisateurs afin d'améliorer le service proposé. » ([Eiffel 2009])

Le schéma suivant en représente les différentes étapes :

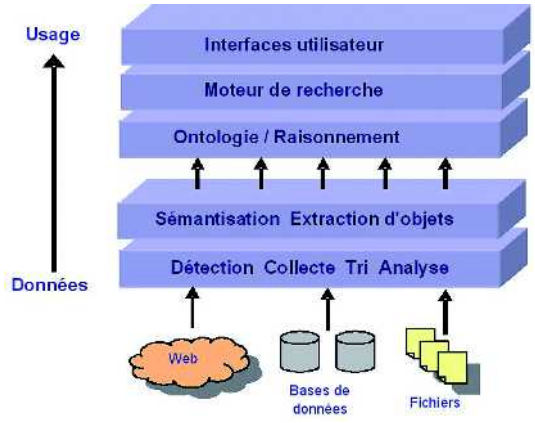


Figure 1 : Projet Eiffel – vue globale [Eiffel 2009]

1 Projet ANR-05-RNTL-007 – EIFFEL – liant deux entreprises (Antidot et Mondeca) et trois laboratoires de recherche (INRIA, LIRMM et MoDyCo) – www.projet-eiffel.org.

Les données sont au cœur du projet Eiffel : données brutes de départ que constituent les pages Web collectées, données extraites dans ces pages, données peuplant l'ontologie, données à fournir à l'utilisateur. Chaque type de donnée nécessite donc un traitement spécifique : quels filtres appliquer pour sélectionner les « bonnes » pages Web de départ ? Comment analyser le contenu de ces pages pour en tirer les « bonnes » informations ? Comment stocker, efficacement, ces informations ? Comment les restituer à l'utilisateur final pour répondre à ses besoins ?

Ces différentes étapes, et les problématiques qui y sont liées, constituent le projet Eiffel dans son ensemble et doivent donc interagir entre elles, pour un bon fonctionnement général. Toutefois, chacune est, dans une certaine mesure, indépendante. C'est ainsi que j'ai été chargée de développer le module de repérage et d'annotation des données dans les pages Web.

Pour répondre à cette demande et développer ce module, appelé Adetoa (pour *Annotateur d'expressions temporelles, d'objets et d'adresses*), j'ai été confrontée à plusieurs problématiques. En effet, plusieurs champs de recherche s'entrecroisent : le domaine de l'extraction d'information d'une part, la gestion de connaissance d'autre part, mais également le traitement de pages Web.

Adetoa étant chargé de repérer et annoter, dans les pages Web, un certain nombre d'informations, les problématiques liées à l'extraction d'information sont donc centrales pour mon travail. Les besoins et contraintes du projet Eiffel définissent le type d'information à repérer : il s'agit des informations pratiques des offres touristiques présentées dans les pages Web – horaires et périodes d'ouverture ou de fermeture, types d'objet et adresses. En m'intéressant plus particulièrement aux informations temporelles, j'ai donc mené une étude pour définir et classifier les différents types d'expressions permettant de formuler ces données. Le texte contenu dans les pages Web étant en langage naturel, les expressions linguistiques permettant de formuler ces informations sont très variées. Tout en prenant en compte les besoins du projet, et ceux de l'utilisateur de la plateforme, cette étude du contenu des pages Web a donc permis de répondre à la question « quoi repérer ? ». Pour répondre à la question « comment repérer et annoter ? », différents outils ont été envisagés mais mon choix s'est orienté vers un système à base d'expressions régulières (sous la forme de transducteurs) permettant de rester au plus près du texte de départ et ainsi de pouvoir profiter de la richesse linguistique des expressions. Le repérage et l'annotation des différents types d'information ne sont pas suffisants, il faut également lier ces informations pour pouvoir les rattacher à un objet. En effet, chaque information est inexploitable individuellement ; pour être stockée dans la base de connaissance, toute information doit être reliée à un objet touristique instancié, ou à instancier.

Que ce soit pour effectuer ce liage ou pour annoter les données dans les pages Web, le domaine de la gestion de connaissance joue un rôle essentiel. L'ontologie du tourisme, qui a été développée dans le cadre d'Eiffel, y tient une place centrale. Adetoa, tout comme les autres modules du projet, s'appuie sur cette ontologie. Les données annotées doivent en effet la respecter pour pouvoir être stockées dans la base de connaissance qui en a la structure. Toutefois, les informations, telles qu'elles sont formulées dans les pages Web, ne sont pas toujours directement transposables dans l'ontologie. J'ai donc mis au point un schéma d'annotation permettant de respecter les spécificités linguistiques des données. Des règles de mapping permettent ensuite de faire le pont entre les données et l'ontologie. La difficulté

réside dans l'équilibre à trouver entre l'annotation d'un maximum de données dans les pages Web et la façon dont celles-ci peuvent ensuite être stockées et ainsi exploitées.

Le fait que les données sur lesquelles j'ai travaillé sont contenues dans des pages Web oriente nécessairement l'angle de recherche. Si le traitement de pages Web peut sembler moins central aux problématiques habituelles de linguistique ou de traitement automatique du langage naturel, il n'empêche que le contenu n'est pas totalement dissociable du support en tant que dispositif technique : certaines spécificités langagières sont directement liées au support que constitue la page Web. Il semblerait que, par rapport à d'autres supports plus traditionnels comme les livres imprimés ou guides touristiques papier, le Web permette une plus grande liberté face à la syntaxe et autres règles de grammaire (texte souvent partiellement non rédigé, abréviations, etc.). De plus, chaque page Web étant indépendante, aucune ligne éditoriale n'est respectée et cela accentue la diversité des contenus possibles. Surtout, la page Web ne se réduit pas au texte qu'elle contient mais constitue un objet technique à part entière. Elle est conçue pour être affichée dans un navigateur qui se charge donc d'en interpréter le code source. Or, l'apparence de la page Web telle qu'affichée par un navigateur est perdue lors de l'analyse de son code source seul. Les problématiques liées aux spécificités des contenus Web sont donc également au cœur de mon travail. En revanche, si l'analyse des pages Web en termes de sémiotique, telle qu'effectuée dans le cadre du projet TramedWeb[2] auquel j'ai également participé, m'a inspirée, cette approche n'est pas centrale à ce travail.

Si ces trois champs de recherche sont bien distincts dans la littérature, ils se rapprochent et s'entremêlent au sein du projet Eiffel et plus particulièrement au sein d'Adetoa, chargé d'effectuer une tâche d'extraction d'information et d'annotation sémantique dans des pages Web. Ces trois champs sont donc ceux auxquels je me suis intéressée dans cette thèse qui, guidée par le projet Eiffel et le développement d'Adetoa, aborde également des questions plus théoriques comme la formulation du temps dans les pages Web touristiques ou certains aspects sémiotiques de ces pages.

Plan de la thèse

Le plan de cette thèse est construit en trois parties, correspondant aux trois aspects principaux de ce travail. La première partie étudie, sur un plan théorique, les données : spécificités des expressions temporelles et étude linguistique, spécificités des pages Web. La deuxième partie s'intéresse à la façon dont ces données peuvent être manipulées : traitement automatique de pages Web pour l'extraction d'information, schéma d'annotation et ontologie. La troisième partie présente les réalisations techniques qui découlent de ce travail : implémentation d'Adetoa, intégration dans la chaîne de traitement du projet Eiffel et évaluation.

I. Présentation Générale des Données et Objets d'Étude

Chapitre 1 – *Objets et Problématiques*

Ce premier chapitre présente les deux principaux objets avec lesquels j'ai travaillé : les expressions temporelles et les pages Web touristiques. C'est la nature générale des expressions temporelles qui est présentée ici. Les pages Web sont présentées selon un point

2 Projet ANR Blanc 2007-2010 – TramedWeb – liant trois laboratoires de recherche : le GRIPIC, le laboratoire Culture et Communication et le laboratoire MoDyCo [Davallon à par.].

de vue sémiotique, mettant en avant la structure visuelle des pages. Une comparaison entre les guides touristiques papier et les pages Web touristiques permet de signaler quelques spécificités du Web.

Chapitre 2 – *Étude Linguistique*

Le deuxième chapitre s'intéresse, sur un plan linguistique, aux expressions temporelles qui entrent dans mon champ de recherche. Il en propose une classification et se préoccupe des difficultés d'interprétation ou d'ambigüité que peuvent poser certains énoncés. Il présente aussi sommairement les expressions correspondant aux adresses et aux types d'objets.

II. Manipulation des Données et Représentation

Chapitre 3 – *Traitement Automatique de Pages Web Touristiques*

Le chapitre 3 aborde les questions qui se posent lorsque l'extraction d'information automatisée concerne des pages Web. Un bref état de l'art sur l'extraction d'information est présenté, avant de passer aux spécificités des pages Web touristiques sur lesquelles j'ai travaillé : présentation des pages qu'Adetoa est amené à traiter, difficultés liées à la conception et à la structure des pages, difficultés liées à leur contenu.

Chapitre 4 – *Ontologie et Schéma d'Annotation*

Le chapitre 4 montre le rôle central que joue l'ontologie, aussi bien dans Eiffel que pour mon travail. Il débute par un état de l'art sur les ontologies avant de présenter plus en détail celle d'Eiffel. J'ai défini un schéma d'annotation, distinct de l'ontologie ; celui-ci est présenté ici, avec les retours qu'il a engendrés sur l'ontologie.

III. Implémentation et Évaluation

Chapitre 5 – *Implémentation d'Adetoa*

Le chapitre 5 présente les outils qui m'ont servi pour le développement d'Adetoa : Unitex, JAVA, technologies XML. Les ressources linguistiques que j'ai constituées pour effectuer le repérage et l'annotation sont ensuite décrites (transducteurs Unitex et dictionnaire des noms de villes), ainsi que les règles de transformations mises au point pour effectuer certains traitements linguistiques.

Chapitre 6 – *Exploitation des Données Annotées dans une Chaîne de Traitement*

Le chapitre 6 montre comment Adetoa a pu s'intégrer dans la chaîne de traitement plus globale que constitue le projet Eiffel. L'accent y est surtout mis sur la façon dont les données annotées que fournit Adetoa peuvent être exploitées : règles de mapping, consolidation et stockage dans la base de connaissance.

Chapitre 7 – *Expérimentation et Évaluation*

Le chapitre 7, très concret, décrit l'évaluation d'Adetoa qui a été réalisée. Après un état de l'art sur les méthodes classiques d'évaluation des outils de TAL, le protocole d'évaluation y est présenté ainsi que les résultats obtenus pour chacune des tâches effectuées par Adetoa (repérage, annotation, transformations et liage).

Cette thèse a donné lieu aux publications : [Weiser 2008], [Weiser et al. 2008], [Weiser et al. 2009], [Noël et al. 2008], [Fortin et al. 2009] et [Garron et al. à par.].

Première Partie

Présentation Générale des Données et Objets d'Étude

CHAPITRE 1. OBJETS ET PROBLÉMATIQUES

Introduction

Ce premier chapitre a pour but de présenter mes objets d'études et les problématiques qui y sont liées. Dans un premier temps, je présenterai et définirai les expressions temporelles sur lesquelles j'ai travaillé. Dans un second temps, nous verrons l'impact du support – page Web – sur les données étudiées.

1. Les expressions temporelles

Dans cette partie, je vais définir, de façon générale, ce que j'ai appelé « expressions temporelles ». Nous verrons ensuite plus précisément le type d'expressions temporelles auxquelles je me suis intéressée pour ce travail. Ces expressions sont toutes issues du domaine du tourisme et apparaissent plus particulièrement sur des pages Web touristiques.

1.1. Quelques généralités sur la formulation du temps

Classiquement, les expressions temporelles sont souvent réparties en deux catégories. La première comprend les expressions temporelles relatives et la deuxième les expressions temporelles absolues. Les expressions temporelles relatives, dites aussi déictiques, sont des unités linguistiques « dont le sens implique obligatoirement un renvoi à la situation d'énonciation pour trouver le référent visé » [Kleiber 1986]. [Moeschler 1993], reprenant les travaux de Milner, considère les déictiques comme des expressions non autonomes. Il les distingue des expressions autonomes qui sont définies ainsi : « des expressions calendaires comme *en 1963, le 22 novembre 1963* déterminent la référence temporelle de l'énoncé de manière autonome : elles n'ont pas besoin d'autres informations pour le calcul de la référence. »

Voici deux exemples d'expressions temporelles déictiques, ou encore non autonomes :

(1) **Demain**, le musée sera fermé.

(2) Le musée ouvrira ses portes **le 8**.

Pour interpréter ces deux expressions, il est nécessaire de connaître le moment

d'énonciation : la date d'aujourd'hui pour le premier exemple, et le mois et l'année courants pour le second.

Comme l'a précisé la définition ci-dessus, les expressions temporelles absolues, ou autonomes, peuvent être interprétées sans faire appel au contexte linguistique ou extra-linguistique. Ces expressions peuvent généralement être positionnées sur un calendrier. Par exemple :

(3) **Demain, le 8 octobre 2008**, le musée sera fermé.

(4) **Le 8 octobre 2008**, le musée sera fermé.

(5) Le musée sera fermé **en octobre 2008**.

Ces trois expressions se comprennent sans difficulté et sans qu'il soit nécessaire de faire appel au contexte. Dans l'exemple (3), le déictique *demain* permet de connaître la date d'énonciation mais l'expression en elle-même est absolue et non déictique : il n'est pas la peine de savoir qu'on est le 7 octobre pour l'interpréter[3]. Les exemples (4) et (5) se distinguent de par leur granularité : le premier fait référence à un jour précis tandis que le second fait référence à un mois entier. Mais, étant donné que l'on peut placer sur un calendrier aussi bien un jour qu'une période plus longue, comme un mois, et l'année étant précisée, ces deux expressions sont bien absolues.

Cette distinction entre expressions relatives et expressions absolues ne suffit pas à classifier toutes les expressions temporelles. Elle est en effet suffisante pour classifier les expressions de type calendaire, mais ne l'est pas pour d'autres types d'expressions temporelles. Dans [Battistelli et al. 2006], sont qualifiées « d'expressions calendaires » les expressions temporelles ayant un lien avec le repérage dans les calendriers. Pour en arriver à cette définition, les auteurs sont partis des définitions classiques des termes « datation » et « calendrier » dont deux éléments importants ressortent :

> « d'une part la notion de chronologie vue comme une séquence d'événements privilégiés permettant de positionner d'autres événements ; d'autre part le caractère essentiellement relationnel des calendriers, à savoir une mise en relation de deux unités. [...] nous définissons deux types de dates : les dates calendaires et les dates événementielles. » ([Battistelli et al. 2006], p. 8)

Les expressions calendaires peuvent être de complexités diverses :

(6) Le 8 octobre 2008.

(7) Le jour où les voies sur berges ont été ouvertes aux piétons.

Mais, comme nous le verrons dans la suite de ce chapitre, les expressions temporelles qui se trouvent sur les pages Web touristiques ne sont pas toujours des expressions calendaires.

(8) Ouvert **tous les jours**.[4]

(9) Ouvert **du lundi au vendredi de 8h00 à 18h00**.

(10) Fermé **deux semaines en février**.

3 Notons aussi que, dès le 8 octobre, cette expression devient incohérente.

4 Il faut aussi noter que ce type d'exemple, a priori très simple, peut avoir plusieurs interprétations selon le domaine métier : si cette expression concerne un commerce par exemple, elle peut signifier que celui-ci est fermé le dimanche, s'il s'agit d'une mairie alors celle-ci a de grandes chances d'être fermée le dimanche, voire même le samedi.

Ces trois exemples sont typiques des expressions temporelles que l'on rencontre dans les pages Web touristiques. Ce sont des expressions que l'on ne peut pas considérer comme calendaires. La distinction « absolue vs. déictique » ne convient pas pour ce genre d'expressions. De plus, si l'on peut rencontrer dans ces pages Web certains types d'expressions calendaires, d'autres sont beaucoup moins probables. Il est en effet peu réaliste d'annoncer, sur une page Web, un événement de la façon suivante :

(11) ?[5]**Le concert aura lieu le jour où les voies sur berges ont été ouvertes aux piétons.**

Cet exemple pourrait être rejeté à cause de la concordance des temps qui n'est pas respectée. Mais il reste peu probable même si celle-ci est rectifiée :

(12) **Le concert aura lieu le jour où les voies sur berges seront ouvertes aux piétons.**

1.2. Les expressions temporelles liées au domaine du tourisme

Les expressions temporelles qui ont attiré mon attention font partie du domaine du tourisme. Ce sont des informations pratiques liées à un objet touristique, comme celles des exemples suivants :

(13) **Ouvert du lundi au vendredi de 8h00 à 18h00.**[6]

(14) **Le concert aura lieu le 12 octobre.**

(15) **Ouvert toute l'année sauf du 15 au 30 janvier.**

Les horaires d'ouverture contenus dans l'exemple (13) peuvent correspondre à un musée, un office de tourisme, etc. ; les dates, comme celle de l'exemple (14), peuvent concerner un concert, un festival ou une pièce de théâtre ; les périodes d'ouverture plus larges comme dans l'exemple (15) peuvent correspondre à des hébergements, hôtels ou campings. Ces expressions se veulent informatives et précises. Elles permettent au touriste de planifier ses activités.

Les dates d'événements ou horaires d'ouverture ne sont pas exclusivement spécifiques au domaine du tourisme et peuvent servir à quiconque souhaitant planifier une sortie ou activité. En revanche, d'autres types d'informations sont réellement propres au domaine du tourisme, par exemple la notion de saison : *hors saison, en saison, haute et basse saison*.

Les exemples d'expressions temporelles donnés ci-dessus sont typiques et encore relativement simples à interpréter. L'exemple (15) laisse entrevoir les difficultés que peut poser l'interprétation des expressions, surtout lorsqu'elle est automatique. En effet, il contient une exception (avec le *sauf*) qu'il faut donc prendre en compte, et de plus, une ellipse apparaît dans la date (*janvier* n'est pas répété deux fois). La complexité des expressions peut être encore plus grande :

(16) **En Juillet et Août ouvert tous les jours. Hors cette période, fermeture le mercredi soir, le jeudi toute la journée et le dimanche soir. Horaires d'ouverture: de 12h00 à 14h00 de 19h00 à 22h00.**

(17) **Fermé durant les vacances scolaires de février. Fermeture hebdomadaire non déterminée.**

5 Le point d'interrogation indique que l'énoncé n'est pas très naturel.

6 À partir d'ici et, sauf mention contraire, les expressions données en exemple sont toutes issues de pages Web touristiques. Elles sont retranscrites telles quelles, c'est-à-dire avec leurs fautes de frappe, leur ponctuation et les majuscules ou minuscules.

Ces exemples permettent donc de montrer que l'interprétation des expressions temporelles touristiques est loin d'être triviale. Une analyse plus détaillée de ces expressions sera faite au chapitre 2.

1.3. Étude des expressions temporelles dans les pages Web touristiques – mise en relation avec l'affiche

Afin de mieux comprendre les spécificités des expressions temporelles présentes sur le Web, il m'a semblé intéressant de comparer ces expressions avec celles apparaissant sur d'autres supports. J'ai choisi l'affiche comme base de comparaison. Je vais donc mettre en évidence ici quelques points communs et quelques différences entre ces deux supports.

Le Web étant vaste et varié, il est impossible d'émettre des généralités. Je me suis donc concentrée, pour cette comparaison, sur un type de pages plus restreint qui sont les pages Web touristiques. Tout d'abord, c'est sur des pages Web touristiques que le projet Eiffel est basé et cela m'a permis d'avoir accès à un grand nombre de pages déjà filtrées et aspirées. De plus, et pour cette partie, je me suis aussi basée sur des pages Web touristiques participatives[7], qui sont les pages sur lesquelles porte le projet TramedWeb [TramedWeb 2006][8], auquel j'ai participé. En ce qui concerne les affiches, j'ai repris les analyses présentées dans [Joly 1993], qui concernent principalement les affiches publicitaires.

Souvent, les expressions temporelles que l'on rencontre dans ces pages Web ont la propriété d'être toujours actualisées, ou du moins de sembler actuelles. En effet, l'avantage d'une page Web est qu'elle peut être mise à jour facilement. On y trouve donc des informations ponctuelles, en temps réel. Dans certains types de pages Web, les expressions temporelles sont déictiques et leur interprétation dépend du moment de consultation. Ce sont aussi souvent des expressions périodiques, répétitives ou encore génériques comme par exemple *ouvert tous les jours*. Lorsqu'il ne s'agit pas d'horaires purement informatifs, les informations temporelles ont souvent pour but d'interpeller l'internaute. Cet emploi se rapproche de ce que l'on peut trouver dans le domaine de la publicité. Un concert peut par exemple être annoncé avec *mardi soir* ou *le 8 octobre*, ce qui a pour but de montrer à l'internaute que la date est proche. On retrouve le même type d'informations sur des affiches, qui ont, elles aussi, une visée informative et qui ont un caractère d'actualité. La différence entre les deux est que la page Web peut être actualisée alors que l'affiche devient obsolète une fois la date passée. Les affiches ont donc une durée limitée et sont remplacées périodiquement, sauf en cas d'affichage sauvage, mais il faut alors émettre une réserve quant à la fiabilité de l'information. Dans [Joly 1993], Martine Joly reprend Roland Barthes selon qui, concernant les affiches, « deux grands cas de figure se présentent : soit le texte a, par rapport à l'image, une fonction d'*ancrage*, soit il a une fonction de *relais* ». Elle en redonne les définitions :

> « La fonction d'ancrage consiste à arrêter cette « chaîne flottante du sens » qu'engendrerait la nécessaire polysémie de l'image, en désignant le « bon niveau de lecture », quoi privilégier parmi les différentes interprétations que peut solliciter l'image seule. La presse offre des exemples quotidiens de cette fonction d'ancrage du message linguistique, qu'on appelle aussi « la légende » de l'image.[...]
>
> La fonction de relais se manifesterait, quant à elle, lorsque le message linguistique viendrait suppléer des carences expressives de l'image, prendre son relais. En effet,

7 Les sites participatifs sont des sites dans lesquels l'internaute peut ajouter son propre contenu ainsi que voir le contenu ajouté par les autres internautes, une notion de communauté y est souvent liée.

8 www.tramedweb.fr

> malgré la richesse expressive et communicative d'un message purement visuel [...], il y a des choses qu'il ne peut pas dire sans recours au verbal. » ([Joly 1993], p. 96)

Ces deux fonctions sont largement utilisées sur le Web, lieu sur lequel se marient très souvent textes et images. Par exemple, dans la page Web de la figure 2, l'image montre un bâtiment qui pourrait être un château, un hôtel particulier, une maison, etc. C'est le texte qui permet de savoir qu'il s'agit d'une mairie, remplissant alors sa fonction d'ancrage. Dans la page Web de la figure 3, l'image montre l'auberge-restaurant et le texte précise la capacité : *100 personnes, 60 et 85 couverts pour groupe*. Cette information ne pourrait en aucun cas être représentée par une image et le texte y remplit donc la fonction de relais.

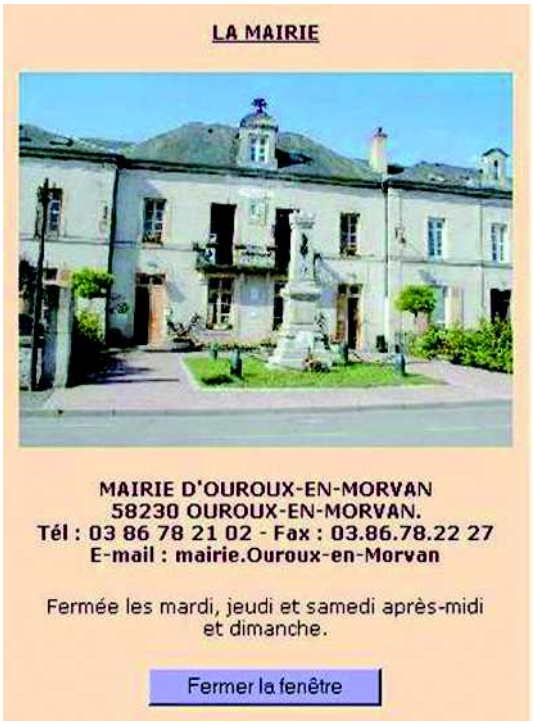


Figure 2 : Page Web d'une mairie



Figure 3 : Page Web d'une auberge

Fresnault-Deruelle, dans *L'image placardée* [Fresnault-Deruelle 1997], a beaucoup travaillé sur l'affiche. Mon étude des pages Web peut se rapprocher de ces travaux théoriques, en particulier lorsqu'ils mentionnent le support sur lequel l'information prend place :

> « Le rapport texte/image entre évidement en ligne de compte, qui peut générer des variantes selon qu'il s'agit d'une affiche ou d'une annonce. La prise en considération du support peut ainsi pousser les concepteurs à calibrer le « rédactionnel » en fonction du support. » ([Fresnault-Deruelle 1997], p.85)

Sur une page Web, les contraintes techniques liées au support sont omniprésentes et influencent sans aucun doute le contenu rédactionnel.

Voici comment cet auteur résume l'analyse d'une affiche :

> « En bref, l'analyse [d'une affiche] consiste :
>
> – outre l'étude du message verbal, à repérer les codes au travail dans l'image : codes formels ou iconiques (degré de réalisme, facture, effet de collage, cadrage, couleur, position des objets, etc.) ; codes « mondains » vestimentaires, physiognomoniques, scénographiques, etc. ;
>
> – ces codes étant repérés, l'analyse peut se poursuivre avec la description de la façon dont les unités, actualisées et regroupées entre elles, concourent à l'effet global produit par le document en situation ;
>
> – en tant que composition originale, l'affiche ne saurait également être approchée sans que cette dernière soit restituée par rapport aux traditions iconographiques dans la lignée desquelles elle semble s'inscrire (pastiche, parodie, citation).
>
> En un mot, « focaliser » sur les formes plutôt que sur les contenus peut parfois

> constituer un point de départ. » ([Fresnault-Deruelle 1997], p.87)

Comme le montrent ces exemples de pages Web, ces propriétés des affiches sont presque directement transposables pour parler de pages Web, sur lesquelles la disposition, les couleurs, les polices donnent des indications aussi bien que le contenu textuel en lui-même.

S'agissant de la temporalité, celle-ci ne s'exprime pas toujours de la même manière dans une page Web ou sur une affiche. En effet, les affiches sont ancrées temporellement mais rarement par des marques linguistiques, tandis que sur des pages Web, ce sont le plus souvent des marques linguistiques qui remplissent cette fonction.

Les polices d'écriture sont souvent utilisées comme marques temporelles dans les affiches. Par exemple, au sujet d'une publicité pour *Marlboro*, Martine Joly écrit :

> « La classification classique des caractères distingue trois grands types de caractères à empâtement : triangulaire, filiforme et rectangulaire, par opposition aux caractères sans empâtement. Ceux-ci sont considérés comme des caractères « modernes ». Le choix des caractères est donc très important dans l'implicite du message. Ainsi le choix de l'empâtement triangulaire fait implicitement référence au développement de la presse au XIX^{e} siècle. On voit donc comment cette allusion, associée à l'image du cow-boy, renvoie à l'univers stéréotypé du typographe de « l'ouest », à l'idée de conquête, d'aventure et de progrès. » ([Joly 1993], p. 98)

Le choix des polices d'écriture est également essentiel lors de la création d'une page Web. Des polices ont d'ailleurs été créées spécifiquement pour le Web (Arial par exemple), avec la caractéristique d'être plus lisibles à l'écran qu'une fois imprimées sur papier. Mais, si les polices d'écriture choisies lors de la création peuvent aussi jouer un rôle temporel, par exemple donner à un restaurant un caractère rustique ou classique, elles ne peuvent pas servir à donner des informations temporelles pratiques.

Dans certains types de pages Web, il est très important de montrer que l'information est actualisée et certains procédés sont souvent utilisés à cette fin. Il s'agit notamment de dater le moment où l'information a été mise en ligne. Cela est très souvent le cas dans les sites participatifs. Dans le site www.webcity.fr, guide touristique, les internautes peuvent commenter les restaurants, bars, hôtels où ils sont allés. Il est alors très important de marquer à quel moment les commentaires ont été placés et un commentaire récent aura plus d'impact qu'un commentaire laissé deux ans auparavant. S'il s'agissait de commenter, par exemple, des peintures de Van Gogh, tel ne serait pas le cas, car les peintures ne changeant pas, les commentaires restent valables dans la durée. Mais dans le monde du tourisme, tout change très vite et, par exemple, un restaurant peut être très bon et peu cher à une date donnée mais changer très rapidement. Voilà pourquoi la datation des informations est importante. Ces sites participatifs mettent donc l'accent sur les dates et utilisent différents procédés pour leur donner plus d'importance. Par exemple un commentaire placé le jour même sera mis en avant et précédé de la mention *aujourd'hui,* voire même de l'heure exacte. La granularité deviendra d'autant moins fine que les commentaires seront anciens : l'heure sera d'abord omise, puis le jour, pour ne laisser que le mois. Les dates sont donc dynamiques et actualisées chaque jour : le commentaire précédé de *aujourd'hui* ne peut évidemment rester ainsi qu'un seul jour et dès le lendemain il faut donc transformer le *aujourd'hui* en *hier* ou en une date précise.

Mais ces procédés sont principalement utilisés dans les sites participatifs et, dans le cadre du

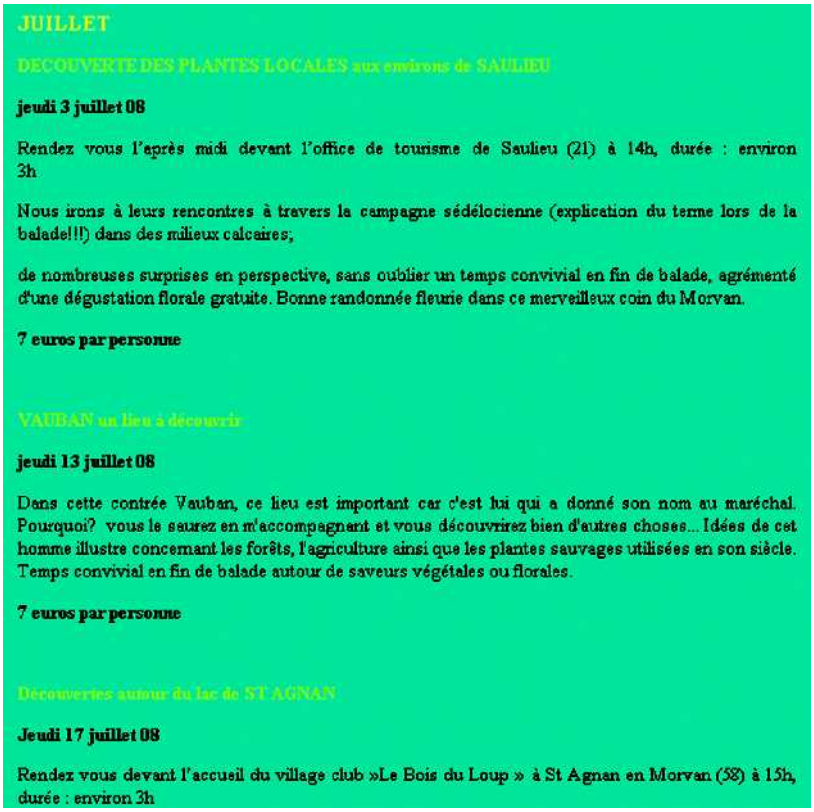
projet Eiffel, ces pages ne sont pas traitées. En effet, comme nous le verrons dans la deuxième partie de ce chapitre, les pages Web traitées sont celles d'offres touristiques et elles sont créées principalement par les acteurs du tourisme eux-mêmes. Ces sites sont plus stables et les dates rarement dynamiques.

2. Pages Web touristiques étudiées

Dans le cadre de cette thèse, je me suis intéressée aux mêmes pages Web que celles du projet Eiffel. Elles avaient l'avantage d'avoir déjà été filtrées et aspirées sur le Web, de façon à constituer un corpus de travail. Ce sont des pages Web touristiques : elles présentent un ou plusieurs objets touristiques. Il ne s'agit pas de toutes les pages à caractère touristique, mais d'un sous-ensemble de celles-ci. Par exemple, les sites de carnets de voyages ou les guides touristiques en ligne ne sont pas considérés. C'est donc un ensemble restreint de pages, mais elles présentent tout de même une grande variété. Les pages auxquelles je m'intéresse présentent un objet touristique précis (page d'un hôtel, d'un musée, etc.) ou plusieurs objets (événements touristiques organisés par une ville, équipements sportifs d'une commune, etc.). Lorsqu'elles ne mentionnent qu'un objet touristique, elles regroupent généralement les différentes informations pratiques qui le concernent : adresse, nom, heures et période d'ouverture. Lorsqu'elles mentionnent plusieurs objets, chacun des objets peut être présenté en détail à l'aide d'un certain nombre d'informations (nom, adresse, horaires, etc.) ou bien un objet principal est présenté et des objets « satellites » sont simplement mentionnés sans être détaillés. Voici quelques exemples de pages Web touristiques, en plus de ceux donnés ci-dessus, dans les figures 2 et 3 :



Figure 4 : Page Web d'un producteur fermier



JUILLET

DECOUVERTE DES PLANTES LOCALES aux environs de SAULIEU

jeudi 3 juillet 08

Rendez vous l'après midi devant l'office de tourisme de Saulieu (21) à 14h, durée : environ 3h

Nous irons à leurs rencontres à travers la campagne sédélocienne (explication du terme lors de la balade!!!) dans des milieux calcaires;

de nombreuses surprises en perspective, sans oublier un temps convivial en fin de balade, agrémenté d'une dégustation florale gratuite. Bonne randonnée fleurie dans ce merveilleux coin du Morvan.

7 euros par personne

VAUBAN un lieu à découvrir

jeudi 13 juillet 08

Dans cette contrée Vauban, ce lieu est important car c'est lui qui a donné son nom au maréchal. Pourquoi? vous le saurez en m'accompagnant et vous découvrirez bien d'autres choses... Idées de cet homme illustre concernant les forêts, l'agriculture ainsi que les plantes sauvages utilisées en son siècle. Temps convivial en fin de balade autour de saveurs végétales ou florales.

7 euros par personne

Découvertes autour du lac de ST AGNAN

Jeudi 17 juillet 08

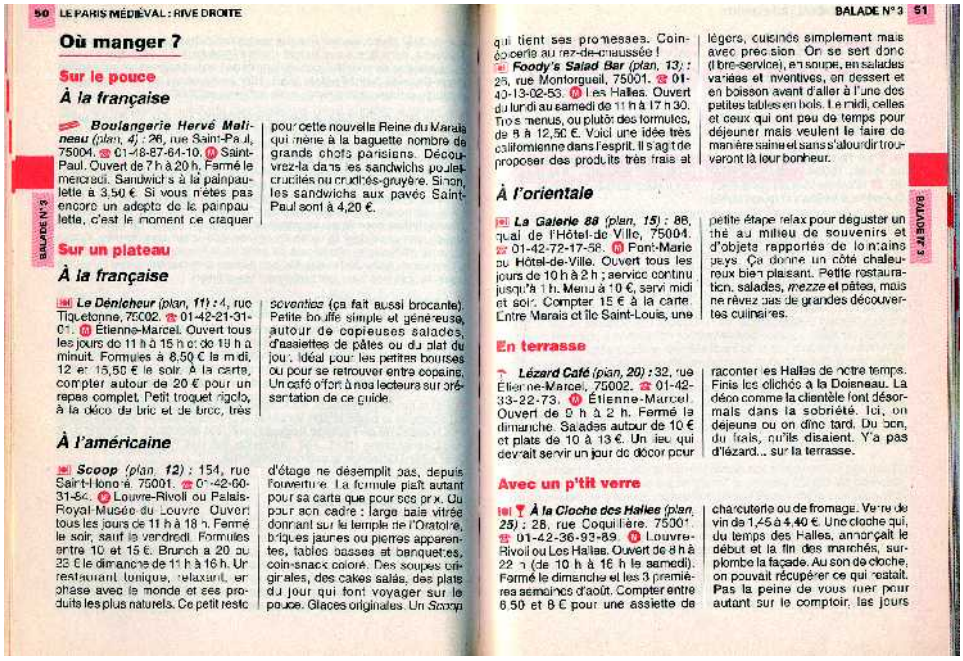
Rendez vous devant l'accueil du village club »Le Bois du Loup » à St Agnan en Morvan (58) à 15h, durée : environ 3h

Figure 5 : Page Web d'un guide de promenades

Les quatre exemples de pages Web donnés sont typiques de ce que l'on peut rencontrer. Les pages des figures 2 et 4 sont relativement simples : elles présentent un objet touristique et ne donnent que les informations utiles : adresse, nom, horaires d'ouverture. La page de la figure 3 est déjà un peu plus complexe : elle contient un encadré coloré, des logos, et le texte est plus long que dans les deux autres pages. La page de la figure 5, quant à elle, est simple dans la disposition, mais au niveau du contenu, elle mentionne plusieurs objets touristiques, ce qui la rend beaucoup plus complexe pour un traitement automatique. Elle contient en effet plusieurs horaires qui devront donc, chacun, être associés à l'objet correspondant.

2.1. Comparaison avec des guides papier

De par leur caractère informatif et pratique, les pages Web qui constituent mon objet d'étude se rapprochent donc, au niveau du contenu, de certains extraits de guides touristiques papier ; une page présentant un objet touristique correspond à l'article ou à l'entrée présentant un lieu dans le guide papier. Mais au niveau de la forme et du type d'expressions temporelles qui s'y trouvent, de nombreuses différences émergent. La figure 6 montre un extrait d'un guide touristique papier [Routard Paris 2007].



50 LE PARIS MÉDIÉVAL : RIVE DROITE

Où manger ?

Sur le pouce

À la française

Boulangerie Hervé Malineau *(plan, 4)* : 26, rue Saint-Paul, 75004. ☎ 01-48-87-64-10. Ⓜ Saint-Paul. Ouvert de 7 h à 20 h. Fermé le mercredi. Sandwichs à la pain-paulette à 3,50 €. Si vous n'êtes pas encore un adepte de la pain-paulette, c'est le moment de craquer pour cette nouvelle Reine du Marais qui mène à la baguette nombre de grands chefs parisiens. Découvrez-la dans les sandwichs poulet-crudités ou crudités-gruyère. Sinon, les sandwichs aux pavés Saint-Paul sont à 4,20 €.

Sur un plateau

À la française

Le Dénicheur *(plan, 11)* : 4, rue Tiquetonne, 75002. ☎ 01-42-21-31-01. Ⓜ Etienne-Marcel. Ouvert tous les jours de 11 h à 15 h et de 19 h à minuit. Formules à 8,50 € le midi, 12 et 15,50 € le soir. À la carte, compter autour de 20 € pour un repas complet. Petit troquet rigolo, à la déco de bric et de broc, très soventes (ça fait aussi brocante). Petite bouffe simple et généreuse, autour de copieuses salades, d'assiettes de pâtes ou du plat du jour. Idéal pour les petites bourses ou pour se retrouver entre copains. Un café offert à nos lecteurs sur présentation de ce guide.

À l'américaine

Scoop *(plan, 12)* : 154, rue Saint-Honoré, 75001. ☎ 01-42-60-31-54. Ⓜ Louvre-Rivoli ou Palais-Royal-Musée du Louvre. Ouvert tous les jours de 11 h à 18 h. Fermé le soir, sauf le vendredi. Formules entre 10 et 15 €. Brunch à 20 ou 23 € le dimanche de 11 h à 16 h. Un restaurant tonique, relaxant, en phase avec le monde et ses produits les plus naturels. Ce petit resto d'étage ne désemplit pas, depuis l'ouverture. La formule plaît autant pour sa carte que pour ses prix. Du pour son cadre : large baie vitrée donnant sur le temple de l'Oratoire, briques jaunes ou pierres apparentes, tables basses et banquettes, coin-snack coloré. Des soupes originales, des cakes salés, des plats du jour qui font voyager sur le pouce. Glaces originales. Un *Scoop* qui tient ses promesses. Coin-épicerie au rez-de-chaussée !

Foody's Salad Bar *(plan, 13)* : 25, rue Montorgueil, 75001. ☎ 01-40-13-02-53. Ⓜ Les Halles. Ouvert du lundi au samedi de 11 h à 17 h 30. Trois menus, ou plutôt des formules, de 9 à 12,50 €. Voici une idée très californienne dans l'esprit. Il s'agit de proposer des produits très frais et légers, cuisinés simplement mais avec précision. On se sert donc (libre-service), en soupe, en salades variées et inventives, en dessert et en boisson avant d'aller à l'une des petites tables en bois. Le midi, celles et ceux qui ont peu de temps pour déjeuner mais veulent le faire de manière saine et sans s'alourdir trouveront là leur bonheur.

À l'orientale

La Galerie 88 *(plan, 15)* : 88, quai de l'Hôtel-de-Ville, 75004. ☎ 01-42-72-17-58. Ⓜ Pont-Marie ou Hôtel-de-Ville. Ouvert tous les jours de 10 h à 2 h ; service continu jusqu'à 1 h. Menu à 10 €, servi midi et soir. Compter 15 € à la carte. Entre Marais et île Saint-Louis, une petite étape relax pour déguster un thé au milieu de souvenirs et d'objets rapportés de lointains pays. Ça donne un côté chaleureux bien plaisant. Petite restauration, salades, *mezze* et pâtes, mais ne rêvez pas de grandes découvertes culinaires.

En terrasse

Lézard Café *(plan, 20)* : 32, rue Étienne-Marcel, 75002. ☎ 01-42-33-22-73. Ⓜ Étienne-Marcel. Ouvert de 9 h à 2 h. Fermé le dimanche. Salades autour de 10 € et plats de 10 à 13 €. Un lieu qui devrait servir un jour de décor pour raconter les Halles de notre temps. Finis les clichés à la Doisneau. La déco comme la clientèle font désormais dans la sobriété. Ici, on déjeune ou on dîne tard. Du bon, du frais, qu'ils disaient. Y'a pas d'lézard... sur la terrasse.

Avec un p'tit verre

À la Cloche des Halles *(plan, 25)* : 28, rue Coquillière, 75001. ☎ 01-42-36-93-89. Ⓜ Louvre-Rivoli ou Les Halles. Ouvert de 8 h à 22 h (de 10 h à 16 h le samedi). Fermé le dimanche et les 3 premières semaines d'août. Compter entre 6,50 et 8 € pour une assiette de charcuterie ou de fromage. Verre de vin de 1,45 à 4,40 €. Une cloche qui, du temps des Halles, annonçait le début et la fin des marchés, surplombe la façade. Au son de cloche, on pouvait récupérer ce qui restait. Pas la peine de vous lever pour autant sur le comptoir, les jours

BALADE N°3 51

Figure 6 : Extrait du Guide du Routard Paris balades

Dans les guides touristiques papier, les entrées sont structurées de façon très régulière. Cette structure se remarque à deux niveaux : au sein d'une entrée particulière, les informations sont structurées, elles apparaissent dans un ordre précis, identique pour toutes les entrées d'une même catégorie ; au niveau supérieur, les entrées sont regroupées et catégorisées, on ne mélange pas des hôtels et des restaurants par exemple. Le guide dans son ensemble peut se rapprocher de pages Web qui présentent plusieurs objets touristiques tandis que chaque entrée peut se rapprocher de pages Web présentant un objet touristique particulier. Je me suis intéressée à ce second rapprochement et j'ai comparé deux des entrées présentées dans ce guide avec les pages Web[9] de ces objets.

Dans ce guide papier, l'organisation est marquée : on peut voir le titre *Où manger ?* qui indique qu'il s'agit de restaurants. Des sous-catégorisations apparaissent ensuite avec le S*ur le pouce*, *sur un plateau*, *en terrasse* et *Avec un p'tit verre*, et encore un niveau en dessous avec *À la française*, *À l'américaine* et *À l'orientale*. Pour chaque entrée, on trouve ensuite les mêmes informations, ordonnées de la même manière : le nom (en gras), la localisation sur le plan (entre parenthèses) puis l'adresse, le numéro de téléphone et la station de métro (indiqués par un petit logo). On trouve ensuite les horaires d'ouverture suivis d'informations complémentaires : le prix, le type de plats, le décor, etc.

9 http://www.chezfoodys.com/ ; http://caxiote.free.fr/



Figure 7 : Page Web du restaurant Foody's



Figure 8 : Page Web du restaurant Scoop

Les pages Web ci-dessus ne sont pas du tout structurées d'une telle manière. Premièrement, chaque page étant autonome et individuelle, on perd nécessairement la structuration de premier niveau : l'information « cette page est la page d'un restaurant » n'est pas clairement énoncée. Deuxièmement, il n'y a pas de régularité entre les différentes pages Web. Les informations sont présentées avec des structures différentes, un même schéma n'est pas respecté. Les informations présentées dans le guide papier n'apparaissent pas toutes dans les pages Web. En effet, si ces deux exemples mentionnent l'adresse, les horaires ne sont pas indiqués. De plus, le type d'objet n'est pas clairement énoncé. Dans la page de *Foody's*, on ne trouve que très peu d'informations : l'illustration est le premier moyen qui permet à l'internaute de savoir qu'il s'agit d'un restaurant. Dans le cas de *Scoop*, en plus du texte de la « bulle » centrale, c'est le texte en anglais en haut à droite qui fournit cette information : *All-Fresh All-HomeMade Always Lunch . Brunch . Dîner*. Ces pages Web n'étant pas issues d'un guide et ne présentant pas toutes les informations mentionnées dans le guide papier, l'objet n'est pas de réaliser une comparaison détaillée. Toutefois, ces pages permettent de montrer que les procédés utilisés pour accéder à l'information ne sont pas les mêmes sur le Web ou sur le papier.

Dans les guides papier, la date d'édition, en général l'année, a son importance et figure en gros sur la couverture. Elle poursuit alors l'objectif d'*ancrage* visant à montrer au lecteur que l'information est actuelle. Les expressions temporelles que l'on trouve à l'intérieur des guides informent sur l'objet qu'elles concernent : date d'un festival annuel, heures et jours d'ouverture d'un musée. Leur valeur vient de leur fiabilité ; ces indications temporelles se veulent informatives. De plus, pour que le guide papier reste crédible dans la durée, les informations temporelles qu'il contient doivent être stables et absolues. Au contraire, dans les pages Web, les informations temporelles sont souvent déictiques.

La différence entre pages Web et guide papier apparaît aussi sous un autre angle en ce qui concerne les expressions temporelles ; il s'agit du facteur d'homogénéité. En effet, un guide papier suit une certaine ligne éditoriale et l'ensemble des expressions temporelles qu'il contient sont cohérentes et homogènes, respectant un même schéma. Au contraire, sur des pages Web, ces informations sont appelées à être dynamiques, à évoluer en fonction du moment de consultation de la page et sont donc beaucoup moins homogènes. Dans les exemples donnés ci-dessus, les informations temporelles sont bien structurées de la même manière dans les deux entrées du guide papier : *Ouvert tous les jours de 11h à 18h. Fermé le soir sauf le vendredi* et *Ouvert du lundi au samedi de 11h à 17h30*.

Dans les guides touristiques papier, l'information est donc bien plus structurée et standardisée que dans des pages Web touristiques, qui sont très variées et ne respectent pas un modèle de page unique. Dans un guide, toutes les entrées correspondant au même type d'objet touristique sont structurées de la même manière et contiennent le même type d'informations. Des marqueurs linguistiques permettent de repérer chaque information à extraire. De plus, la ponctuation est très rigoureuse, ce qui facilite grandement l'interprétation, surtout si celle-ci est automatisée. Les points et points-virgules sont bien utilisés : les points séparent les informations qui sont de natures différentes tandis que les points-virgules séparent les différentes informations de même type (les différents prix par exemple). Dans les pages Web, au contraire, la ponctuation – s'il y en a – est rarement utilisée à bon escient. La plupart du temps, ce sont des espaces blancs ou des « lignes sautées » qui servent de séparateurs.

Bien sûr, l'extraction d'information dans des documents contenant du texte rédigé n'est pas

triviale, mais il faut souligner qu'il existe déjà de nombreux outils permettant de faciliter l'extraction automatique : par exemple, des outils d'analyse syntaxique partielle, de chunking ou de parsing sont disponibles. Il subsiste toutefois des difficultés, comme la résolution d'anaphores, qui rendent l'analyse automatique de textes complexe. Ce qui m'intéresse, ce sont les difficultés que l'on peut rencontrer lors de l'analyse automatique de pages Web et celles-ci sont très différentes des difficultés que présente une analyse automatique de texte rédigé. Par exemple, dans *Le guide du routard* [Routard Paris 2007], comme illustré dans la figure 6, l'entrée de chaque restaurant contient son nom suivi de son adresse, son numéro de téléphone et sa station de métro puis ses indications d'ouverture comme *ouvert tous les jours*. Une fois le format identifié, chaque entrée est facile à analyser automatiquement. Le problème est que, sur une page Web, le même type d'information prendrait plutôt la forme présentée sur la page Web de la figure 9.

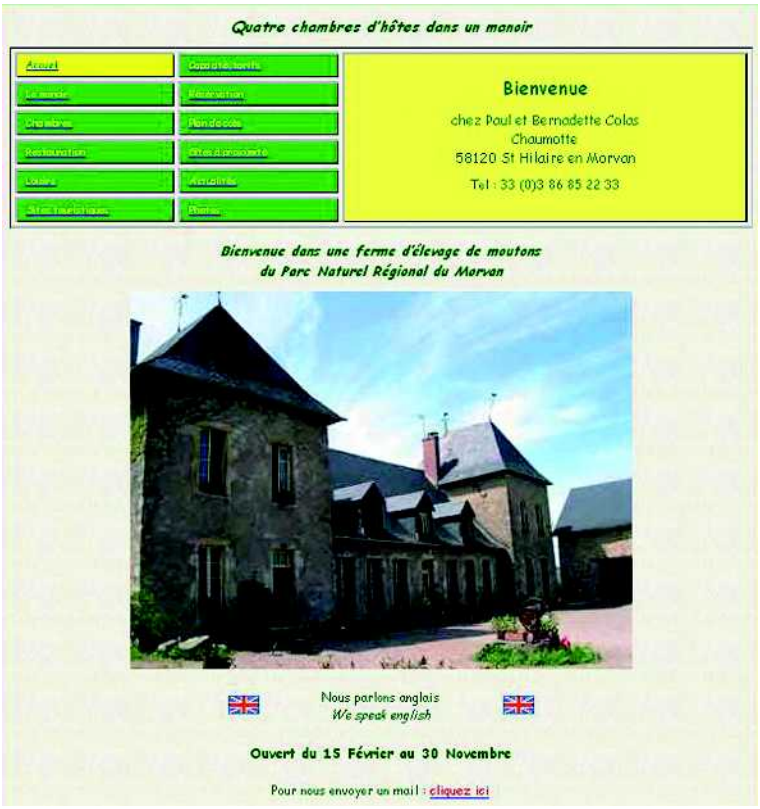


Figure 9 : Page Web d'une ferme-auberge

Pour l'instant, aucun outil n'est capable d'interpréter que *Chez Paul et Bernadette Colas* est le nom d'une auberge ou que *Ouvert du 15 Février au 30 Novembre* correspond aux informations d'ouverture de cette auberge en particulier. C'est pour cela que les méthodes d'extraction d'information dans des textes structurés ne peuvent pas être directement appliquées à des pages Web, dans lesquelles la syntaxe n'est souvent pas conforme à la norme et où la structure n'est pas standardisée.

Mon travail est proche de [Tenier et al. 2006a] dont l'objectif est d'extraire, dans des pages Web, les informations concernant des équipes de recherche. Mon approche est plus générale

en ce sens que les pages que ces auteurs analysent respectent des règles de disposition spatiale, contrairement aux pages Web que j'étudie, qui sont entièrement libres. Je me rapproche également de [Bry et al. 2003], qui ont travaillé sur des modules de raisonnement temporel et de localisation pour des pages Web. Mais la différence entre leur travail et le mien est que ces auteurs travaillent sur des documents XML, dans lesquels l'information est déjà sémantiquement structurée ; les expressions temporelles sont déjà marquées et n'ont donc pas à être repérées.

Le langage TimeML[10] est un langage de marquage d'expressions temporelles et d'événements. J'ai choisi de ne pas m'appuyer sur ce langage car, pour les besoins du projet Eiffel et du présent travail, ses annotations sont trop riches et elles ne sont pas spécifiques à un domaine. En effet, ce format permet d'annoter en détail des expressions temporelles, de les ordonner et de faire certains raisonnements, mais cela ne répond pas aux besoins du projet. Par ailleurs, TimeML a été conçu pour annoter des textes rédigés puisque ses principales fonctions sont d'ancrer temporellement les événements, de les ordonner les uns par rapport aux autres, de faire des raisonnements sur des expressions temporelles sous-spécifiées (par exemple *la semaine dernière*) et de faire des raisonnements sur la durée des événements. Les informations que je cherche à annoter ne concernent pas des événements (sauf par exemple pour les annonces de spectacles). Mais surtout, elles ne prennent pas place dans un discours, sous la forme d'un texte narratif dans lequel plusieurs événements seraient mentionnés. Si plusieurs expressions temporelles sont présentes dans les pages Web que je traite, comme les horaires de deux piscines, celles-ci n'ont aucun lien entre elles. De plus, ces informations sont spécifiques au domaine du tourisme et ont donc des caractéristiques propres qu'il serait dommage d'ignorer. Elles prennent par exemple la forme de tranches horaires journalières mais on ne trouvera jamais, au sujet d'un objet touristique, *à la veille de la naissance de Jean*. La plupart des expressions traitées auraient pu être annotées à l'aide de TimeML, mais cela aurait considérablement alourdi les résultats sans que les spécificités de ce format soient exploitées. Par ailleurs, cela n'aurait pas permis de remplir tout à fait les besoins du projet Eiffel, pour lequel il ne suffit pas de décrire les expressions temporelles. En effet, il faut également les interpréter pour les typer en tant qu'ouvertures ou fermetures par exemple, et cela n'est pas prévu dans TimeML. C'est donc à un niveau plus sémantique que le format d'annotation doit être mis au point.

La différence entre mes travaux et la plupart des travaux sur le traitement du temps est que ces derniers s'appuient sur le traitement des événements. Les expressions temporelles sur lesquelles je travaille ne définissent pas des événements, au sens de [Davidson 1969]. Ainsi, dans [Gayral & Grandemange 1992], on peut lire :

> « Nous proposons une ontologie temporelle adaptée au traitement des différents phénomènes temporels apparaissant dans des énoncés en langage naturel. II s'agit de rendre compte des événements, de leur durée, de leur localisation et aussi des relations entre ces événements, qu'elles soient purement temporelles (d'inclusion, de précédence...) ou qu'elles correspondent à des liens de structuration (relation d'un événement à ses sous-événements) ou de causalité. » (p.296)

Cet amalgame entre temporalité et événements est souvent effectué, mais ne peut pas s'appliquer aux expressions étudiées ici. En effet, les expressions temporelles que je prends en compte ne servent pas à définir des événements.

10 http://www.timeml.org

2.2. Comment nomme-t-on dans une page Web ?

Une autre différence majeure entre pages Web et guides touristiques papier n'a pas été mentionnée dans le point précédent. Elle concerne la façon de nommer et de typer les objets touristiques dont il est question. En effet, dans un guide papier, il est annoncé dès la couverture de quoi il va être question : par exemple *Guide chambre d'hôte prestige*. En revanche, tel n'est pas le cas dans une page Web. Si l'objet est parfois annoncé comme dans un guide papier, c'est loin d'être toujours le cas. De plus, il ne figure pas à un endroit précis qui pourrait être apparenté à la couverture du livre, il peut apparaître n'importe où dans la page. Parfois le type de l'objet n'apparaît même pas et seul le contexte permet de l'identifier. Par exemple, un site présentant un restaurant pourra ne contenir que son nom *La fourchette*, son adresse et ses jours ou horaires d'ouverture. En premier lieu, la sémantique du nom de l'objet touristique suggérera à l'internaute qu'il s'agit d'un restaurant. En deuxième lieu, les moments d'ouverture peuvent le confirmer : *ouvert tous les jours, midi et soir, sauf le dimanche*. De tels horaires ne peuvent en effet que difficilement concerner un objet touristique d'un autre type. Enfin, en troisième lieu, une image, photo ou illustration, peut figurer sur la page et ainsi donner une indication à l'internaute, si elle représente par exemple une salle de restaurant ou sa terrasse, comme c'est le cas dans la page de la figure 7. Un internaute n'aura donc aucun mal à interpréter un tel site, mais il n'en est pas de même pour un traitement automatique.

2.3. La théorie sémiotique et son application aux pages Web

Jusqu'à maintenant, je me suis principalement intéressée aux contenus textuels des pages Web. Or, les pages Web sont loin de ne contenir que du texte. Elles contiennent aussi de nombreux logos et images, et surtout la mise en page, la mise en forme, la disposition, les agencements et les couleurs y font sens. En plus de relever de la linguistique, ces objets relèvent donc aussi de la sémiotique, science qui consiste à étudier la signification. Voici la définition qu'en donne Joly, qui distingue, à la suite des travaux de Saussure [Saussure 1974] et Peirce [Peirce 1978], ce terme de son voisin « sémiologie » :

> « [Les termes « sémiotique » et « sémiologie »] ne sont pas pour autant synonymes : le premier, d'origine américaine, est le terme canonique qui désigne la sémiotique comme philosophie des langages. L'usage du second, d'origine européenne, est plutôt compris comme l'étude de langages particuliers (image, gestuelle, théâtre, etc.). » ([Joly 1993], p. 22)

Mais dans l'usage, le terme *sémiotique* tend à se généraliser. L'extrait suivant de l'ouvrage *Sémiotique des langages d'icônes* ([Vaillant 1999]), précise cette tendance :

> « La théorie générale des signes a été baptisée *sémiologie* par Saussure, ou plus près de nous par Buyssens, Mounin, Barthes, et même encore par Eco en 68, avant que l'usage n'entérine la collision de ce terme avec celui de sémiotique, d'origine anglo-saxonne (Locke, Peirce) (cf. Rastier [1997], annexe[11]). Aujourd'hui, le second terme prédomine dans ce sens. Il fallait donc que le premier se cantonne dans un sens plus spécialisé ; ce fut celui de la description spécifique de systèmes de signes particuliers (cf. Joly [1994, pp. 16--18[12]). Comme le fait d'ailleurs remarquer Eco (1968)[13], cet emploi est déjà

11 François Rastier, 1997. « Problématiques du signe et du texte ». Intellectica.

12 Martine Joly, 1994. *L'image et les signes. Approche sémiologique de l'image fixe*. Paris : Nathan (coll. « Fac/Image »).

13 Umberto Eco, 1968. *La structure absente. Introduction à la recherche sémiotique*. Paris : Mercure de France, 1984. (trad. fr. de *La struttura assente. La ricerca semiotica e il metodo strutturale*. Milan : Bompiani, 1968).

> contenu dans celui, plus précis, de Hjelmslev, pour qui une sémiologie est une sémiotique dont le plan du contenu est lui-même une sémiotique. Cette distinction est d'une certaine manière reflétée ici. D'une démarche plus consciente, nous avons voulu, dans l'expression « système sémiologique » par exemple, introduire entre *sémiotique* et *sémiologique* la même nuance que celle qui existe entre *phonétique* et *phonologique* (on aurait dit en anglais ``*semiomics*'', suivant la distinction `*etic*'/`*emic*' chère à Eco) : une nuance entre la science de la substance et celle de la forme. »

Je suivrai la tendance en utilisant donc de préférence le terme *sémiotique*.

Pour entrer plus concrètement dans la sémiotique, je vais étudier les différents types de signes que l'on rencontre dans les pages Web, la façon dont ils interagissent et la façon dont ils peuvent être produits et interprétés. En suivant Joly, je vais reprendre les théories de Saussure et de Peirce et sa classification des signes. Mais de quoi relèvent les signes qui permettent d'établir les relations entre les différentes parties d'une page Web et les agencements ? Comment certains agencements peuvent-ils faire sens ?

Saussure a choisi la linguistique, étude systématique de la langue, comme pilier de la « sémiologie », « science générale des signes » [Saussure 1974]. Il a donc tenté d'isoler les unités minimales de signification dans la langue, les monèmes. Pour lui, chaque signe linguistique est constitué de deux éléments indissociables : le signifié et le signifiant ; le signifiant (image acoustique d'un mot) étant arbitrairement lié au signifié (concept – représentation mentale d'une chose). Cela est représenté par le schéma suivant :

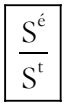


Figure 10 : Schéma du signe selon Saussure

Saussure a également développé les outils méthodologiques de permutation, opposition et commutation, largement utilisés en linguistique, mais également adaptables à d'autres systèmes de signes que la langue.

Peirce, quant à lui, a essayé de mettre au point une théorie générale des signes et une typologie très générale dans une perspective plus large. La langue en fait partie mais sans être mise en avant. Voici un extrait de la théorie de Peirce, telle qu'elle est exposée dans [Joly 1993] :

> « Un signe a une matérialité que l'on perçoit avec l'un ou plusieurs de nos sens. [...] cette chose que l'on perçoit tient lieu de quelque chose d'autre : c'est la particularité essentielle du signe : être là, présent, pour désigner ou signifier autre chose, d'absent, concret ou abstrait. [...]
>
> Pour Peirce, un signe est « quelque chose, tenant lieu de quelque chose pour quelqu'un, sous quelque rapport, ou à quelque titre ».
>
> Cette définition a le mérite de montrer qu'un signe entretient une relation solidaire entre trois pôles au moins (et non plus seulement deux comme chez Saussure) : la face perceptible du signe : « representamen » ou signifiant (S^t), ce qu'il représente : « objet » ou référent, et ce qu'il signifie : « interprétant » ou signifié ($S^é$). » ([Joly 1993], p. 25-26)

Voici la représentation de cette triangulation :

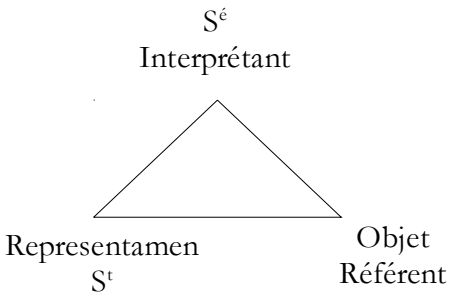


Figure 11 : Schéma du signe selon Peirce

Joly, dans *Introduction à l'analyse de l'image*, distingue trois types de signes : les signes iconiques, les signes plastiques et les signes linguistiques. Les signes iconiques correspondent à des « images », au sens théorique du terme. Les signes plastiques couvrent les textures, couleurs, formes et composition interne. Enfin, les signes linguistiques correspondent au langage verbal.

Ces différents types de signes se retrouvent aussi bien dans les pages Web que j'étudie que dans les images ou publicités étudiées par Joly. Mais peut-être ne suffisent-ils pas à les caractériser. D'autres types de signes et surtout la façon dont ces signes interagissent entre eux pour créer du sens ne sont pas explicités.

En parlant de la composition, Joly écrit :

> « En ce qui concerne l'image publicitaire, la composition est étudiée de manière telle « que le regard sélectionne dans l'annonce les surfaces porteuses des informations clés » [Note : « Georges Péninou, « Physique et métaphysique de l'image publicitaire », in *Communications*, n°15, Seuil, 1970. »], d'autant plus que l'on sait qu'il existe des modèles, ou *patterns*, de lecture qui ne confèrent pas la même valeur aux différents emplacements de la page. » ([Joly 1993], p. 85)

Ce type de procédé est sans aucun doute utilisé sur des pages Web mais d'autres sont aussi présents. Par exemple, si deux blocs d'information sont mis en forme de la même manière (même police de caractère, même couleur), on peut souvent en déduire qu'ils sont liés : qu'ils concernent le même objet ou sont de même nature. S'il est possible de faire cette déduction, c'est que quelque chose donne une signification à l'interaction qui a lieu entre le message linguistique et les signes plastiques : la combinaison des deux peut être considérée comme un signe. Aucun des signes présentés par Joly n'a cette propriété. Le Web donne donc lieu à de nouveaux types de signes qui ne figurent pas dans les affiches ou publicités. Ces signes ont été étudiés dans les travaux menés par Rastier et Valette dans le cadre du projet PRINCIP [Valette 2004], visant la détection de sites racistes. Ils se sont notamment appuyés sur le rôle des polices de caractères, couleurs et autres informations de mise en forme. Ils montrent ainsi que le choix des couleurs ou polices peut constituer un indice pour la classification des sites. Toutefois, si visuellement sur une page Web, la combinaison du message linguistique et des signes plastiques peut être significative, la problématique que je pose est que, sur un plan technique, les deux sont disjoints (voir sur ce point le chapitre 3).

2.4. Éléments d'une page Web

Un document de texte rédigé, imprimé ou non, se lit de façon ordonnée, de la première à la dernière ligne. Même s'il peut être constitué de plusieurs paragraphes, une partie de ce texte

isolée n'est normalement pas compréhensible ; elle ne se suffit pas à elle-même. En revanche, une page Web se consulte. Elle est constituée de différents éléments plus ou moins indépendants les uns des autres. Ces éléments sont la plupart du temps déplaçables dans la page Web, et cela sans nuire à sa compréhension. S'il y a également des images dans les pages Web, je ne parle ici que des éléments textuels. J'entends par *élément* une zone de texte formant une unité de sens, comme par exemple une adresse ou les horaires.

J'utilise la page Web de la figure 9 pour faire une démonstration. Dans la figure 12, j'ai encadré ce que je considère comme des blocs significatifs.

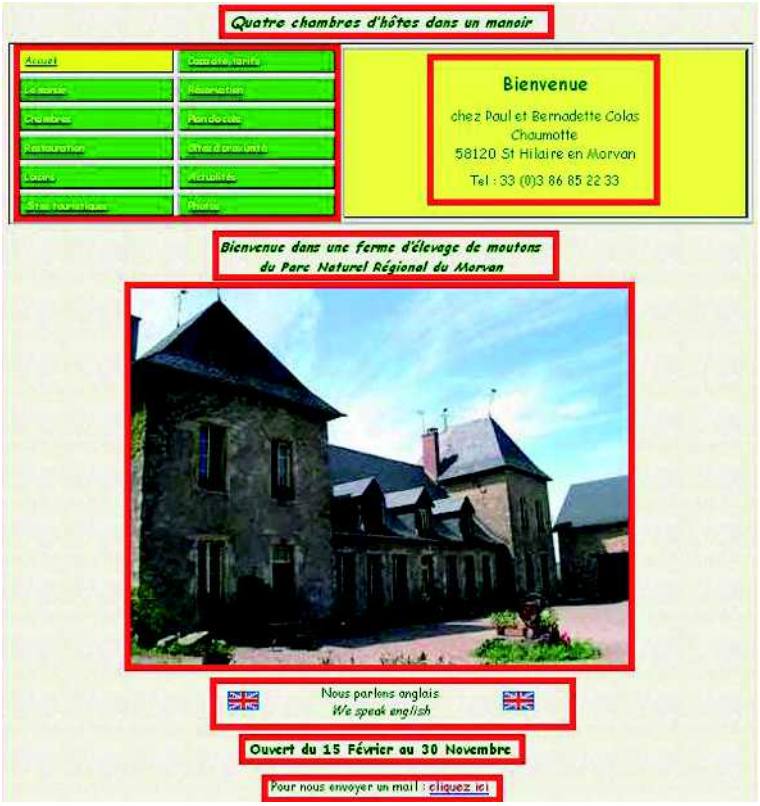


Figure 12 : Page Web dans laquelle les blocs significatifs ont été marqués

Dans cette page, chaque bloc encadré représente donc une unité de signification. Les blocs significatifs ont été délimités de manière intuitive, d'autres découpages étant probablement possibles. Je vais montrer que ces blocs peuvent être déplacés dans la page sans nuire à sa compréhension. Voici donc deux exemplaires de la même page, dans lesquels certains blocs ont été déplacés.



Figure 13 : Autre disposition pour le même contenu - première possibilité

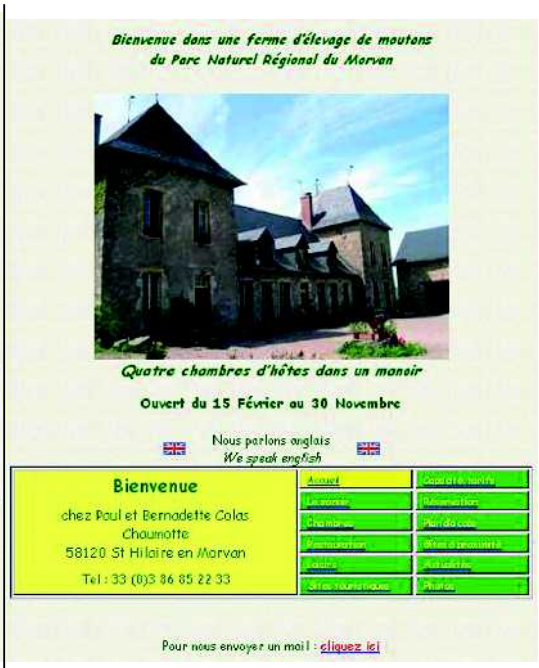


Figure 14 : Autre disposition pour le même contenu - deuxième possibilité

Il est intéressant de noter que le choix des blocs significatifs est important, car si j'avais déplacé des blocs plus petits, l'unité de la page Web et sa compréhensibilité n'auraient pas été respectées. Ainsi, dans l'exemple suivant, des blocs plus petits ont été déplacés et la page Web perd de sa cohérence. Dans cette page, au niveau du code HTML, la plus petite unité considérée comme un bloc est un paragraphe (contenu de la balise <p>) ; il ne s'agit pas de jouer avec la syntaxe du français en déplaçant des morceaux de phrases, mais bien de manipuler la syntaxe du HTML, en déplaçant des blocs balisés. Cependant, étant donné la liberté que laisse le HTML au concepteur de la page Web, les blocs balisés ne correspondent pas nécessairement à des blocs significatifs et, inversement, un bloc significatif n'est pas forcément exactement encadré par une paire de balises. Si les manipulations que j'ai effectuées touchent directement à la syntaxe du code HTML, elles s'appuient tout de même sur le sens du contenu textuel de la page pour délimiter les blocs significatifs.

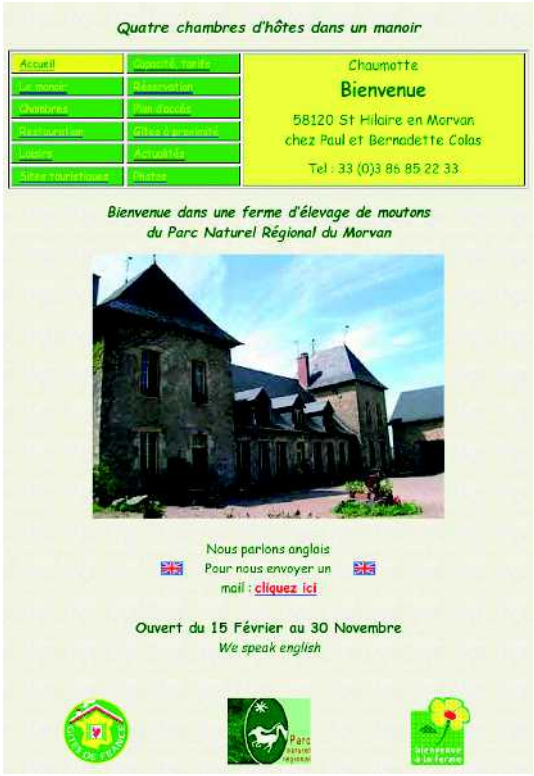


Figure 15 : Autre disposition pour le même contenu - troisième possibilité – mais avec perte de cohérence

Tout d'abord, dans la grande case du premier tableau, l'adresse n'apparaît plus clairement. De plus, au niveau de la mention *We speak english*, celle-ci n'apparaît plus à proximité de sa traduction en français, et surtout elle est loin du logo (petit drapeau britannique) qui lui correspond. On ne peut pas dire que cela empêche totalement la compréhension de la page, mais cela la complique et la rend moins naturelle.

J'aurais pu aller encore plus loin en faisant « exploser » le tableau, et en répartissant, dans la page, les différentes cases qui le composent. Néanmoins, si la dernière page proposée perd déjà de sa compréhensibilité, il est certain que sans le tableau, le menu ne serait plus exploitable.

La limite est donc bien l'unité considérée comme minimale : jusqu'à un certain point, les éléments qui composent la page sont déplaçables à volonté, sans que cela nuise à la compréhension globale. Mais si ces éléments minimaux sont encore divisés, alors ce sont ces éléments eux-mêmes qui perdent leur sens et la page, par voie de conséquence, aussi.

Conclusion

En présentant les objets sur lesquels cette thèse s'est concentrée, ce chapitre a permis de mettre en avant les problématiques qui sont développées dans cette thèse.

En premier lieu, les expressions temporelles touristiques montrent une grande variété et ne

correspondent pas à des expressions calendaires au sens classique du terme. Le chapitre suivant présentera plus en détail ces expressions, à l'aide d'une étude linguistique.

En second lieu, les données sur lesquelles j'ai travaillé se trouvent sur des pages Web, et cela joue un grand rôle. Deux problématiques sont à poser. La première est que les textes écrits sur des pages Web ne respectent pas toujours les standards du français : grammaire, orthographe, ponctuation. De plus, il ne s'agit pas souvent de textes rédigés mais plutôt de fragments, surtout lorsqu'il s'agit d'énoncer les informations pratiques. La seconde est liée au statut même de la page Web en tant qu'objet technique. Celle-ci a en effet deux facettes : l'une que constitue la page Web telle qu'affichée dans un navigateur que l'internaute peut consulter ; l'autre que constitue le code source, sur lequel j'ai travaillé. La difficulté vient du fait que le lien entre le rendu visuel d'une page affichée dans un navigateur et son code source n'est pas très marqué : pour un même rendu visuel, plusieurs codes sources sont possibles. Or, pour interpréter une page affichée, l'internaute s'appuie également sur la représentation visuelle de la page.

CHAPITRE 2. ÉTUDE LINGUISTIQUE

Introduction

Ce chapitre a pour but de présenter, sur un plan linguistique, les expressions sur lesquelles j'ai travaillé. Les expressions qu'Adetoa doit repérer et annoter (qui sont présentées plus en détail aux chapitres 4 et 5) en font partie. Une classification théorique des expressions temporelles est proposée ici. Les expressions de localisation et de typage des objets touristiques sont également sommairement présentées.

Dans la seconde partie du chapitre, l'accent est mis sur les problèmes d'interprétation que peuvent poser les expressions temporelles et les solutions qui ont été choisies dans le cadre du projet Eiffel.

1. Caractérisation et classification des expressions

Les informations auxquelles je me suis intéressée dans le corpus de pages Web touristiques du projet Eiffel sont de trois types. Il s'agit d'expressions temporelles et d'expressions spatiales, ainsi que d'expressions dénotant des objets touristiques. Ces informations pratiques sont utiles et souvent nécessaires dans le monde du tourisme, notamment pour la création d'une plateforme touristique telle celle réalisée dans le cadre du projet Eiffel. Les expressions temporelles étant les plus complexes et les plus intéressantes sur un plan linguistique, ce sont elles qui constituent le cœur de cette étude.

1.1. Expressions temporelles

Les expressions temporelles auxquelles je me suis intéressée ont une visée informative et pratique. Il ne s'agit pas de dates historiques ou d'expressions descriptives du type *la nuit d'avant*, mais d'informations pratiques dans le domaine du tourisme. Il peut ainsi s'agir d'horaires d'ouverture ou de fermeture, de dates, de périodes, etc., chaque information temporelle étant liée à un objet touristique.

Les expressions temporelles touristiques peuvent être génériques ou propres à un type objet en particulier. Par exemple, une date comme *le 31 octobre 2008* peut aussi bien convenir à un concert qu'à une représentation de théâtre ou à l'ouverture d'une patinoire. En revanche, une expression comme *ouvert midi et soir sauf le lundi* peut difficilement s'appliquer à autre chose qu'à un restaurant.

Afin de présenter un panorama assez complet des différentes expressions concernées, j'ai

choisi de les classifier selon le type d'objet auquel elles se rapportent. Ainsi, après avoir défini cinq catégories d'objets touristiques, je présente, pour chacune, les expressions temporelles qui s'y rapportent. Cela permet de mettre en avant certaines caractéristiques des expressions, qui sont en général assez proches lorsqu'elles concernent un même type d'objet. Ces cinq catégories sont les suivantes : la première regroupe les pages d'offices de tourisme, mairies, magasins, etc. ; la deuxième les hébergements ; la troisième comprend les restaurants ; la quatrième les manifestations touristiques et, enfin, la dernière les « pages-agenda ». Ces catégories ont émergé à la suite d'une étude de corpus ; les regroupements effectués sont fonction du type d'expressions temporelles qui peuvent concerner les objets de chacune des catégories.

1.1.1. « Office de tourisme / mairie / magasin »

La catégorie « Office de tourisme / mairie / magasins » regroupe les sites d'offices de tourisme, de mairies, mais aussi les sites de magasins, de musées, d'activités ou de lieux à visiter. Elle regroupe plus précisément les pages dont les expressions temporelles correspondent à des horaires d'ouverture (ou de fermeture) hebdomadaires, complétés parfois par des dates.

Comme on peut le voir dans les exemples suivants, les sites de mairies, d'office de tourisme ou de magasins contiennent la plupart du temps des informations d'ouverture précises avec jours et heures. Des informations plus complexes peuvent aussi apparaître : périodes de l'année, basse ou haute saison, etc. De plus, une marque lexicale du type *ouvert*, *fermé*, *horaire d'ouverture*, *horaire* est toujours présente. Cette marque est très importante car elle peut servir à déclencher le repérage de l'expression et à l'encadrer.

(18) **L'office de tourisme est ouvert du mardi au samedi, de 10h30 à 12h30 & de 14h à 17h**

(19) **Horaires d'ouverture Lundi : 13h30-17h30 Du mardi au vendredi : 8h30 -12h00 14h30-17h30 Samedi : 8h30 -12h00**

(20) **HORAIRES 2007 : Basse saison : Du 1er octobre 2006 au 31 mars 2007 : du lundi au samedi de 9h à 12h et de 14h à 18h Fermé le dimanche et les jours fériés Haute saison : Du 1er avril 2007 au 30 septembre 2007 : Du lundi au samedi de 9h à 18h30 Dimanche et jours fériés : 10hà 13h et de 15h à 18h L'Office de tourisme sera fermé les 1er janvier, 1er mai, 1er novembre, 11 novembre et 25 décembre 2007**

(21) **La bibliothèque de Lormes vous ouvre ses portes : Lundi 15h/18h , Mardi 10h/12h et 14h/19h , Mercredi 15h/18h , Jeudi 10h/12h , Vendredi 15h/18h , le 2ème et le 4ème samedi du mois de 15 h à 17 h**

(22) **Visite à la ferme Du 1er février au 30 novembre : Du lundi au samedi : 7h-12h et 14h-19h Dimanche : 7h-13h et 16h-19h30**

(23) **Ils vous ouvrent leurs caves les samedis 27 mai , 15 juillet et 12 août 2006 , de 10h à 12h et de 14h à 19h Puis les vignerons se relaient pour vous accueillir de 15h à 19h , du 03 Juillet au 31 Août 2006**

(24) **Horaires lundi mardi mercredi jeudi vendredi samedi matin 08h30 - 12h00 08h30 - 12h00 08h30 - 12h00 08h30 - 12h00 08h30 - 12h00 10h00 - 11h30 ap. midi 14h00 - 18h00 14h00 - 18h00 14h00 - 18h00 14h00 - 18h00 14h00- -**

Ces expressions apparaissent telles quelles sur les pages Web des objets touristiques, à l'exception du dernier exemple. Sur la page Web, le texte de l'expression (24) est présenté dans un tableau (figure 16), je reviendrai sur ce tableau au chapitre 3, page 68.

Horaires	lundi	mardi	mercredi	jeudi	vendredi	samedi
matin	08h30-12h00	08h30-12h00	08h30-12h00	08h30-12h00	08h30-12h00	10h00-11h30
ap. midi	14h00-18h00	14h00-18h00	14h00-18h00	14h00-18h00	14h00-	-

Figure 16 : Horaires d'ouverture présentés dans un tableau

1.1.2. « Hébergements »

La catégorie « Hébergements » regroupe tous les types d'hébergements : camping, hôtel, chambre d'hôtes, etc. Les expressions temporelles que l'on trouve dans les pages Web dédiées à des hébergements correspondent pour la plupart à des périodes d'ouverture ou de fermeture annuelles.

(25) **Ouvert du 31 Mai au 15 Octobre**

(26) **Ouvert de Mai à Octobre**

(27) **Ouverture : De mi-avril à mi-novembre. Fermé le mardi sauf jour férié**

(28) **Période de location de juin à septembre**

(29) **Ouvert toute l'année**

(30) **Ouverture du camping En principe le camping est ouvert du premier avril jusqu'au mi-novembre, mais ces dates sont surtout fonctions de la présence de campeurs sur le terrain, nous pouvons ouvrir avant et après. S'il fait beau ou sur réservation les dates d'ouverture peuvent être un peu plus amples.**

Pour les hébergements, les horaires d'ouverture sont assez standardisés (mis à part le dernier exemple). Dans de nombreuses pages Web présentant des hébergements, les expressions temporelles sont au même format que celles des exemples (25) ou (26). La plupart du temps, on trouve, dans les expressions temporelles liées à des hébergements, une période de l'année, exprimée plus ou moins précisément (granularité mensuelle ou journalière). Une nouvelle fois, des marques lexicales précises permettent le plus souvent de déclencher le repérage des expressions : *ouvert*, *ouverture*.

1.1.3. « Restaurants »

Les restaurants et autres lieux de restauration constituent une catégorie à part entière car leurs horaires d'ouverture sont particuliers. La journée n'est plus découpée en tranches « matin », « après-midi », etc., mais en « midi » et « soir ». La granularité est différente de celle des autres catégories. Les expressions temporelles associées à des restaurants indiquent des tranches horaires journalières d'ouverture, des informations hebdomadaires et parfois également des périodes d'ouverture annuelles. Les expressions indiquant les horaires des restaurants peuvent être complexes. Par exemple, pour *ouvert tous les jours sauf le mercredi midi*, il faut interpréter que, le mercredi, le restaurant est ouvert le soir.

(31) **Restaurant fermé mardi et mercredi hors saison Ouvert tous les jours en juillet et août**

(32) **Fermé le vendredi soir, le dimanche soir et le soir des jours fériés.**

(33) **Fermé du 01/02/2007 au 01/03/2007, du 01/10/2007 au 18/10/2007 et le lundi et le mardi sauf juillet et août.**

(34) **Fermé durant les fêtes de fin d'année et le mardi soir et le mercredi.**

(35) **Fermé de fin novembre à fin février. Fermé le dimanche soir, le lundi et le mardi de fin février au 21/03 et le lundi, le mardi et le mercredi du 12/11 à fin novembre.**

(36) **Fermé du 20/12/2006 au 19/01/2007 et le lundi et le mardi midi.**

(37) **Ouvert toute l'année.**

(38) **Ouverture : toute l'année, du jeudi au dimanche midi et soir sur réservation. Juillet et août tous les soirset[14] d'octobre à avril du vendredi au dimanche sauf groupes.**

Ces exemples montrent que, le plus souvent, les informations temporelles concernant les restaurants sont exprimées en fermetures. En effet, lorsqu'il s'agit d'un restaurant, il est rarement nécessaire de donner des horaires d'ouverture précis (par exemple pour le midi : 11h30-15h00 et pour le soir 19h-23h). Les personnes souhaitant s'y rendre font en effet appel à leur connaissance du monde, du domaine, ou encore à leur culture, pour estimer les horaires. En revanche, les restaurants ont généralement une ou plusieurs plages de fermeture dans la semaine (voir horaires prototypiques au chapitre 6), et ce sont elles qui sont indiquées sur les pages Web.

1.1.4. « Manifestations touristiques »

La catégorie « Manifestation touristique » comprend tout objet touristique se produisant à une date précise : concert, festival, spectacle, défilé, etc. Les informations temporelles correspondant aux manifestations touristiques sont en général précises et peuvent descendre à une granularité horaire. En revanche, elles sont souvent relatives, c'est-à-dire qu'il faut connaître l'année ou le mois en cours pour les interpréter. Nous avons vu au chapitre 1 que, sur le Web, les dates sont souvent déictiques, mais, comme les informations peuvent être renouvelées régulièrement, les pages sont ainsi toujours « à jour ».

(39) **Inauguration du Musée le 7 Juillet 2006**

(40) **Concert les vendredi 13 et samedi 14 mai**

(41) **Quartet vendredi 19 janvier 2007 20h30**

La différence que l'on peut observer entre ces expressions temporelles et celles des autres catégories est que, pour les manifestations, le nom de l'objet touristique, ou du moins son type (musée, concert), est souvent repris à proximité de l'expression. Sur un plan technique, cela pourrait contribuer à déclencher le repérage des expressions.

1.1.5. Pages mixtes

Cette catégorie est plus diversifiée : elle comprend les sites qui répertorient plusieurs objets touristiques, par exemple sous la forme d'agenda. Les manifestations sont souvent reprises dans des pages-agenda. Des dates (la plupart du temps absolues) sont toujours indiquées.

14 La faute de frappe (absence d'espace) est retranscrite telle qu'elle apparaît sur la page Web du restaurant.

Les exemples qui suivent permettent de constater que, pour les spectacles et conférences, dates et heures sont le plus souvent indiquées, tandis que pour d'autres activités, seule la date est mentionnée. L'agenda reprenant les horaires d'ouverture d'un musée adopte la même formulation que celles que l'on a pu rencontrer dans la catégorie « Office de tourisme / mairie / magasin ». Les agendas comprennent des formulations très variées, mais certaines expressions apparaissent tout de même parfois plusieurs fois au sein d'une même page, comme *rendez-vous devant X à Y heure*. De plus, sauf exception, les dates figurant dans un agenda sont absolues.

(42) **JUILLET DECOUVERTE DES PLANTES LOCALES aux environs de SAULIEU jeudi 5 juillet 07 Rendez vous l'après midi devant l'office de tourisme à 15h**

(43) **Jeudi 3 mai à Prémery Concert à 19h**

(44) **Autour du château de BAZOCHES Samedi 14 juillet 07 Rendez vous l'après midi devant l'église de Bazoches (58) à 15h**

(45) **Découvertes autour du lac de ST AGNAN Mardi 17 juillet 07 Rendez vous devant l'accueil du village club »Le Bois du Loup » à St Agnan en Morvan (58) à 15h**

(46) **VAUBAN un lieu à découvrir Samedi 21 juillet 07 Rendez vous au pied de la croix devant le château (58) à 15h**

(47) **15 AVRIL 2007 : VI ème MARCHE DE PRINTEMPS**

(48) **Prochain concert le 12 mai avec Pêcheurs de Lune**

Comme les exemples le montrent, les expressions temporelles que l'on trouve dans les pages recensant plusieurs objets touristiques sont les plus variées car elles concernent des objets touristiques de natures différentes. Ce qui les caractérise, cependant, par rapport aux autres types de pages, c'est qu'un cadre général peut être donné. Par exemple, on trouvera souvent, sur les pages de ce type, un titre comme *Concerts du mois de juin* ou *Festivals de l'été*, jouant un rôle cadratif. « Cadratif » peut être pris ici dans un sens se rapprochant, sur un plan théorique, de la notion de cadre proposée par [Charolles & Péry-Woodley 2005], et plus spécifiquement encore de la notion de cadres temporels [Le Draoulec & Péry-Woodley 2005]. En effet, si dans les travaux de ces auteurs, les cadres temporels sont le plus souvent marqués par des adverbiaux temporels, leur rôle est très similaire à celui joué par les titres de ces pages.

1.1.6. Bilan

Ce type de classification pose un certain nombre de questionnements. Tout d'abord, est-ce que toutes les pages présentant des objets touristiques et contenant des expressions temporelles trouvent une place dans cette classification ? Deuxièmement, cette classification est-elle assez fine ? Cette classification semble assez générique, pour que, en généralisant si besoin, chaque page touristique puisse y trouver sa place ; si la classification était trop fine, certaines pages risqueraient de ne plus pouvoir y trouver place. Par exemple, si, au lieu de la catégorie « hébergements », se distinguaient « hôtel », « camping », « chambre d'hôtes », « location meublée », etc., serait-il encore possible de classer « mobile-home » ?

Cette classification, qui m'a principalement servi à présenter les expressions auxquelles je me suis intéressée, pourrait également jouer un rôle au niveau du repérage de ces expressions dans les pages Web, et ce, à deux niveaux. D'un côté, lorsqu'une expression linguistique correspondant à un type d'objet est repérée dans une page et qu'il faut y associer des

informations temporelles, la catégorisation pourrait indiquer quel type d'expression temporelle chercher. De l'autre côté, s'il n'est pas possible de trouver directement, dans la page Web, le type d'objet touristique dont il est question, mais qu'une expression temporelle est repérée, alors la catégorisation pourrait permettre de retrouver le type d'objet, à partir du type d'expression temporelle (par exemple l'expression *ouvert midi et soir* permet de déduire qu'il s'agit d'un restaurant).

1.2. Expressions spatiales

En ce qui concerne les expressions de localisation spatiale, je me suis principalement intéressée aux adresses prototypiques comme *8, rue de l'église, 58000 Nevers* et quelques unes de leurs variantes (pas de numéro, une information de boîte postale etc.). Le schéma de ce type d'adresse correspond à un numéro, un marqueur lexical indiquant le type de voie, un nom de voie, puis un nombre pour le code postal et enfin un nom de ville.

(49) **34 Rue Saint Gildard 58000 NEVERS**

(50) **Rue de la Préfecture - 58039 NEVERS**

Les nombreuses informations de localisation se trouvant dans les pages Web étudiées sont rarement aussi complètes que les deux exemples donnés ci-dessus. Toutefois, une adresse partielle (comprenant au moins le code postal et le nom de la ville) est le plus souvent mentionnée. Certaines informations qui se trouvent dans des adresses n'apportent rien à la localisation mais sont utiles aux services postaux. Il s'agit en particulier du cedex et du numéro de boîte postale.

Une étude de ces informations de localisation a permis de mettre au jour une difficulté. Elle concerne les noms de villes qui peuvent avoir plusieurs formes, notamment lorsqu'ils comprennent des accents ou des traits d'union. Une lettre accentuée perd souvent son accent si elle est en majuscule et, en ce qui concerne les noms composés, l'usage du trait d'union n'est pas très rigoureux dans les pages Web et il serait dommage de ne pas repérer un grand nombre d'adresses sous prétexte que les traits d'union ne sont pas bien employés.

Le dictionnaire qui a été mis au point (voir chapitre 5) doit donc pouvoir reconnaître plusieurs formes pour un même nom de ville : avec et sans accent et avec et sans trait d'union.

En plus des expressions prototypiques, les informations de localisation partielles comme les suivantes sont fréquentes :

(51) **Office de tourisme 58230 Brisson**

(52) **Le champ Thierry 58170 Luzy**

Pour interpréter ce genre d'expression, il faudrait donc combiner les expressions de localisation et les types d'objets touristiques.

Par ailleurs, d'autres types d'informations de localisation sont parfois mentionnés dans les pages Web, comme l'exemple suivant :

(53) **Au centre du village de Tamnay en Bazois (D978 direction Nevers)**

Il serait intéressant de repérer des informations comme celle donnée dans l'exemple (53) pour lier les différents objets touristiques entre eux et donc pouvoir proposer des activités complémentaires aux utilisateurs de la plateforme mais il faudrait pouvoir les modéliser dans

l'ontologie et adapter le système de requêtes en conséquence. Le repérage de ce type d'information n'a pas été implémenté dans Adetoa (voir le chapitre 4 pour plus de détails sur les informations qui sont repérées).

1.3. Objets touristiques

Cet intitulé vise ici le type de l'objet dont il est question dans une page Web (restaurant, musée, hôtel, etc.). Ces types d'objet doivent en effet faire l'objet d'un repérage dans les pages Web et les expressions qui permettent de les annoncer sont donc brièvement présentées ici.

Repérer le type d'objet touristique dont il est question dans une page Web peut sembler trivial. En effet, les types d'objets possibles ne sont pas en nombre infini et le vocabulaire qui y correspond peut donc paraître restreint, facilitant un repérage automatique basé sur une liste de ces types d'objets. Différents problèmes se posent néanmoins. Premièrement, le langage naturel étant ce qu'il est, le nombre de formulations possibles est en réalité bien plus grand et de nombreuses ambigüités (réelles ou virtuelles) peuvent apparaître dans les pages Web. Deuxièmement, le type de l'objet dont il est question dans une page Web n'apparaît pas toujours textuellement au sein même de cette page. En effet, une image, le nom de l'objet touristique ou d'autres informations suffisent parfois à l'internaute pour comprendre de quoi il s'agit ; le type de l'objet n'est alors pas nécessairement explicitement énoncé. Troisièmement, la question de la finesse de l'annotation attendue se pose : suffit-il que l'annotation signifie qu'il s'agit d'un hébergement ou faut-il qu'elle précise qu'il s'agit d'un camping, ou encore plus finement du type de camping (mobil-home, emplacements pour tentes, etc.). Les exemples suivants permettent d'illustrer plus concrètement les problèmes qui se posent.

(54) Auberge

(55) Hôtel-restaurant

La catégorie d'un objet touristique peut être ambiguë ou double. Par exemple, une auberge peut être un restaurant et un type d'hébergement, il y a donc là une ambigüité. Le contexte permet en général à l'internaute de lever l'ambigüité, mais cela peut devenir très complexe pour un traitement automatique. Pour un hôtel-restaurant, un café-théâtre ou encore un camping-piscine, le type est double : les deux types doivent-ils alors être attribués à l'objet ?

(56) Stade

Stade, visite ou encore *balade* peuvent-ils être considérés comme des objets touristiques ? Par exemple, le problème du terme *stade* est que, s'il peut en effet faire référence à un équipement sportif (tel que modélisé dans l'ontologie), son homonyme (*stade* en tant que niveau, état) est un terme courant de la langue qui ne doit alors pas être repéré.

Certaines pages présentent un objet en particulier mais font en plus référence à de nombreux autres objets « satellites » qui risquent alors de nuire au repérage. En effet, l'objet principal dont il est question est présenté avec différentes informations pratiques (adresse, horaires), tandis que, la plupart du temps, les autres objets sont simplement nommés et ne peuvent alors pas être exploités. Par exemple, la page d'un hôtel indique ce qui se trouve à proximité de l'hôtel : des restaurants, une piscine, etc. Il faudra alors trouver un moyen de déterminer quel est l'objet principal à associer avec l'adresse trouvée. Il semblerait que, dans de nombreux cas, il s'agisse simplement du premier objet mentionné dans la page, mais une telle généralisation semble risquée.

Certaines pages présentent plusieurs objets principaux, avec pour chacun des informations pratiques, un réel travail d'appariement est alors nécessaire étant donné que le but est, pour chaque objet, de regrouper toutes les informations le concernant.

Certaines expressions peuvent être considérées comme désignant le type d'un objet sans que cela soit le cas. Prenons en exemple l'expression *hôtel de ville*. Dans le cadre du projet Eiffel, les mairies sont considérées comme des objets touristiques au même titre que les offices de tourisme. En effet, dans certaines communes, les mairies peuvent jouer le rôle d'office de tourisme ou de syndicat d'initiative. Deux problèmes se posent alors. Le premier est que *l'hôtel de ville* ne doit pas être considéré comme un hôtel en tant qu'hébergement, ce qui n'est pas trop délicat dans la mesure où l'expression *hôtel de ville* apparaît en général telle quelle dans la page Web. Le second est que les expressions *hôtel de ville* ou *mairie* peuvent facilement apparaître dans des pages qui ne concernent pas la mairie et générer alors des repérages fautifs. C'est le cas par exemple dans certaines adresses : *place de l'hôtel de ville*. On peut aussi trouver des expressions comme *bar de l'hôtel de ville*. Éviter le repérage fautif semble dans ces cas plus difficile.

Le type d'objet touristique qu'une page Web présente n'est pas toujours textuellement énoncé sur celle-ci. Il peut être implicite. Par exemple, le nom *Le jardin des pâtes,* laisse entendre qu'il s'agit d'un restaurant, mais, comme la page Web présentée dans la figure suivante le montre, le terme *restaurant* ne figure pas. Les termes *menu*, *carte*, *cuisine*, *chef*, *réservation* pourraient être exploités par un traitement automatique, mais cela risquerait d'engendrer des repérages fautifs.



Figure 17 : Page Web du restaurant « Le jardin des pâtes »

2. Inventaire des problèmes d'interprétation de certaines expressions temporelles

Dans cette partie, j'étudie un échantillon d'expressions temporelles typiques de celles qui se trouvent sur les pages Web touristiques. Ces expressions sont de niveaux de complexité très variés : certaines semblent extrêmement simples, d'autres au contraire paraissent presque incompréhensibles. J'aborderai les problèmes que ces expressions posent, aussi bien au niveau linguistique que pour un traitement automatique, la façon dont ces expressions peuvent être modélisées dans une ontologie et les solutions qui ont été envisagées pour les traiter dans le cadre du projet Eiffel. L'ontologie sur laquelle s'appuie le projet Eiffel et à laquelle quelques références sont faites dans cette partie est présentée au chapitre 4.

2.1. Étude linguistique

Cet inventaire reprend quelques expressions temporelles rencontrées dans les pages Web touristiques et a pour but d'en faire une étude linguistique. Certaines de ces expressions présentent des difficultés d'interprétation : ambigüités virtuelles ou même réelles[15], dates imprécises, etc.

(57) Le 21 avril

Cette date figure dans une page présentant une manifestation touristique. Dans l'ontologie, les objets de type <evenement_manifestation>[16] attendent, entre autres, les propriétés <date_debut> et <date_fin>. Comment doit-on dès lors procéder lorsque nous n'avons qu'une seule date, correspondant à une journée entière ? Deux possibilités : dans le premier cas, dédoubler la date et la mettre à la fois en tant que <date_debut> et en tant que <date_fin> ; dans le second cas, ne remplir que <date_debut> et établir une règle du type « si <date_fin> n'est pas indiquée, alors sa valeur est la même que celle de <date_debut> ». Mais cette règle risquerait d'induire des erreurs, notamment pour les expressions du type *le festival commence le 21 avril*.

(58) Du 10 au 20 mars

Cette expression indiquant un intervalle de temps concerne une manifestation touristique, mais elle pourrait également correspondre à la période de fermeture d'un magasin, d'une bibliothèque, etc. Cette expression, simple en apparence, peut devenir complexe pour un traitement automatique car elle comporte une ellipse. En effet, lorsqu'un lecteur humain lit cette expression, il comprend sans hésitation qu'il s'agit du 10 mars et non pas du 10 février, du 10 décembre ou de 10 bananes, car il s'agit d'une factorisation. Un élément précis est donc factorisé et pas un type d'élément : seul un mois en particulier et non un mois en général peut faire l'objet d'un tel phénomène. Les tests suivants permettent de montrer que l'interprétation d'une telle expression est régulière.

15 Ambigüité réelle signifie que l'expression présente plusieurs interprétations possibles, aussi bien pour un locuteur natif que pour une machine. Une expression virtuellement ambiguë n'est ambiguë que pour une machine.

16 Les chevrons indiquent qu'il s'agit d'un nom de classe dans l'ontologie.

(59) *[17]30 au 4 avril

(60) du 10 janvier au 20 mars

(61) du 10 mars au 20 mars

L'exemple (59) n'est pas naturel car il comporte une rupture de granularité : si la factorisation fonctionnait comme dans l'exemple (58), on obtiendrait *du 30 avril au 4 avril*, ce qui n'est pas cohérent. Cette expression devrait donc s'interpréter comme signifiant *du 30 mars au 4 avril*, mais il n'est pas possible de l'abréger de cette manière. De la même façon, deux mois différents (janvier et mars) sont mentionnés dans l'exemple (60). Cette expression ne peut donc en aucun cas être considérée comme équivalant sémantiquement à l'exemple (58), tandis que l'exemple (59) l'est. L'expression de l'exemple (58) peut uniquement signifier *du 10 mars au 21 mars*. Ces phénomènes de factorisation peuvent concerner d'autres types d'expressions. On peut en effet rapprocher ces exemples des deux suivants :

(62) Elle mange des pommes et lui des bananes.

(63) *Elle mange des pommes et lui du vin.

De la même façon qu'un mois était factorisé dans les exemples précédents, ici c'est le verbe *manger* qui fait l'objet d'une factorisation. Le deuxième exemple n'est donc pas naturel car on ne dit pas *manger du vin*. Ce n'est pas le type d'action (en l'occurrence *ingérer*) qui est factorisé mais bien une action particulière (*manger*).

(64) Du lundi au vendredi, 9h – 11h

Cette expression indique les horaires d'ouverture ou d'accès d'un lieu touristique. La difficulté d'interprétation de cette expression vient du fait qu'elle est factorisée. Il faut en effet comprendre que les horaires s'appliquent à chaque jour, du lundi au vendredi. L'ontologie ne permet pas de modéliser des intervalles sans date, avec uniquement des jours de la semaine. Pour pouvoir stocker les informations contenues dans cette expression, il faudrait donc énumérer les jours de la semaine, tout en mentionnant que les horaires correspondent à chacun des jours. C'est une règle de transformation qui peut effectuer cette énumération. Une règle pourrait également permettre d'indiquer les fermetures, en inférant que si c'est ouvert du lundi au vendredi, alors c'est fermé le samedi et le dimanche. Toutefois, si cette inférence semble simple dans un cas comme celui-ci, elle peut s'avérer plus compliquée dans d'autres cas, comme dans l'exemple (65). Ne pas inférer de jours de fermeture réduirait le risque d'erreurs. Les choix qui ont été effectués et les règles de transformations que j'ai développées pour ce type d'expressions sont présentés au chapitre 5.

(65) Ouvert du lundi au vendredi de 9h à 11h et le dimanche en haute saison.

Cet énoncé, qui peut sembler proche du précédent, est en réalité bien plus complexe, étant donné qu'il comprend une exception. En effet, aucun marqueur lexical propre à l'exception (comme *sauf* ou *à l'exception de*) n'est explicitement mentionné. De plus, même si *en haute saison* peut être considéré comme un marqueur d'exception, un contexte extralinguistique reste nécessaire pour l'interpréter (la haute saison n'est pas la même à la mer et à la montagne).

17 L'astérisque indique l'impossibilité de l'expression.

(66) **En Juillet et Août ouvert tous les jours. Hors cette période, fermeture le mercredi soir, le jeudi toute la journée et le dimanche soir. Horaires d'ouverture: de 12h00 à 14h00 de 19h00 à 22h00.**

Cet énoncé est particulièrement complexe, même pour un lecteur humain. Il s'agit des horaires d'ouverture d'un restaurant, ce qui est marqué par l'utilisation du mot *soir* et par les horaires. Les difficultés sont multiples. Le *hors cette période* doit être interprété : il correspond aux mois de janvier, février, mars, avril, mai, juin, septembre, octobre, novembre et décembre. Les horaires de fermeture ne s'appliquent qu'à ces mois-là et peuvent faire assez simplement l'objet d'une annotation automatique. En revanche il semble assez complexe de passer de ces horaires de fermeture à des horaires d'ouverture. En effet, si le restaurant est fermé le dimanche soir, on peut en déduire qu'il est ouvert le dimanche midi, mais cette inférence n'est pas simple à mettre en place. Les horaires d'ouverture, quoique assez simples à interpréter, apportent aussi leur lot de difficultés. En effet, il faut comprendre que ces horaires correspondent à tous les jours de la semaine, en juillet et août, et aux jours ou parties de journées qui ne font pas l'objet d'indications particulières pour le reste de l'année.

(67) **Visites de juin à septembre les après-midi, sauf lundi et mardi. Sur rendez-vous le reste du temps.**

Dans cette expression, la difficulté linguistique provient des exceptions. *Sauf lundi et mardi* indique que ce lieu est fermé toute la journée les lundi et mardi, et non pas que c'est ouvert uniquement le matin. De plus, *sur rendez-vous le reste du temps* est inqualifiable ; cette expression ne peut être formalisée et doit rester sous forme textuelle pour pouvoir être conservée.

(68) **Fermeture annuelle début Février à début Mars.**

Cette expression est complexe car elle est floue : comment passer des expressions *début février* et *début mars* à des dates ? S'agit-il du 1er février et du 1er mars ou, par exemple, du premier lundi de février et du premier dimanche de mars ? On peut même imaginer que *début février* corresponde en fait à l'un des derniers jours de janvier. Cette ambigüité est réelle et ne peut pas être levée. [Battistelli et al. 2006] distinguent les dates absolues et les dates relatives, les secondes ayant besoin d'un calcul pour être inscrites dans une représentation calendaire, contrairement aux premières. Mais ils constatent également que cette distinction n'est pas suffisante :

> Mais cette distinction ne suffit pas pour être à même de représenter leur sémantique [des expressions calendaires] et, au delà, pour pouvoir envisager de représenter le parcours dans le calendrier qu'appréhende un lecteur : il faut en effet savoir quelles *zones temporelles* elles désignent dans ce calendrier. Ces zones vont dépendre des marques autres que les grains qu'elles comprennent : articles, déterminants, locutions prépositionnelles, connecteurs, quantificateurs, etc. Les zones désignées pourront alors être précises ou au contraire « floues », être perçues comme des « points » ou des intervalles, comprendre ou non les bornes initiale et/ou finale. Or ceci relève directement de la catégorie de l'aspect. ([Battistelli et al. 2006], p.20)

Cette expression est en effet « floue » et ne peut pas être aisément transformée en date calendaire. Un traitement des expressions floues est proposé dans [Fortin et al. 2009], voir chapitre 6.

(69) **Ouvert du mardi au samedi de 9h00 à 12h30 et de 15h00 à 19h00 Le dimanche matin de 10h00 à 12h30**

Cette expression est assez directe à interpréter et ressemble aux horaires d'ouverture vus dans les exemples (64) et (66). Du point de vue linguistique, on peut se demander à quoi sert la redondance de l'information *matin* : on aurait pu se contenter de *dimanche matin* ou de *dimanche de 10h00 à 12h30*. Cette répétition est une forme d'insistance pour marquer que le dimanche est différent des autres jours, où c'est également ouvert l'après-midi.

(70) **Fermée les mardi, jeudi et samedi après-midi et dimanche.**

Cette expression présente une ambigüité réelle. S'agit-il de mardi après-midi et jeudi après-midi ou de mardi et jeudi toute la journée ? De même pour dimanche, s'agit-il de toute la journée ou seulement de l'après-midi. L'ambigüité provient du fait que *après-midi* peut, ou non, faire l'objet d'une factorisation. Elle est impossible à lever[18] et il faudra donc faire un choix dans l'interprétation de ce type d'énoncé.

(71) **Horaires d'ouverture : lundi, mercredi 13h30_17h30 mardi, jeudi, vendredi 8h-12h**

Ces horaires d'ouverture ne sont pas très intuitifs mais ne présentent pas d'ambigüité. À la première lecture, on peut penser qu'il s'agit de « du lundi au mercredi : 13h30-17h30 » mais étant donné que des horaires pour mardi sont donnés dans la suite de l'énoncé, ce n'est pas le cas. On a donc bien d'un côté les horaires pour les lundi et mercredi, et de l'autre les horaires pour les mardi, jeudi et vendredi. Cet énoncé est rendu particulier car les jours de la semaine sont ordonnés et qu'il est donc habituel de les énumérer dans cet ordre, qui n'est pas respecté ici. C'est donc l'absence de respect du « prototype », principe introduit par Kleiber, qui rend cette expression moins intuitive.

> « Les tests et expériences décrits dans les premiers travaux d'E. Rosh (voir bibliographie) introduisent la notion de prototype comme étant le meilleur exemplaire ou encore la meilleure instance, le meilleur représentant ou l'instance centrale d'une catégorie. Il s'agit donc d'une acception technique différente du sens courant de 'premier exemplaire d'un modèle (de mécanisme, de véhicule) construit avant la fabrication en série'. » ([Kleiber 1990], p. 47-48)

En effet, cette notion définie ainsi par Kleiber peut très bien s'appliquer aux jours de la semaine.

2.2. Solutions envisagées pour Eiffel

Cette étude des expressions temporelles trouvées dans les pages Web soulève plusieurs problèmes. S'ils sont intéressants sur le plan théorique, ils nécessitent également une solution simple et implémentable dans le cadre du projet Eiffel. Voici donc un inventaire des problèmes traités et des décisions qui ont été prises. Les décisions sont présentées ici sur un plan théorique. Leur implémentation pratique est présentée aux chapitres 4 et 5.

18 Après vérification auprès de la mairie en question, celle-ci est ouverte le mardi jusqu'à 12h et le jeudi jusqu'à 12h.

2.2.1. Les dates seules

L'ontologie d'Eiffel modélise des périodes. Pour reconstruire une période à partir d'une date seule, il a alors été décidé d'instancier la même date en tant que date de début et date de fin. Si cette solution peut facilement être mise en œuvre, il faut toutefois noter qu'elle peut engendrer des erreurs. En effet, lorsqu'une date seule est indiquée, cela n'implique pas nécessairement qu'il s'agit d'une seule journée. Il pourrait en réalité s'agir d'une date de début, sans que la date de fin soit mentionnée (*le festival débutera le 20 janvier*). L'étude linguistique des pages Web a montré que ce second cas était rare dans le type de pages Web qui est pris en compte. Le choix d'inférer que, lorsqu'une date seule est trouvée, il s'agit alors d'une seule journée semble ne pas engendrer trop d'erreurs.

2.2.2. Les zeugmes

Avant de pouvoir instancier les expressions du type *du 10 au 21 mars* dans l'ontologie, il faut reconstruire l'expression complète, afin de pouvoir convertir la date de début en « 10 mars » et la date de fin en « 21 mars ». Un module réalisé en JAVA, à base de règles, permet de reconstruire ces expressions (chapitre 5).

2.2.3. La distribution

Pour les expressions du type *ouvert du lundi au vendredi de 8h à 18h*, il faut associer les horaires d'ouverture à chacun des jours d'ouverture. L'intervalle de jours doit alors être converti en énumération. C'est le module de transformations qui se charge d'effectuer cette énumération (chapitre 5).

2.2.4. Les exceptions

Les expressions comprenant des exceptions comme *ouvert tous les jours sauf le lundi* ou *fermé le mardi sauf en haute saison* sont délicates à traiter. Un traitement linguistique effectuant des inférences sur ce type d'expression est difficilement implémentable et le risque d'erreur resterait grand. Il a donc été décidé de ne pas toujours essayer d'interpréter ces expressions, mais de les garder sous la forme de chaînes textuelles telles qu'elles apparaissent dans les pages Web lorsqu'aucune interprétation n'est possible. Ainsi stockées dans la base de connaissance, elles peuvent être fournies à l'internaute effectuant une requête sur l'objet touristique concerné. Il en est de même pour les informations complémentaires du type *sur rendez-vous*.

2.2.5. Le flou

Le traitement des expressions floues ou imprécises s'effectue en deux temps. Dans un premier temps, lors de l'annotation, une balise <incertitude> est insérée lorsqu'une information semble floue. Par exemple, pour les expressions *début février* ou *visites de juin à septembre*, il n'est pas possible de déduire directement des dates précises. Pour pouvoir stocker correctement ces informations dans la base de connaissance, des inférences doivent avoir lieu. Un modèle possibiliste qui permet un traitement flou et associe un score selon les probabilités d'ouverture ou de fermeture, proposé dans [Fortin et al. 2009], peut être mis en place. Ainsi, pour *ouverture début février*, par exemple, un score d'ouverture plus élevé sera attribué au 6 février qu'au 1er février.

Conclusion

Ce chapitre a permis de montrer la complexité et la diversité des expressions que j'ai étudiées et pour lesquelles un module de repérage et d'annotation a été mis au point, Adetoa.

Le chapitre 4 montre la mise au point d'un schéma d'annotation permettant de rendre compte de ces expressions en respectant ce qui est donné sur un plan linguistique. Le chapitre 5 présente les transducteurs que j'ai mis au point pour pouvoir repérer et annoter ces expressions et qui doivent donc prendre en compte leur complexité. Cette étude linguistique a donc servi au développement des transducteurs.

Les règles de transformations que j'ai mises au point en découlent également directement.

Deuxième Partie

Manipulation des Données et Représentation

Chapitre 3. Traitement Automatique de Pages Web Touristiques

Introduction

Dans ce chapitre, je m'intéresse aux caractéristiques de l'extraction d'information dans des pages Web. Après un rapide état de l'art sur l'extraction d'information et l'extraction d'information dans des pages Web, je présenterai plus en détail le corpus sur lequel j'ai travaillé. Ce corpus est celui du projet Eiffel et j'indiquerai donc comment il a été constitué, ainsi que les propriétés des pages qui le composent.

Plus concrètement, je donnerai ensuite quelques exemples de pages afin de montrer comment elles sont traitées par Adetoa, avant d'exposer les difficultés engendrées par certains types de pages, que ce soit au niveau de leur structure ou de leur contenu.

1. État de l'art autour de l'extraction d'information

L'extraction d'information est la discipline qui consiste à repérer, dans des textes, les données permettant de répondre à un certain nombre de questions. Ces questions se présentent comme un formulaire dont il faut remplir les cases. En s'appuyant sur Pazienza, Poibeau donne la définition suivante :

> « L'extraction d'information désigne l'activité qui consiste à remplir automatiquement une banque de données à partir de textes écrits en langue naturelle ». ([Poibeau 2003] p.13)

Le but de l'extraction est donc de chercher, dans un certain nombre de textes, des informations précises et prédéfinies. Par « prédéfinies », on entend que l'on sait à quelle question l'information doit répondre. En revanche, ce que l'on ne sait pas, c'est où l'information se trouve, et sous quelle forme ou formulation elle se présente. L'extraction d'information est donc toujours guidée par le but. En effet, même si certains systèmes d'extraction d'information comprennent des étapes génériques (comme une analyse syntaxique par exemple), la plupart d'entre elles sont propres à un domaine particulier.

1.1. Extraction automatique d'information – généralités

1.1.1. Extraction et entités nommées

L'extraction automatique d'information est vaste et variée, étant donné qu'elle dépend du type d'information que l'on cherche à extraire et du support sur lequel celle-ci se trouve. L'extraction est en effet différente selon le type de corpus à analyser (corpus journalistique, textes médicaux, etc.).

Le repérage des entités nommées est directement lié à l'extraction automatique d'information. Depuis la conférence MUC 6[19] [MUC-6 1995], cette tâche a été ajoutée aux défis proposés. Dans ce cadre, les entités nommées ont été définies ainsi :

> The Named Entity task consists of three subtasks (entity names, temporal expressions, number expressions). The expressions to be annotated are "unique identifiers" of entities (organizations, persons, locations), times (dates, times), and quantities (monetary values, percentages). ([MUC-6 1995][20])

> La tâche relative aux entités nommées se compose de trois sous-tâches (noms d'entités, expressions temporelles et expressions numériques). Les expressions à annoter sont les « identifiants uniques » d'entités (entreprises, personnes, lieux), de moments (dates, heures) et de quantités (valeurs monétaires, pourcentages). [Ma traduction]

Les entités nommées incluent donc les noms propres : noms de personnes, d'entreprises ou de lieux. D'autres expressions sont aussi souvent considérées comme des entités nommées : certaines informations temporelles (date et heure) et certaines données numériques (valeurs monétaires, pourcentages, etc.). À ce titre, le repérage d'entités nommées est souvent une tâche centrale dans les projets d'extraction d'information. Les bases de données qui doivent être remplies à l'aide des résultats de l'extraction attendent en effet souvent des entités nommées : noms de personnes, dates et lieux dans le cadre d'une analyse de dépêches journalistiques, valeurs numériques dans l'analyse de corpus économiques, etc. Voir aussi [Poibeau 2008] pour plus de détails sur les entités nommées et leur rôle en extraction d'information.

Toutefois, le domaine de l'extraction d'information est plus vaste que le repérage d'entités nommées : par exemple, l'extraction de phrases contenant l'expression de sentiments n'a aucun lien avec les entités nommées. Dans le cadre de mes travaux, je me rapproche des entités nommées avec le repérage des adresses et des horaires.

1.1.2. Extraction automatique d'informations temporelles

Dans le domaine de l'extraction d'information, les informations temporelles ont été largement étudiées. [Mani 2004] fait un inventaire de ces travaux. Deux tâches principales sont concernées : le repérage des informations et leur interprétation, ou plutôt leur représentation. [Battistelli et al. 2006], qui proposent un modèle de représentation des expressions calendaires, soulignent le fait qu'il n'existe pas de définition consensuelle des expressions calendaires. De plus, ce ne sont pas les expressions calendaires en elles-mêmes qui ont le plus été étudiées.

En effet, de nombreux travaux se sont plutôt intéressés à la temporalité des événements

19 Les conférences MUC *(Message understanding conference),* dont la première a eu lieu en 1987, reflètent bien l'évolution de l'extraction d'information.

20 http://cs.nyu.edu/cs/faculty/grishman/muc6.html

décrits dans les textes afin de pouvoir les ordonnancer [Hobbs & Pustejovsky 2003], tâche utile dans l'analyse de corpus journalistiques [Mani & Wilson 2000] ou pour des études biographiques. Les travaux liés à TimeML ont montré l'importance du traitement de la temporalité, mais ils se sont également intéressés principalement aux événements [Pustejovsky et al. 2003].

Les expressions auxquelles je me suis intéressée ont des caractéristiques très particulières, liées au domaine du tourisme. Elles contiennent des informations temporelles, mais celles-ci ne sont pas nécessairement calendaires. Elles ne requièrent donc pas une représentation calendaire. De plus, ces informations ne concernent pas des événements, tels ceux considérés dans TimeML.

1.1.3. Principales méthodes d'extraction d'information

Les techniques d'extraction d'information peuvent être classées dans deux catégories principales : les méthodes à base de règles construites manuellement et les méthodes qui reposent sur un apprentissage. Ces méthodes ne sont cependant pas totalement disjointes puisque le produit de l'apprentissage est en général de nouvelles règles. De nombreux systèmes d'extraction utilisent un système hybride dans lequel les règles créées automatiquement sont ensuite vérifiées manuellement.

La première catégorie (système fondé sur des règles) est la plus ancienne. Le principe de cette méthode est que le concepteur du système établit manuellement un ensemble de règles permettant de repérer et d'extraire les données voulues. Ces règles sont des patrons d'extraction, souvent mis en œuvre à l'aide d'automates mais la création de ces patrons est un travail long et coûteux. C'est pour cela que des méthodes basées sur un apprentissage ont été mises au point.

Les méthodes fondées sur un apprentissage consistent généralement en une analyse automatique et statistique d'un corpus annoté. Souvent, elles permettent d'établir des patrons d'extraction. L'apprentissage peut être supervisé ou non.

De plus, une fois les entités voulues extraites, celles-ci doivent également être mises en relation les unes avec les autres. Par exemple, pour une application voulant déterminer quels chercheurs travaillent dans quels laboratoires, il faut, bien entendu, repérer les noms des chercheurs et ceux des laboratoires, mais sans lien entre ces entités, l'extraction n'est pas pertinente. La tâche de liage des différentes entités repérées est donc importante et directement liée à celle de l'extraction d'information. Un historique des systèmes d'extraction d'information est donné dans [Nguyen 2006].

1.2. Extraction automatique d'information – dans des pages Web

Depuis quelques années, et avec l'expansion du Web, de nombreux travaux d'extraction d'information se sont orientés vers le Web. Étant donnée la structure propre des pages Web, les systèmes développés ne reposent pas sur les mêmes méthodes que celles de l'extraction d'information dans des ressources textuelles. [Laender et al. 2002] proposent une étude des différents outils d'extraction d'information dans les pages Web.

La méthode des *wrappers* est la plus répandue. Un wrapper, aussi appelé extracteur, est un programme qui extrait des informations dans un document et les transforme en informations structurées pouvant intégrer une base de données. Cette méthode ne se base pas sur des connaissances linguistiques mais sur la structure des documents. Comme le

mentionnent [Chang & Lui 2001], ces méthodes nécessitent de connaître *a priori* le format des informations à extraire, mais aussi celui des pages Web.

De nombreux systèmes sont également basés sur des méthodes d'apprentissage. Cet apprentissage permet souvent de générer de nouveaux wrappers. Il s'agit alors d'induction de wrappers.

Un exemple plus concret de projet d'extraction d'information sur le Web est l'étude menée par [Tengli et al. 2004], dans laquelle ils cherchent à extraire les informations liées aux inscriptions dans les universités américaines, ces informations se trouvant dans des tableaux. Leur système fonctionne grâce à un apprentissage et ne concerne que des tableaux d'un certain type, sur un certain type de pages Web, en HTML. Les pages sont celles d'universités américaines qui présentent les statistiques d'inscriptions pour une année donnée (pourcentage homme / femme, niveau d'étude, etc.). Les données sont donc sensiblement du même type, d'un tableau à un autre. De plus, les universités américaines se basent, pour présenter leurs informations sur le CDS (Common Data Set), produit d'une initiative commune de plusieurs d'entre elles. Celle-ci est définie sur la page Web de l'Université de Stanford :

> « CDS The Common Data Set Initiative is a collaborative effort between publishers and the educational community to improve the quality and accuracy of information provided to all involved in a student's transition into higher education, as well as to reduce the burden on colleges of compiling and reporting information. Questions and definitions used by the U.S. Department of Education in its college surveys are a guide in the development of CDS items. » ([site de Stanford University][21])
>
> CDS : le « Common Data Set » est un effort de collaboration entre les éditeurs et la communauté éducative pour améliorer la qualité et l'exactitude des informations fournies à toutes les personnes impliquées dans la transition de l'élève vers l'enseignement supérieur, ainsi que pour réduire la charge des universités en ce qui concerne la compilation et la diffusion d'informations. Les questions et les définitions utilisées par le département américain de l'éducation dans ses sondages effectués dans les universités sont un guide pour l'élaboration des données de la CDS. [Ma traduction]

Toutefois, ces auteurs remarquent que, bien que les tableaux de leur corpus soient proches en ce qui concerne leur contenu, ils varient beaucoup au niveau de la formulation des titres de colonnes, mais aussi parce que certaines informations sont présentes dans certains tableaux mais pas dans d'autres. Enfin, et surtout, le formatage en HTML n'est pas du tout le même d'un tableau à un autre. Ainsi, ils montrent bien qu'une généralisation pour un traitement automatique n'est pas aisée, malgré le fait que leur système n'analyse que les tableaux respectant le CDS et créés à l'aide de la balise <table>. Leurs tableaux ont donc une structure standard du type :

```
<Table>
   <Tr>
      <Td> ... </Td>
      <Td> ... </Td>
   </Tr>
</Table>
```

Ne s'intéressant qu'à ce type de tableaux, l'approche de [Tengli et al. 2004] n'est pas toujours applicable. En effet, comme le montrent [Gatterbauer et al. 2007], les mêmes données peuvent avoir la même représentation visuelle mais être codées avec des balises différentes.

21 http://ucomm.stanford.edu/cds/cds_2009.html

Par exemple, des balises , associées à des informations de positionnement, peuvent permettre d'agencer les données sous la forme d'un tableau identique à celui qui aurait été obtenu à l'aide de la balise <table>. Les informations de positionnement peuvent être contenues dans un attribut de la balise ou dans une feuille de style[22].

```
<DIV Information de position / référence css>
   ...
</DIV>
<DIV Information de position / référence css>
   ...
</DIV>
```

L'utilisation des feuilles de style reflète une évolution du Web et des méthodes de conception des pages. Les informations de disposition spatiale ne sont plus contenues dans le code HTML mais dans la feuille de style associée. De ce fait, les informations de mise en page sont perdues si l'on ne regarde que le code source. Voir [Garron et al. à par.] pour plus de détails sur l'évolution du Web et les conséquences qui en découlent pour une analyse des pages Web.

Les pages de mon corpus ne respectent aucune norme prédéfinie et les tableaux y sont donc codés de différentes façons ; je reviendrai sur les problèmes que posent les tableaux au paragraphe 2.3.1.a.

Ces travaux d'extraction d'information soulèvent différents problèmes : le traitement des tableaux, le liage des informations repérées, la normalisation des informations, etc. [Nagy et al. 2009] présentent ces difficultés plus en détail. Si je suis également confrontée à ces difficultés, mes travaux sont toutefois différents. Comme nous le verrons ci-dessous, la différence réside principalement dans le choix de mon corpus : celui-ci n'a en effet pas été sélectionné en fonction de la structure des pages. Ces pages ne sont d'ailleurs souvent pas structurées ; il en résulte que les méthodes à base de wrappers ne sont pas applicables.

2. Pages Web touristiques

Dans le cadre de l'extraction d'information, le cas du Web est particulier. Celui des pages Web touristiques est un cas particulier du cas particulier. Et pourtant, cela reste très vaste, aussi bien en ce qui concerne le nombre de pages qu'en ce qui concerne leur variété. Dans cette partie, je vais présenter les spécificités du corpus constitué par le projet Eiffel et sur lequel je me suis appuyée. J'exposerai ensuite quelques exemples de pages et la façon dont elles sont traitées par Adetoa. Enfin, j'aborderai les difficultés liées à la constitution de pages Web.

2.1. Spécificités du corpus Eiffel

Les pages Web sur lesquelles j'ai travaillé dans le cadre de cette thèse sont celles qui ont servi au projet Eiffel. Je me suis basée sur ce corpus car il a l'avantage d'avoir été collecté selon certains critères et d'avoir été déjà pré-traité et converti en XML, comme nous le verrons ci-dessous. Ainsi, cela m'a évité de « choisir » mes pages de manière partiale. Les pages du corpus Eiffel sont toutes[23] en français et liées au domaine touristique. Il s'agit donc, par

22 Feuille de style CSS (*casacding style sheet*).

23 Quelques pages ne respectant pas ces critères subsistent dans le corpus et sont enlevées manuellement pour les évaluations.

exemple, de pages d'hôtels, de clubs de loisirs, d'offices de tourisme, etc.

Antidot, qui a constitué ce corpus, m'a fourni plus de 8000 pages. Si le besoin s'en était ressenti, le corpus aurait pu être élargi.

2.1.1. Crawl des pages Web

Antidot a effectué le crawl des pages Web selon deux points d'entrée :

- Les pages Web des objets touristiques fournis par divers offices de tourisme et autres référentiels. Dans ce cas, seule la page référencée est crawlée.
- Un périmètre spécifié pour l'utilisation d'AFS[24]. Toutes les pages des sites répertoriés sont alors crawlées.

Les pages Web qui constituent le corpus d'Eiffel sont indépendantes les unes des autres, c'est-à-dire qu'elles sont isolées du site duquel elles proviennent. L'étape de crawl a parfois absorbé plusieurs pages d'un même site mais celles-ci ne sont ensuite plus liées les unes aux autres. Cela peut avoir des conséquences car, si certains sites ne sont constitués que d'une seule page, ce n'est pas le cas de la majorité des sites, généralement composés de plusieurs pages destinées à « fonctionner » ensemble. Par exemple, sur le site Web d'un musée, une page peut contenir toutes les informations pratiques mais ne pas redonner le nom du musée. Lors d'une utilisation « standard » du Web, l'internaute n'accède à cette page que depuis le site du musée et sait donc que les informations présentées sont propres au musée en question. Il n'est pas possible de connaître le chemin d'accès qui a mené aux pages du corpus. Néanmoins, les URL[25] des pages sont présentes dans le fichier crawlé et elles peuvent parfois donner des indications sur le chemin d'accès et sur le site dont elles sont issues.

2.1.2. Tous types de pages touristiques, sans régularité

Malgré l'utilisation d'un outil de crawling, les pages qui constituent le corpus s'avèrent très variées. Elles concernent de nombreuses catégories d'objets touristiques et n'ont pas toutes la même visée. Si la page d'un hôtel a pour but d'attirer le client et de lui donner toutes les informations pratiques dont il aura besoin, il n'en est pas de même pour la page d'une mairie ou d'un office de tourisme, qui aura une visée plus globale : présenter l'agenda des sorties dans la ville, donner les adresses des bureaux de poste, etc.

Par ailleurs, les pages ne sont pas toutes conçues par le même type de personnes : certaines sont conçues par des professionnels, mais la plupart d'entre elles sont créées par des amateurs, plus ou moins confirmés. De cela résulte une grande diversité au niveau de la complexité du code des pages Web.

De plus, les concepteurs n'utilisent pas tous les mêmes outils pour créer leurs pages : certains tapent directement leur code HTML, d'autres utilisent des éditeurs permettant de visualiser le résultat. Le langage HTML n'étant pas très contraignant, plusieurs codes HTML sont possibles pour un même contenu affiché d'une seule façon. Cela est illustré dans [Cohen et al. 2002].

Les pages Web qui constituent le corpus Eiffel ne présentent donc pas de régularité : aucun patron générique n'est utilisé. Ce manque concerne aussi bien l'aspect visuel de la page pour

24 AFS : Antidot Finder Suite, solution de recherche de référence à base d'agents logiciels.

25 Uniform Resource Locator : standard permettant de localiser une ressource ; dans le cas des pages Internet, on parle d'adresses Web.

l'internaute qui s'y rend, que le code de la page. Cela s'applique aussi aux données qui figurent sur les pages : elles ne contiennent pas toutes le même type de données. Certaines font une présentation générale de la région, d'autres ne donnent que des informations pratiques, et au niveau des informations pratiques, certaines pages indiquent des horaires très détaillés, les prix, l'adresse, etc., tandis que d'autres ne fournissent, par exemple, que l'adresse. Cette liste de différences est loin d'être exhaustive. Le seul point commun qui existe véritablement entre les pages Web du corpus est qu'elles se rapportent toutes, plus ou moins directement, au domaine touristique.

2.1.3. Spécificité technique : passage au XML

Après avoir effectué le crawl des pages Web, Antidot s'est chargé de les transformer en documents XML. Le choix du langage XML a été fait au sein du projet Eiffel pour pouvoir profiter des outils préexistants liés à cette technologie (parcours d'arbres, JDOM, etc.).

Pour le passage au XML, la structure du code HTML a été conservée mais de nombreuses balises ont été supprimées, de façon à « alléger » le code. En effet, ces balises servent à afficher la page Web dans un navigateur mais la plupart d'entre elles ne sont pas utiles dans le cadre d'un traitement automatique. Nous[26] avons donc décidé de ne conserver, lors du passage au XML, que les balises qui pouvaient nous être utiles : celles pouvant permettre une analyse de l'organisation, de la structure et de la mise en forme de la page (marques de formatage, police, taille et couleur, marques de tableaux, titres, etc.). Ces balises ont été conservées car elles peuvent servir à des réflexions théoriques et à une analyse sémiotique des pages Web. Elles ne sont cependant pas exploitées pour le traitement automatique dans le cadre du projet Eiffel, pour lequel une analyse du contenu textuel du document, sous forme de flot de caractères, a finalement semblé suffisante.

En effet l'analyse du code source des pages Web, en HTML, n'apporte pas d'informations sémantiques facilement exploitables. Les balises HTML (qui sont conservées dans le document XML) n'ont pas de sémantique fixe et chacun est libre de les utiliser comme bon lui semble. Par exemple, la balise <H1>, prévue à l'origine pour marquer les titres, est souvent utilisée uniquement pour afficher une portion de texte en grand, sans qu'il s'agisse pour autant d'un titre. Les balises HTML ne servent pas à typer les informations qu'elles englobent. Il est donc difficile, voire impossible, de les interpréter. De plus, l'ordre des informations telles qu'elles apparaissent dans le code source ne reflète pas nécessairement l'ordre de lecture de la page affichée dans un navigateur.

2.2. Exemples de pages traitées par Adetoa

Maintenant que le corpus a été décrit sur un plan technique, je vais présenter plus concrètement des exemples de pages sur lesquelles Adetoa a été appliqué. Je pourrai ainsi montrer sur quel type de pages il fonctionne bien avant d'aborder, dans la partie suivante, les difficultés que j'ai rencontrées. Comme cela a déjà été présenté, les expressions temporelles touristiques sont au cœur de mes préoccupations. Néanmoins, pour les besoins du projet Eiffel, je me suis aussi intéressée aux expressions de localisation (adresse) et aux objets touristiques (type de lieu : hôtel, restaurant, musée, etc.). Ces trois types d'informations doivent donc être repérés dans les pages Web, annotés et liés les uns aux autres.

26 Décision prise avec les différents membres du projet (Mondeca – Antidot – Modyco).

2.2.1. Pages présentant un objet

Dans les pages suivantes, sont encadrées (en bordeaux) les informations qui sont correctement repérées et annotées par Adetoa.

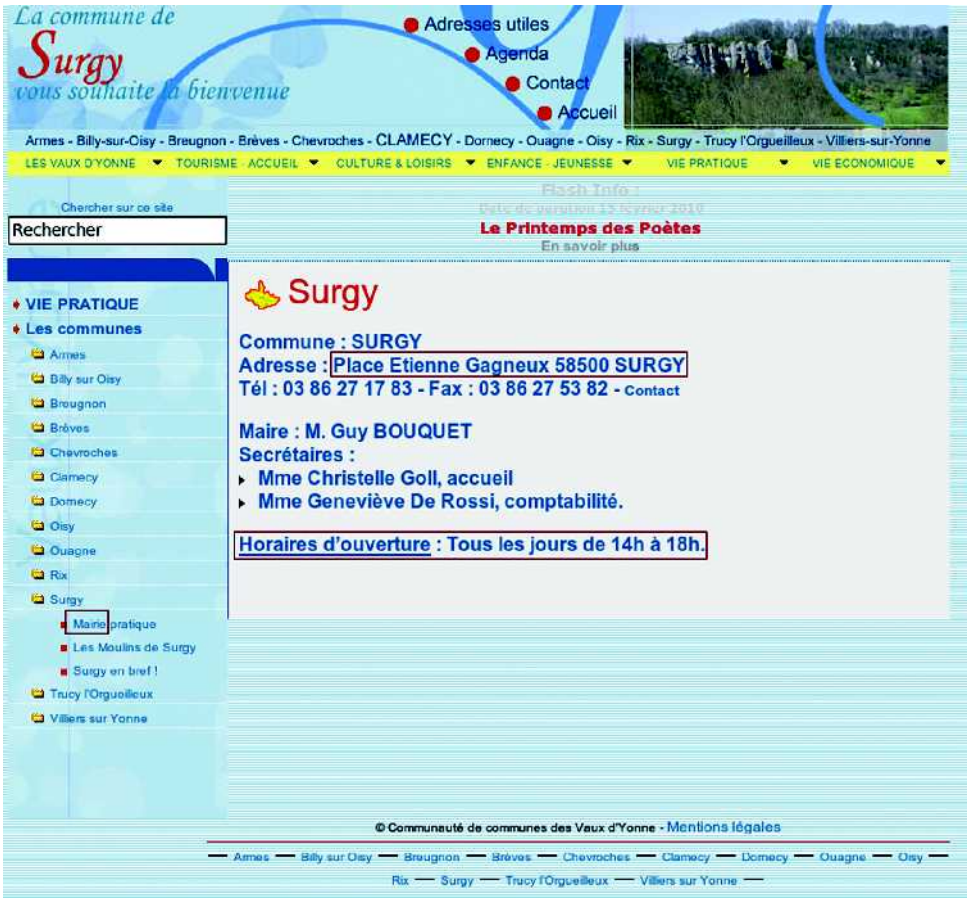


Figure 18 : Page d'une mairie

Dans cette page qui présente les informations pratiques d'une mairie, toutes les informations intéressantes dans le cadre d'Eiffel sont correctement repérées et annotées. Ainsi, sont repérés l'objet (*mairie*), ses horaires d'ouverture et son adresse.

Notons toutefois que, si les horaires indiquent *ouvert tous les jours*, la mairie n'est probablement pas ouverte le dimanche. C'est malgré tout ce qui est mentionné sur la page et c'est donc l'information qui sera annotée et stockée dans la base de connaissance. Il incombe ensuite à l'utilisateur de s'assurer des données fournies – en téléphonant ou bien en faisant simplement référence à ses connaissances du monde (une mairie est fermée le dimanche !).

Les trois pages suivantes ont déjà été présentées au chapitre 1. Elles sont typiques des pages à traiter et permettent de bien illustrer comment fonctionne Adetoa.



Figure 19 : Page Web d'un producteur fermier

La page de la figure 19 donne toutes les informations pour visiter une ferme. Adetoa y repère le type d'objet (*producteur fermier, visite*) et l'annote correctement. L'adresse et les horaires d'ouverture sont également bien repérés et bien annotés. Malgré la complexité des horaires d'ouverture, cette page ne pose pas de problème et est bien traitée par Adetoa.

Dans la figure 20, la page présente une ferme-auberge. Toutes les informations pratiques y sont repérées par Adetoa : type (*chambres d'hôtes – gîtes*), adresse (*58120 St Hilaire en Morvan*) et période d'ouverture (*ouvert du 15 Février au 30 Novembre*). Ces informations sont également bien annotées.

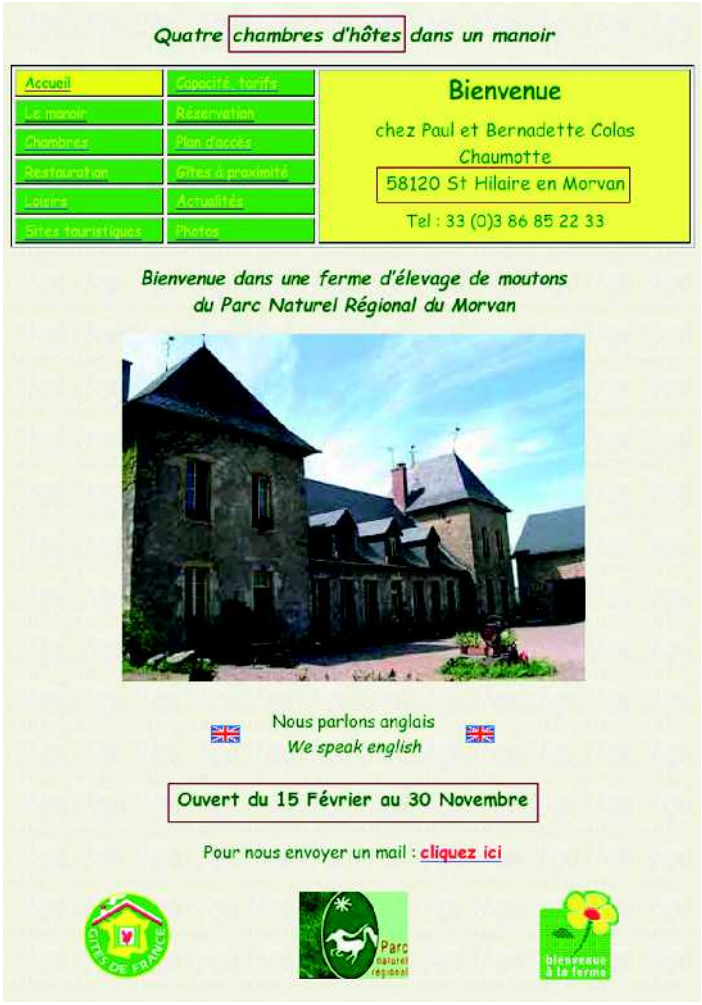


Figure 20 : Page Web d'une ferme-auberge

La page de la figure 21 présente une mairie. L'objet *mairie* y est repéré par Adetoa, tout comme l'adresse et les horaires de fermeture. Ces informations y sont correctement annotées. Comme cela a été vu au chapitre 2, l'expression temporelle est ambiguë : *après-midi* se rapporte-t-il uniquement à samedi ou aux autres jours aussi ? Il est impossible de le savoir ; pour annoter cette information, Adetoa tranche et distribue *après-midi* à *mardi, jeudi et samedi*.



Figure 21 : Page Web d'une mairie - 2

Ces exemples montrent que, même si les pages Web semblent simples, elles ne se ressemblent pas et ont chacune leurs spécificités propres. À cela s'ajoute le fait que les expressions temporelles sont souvent très complexes (comme vu au chapitre 2), ce qui rend la tâche de repérage et d'annotation non triviale. Si ces tâches sont relativement bien effectuées sur ce type de pages, d'autres posent problème et sont présentées dans la suite de ce chapitre.

2.2.2. Pages présentant plusieurs objets

Dans les pages Web qu'Adetoa est amené à traiter, nombreuses sont celles qui présentent (ou au moins mentionnent) plusieurs objets touristiques. C'est le cas des deux exemples ci-dessous qui sont, malgré tout, bien traités par Adetoa. Les zones encadrées ont été correctement repérées et annotées, tout comme d'autres objets (comme *randonnée*, *VTT* ou *tennis*) qui sont également repérés dans ces pages.



Figure 22 : Page présentant deux hôtels



Figure 23 : Page présentant plusieurs objets

La page de la figure 22 a pour but de proposer des hôtels suite à une recherche. Dans ce cas précis, deux hôtels sont proposés. Leurs adresses et périodes d'ouverture sont bien repérées et annotées par Adetoa et elles sont correctement liées. Le fait que les informations sont présentées sous la forme d'un tableau ne pose pas de problème car, dans le code source de la page, les détails concernant chacun des deux hôtels se suivent (voir le point 2.3.1.a. pour plus de détails au sujet des tableaux dans les pages Web).

La page de la figure 23 est celle d'un gîte qui présente les activités accessibles aux alentours. Deux objets y sont correctement annotés et repérés : il s'agit des deux piscines. Pour le troisième objet dont l'adresse est bien repérée et annotée, le type n'est pas parfaitement défini puisque seul *rafting* est repéré. Cela permet toutefois d'identifier que l'adresse correspond à un lieu proposant des sports d'eau vive. Le liage des différents objets est également bien effectué.

Ces exemples permettent donc de montrer la diversité des pages qu'Adetoa arrive à traiter, y compris quand leur complexité est plus grande.

2.3. Difficultés liées à la conception des pages

Comme on l'a vu dans la présentation du corpus, le fait que le contenu de la page Web est « aplati » sous forme de flot de caractères simplifie le traitement automatique. Il faut cependant bien garder en mémoire qu'il s'agit du contenu d'une page Web touristique et non d'un texte narratif. On ne peut donc pas appliquer d'outils linguistiques, tels que des analyseurs syntaxiques, des systèmes d'étiquetage ou d'extraction existants, prévus pour des textes construits et rédigés. Étant donné le type d'informations que je souhaite extraire et la façon dont elles sont présentes sur des pages Web, de tels outils ne seraient pas nécessairement utiles. En effet, sur les pages Web, les horaires d'ouverture, les adresses et les types d'objets constituent le plus souvent de petites zones de texte autonomes. C'est pourquoi une analyse linguistique plus globale de la page n'a pas semblé obligatoire.

Toutefois, ces « zones de texte autonomes » sont parfois morcelées par la structure de la page Web et c'est le problème que posent les tableaux ou les pages dans lesquelles l'ordre des informations dans la page affichée n'est pas le même que dans son code source. De plus, au niveau de leur contenu, certaines pages contiennent de nombreuses informations qui ne concernent pas le même objet. Nous verrons ci-dessous comment cela rend l'analyse plus complexe.

2.3.1. Difficultés structurelles

Les difficultés structurelles sont intrinsèques à la conception des pages Web. Elles sont liées au fait que le code source d'une page Web ne correspond pas directement à la page qui s'affiche dans un navigateur, puisqu'il doit, pour cela, être interprété. Comme nous l'avons déjà précédemment indiqué, nous avons décidé de n'utiliser, pour Eiffel, que le contenu textuel des pages Web, sans en analyser le code source. Les caractéristiques structurelles des pages sont donc perdues, ce qui pose parfois des difficultés pour leur traitement.

a. Le problème des tableaux

Les tableaux sont très fréquents dans les pages Web. S'ils sont présents dans le code source des pages, ils ne sont pas toujours visuellement représentés dans les pages une fois affichées

dans un navigateur. C'est-à-dire que les tableaux sont aussi utilisés par les concepteurs pour mettre en forme les pages Web. Il faut donc distinguer ceux dont l'unique fonction est de structurer spatialement les pages (visée de mise en forme visuelle des informations) de ceux qui jouent un rôle sémantique en structurant l'information elle-même. Ces deux types de tableaux ont donc une vocation différente. Néanmoins, en ce qui concerne le code source de la page, ils sont construits avec les mêmes balises, ce qui ne permet pas de les différencier facilement.

Les codes source de certaines des pages présentées au début de ce chapitre contiennent des tableaux, notamment celui de la page de la figure 18. Ce sont des tableaux de mise en forme, tandis que le tableau suivant est un tableau structurant. [Wang & Hu 2002] distinguent ainsi les « genuine tables » des « non-genuine tables » (« vrais » tableaux et « faux » tableaux).

Horaires	lundi	mardi	mercredi	jeudi	vendredi	samedi
matin	08h30-12h00	08h30-12h00	08h30-12h00	08h30-12h00	08h30-12h00	10h00-11h30
ap. midi	14h00-18h00	14h00-18h00	14h00-18h00	14h00-18h00	14h00-	-

Figure 24 : Tableau contenant des horaires

Les tableaux structurants jouent effectivement un rôle sémantique. En effet, si on ne garde que le texte et que l'on supprime toutes les balises, les données sont alors présentées ainsi :

(72) **Horaires lundi mardi mercredi jeudi vendredi samedi matin 08h30 - 12h00 08h30 - 12h00 08h30 - 12h00 08h30 - 12h00 08h30 - 12h00 10h00 - 11h30 ap. midi 14h00 - 18h00 14h00 - 18h00 14h00 - 18h00 14h00 - 18h00 14h00 - -**

Un tel flot de caractères est totalement incohérent et incompréhensible : sans la structure, le sens est perdu. Toutefois, il faut noter que cela dépend de la granularité de l'information que le tableau structure. En effet, le sens est ici perdu car la granularité est très fine et qu'une information que l'on pourrait considérer comme constituant une unité significative (voir chapitre 1 sur les unités minimales) est divisée par le tableau. En revanche, un tableau peut très bien être structurant mais à un niveau moins fin, de façon à ce que chaque case contienne une unité significative dans son ensemble. Il reste alors interprétable. C'est le cas par exemple du tableau présent dans la page de la figure 22.

Par ailleurs, les tableaux peuvent prendre différentes formes. Ainsi, les informations données dans le tableau de la figure 24 pourraient également être présentées comme dans le tableau 1.

Horaires	Matin	Ap. midi
Lundi	8h30-12h00	14h00-18h00
Mardi	8h30-12h00	14h00-18h00
Mercredi	8h30-12h00	14h00-18h00
Jeudi	8h30-12h00	14h00-18h00
Vendredi	8h30-12h00	14h00-
Samedi	10h00-11h30	-

Tableau 1 : Horaires en tableau vertical

De plus, si la présence des traits qui délimitent les lignes et colonnes du tableau indique qu'il s'agit d'un tableau structurant, elle n'est pas indispensable. Les « vrais tableaux » en sont parfois dépourvus, ce qui ne les empêche pas de conserver leur fonction de structuration. Plusieurs auteurs se sont intéressés à ce problème. [Wang & Hu 2002] ont développé un système à base d'apprentissage pour le repérage des tableaux structurants. Le système de [Gatterbauer et al. 2007] remplit sensiblement la même fonction mais la méthode est très différente : elle se base sur le rendu visuel de la page Web et non sur son code source.

Ces systèmes proposent donc des solutions pour repérer les tableaux structurant l'information dans les pages Web. Néanmoins, ils ne proposent pas de solution pour les interpréter. Dans le cadre de mes travaux, le repérage seul n'est pas suffisant ; les données doivent aussi être interprétées et ce n'est pas une tâche triviale. En effet, pour donner du sens au contenu d'un tableau, il faut pouvoir trouver la signification des lignes et des colonnes et les associer afin d'en restituer le sens. Par exemple, le tableau de la figure 24 n'est utile que si l'on peut reconstruire les horaires et obtenir l'information suivante :

(73) **Horaires : du lundi au jeudi, de 8h30 à 12h00 et de 14h à 18h. Le vendredi de 8h30 à 12 h et de 14h à x[27] . Le samedi de 10h à 11h30.**

La question de la reconstruction du sens des tableaux, notamment de ceux présents dans des pages Web, est importante et de nombreux auteurs s'y sont intéressés. Comme cela a déjà été présenté, [Tengli et al. 2004] ont travaillé sur l'interprétation des tableaux dans des pages Web. Leur système est fondé sur une méthode d'apprentissage qui leur a permis de traiter un grand nombre de tableaux dont l'organisation et le contenu sont proches. Les autres méthodes d'extraction d'information dans les tableaux de pages Web, à base de wrappers (voir [Cohen et al. 2002], [Nguyen 2006]) ou d'analyse de patrons (voir [Crescenzi & Mecca 2004], [Chang & Lui 2001]) s'appuient également sur la régularité des tableaux.

Or, comme je l'ai déjà mentionné, mon corpus ne présente aucune régularité entre les différentes pages qui le constituent ; il n'y en a donc pas non plus au niveau des tableaux. Ces méthodes ne sont, de fait, pas réellement appropriées à mes données.

La mise au point d'un système propre à mon corpus pour le repérage et l'interprétation des tableaux structurant l'information est donc très complexe. De plus, le bénéfice est mince compte tenu de la faible quantité de tableaux structurants dans les pages que j'ai à traiter.

27 Information incomplète dans le tableau.

En ce qui concerne les tableaux sans sémantique structurelle, c'est-à-dire les tableaux qui ne servent qu'à la mise en forme et non pas à la structuration de l'information, la difficulté est moindre. Dans mon corpus, de nombreuses informations sont contenues dans de tels tableaux et cela aurait donc pu être très problématique. Néanmoins, l'étude de corpus et l'analyse du code source des pages m'ont révélé que ces tableaux ne « divisent » pas l'information. Chaque cellule contient en fait une « zone de texte autonome » : si le tableau n'a pas pour rôle de structurer l'information, alors cette information n'est pas morcelée dans plusieurs cellules. Pour les tableaux de mise en forme, qui sont les plus fréquents, une analyse du contenu textuel seul est donc la plupart du temps suffisante.

Ces observations m'ont amenée à m'interroger sur l'utilité du traitement des tableaux. L'investissement nécessaire pour le repérage et l'interprétation des tableaux structurant l'information est important mais n'est pas rentable étant donnée leur marginalité. J'ai donc choisi de ne pas prendre en compte les informations structurelles données par les tableaux.

b. Le problème de l'ordre des données

La page donnée en figure 25 (page 72) est complexe à interpréter et les difficultés se situent à plusieurs niveaux. Il s'agit de la page d'une crêperie située dans un manoir. La page indique l'adresse de ce manoir et les horaires d'ouverture de la crêperie. Mais ensuite, d'autres informations sont données : des détails sur deux manoirs différents, des informations au sujet d'hébergements dans la région. Il est donc difficile de relier les informations aux objets qu'elles concernent effectivement.

La connaissance de la disposition visuelle d'une page Web telle qu'elle est affichée par un navigateur pourrait être utile ; on pourrait alors établir des règles du type « l'objet le plus haut dans la page est l'objet dont il est question » ou « l'adresse apparaît toujours en dessous de l'objet auquel elle est rattachée ». Mais le code source de la page, en HTML, ne contient pas ces informations. En effet, l'ordre du texte dans le code source n'est pas toujours le même que celui qui est affiché dans le navigateur.

Par exemple, dans le code de cette page (représenté dans la figure 26, page 73), le texte *Conception, Réalisation et Hébergement* se trouve avant le texte *Crêperie à la ferme Manoir de Trouzilit* (encadrés en bleu dans l'image). L'ordre de ces informations n'est ici pas intéressant mais cet exemple me permet d'illustrer ce phénomène d'ordre de lecture non respecté qui peut parfois être plus problématique.

Différentes informations sont tout de même repérées dans cette page : la localisation (code postal et ville) et une partie de l'expression temporelle. L'expression complète est complexe : ***Toute l'année*** *en VSD midi et soir. Tous les jours juillet et aout*. Seule la partie en caractères gras est repérée car *en VSD* n'est ni répandu, ni très « français » et n'est donc pas prévu dans Adetoa. Cela bloque le repérage du reste de l'expression. Au niveau des objets, plusieurs informations sont repérées mais le terme *location* est interprété comme un hébergement alors qu'il s'agit ici d'une location de salle. De plus, la page présente en réalité deux objets qui sont liés (le manoir, que l'on peut visiter, qui est un gîte et la crêperie, qui est située à l'intérieur de ce manoir). Une seule adresse correspond aux deux objets. Les horaires affichés en haut correspondent à la crêperie, l'expression *ouvert toute l'année* correspond au manoir en lui-même. La question des pages présentant plusieurs objets sera discutée par la suite mais les problèmes que cela pose, dans ce cas précis, ne concernent pas directement le fait que

plusieurs objets sont présentés, mais plutôt l'organisation globale de la page dans laquelle les informations sont mêlées et où le liage est donc difficile à effectuer.

Par ailleurs, plus bas dans la page, un encadré mentionne encore les caractéristiques du lieu (il n'apparaît pas dans l'image ci-dessus) : type de gîte et prix des chambres d'hôtes. Un nouvel objet est certes repéré mais cela ne permet pas davantage de bien lier les différentes informations entre elles.

Dans ce type de pages, et compte tenu du fait que la méthode utilisée ne s'intéresse en réalité qu'à de petits fragments correspondant, comme indiqué précédemment, à des « zones de texte autonomes », l'ordre des informations n'est que rarement dommageable pour le repérage et l'annotation. Il l'est un peu plus pour ce qui est du liage des informations, mais l'utilisation d'une règle simple, présentée au chapitre 5 et évaluée au chapitre 7, s'est tout de même souvent avérée pertinente.



Figure 25 : Page Web d'une crêperie

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><head>
<title>Auberge - Chambre d'hote - Ferme - Auberge du Trèfle</title>
[...]
</head><body onload="[...]">
<table border="0" cellpadding="0" cellspacing="0" height="100%"
width="100%">
<tbody><tr>
<td align="left" valign="top"><table border="0" cellpadding="0"
cellspacing="0" height="100%" width="100%">
<tbody>
[...]
</tbody></table></td>
</tr>
<tr>
<td align="left" valign="top"><table border="0" cellpadding="0"
cellspacing="0" height="100%" width="760">
<tbody><tr>
<td align="left" valign="top"><table border="0" cellpadding="0"
cellspacing="0" height="100%" width="100%">
<tbody><tr>
<td align="left" valign="top"> <table border="0" cellpadding="0"
cellspacing="0" height="100%" width="150">
<tbody><tr>
[...]
<td class="fondcellule4" align="left" height="200" valign="top">
<div align="center">
<p> </p>
<p><a onmouseover="MM_swapImage('Image29','','images/marques/
logo32x32-over.gif',1)" onmouseout="MM_swapImgRestore()" href="http://
www.virtual-creation.com/" target="_blank">
<span class="unnamed1">
Conception, Réalisation <br>et Hébergement <br>
VIRTUAL CREATION</span></a></p>
<p> </p>
</div></td>
</tr>
</tbody>[...]</table></td>
<td align="left" valign="top"><table class="fondcellule3" border="0"
cellpadding="0" cellspacing="0" height="430" width="100%">
<tbody><tr>[...]
<td align="left" valign="top"><table border="0" cellpadding="0"
cellspacing="0" height="100%" width="100%">
<tbody><tr>
<td width="15"> </td>
<td align="left" valign="top"><table border="0" cellpadding="0"
cellspacing="0" height="100%" width="100%">
<tbody><tr>
<td align="left" valign="top">
<table border="0" cellpadding="4" cellspacing="1" width="600">
<tbody><tr>
<td class="gdtitre" align="center"><strong><font size="5">Crêperie à
la ferme MANOIR DE TROUZILIT</font></strong></td>
[...]
</tbody></table>
</body></html>
```

Figure 26 : Code source simplifié de la page de la ferme-auberge

2.3.2. Difficultés liées au contenu des pages

Les pages qu'Adetoa peut traiter sont des pages touristiques comprenant des informations pratiques. Toutefois, cette définition reste vaste et de nombreuses pages y correspondent sans pour autant pouvoir être traitées, ou du moins facilement. Il s'agit principalement de pages présentant plusieurs objets touristiques, sous la forme d'un agenda ou autre. Une difficulté supplémentaire, concernant les informations contenues dans des images, est ensuite présentée.

a. Le problème des pages-agenda

Les pages présentant un agenda posent des problèmes d'interprétation car elles se rapportent à plusieurs objets touristiques. Il faut donc lier correctement les différentes informations correspondant à un objet (la date avec le lieu et l'objet). La page présentée dans la figure 27, pose des problèmes pour une interprétation automatique : cette page est en effet simple à comprendre pour un lecteur humain mais je vais montrer qu'elle ne l'est pas pour un traitement automatique.



Figure 27 : Exemple de page-agenda

Cette page présente l'agenda d'une communauté de communes. À chaque événement sont associés une date, le type d'événement (brocante, balade, etc.) et un lieu (nom de la commune). Visuellement, il est alors facile d'interpréter que la date indiquée en gras concerne l'événement qui suit. Toutefois, pour la traiter de façon automatique, il faudrait en tirer une règle générale du type « associer chaque date à l'élément qui suit » ou « lier toutes les informations qui se trouvent dans une même cellule de tableau » (il faut en effet noter que, même s'il n'est pas visible, c'est un tableau qui structure cette page). Le problème que posent les règles de ce type est qu'elles ne sont en réalité pas généralisables. En effet, sur de nombreuses pages Web, des tableaux contiennent des dates qui ne sont pas nécessairement à lier aux autres informations de la cellule. La première règle proposée n'est pas non plus généralisable : les dates figurent parfois après l'événement auquel elles se rapportent. Si le liage est évident pour un humain qui consulte la page, comment effectuer ce liage automatiquement ? Voici quelques exemples qui illustrent l'impossibilité d'établir des règles d'interprétation généralisables.

Les quatre pages présentées ici contiennent un agenda. Toutes les entrées de ces agendas

contiennent au moins trois éléments : la date (et éventuellement l'heure), le lieu (ville, salle, etc.) et l'objet touristique (nom de l'artiste en concert, événement, activité, etc.). Pour un humain, le liage de ces différentes informations est donc évident grâce à l'organisation visuelle des pages : le jeu des polices de caractères, la disposition des informations, les puces sont autant d'indices qui permettent d'interpréter la page. Si l'on regarde ces pages méthodiquement, il est possible d'établir des règles d'interprétation pour chacune d'entre elles. Par exemple, pour la page de la figure 27, on peut reprendre les deux règles proposées plus haut :

- Lier la date à l'événement et au lieu qui suivent
- Lier toutes les informations qui se trouvent dans une même cellule de tableau



Figure 28 : Extrait d'une page-agenda (1)

Néanmoins, des règles de ce type ne sont pas généralisables et ne peuvent pas s'adapter aux autres pages, même si celles-ci présentent aussi un agenda.

Par exemple, pour la page de la figure 28, l'ensemble de la colonne constitue une cellule d'un tableau, donc la deuxième règle ne peut pas s'appliquer. En ce qui concerne la première, l'ordre des informations n'est pas le même. Sur cette page, l'ordre est « nom – date – lieu » et non plus « date – nom – lieu ». L'utilisation de la règle proposée ci-dessus mènerait à faire le liage erroné suivant : « Vendredi 28 août 2009 – 17h30 Jardin des Bénédictins - Concert gratuit Bluetones ».



Figure 29 : Extrait d'une page de ville avec agenda

Dans la page de la figure 29, l'ensemble du cadre « Agenda » constitue aussi une cellule de tableau et n'est pas subdivisé. La deuxième règle n'est donc pas pertinente. En ce qui concerne l'ordre des éléments, c'est en revanche le même que dans la page de la figure 27. Il serait donc possible d'utiliser la même règle. Le problème qui se pose alors est le suivant : comment savoir si une règle peut être appliquée ou non ? D'autres règles pourraient être établies pour cela, comme par exemple : « si le premier élément est une date alors la règle est applicable ». Mais quel est le premier élément ? Celui de la page, de la cellule du tableau ou du paragraphe ? Dans la figure 29, le premier élément du cadre serait le bon si le titre *Agenda* qui le précède ne faisait pas déjà partie du cadre. Dans la figure 27, *Agenda* et *en Juillet* figurent avant la première date. Beaucoup de paramètres rentrent donc en ligne de compte pour savoir quelle règle est appropriée et déterminer ensuite si et comment elle est applicable.

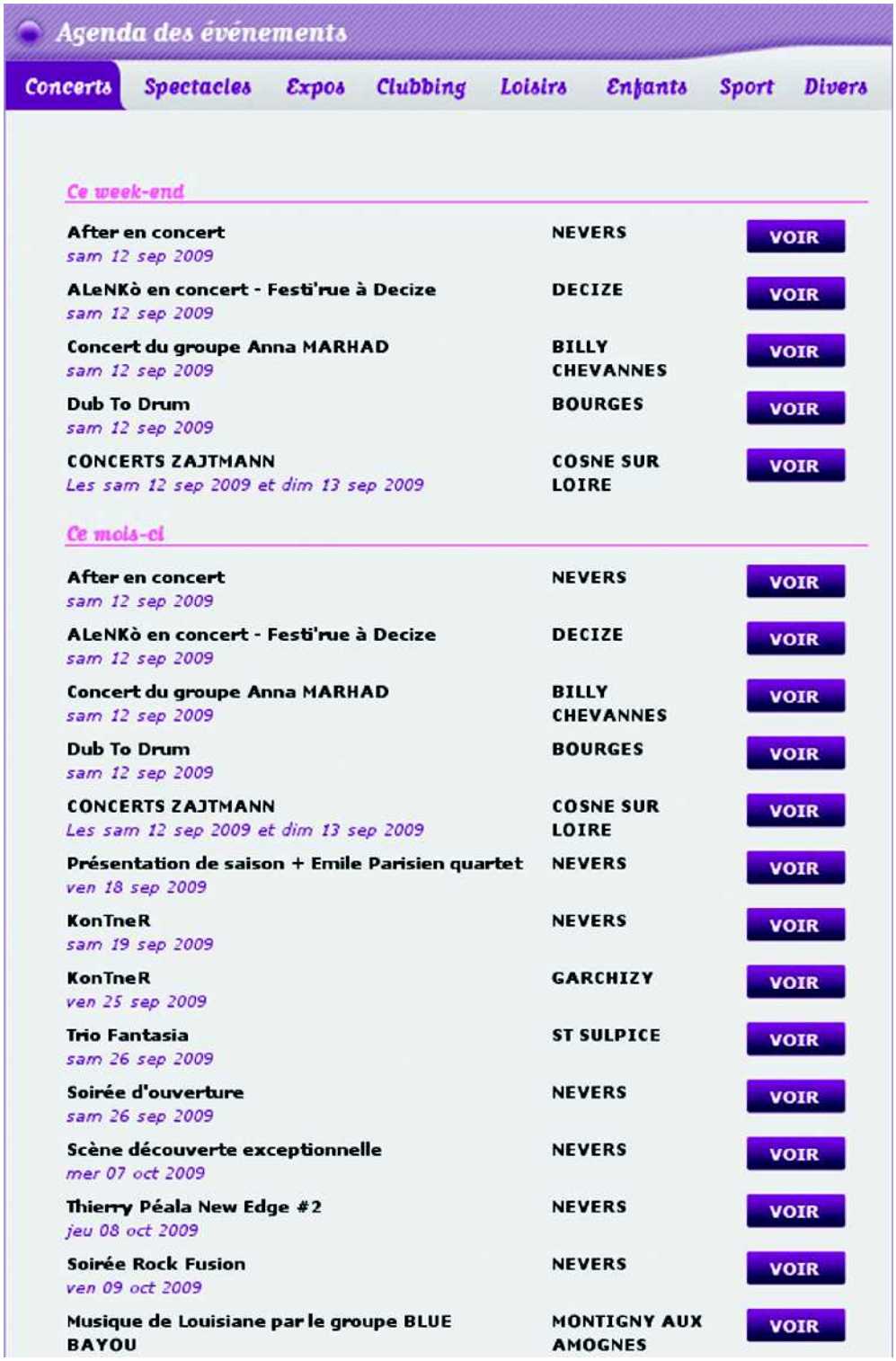


Figure 30 : Extrait d'une page-agenda (2)

La structure de l'agenda de la page de la figure 30 est un peu différente car il s'agit d'un tableau à plusieurs colonnes. La première règle, qui évoque le liage de toutes les informations d'une même cellule, n'est donc pas applicable. Ici, il faudrait lier toutes les informations d'une même ligne de tableau. L'ordre des données est perturbé par la structure du tableau : si on lit celui-ci de gauche à droite, sans tenir compte des cellules, l'ordre est alors « nom – lieu – date », ordre non rencontré dans les exemples précédents. Si on lit le tableau par cellule, on obtient l'ordre « nom – date – lieu », comme dans la page de la figure 30.

L'analyse de ces différentes pages-agenda révèle que, si leur interprétation semble simple pour un internaute humain, elle ne l'est pas pour un système automatique. De nombreux paramètres sont à prendre en compte et les pages Web peuvent prendre des formes très variées. En effet, les concepteurs de pages Web font en sorte que celles-ci soient claires visuellement et utilisent pour cela des procédés difficilement interprétables automatiquement. Ainsi, l'organisation visuelle des pages-agenda est importante et joue un grand rôle dans leur bonne interprétation. Ces pages ont d'ailleurs des points communs dans leur mise en forme. Comme précisé plus haut, chaque entrée d'agenda contient au moins trois informations. Chacune de ces informations possède, en général, une mise en forme spéciale qui se répète pour chaque entrée. Ainsi, dans la page de la figure 27, la date est en gras, le nom de l'événement, en caractères standard, et la ville, en lettres capitales. Dans l'agenda de la figure 30, seul le nom de l'événement (en l'occurrence nom du groupe en concert) se distingue du reste et est affiché en bleu, tandis que le reste est en noir. Un saut de paragraphe et le nom du groupe en bleu permettent toutefois de distinguer les différentes entrées. Dans la page de la figure 29, les informations sont d'une part structurées par des puces qui indiquent le début d'une nouvelle entrée, d'autre part, par un code de couleurs : la date et le nom de l'événement sont en orange tandis que le reste est en noir. Enfin, dans l'agenda de la figure 30, le nom du groupe en concert est en gras, la date est en violet et en italiques et le lieu est en lettres capitales.

Comme je viens de l'illustrer, l'utilisation de polices de caractères et de couleurs différentes permet de structurer facilement les informations en faisant apparaître les « groupes d'informations », c'est-à-dire les informations qui sont à lier entre elles. Néanmoins, cette organisation n'est claire que pour un utilisateur humain et reste difficilement formalisable pour un traitement automatique. De plus, de nombreux outils techniques existent pour mettre en forme des pages Web et les codes HTML qui en résultent peuvent différer. Ainsi, plusieurs codes source sont possibles pour un même rendu visuel.

Certains travaux d'extraction d'information dans les pages Web peuvent se rapprocher du cas des pages-agenda. [Novotný et al. 2009] ont travaillé sur des sites marchands : sur une page contenant plusieurs articles, ils ont, par exemple, extrait toutes les informations qui se rapportaient à chaque article. Le système décrit dans [Habegger & Qualafou 2004] a été testé sur l'extraction des descriptions de DVD sur le site d'Amazon[28]. Le point commun entre ces travaux est qu'ils fonctionnent sur des pages Web structurées, voire semi-structurées. Les informations liées à un même objet apparaissent dans une structure donnée qui est identifiable. Des structures peuvent être repérées dans certaines pages-agenda. Néanmoins, une fois de plus, la régularité fait défaut et il n'est donc pas possible de généraliser ces structures. La mise en œuvre d'un système d'extraction propre aux pages-agenda nécessiterait probablement un algorithme d'apprentissage.

Par ailleurs, les résultats des études en analyse du discours comme [Ho-Dac & Péry-Woodley

28 www.amazon.fr

2009] qui proposent de considérer les adverbiaux temporels comme des marqueurs de segmentation discursive pourraient être transposables aux pages-agenda. Par exemple, pour la page de la figure 30, les titres *ce week-end* et *ce mois-ci* permettraient de poser des cadres. Les informations se trouvant à l'intérieur seraient sous la portée de l'expression titre. Toutefois, pour que cela soit réellement applicable, il faudrait que la structure du code source de la page respecte ces cadres et que cela se reflète dans l'arbre. Or, comme de nombreux exemples l'ont montré, la structure de l'arbre du code de la page ne respecte pas nécessairement la logique des données qu'il contient. Les cadres temporels sont bien posés linguistiquement par les titres *ce week-end* ou *ce mois-ci* mais ils ne sont pas représentés par un élément technique dans le code source de la page.

Cette approche discursive et les approches à base d'apprentissage présentées ci-dessus m'ont semblé trop éloignées de mes travaux et difficilement intégrables à la chaîne de traitement prévue dans le cadre d'Eiffel. J'ai donc choisi d'exclure ce type de pages de mon champ d'application.

b. Le problème des pages mentionnant plusieurs objets

La page présentée dans la figure 31 est complexe à interpréter car, en plus de présenter un objet principal, elle mentionne de nombreux autres objets touristiques, objets satellites. De plus l'objet principal est difficile à identifier.



Figure 31 : Page d'un centre de loisirs mentionnant d'autres objets

En effet, il est question, dans cette page, d'un parcours d'aventures, d'un rando-raid et d'une journée nature. Sont également mentionnées de nombreuses activités : VTT, canoë-kayak, randonnée et, dans la colonne à gauche : rafting, quad, équitation, etc. De nombreux objets sont ainsi repérés et annotés par Adetoa mais il est difficile de savoir de quoi il est exactement question. En réalité, cette page est la page d'un centre de loisirs *ABLoisirs* qui propose différentes activités. Ce centre constitue donc l'objet touristique principal, et c'est lui qui devrait être repéré, ainsi que son adresse. Mais comment le repérer automatiquement ? La difficulté réside dans le fait que seul son nom apparaît, mais pas son type : l'expression *centre de loisirs* n'apparaît nulle part dans la page. Même en se basant sur le nom, contenant le mot *Loisirs*, il est impossible d'en déduire automatiquement qu'il s'agit d'un centre de loisirs.

D'autres noms sont en effet construits de la même façon mais ne concernent pas des centres de loisirs, comme par exemple *France Loisirs* qui désigne une librairie / un club de lecture. Ce sont les connaissances du monde et les sens cognitifs humains qui permettent à un internaute de comprendre qu'il est, dans cette page, question d'un centre de loisirs.

Néanmoins, ces pages sont assez marginales et les objets sont, la plupart du temps, clairement énoncés. Un critère de proximité permet de décider comment lier les informations les unes aux autres : chaque localisation ou information temporelle est à lier à l'objet le plus proche, qui la précède ou qui la suit. Cette mise en œuvre sera présentée plus en détail au chapitre 5.



Figure 32 : Page présentant plusieurs objets

Dans la page de la figure 32, plusieurs objets touristiques sont présentés. Chacun est à lier à une adresse et, éventuellement, à des horaires. La règle de proximité permet de lier les bonnes informations entre elles : le premier objet est à lier à la première adresse et aux premiers horaires.

Notons que cette règle ne peut fonctionner que lorsque les informations à repérer apparaissent, sur la page Web, dans le même ordre que dans son code source. Comme je l'ai déjà montré, l'ordre du texte, tel qu'il apparaît dans un navigateur, n'est pas toujours le même que dans le code source. Toutefois, si l'ordre peut différer lorsque la page contient, par exemple, des tableaux structurants, les informations qui sont proches dans la page Web affichée le sont généralement aussi dans le code source.

Cette règle de proximité me permet ainsi de traiter de nombreuses pages présentant ou mentionnant plusieurs objets touristiques, sans qu'un traitement lourd soit nécessaire. Son implémentation sera présentée au chapitre 5.

c. Le problème des informations contenues dans des images

Dans certaines pages, des informations, y compris pratiques, se trouvent dans des images. Cela pose problème car elles ne sont alors pas du tout accessibles : elles ne figurent, en général, pas sous forme textuelle. Dans certaines pages, les images peuvent avoir, dans le code source, un nom ou un <caption> contenant une partie de l'information, mais ce champ est généralement mal renseigné.

Dans la page de la figure 33, le type d'objet (*hotel, resto*), ainsi que les informations d'ouverture sont bien repérés et annotés par Adetoa (informations encadrées en bordeaux). Toutefois, le nom et le type, dans le bandeau du haut de la page, se trouvent, en réalité, dans une image. Il en est de même pour le bandeau du bas qui contient l'adresse (informations encadrées en bleu). Le type étant répété sous forme de texte ailleurs dans la page, il est repéré mais l'adresse n'apparaît que sous forme d'image et est donc inaccessible à tout traitement automatique basé sur du texte. De plus, lorsqu'une image se trouve sur une page Web, un traitement textuel ne permet pas de savoir si elle contient des données informationnelles ; seul un traitement poussé d'analyse d'image avec reconnaissance de caractères peut avoir accès à ces informations.



Figure 33 : Page d'un hôtel-bar-resto

Les pages contenant des informations situées dans des images ne peuvent donc pas être exclues. Elles sont analysées comme les autres par Adetoa qui en analyse le contenu textuel. Les cas où les informations de localisation et les expressions temporelles figurent uniquement dans des images, et nulle part ailleurs dans la page Web, restent relativement marginaux. Les noms des ressources touristiques et leur type se trouvent plus souvent dans des images mais sont aussi souvent répétés dans la page, ce qui permet tout de même de les repérer.

Conclusion

Ce chapitre a permis de mettre en avant deux aspects du traitement automatique des pages Web.

Le premier, propre au fonctionnement d'Adetoa, est le suivant : en définitive, le seul critère permettant d'indiquer si une page est correctement traitable ou non est lié à l'organisation du contenu dans le code source de la page. Ainsi, si toutes les informations y figurent dans un « ordre logique », le traitement peut avoir lieu, tandis que si l'ordre et bouleversé, notamment

par des informations imbriquées, le traitement ne pourra pas être correctement effectué. L'arbre ci-dessous permet de visualiser les cas dans lesquels le traitement peut être bien mené par Adetoa.

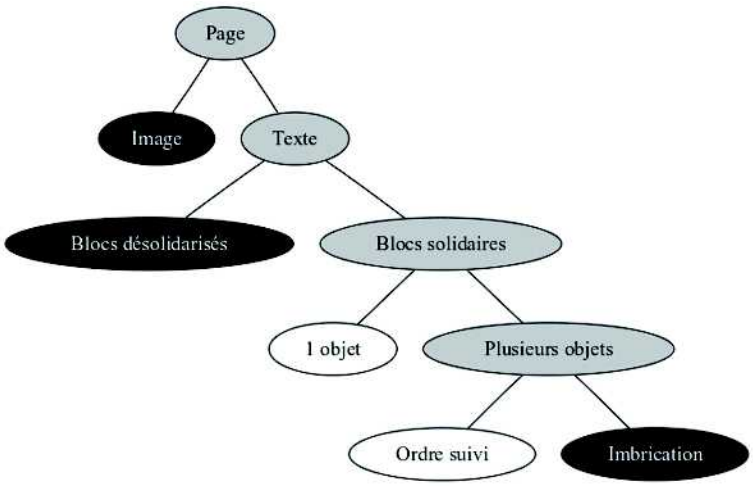


Figure 34 : Arbre de décision montrant les pages traitables par Adetoa

Les feuilles noires indiquent les cas non traitables tandis que les blanches montrent les cas pouvant être traités avec succès par Adetoa. Les nœuds « blocs solidaires » et « blocs désolidarisés » font référence aux entités minimales : si une unité de signification apparaît de façon suivie dans le code source de la page, alors elle constitue un « bloc solidaire », tandis que si elle ne l'est pas, elle constitue un « bloc désolidarisé » et le traitement ne peut pas être effectué. La distinction entre les pages présentant un seul objet et les pages qui en présentent plusieurs permet de montrer que l'ordre général n'a pas d'incidence sur les pages contenant un seul objet, à condition que les blocs significatifs ne soient pas désolidarisés. En revanche, si une page présente plusieurs objets, toutes les informations concernant un objet doivent se suivre car si ce n'est pas le cas et qu'elles sont, par exemple, imbriquées, le traitement ne peut pas être efficace.

Par ailleurs ces traitements sont possibles car les pages étudiées sont en HTML et que le contenu textuel est accessible dans le code source de la page. Les nouvelles technologies utilisées pour la conception de pages Web (JavaScript – AJAX[29] – Flash) empêchent de tels traitements car le contenu textuel n'est pas toujours présent dans le code source mais peut être construit, à la volée, grâce à l'exécution d'un code source et à la récupération de données par interrogation d'un serveur distant [Garron et al. à par.].

Le deuxième aspect est lié au rendu visuel des pages Web. À la simple vue d'une page Web, rien ne permet de savoir si celle-ci est traitable par Adetoa ou non. Une page avec un rendu visuel extrêmement simple peut être codée en HTML de façon très complexe, avec des tableaux, des imbrications, etc., et poser alors des difficultés de traitement. L'inverse est également possible : une page visuellement complexe, contenant de nombreuses informations, affichées sous forme de tableaux etc., peut être codée très simplement, sans

29 AJAX (*Asynchronous JavaScript and XML*) désigne un ensemble de technologies utilisées pour rendre dynamique le contenu d'une page Web.

bouleverser l'ordre logique du contenu textuel et permettre ainsi un traitement efficace par Adetoa.

Visuellement, les deux pages ci-dessous sont très différentes. La première (figure 35), qui semble très simple, contient deux images et du texte. Celui-ci est centré, sans mise en page particulière, et ne contient aucune colonne. La page de la figure 36 semble en revanche bien plus complexe puisque le texte est réparti sur plusieurs colonnes, qu'il y a un bandeau supérieur, etc. Or, il se trouve qu'au niveau du code source, ces deux pages sont relativement simples et qu'elles sont toutes les deux traitées correctement par Adetoa. Pourtant, chacune des deux aurait pu être codée très différemment et avoir exactement le même rendu visuel. Avec des codes sources différents, elles auraient alors pu ne pas être traitables. Par exemple, pour la deuxième page, les horaires auraient pu être stockés dans un tableau dans lequel les informations auraient figuré dans un ordre différent de celui de la lecture, ce qui aurait gêné le repérage.

De façon plus générale, cela m'a permis de montrer qu'il n'est pas possible d'établir un lien fort entre l'apparence visuelle d'une page Web et la structure technique dont est composé son code source. Il n'est donc pas possible de déterminer si une page est traitable ou non par Adetoa en fonction de son rendu visuel.



Figure 35 : Page « simple »



Figure 36 : Page « complexe »

Chapitre 4. Ontologie et Schéma d'Annotation

Introduction

Maintenant que les données sur lesquelles j'ai travaillé ont été présentées (expressions temporelles, chapitre 2) et que les contraintes liées au support (pages Web touristiques, chapitre 3) ont été exposées, je vais m'intéresser à la façon dont les données peuvent être annotées afin d'être exploitées. Pour ce faire, les données doivent tout d'abord être repérées et extraites parmi toutes les informations disponibles. Elles doivent ensuite pouvoir être modélisées dans l'ontologie pour être stockées dans la base de connaissance, comme représenté dans la figure 37.

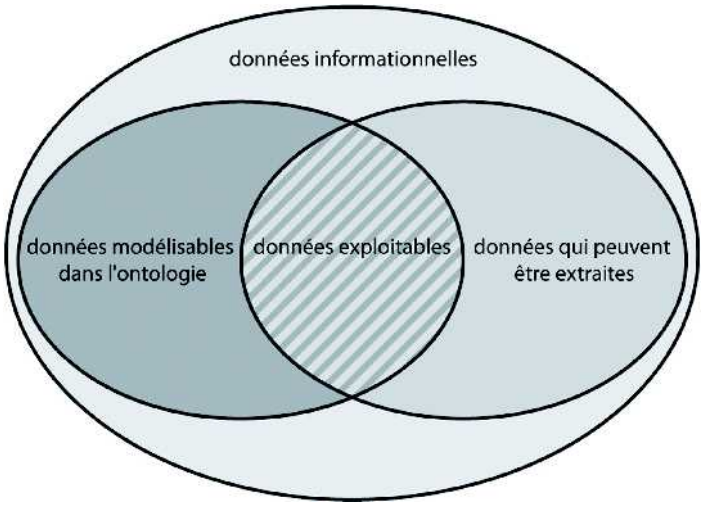


Figure 37 : Ensembles des données

Je vais donc m'intéresser aux ontologies qui sont, la plupart du temps, centrales dans des projets d'annotation. Tout d'abord, un bref état de l'art rappellera les différentes notions utiles à la compréhension de ce chapitre. Je présenterai ensuite plus en détail l'ontologie développée dans le cadre d'Eiffel et comment les données temporelles touristiques y sont modélisées. Le projet Eiffel constitue un cadre pour ce chapitre, sans toutefois le limiter :

Adetoa pourrait être adapté à d'autres applications et des données plus larges que celles utilisées dans le projet sont également présentées.

Enfin, je décrirai le schéma d'annotation que j'ai mis au point pour Eiffel et sur lequel sont basées les annotations effectuées par Adetoa.

Ce chapitre permet de montrer le rôle central de l'ontologie, ainsi que l'influence que les annotations peuvent avoir sur cette dernière, et donc la manière dont elle a évolué au cours du projet.

1. État de l'art autour des ontologies et de l'annotation sémantique

Dans le cadre du projet Eiffel, une ontologie a été développée. Cette ontologie est centrale au projet car elle permet d'en guider plusieurs étapes : elle guide aussi bien le repérage et l'annotation que l'interrogation du système par l'utilisateur. Avant de présenter plus en détail cette ontologie, je vais d'abord faire un rappel de quelques notions concernant les ontologies et autres ressources ontologiques et terminologiques (RTO).

1.1. Ontologies – ontologie et autres RTO

Ce n'est qu'au début des années 90 que le terme d'« ontologie » se répand dans les domaines de l'informatique, de l'intelligence artificielle et des sciences de l'information. Il est emprunté au domaine de la philosophie, et défini ainsi dans le Petit Robert 2001 : « Partie de la métaphysique qui s'applique à l'être en tant qu'être, indépendamment de ses déterminations particulières ». On distingue alors l'Ontologie (avec un « O » majuscule) en tant que discipline philosophique et les ontologies (avec un « o » minuscule) que s'est approprié le domaine de l'ingénierie des connaissances. Comme on le verra à l'aide de définitions plus formelles des ontologies, l'emprunt de ce terme est motivé car les ontologies permettent de modéliser des objets existants et de représenter les concepts correspondant à ce qui existe dans un domaine. [Charlet et al. 2004] donnent cette première définition du terme :

> « Ensemble des objets reconnus comme existant dans le domaine. Construire une ontologie c'est aussi décider de la manière d'être et d'exister des objets. »

Cette définition facilite le lien entre l'Ontologie philosophique et les ontologies de l'ingénierie des connaissances en ce sens que ces deux termes s'intéressent à la notion d'existence. Mais elle reste très abstraite, tandis que les ontologies mènent à des réalisations concrètes. Je vais donner ici quelques définitions trouvées dans la littérature au sujet des ontologies dans le domaine de l'ingénierie des connaissances[30].

> « Une ontologie est une modélisation d'un domaine donné selon une certaine vue du monde. Cette ontologie est conçue comme la spécification d'un ensemble de concepts – *e.g.* entités, attributs, processus –, leurs définitions, leurs interrelations et différentes propriétés et contraintes associées. » ([Laublet 2007])

Cette première définition simplifiée de Laublet est proche de celle de [Charlet et al. 2004] qui complètent avec :

> « Une ontologie peut prendre différentes formes mais elle inclura nécessairement un

30 Voir aussi la thèse de Florence Amardeilh [Amardeilh 2007] pour un état de l'art plus complet des ressources terminologiques et ontologiques.

> vocabulaire de termes et une spécification de leur signification. »

On reviendra plus tard sur la notion de *vocabulaire*.

[Bourigault et al. 2004] proposent une définition plus applicative, liée aux travaux concernant les Systèmes à Base de Connaissance et le Web sémantique :

> « Une ontologie est une conceptualisation des objets du domaine selon un certain point de vue, imposé par l'application. Elle est conçue comme un ensemble de concepts, organisés à l'aide de relations structurantes, dont la principale, celle avec laquelle est construite l'ossature taxonomique de l'ontologie, est la relation *is-a*. Cette conceptualisation est écrite dans un langage de représentation des connaissances, qui propose des « services inférentiels » (classification de concepts, capacité de construire des concepts définis à partir de concepts primitifs, etc.). »

Plus concrètement encore, [Gruber 2008] définit les ontologies sur le plan technique :

> « In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourses. »
>
> Dans le contexte de l'informatique et des sciences de l'information, une ontologie définit un ensemble de primitives de représentation avec lesquelles modéliser un domaine de connaissance ou de discours. [Ma traduction]

Ces différentes définitions permettent de mettre en avant les nombreuses facettes que les ontologies révèlent et les utilisations auxquelles elles peuvent mener. De plus, elles laissent entrevoir l'existence d'un langage formel de représentation des connaissances. En effet, la création d'ontologie nécessite l'existence d'un langage formel grâce auquel il est possible de modéliser, pour un domaine, les différents concepts qui le définissent et les relations sémantiques qui existent entre ces concepts. En reprenant des travaux antérieurs, [Laublet et al. 2009] montrent bien l'ambivalence des ontologies :

> « les ontologies telles que définies dans le cadre de l'Ingénierie des Connaissances, représentent à la fois des objets de consensus pour les humains et des artefacts formels permettant leur exploitation par un agent logiciel (Laublet 2007) » ([Laublet et al. 2009])

Ainsi, si les ontologies sont un objet de consensus permettant aux humains de partager des connaissances, elles ont aussi pour but d'être utilisées par des applications informatiques variées, de sorte qu'elles puissent avoir accès à la même information. Cela exclut le langage naturel pour les représenter ; elles doivent en effet être interprétables par une machine. Les langages OWL[31] et RDF[32] permettent de modéliser des ontologies selon des représentations standards et consensuelles et sont recommandés par le W3C[33].

Ces définitions mentionnent également certaines notions sur lesquelles il est intéressant de revenir : les ressources terminologiques et ontologiques d'une part, et les notions représentées dans les ontologies d'autre part.

En ce qui concerne les éléments d'une ontologie, il s'agit des concepts, des attributs et des relations [Amardeilh 2009]. Les concepts représentent les notions du monde réel que l'ontologie modélise. Il peut s'agir d'objets concrets ou abstraits, simples ou complexes. La structuration des concepts de beaucoup d'ontologies se limite à une taxonomie, dans le sens d'une arborescence (nous verrons par la suite plus précisément ce qu'est une taxonomie). Par

31 Web Ontology Language - http://www.w3.org/TR/owl-features/
32 Resource Description Framework - http://www.w3.org/RDF/
33 http://www.w3.org/2004/OWL/

exemple, dans une ontologie du tourisme, le concept « Infrastructure » aura comme sous-classe le sous-concept « Hébergement » qui aura lui-même des sous-concepts plus précis.

Une relation représente le lien sémantique entre deux concepts qui sont en fait le domaine et le codomaine de cette relation. Les relations permettent des représentations plus complexes de la connaissance du domaine. Par exemple, toujours dans le domaine du tourisme, le concept « Hôtel » pourra entretenir une relation avec le concept « Période_Ouverture ».

Les attributs sont les caractéristiques liées à un concept afin de le définir. Par exemple, le concept « Hôtel » peut avoir les attributs « nom », « nombre_étoiles », « nombre_de_chambres », etc. Les valeurs des attributs sont littérales, il peut s'agir d'une chaîne de caractères, d'un nombre entier, d'un booléen, etc.

En OWL, les concepts sont appelés classes et les relations et attributs sont appelés propriétés. On distingue toutefois les object-properties pour les relations des data-type-properties pour les attributs.

Ces considérations sur les ontologies sont liées à ce que l'on appelle les ressources terminologiques ou ontologiques (RTO) [Bourigault et al. 2004] qui permettent de situer les ontologies dans un contexte plus large. En effet, comme on a pu le remarquer dans cette partie, les ontologies sont basées sur différents éléments. Par exemple, on a pu lire que les ontologies incluaient nécessairement un vocabulaire (en fait au sens de la logique), mais cette notion de vocabulaire est plus diverse dans le contexte des systèmes d'information et de la documentation. De plus, les ontologies informatiques peuvent sembler, en partie, issues de méthodes plus anciennes utilisées par les documentalistes. Je ne pouvais donc pas parler des ontologies sans m'arrêter sur ces différentes notions.

La notion de ressources terminologiques ou ontologiques permet de relier les domaines de la terminologie et de l'ingénierie des connaissances, suite à l'informatisation de documents et au développement de nouvelles applications terminologiques pour l'informatique. Différentes RTO peuvent fonctionner ensemble pour représenter un domaine. On a ainsi déjà détaillé les ontologies qui constituent l'une des principales RTO. Il existe également des index, des glossaires et des bases de données lexicales, ainsi que des taxonomies et des thésaurus que je vais présenter maintenant.

Une taxonomie est une classification des éléments qui constituent un domaine. [Bourigault et al. 2004] en donnent la définition suivante :

> « Une *classification* est la répartition systématique en classes, en catégories, d'êtres, de choses ou de notions ayant des caractères communs notamment afin d'en faciliter l'étude ; c'est aussi le résultat de cette opération. »

Pour [Amardeilh 2007], le but d'une taxonomie est « de conceptualiser les objets du monde et de les organiser hiérarchiquement les uns par rapport aux autres. ». Il existe des taxonomies de concepts et des taxonomies de termes ; les concepts des ontologies étant souvent organisés en taxonomie. De plus, étant donné que les taxonomies sont hiérarchiques, elles peuvent être modélisées à l'aide d'une représentation arborescente. Ainsi, plus un concept se trouve proche de la racine de l'arbre, plus il est général. À l'inverse, plus il est proche des feuilles, plus il est spécifique. La relation qui est au cœur des taxonomies de concepts est la relation de subsomption, dite aussi de généralisation / spécialisation. Les taxonomies ont été largement utilisées dans le domaine des sciences naturelles afin de classifier des espèces animales ou végétales. Dans une taxonomie (simplifiée) du monde animal, « animal » sera à la racine, il pourra avoir les fils « vertébré » et « invertébré ». Sous

« vertébré », on pourra trouver « mammifère », et ainsi de suite jusqu'à arriver à une espèce particulière comme le « labrador golden retriever ».

Les thésaurus sont des ressources qui étaient à l'origine utilisées par des terminologues et des documentalistes. Reprenons la définition de [Bourigault et al. 2004] : « Un thésaurus est un langage documentaire fondé sur une structuration hiérarchisée. Les termes y sont organisés de manière conceptuelle et reliés entre eux par des relations sémantiques. ». Un thésaurus est donc, pour un domaine donné, « un vocabulaire contrôlé et structuré » [Amardeilh 2007]. Les termes qui prennent place dans un thésaurus sont en général les termes préconisés pour un domaine précis. L'utilisation du thésaurus évite, par exemple, l'utilisation de synonymes qui pourraient prêter à confusion. Ces termes sont liés par des relations qui les organisent, comme les relations « narrower », « broader » ou encore « see-also ». La relation d'hyperonymie y tient souvent une place importante, comme dans les taxonomies mais, contrairement à ces dernières, un thésaurus n'est pas limité à un seul type de relation. Il ne s'agit plus d'une hiérarchie arborescente mais plutôt d'un graphe, d'un réseau entre termes.

Si ces différentes RTO sont regroupées c'est parce qu'elles peuvent fonctionner ensemble. Ainsi, les ontologies qui constituent les RTO les plus évoluées peuvent être associées à des taxonomies et à des thésaurus, aussi bien pour aider à leur construction que pour les phases d'utilisation ultérieures. Ces RTO sont ensuite utilisées dans des systèmes à base de connaissance ; le projet Eiffel en est un exemple.

1.2. Rôle d'une ontologie pour l'annotation sémantique

[Amardeilh 2007] définit l'annotation sémantique comme « une représentation formelle d'un contenu, exprimée à l'aide de concepts, relations et instances décrits dans une ontologie, et reliée à la ressource documentaire source ». Il faut noter l'importance, dans cette définition, des termes « ontologie » et « ressource documentaire ». Par ressource documentaire source, elle entend le document que l'on veut annoter. En effet, une annotation n'a de sens que si elle est liée à la ressource à laquelle elle se rapporte.

L'ontologie, quant à elle, permet de guider l'annotation : elle donne la base de connaissance qu'il faut pouvoir remplir avec les données annotées. Amardeilh en définit ainsi le rôle, dans le cadre de l'annotation sémantique :

> « Dans le processus d'annotation sémantique, les ontologies jouent un rôle primordial puisqu'elles modélisent les concepts, leurs attributs et les relations utilisées pour annoter le contenu des documents. » ([Amardeilh 2007])

Les tâches d'annotation sémantique sont donc directement liées à des ontologies. Par exemple, le projet décrit dans [Tenier et al. 2006b] consiste à annoter, dans des pages Web, les informations concernant les chercheurs et les équipes de recherche. Ils ont mis au point une méthode à base d'apprentissage qui permet d'annoter des éléments du document sous forme d'instances de concepts et de rôles de l'ontologie fournie. Leur système d'annotation dépend entièrement de cette ontologie. Pour d'autres exemples de travaux se basant sur une ontologie, voir également les travaux de Florence Amardeilh et Ines Jilani ([Amardeilh et al. 2005], [Jilani & Amardeilh 2009], [Amardeilh & Francart 2006]), ceux de Thierry Hamon et Adeline Nazarenko ([Nédellec & Nazarenko 2005], [Hamon et al. 2007]), ainsi que les travaux de Siegfried Handschuh ([Handschuh 2005], [Uren et al. 2006]).

Le fait d'utiliser une ontologie pour l'annotation sémantique contraint et assure la cohérence des annotations. C'est donc pour cela que le système d'annotation que j'ai développé pour le

projet Eiffel s'appuie sur une ontologie, comme nous le verrons au point suivant.

2. L'ontologie Eiffel : le pivot du projet

Au regard de ces considérations théoriques et techniques, une ontologie a été développée pour Eiffel. Modélisée par Mondeca, cette ontologie du tourisme répond spécifiquement aux besoins du projet. Elle modélise à la fois le domaine du tourisme et du territoire, et un volet temporel lui a été ajouté pour pouvoir modéliser les informations temporelles propres au domaine touristique.

Je parle ici de cette ontologie comme « pivot du projet » car elle est en effet centrale et plusieurs « briques » du projet s'appuient sur elle. Comme nous le verrons plus en détail dans la suite de ce chapitre, l'ontologie guide les opérations de repérage et d'annotation effectuées par Adetoa. Elle aide en quelque sorte, à répondre aux questions « quoi repérer ? » et « comment annoter ? ». Par ailleurs, la base de connaissance du projet dans laquelle sont stockées toutes les informations repose aussi sur cette ontologie. Ainsi, l'ontologie guide les opérations de mapping qui consistent à intégrer à la base de connaissance, les données annotées par Adetoa. Enfin, le système d'interrogation du projet, qui permet à l'utilisateur final d'effectuer des recherches sur la plateforme, s'appuie également sur l'ontologie dans le sens où celle-ci indique ce qui est interrogeable. C'est dans cette optique qu'elle a été conçue : elle permet de modéliser le domaine tout en tenant compte des usages qui en seront faits.

L'ontologie assure donc la cohérence et l'interopérabilité sémantique des différents composants de la plateforme Eiffel et c'est pour cela qu'elle a été développée avec soin, en tenant compte des besoins de chacun des partenaires.

Plus précisément en ce qui concerne mon travail, l'ontologie guide à la fois le repérage et l'annotation. Elle guide le repérage en ce sens qu'elle indique les informations qui doivent être repérées et l'annotation en ce sens qu'elle indique la structure à donner si possible à ces informations. Je me suis également appuyée sur les données en elles-mêmes (pages Web) pour savoir ce qu'il était intéressant de prendre en compte, mais l'ontologie m'a fourni un certain cadre, qui tout en pouvant être dépassé, m'a permis de me focaliser sur les données les plus importantes. Mon but était en effet de prendre en compte un maximum de données dans les pages Web tout en les rendant exploitables par la suite, ce qui impliquait qu'elles soient en adéquation avec l'ontologie et puissent ainsi être stockées dans la base de connaissance.

Nous verrons par la suite que l'analyse linguistique des données et la mise en place d'un schéma d'annotation ont eu des conséquences sur l'ontologie qui a évolué au cours du projet. C'est pourquoi je présente ici la première version de cette ontologie. La deuxième version est présentée à la fin de ce chapitre.

2.1. Structure générale de l'ontologie d'Eiffel

L'ontologie d'Eiffel, décrite dans [Vatant 2008] permet de modéliser l'offre touristique dans ses aspects territoriaux. Elle décrit, de façon fine et extensible, les objets touristiques : territoires, hébergements, patrimoine, activités, etc., et ce, à l'aide d'éléments objectifs et quantitatifs (localisation, tarifs, horaires ...), d'éléments de classification (catégories, labels, mots-clés ...) et de relations sémantiques (voisinage, activité associée ...).

Cette ontologie a été conçue comme un assemblage de trois modules distincts et

réutilisables : une ontologie de domaine qui comprend la description des objets métiers, une ontologie fonctionnelle (gérant le flux d'indexation et de publication) et des composantes terminologiques correspondant à des thésaurus. Tout cela est formalisé dans un format uniforme.

L'ontologie de domaine « tourisme territoire » est la partie de l'ontologie Eiffel la plus importante pour Adetoa, étant donné que c'est sur celle-ci que s'appuie le schéma d'annotation que j'ai mis au point. Elle est constituée de trois composants : les objets TourinFrance[34], les codes géographiques et les informations temporelles. Le label TourinFrance définit un format de description des données touristiques. Déjà largement utilisé, il a servi de base à l'ontologie pour la description des objets touristiques. L'ontologie des codes géographiques pour l'INSEE[35] qui permet d'identifier les territoires français (hiérarchisation de la région à la commune) a été intégrée à l'ontologie Eiffel pour la localisation spatiale des objets touristiques. Le dernier composant est le plus important dans le cadre de mes travaux puisqu'il concerne la modélisation des données temporelles.

Calquée sur l'ontologie d'Eiffel, la base de connaissance du projet permet de présenter les ressources touristiques dans leur contexte, de suggérer aux utilisateurs des voyages, des itinéraires, des séjours ou des activités. Elle permet au territoire de valoriser des ensembles cohérents de ressources, de créer des offres nouvelles et composites (route de l'huile d'olive, circuit des peintres, itinéraires de l'aventure...) et de piloter le marketing du territoire par d'autres biais que le prix. Plus précisément, pour chaque type de ressource, sont modélisées dans l'ontologie les informations qui peuvent y correspondre (horaires, dates, adresse, etc.).

Les descriptions d'objets touristiques issues d'Adetoa sont contrôlées et enrichies par un moteur de raisonnement. Cela permet d'une part d'éliminer les descriptions incohérentes par rapport à l'ontologie (un rapport d'erreur est alors généré et permet une éventuelle réparation), et d'autre part de compléter les descriptions par inférence à partir d'un jeu de règles modélisant les connaissances implicites du domaine.

Plutôt que de présenter ici, sans les illustrer, les classes de l'ontologie ainsi que leurs propriétés, je les décrirai au fil des paragraphes suivants, en fonction des besoins.

2.2. Représentation du temps dans l'ontologie Eiffel

Comme on l'a vu au début de ce chapitre, les ontologies permettent de modéliser un domaine donné, et, dans le cadre d'Eiffel, celui-ci est le tourisme et le territoire. Les ontologies de domaine ne sont toutefois qu'un type d'ontologie et il en existe d'autres. De nombreux auteurs classifient les ontologies en quatre catégories ([Studer et al. 1998], [Guarino 1998], [Heijst et al. 1997]). Ces quatre catégories, outre les ontologies de domaine, comprennent les ontologies d'application (propres à une application, dans un domaine donné), les ontologies de représentation (qui « organisent les primitives de la théorie logique » [Charlet 2002] sans être propres à un domaine) et les ontologies génériques (qui ont une vision théorique, ou du moins une visée plus large).

Cette classification est encore reprise dans des travaux plus récents ([Laublet 2007], [Amardeilh 2007]) qui soulignent la possibilité d'une cinquième catégorie : les ontologies de tâches (ou de méthode de résolution de problème) qui explicitent les concepts liés au raisonnement.

34 http://www.tourinfrance.net/TourinFrance22/index.htm
35 http://rdf.insee.fr/geo/

Si une ontologie de domaine a été développée pour Eiffel, je me suis aussi intéressée aux ontologies génériques (aussi appelées *Upper-Level ontologies*) dont le but est de modéliser des concepts généraux, propres à différents domaines : « Elles permettent par exemple de formaliser les aspects temporels ou spatiaux des objets du monde réel » [Amardeilh 2007]. L'ontologie d'Eiffel ayant un volet permettant de modéliser des données temporelles, elle aurait pu faire appel à une ontologie générique du temps. Cependant, les informations temporelles dont il est question dans Eiffel ne peuvent pas être considérées comme universelles, car elles sont spécifiques au domaine du tourisme. Les notions génériques du temps, comme la chronologie d'événements, les notions de passé et de futur ne sont pas impliquées ici. Sont en jeu uniquement des connaissances temporelles très spécifiques (et le chapitre 2 l'a déjà montré), comme des horaires d'ouverture, des exceptions, etc. C'est pour cela qu'une ontologie du temps comme OWL-Time[36] n'a pas été adoptée. Celle-ci aurait en effet pu être intégrée à l'ontologie d'Eiffel mais cela aurait posé des difficultés techniques. Tout d'abord, l'ontologie préexistante sur laquelle le projet s'est appuyé permettait de stocker et d'interroger des informations temporelles découpées en années, mois, jours, heures, minutes, secondes mais ne prévoyait pas d'autres découpages temporels comme le découpage hebdomadaire. De plus, l'ontologie OWL-Time a un fort pouvoir expressif mais elle nécessite pour cela la création d'un réseau complexe d'objets intermédiaires :

> « Un objet chrono-localisé est relié à un agrégat[37] temporel qui est une séquence d'entités temporelles qui ont elles-mêmes chacune une description complexe. On a donc trois intermédiaires entre l'objet à renseigner et à interroger, et les données temporelles finales. » ([Vatant 2006])

Ces différents niveaux intermédiaires posent des difficultés, aussi bien au moment de stocker des informations, qu'au moment de les interroger. C'est pourquoi, au sein du projet Eiffel, la décision a été prise d'adopter une approche plus fonctionnelle pour représenter précisément les informations temporelles utiles au projet.

3. Schéma d'annotation

C'est en respectant l'ontologie et en observant les données contenues dans les pages Web que j'ai pu mettre au point le schéma d'annotation utilisé dans Adetoa. Ce schéma étant le cœur des échanges entre Adetoa et les autres « briques » du projet Eiffel, il est le résultat de nombreux échanges avec les autres partenaires et a été validé par eux.

Toutefois, dans la plupart des systèmes d'annotation basés sur une ontologie, c'est l'ontologie qui sert directement de schéma d'annotation. Les raisons suivantes m'ont poussée à proposer un schéma distinct de l'ontologie. Nous le verrons, ce schéma reste proche de l'ontologie, mais il n'en est pas un calque direct. En effet, le but de mon schéma d'annotation est de permettre d'annoter, dans les pages Web, les expressions temporelles telles qu'elles y sont formulées. J'ai donc choisi que ce schéma reflète au mieux la langue et respecte la façon dont sont exprimées les informations. Ainsi, par exemple, sont modélisés dans l'ontologie les jours de la semaine en tant qu'ouverture ou fermeture. Une expression comme *ouvert du lundi au jeudi* n'y est pas directement modélisable. Une transformation est nécessaire pour aboutir à la suite de jours *lundi, mardi, mercredi, jeudi* qui, elle, est directement modélisable. Ayant choisi

36 http://www.w3.org/TR/owl-time/

37 La notion d'agrégat temporel est celle de Pan et Hobbs, voir [Pan & Hobbs 2005] et [Hobbs & Pan 2004].

d'annoter les informations explicitement présentes dans les pages Web, sans transformation, j'ai donc dû m'éloigner de l'ontologie et proposer les balises <JOUR_DEBUT> et <JOUR_FIN> pour annoter ce type d'expressions. Le schéma d'annotation que j'ai proposé permet donc d'annoter les informations à un niveau linguistique. Nous le verrons aux chapitres 5 et 6, des règles de transformations et de mapping permettent ensuite de transformer l'information pour la rendre conforme à l'ontologie et la stocker dans la base de connaissance.

Pour pouvoir développer ce schéma, j'ai dû prendre en compte différents paramètres. Tout d'abord, une analyse des pages Web a permis d'identifier les informations qui pouvaient être intéressantes. Cependant, et comme cela a déjà été mentionné, les données exploitables sont celles qui peuvent, d'une part être repérées, et d'autre part être stockées. Le but est, comme nous le verrons ci-dessous, que le schéma d'annotation permette d'annoter des données le plus finement possible, tout en respectant l'ontologie : inutile, pour le projet, d'annoter des informations qui ne peuvent être instanciées dans la base de connaissance (dont la structure reprend celle de l'ontologie). De plus, les usages sont aussi à prendre en compte : inutile de stocker des données qu'aucun utilisateur n'interrogerait. Mon travail a donc été guidé à la fois par les données textuelles contenues dans les pages Web, les besoins du projet, les usages et la modélisation de l'ontologie, comme le montre le schéma de la figure 38.

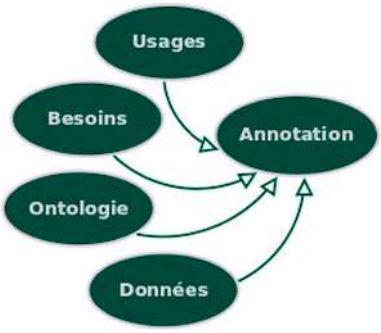


Figure 38 : Éléments influençant l'annotation

De plus, et nous le verrons ci-dessous, ce travail d'élaboration d'un schéma d'annotation a révélé que l'ontologie d'Eiffel, telle qu'elle avait été conçue, n'était pas totalement adaptée. Elle a donc été ajustée de façon à pouvoir mieux prendre en compte les données annotées. Le schéma de la figure 39 illustre la mise au point du schéma d'annotation et le retour qu'il a provoqué sur l'ontologie.

Le schéma d'annotation présenté dans ce chapitre décrit comment annoter les données qu'Adetoa a repérées. Ces données sont ensuite transformées et stockées dans la base de connaissance à l'aide de règles de mapping qui seront exposées au chapitre 6.

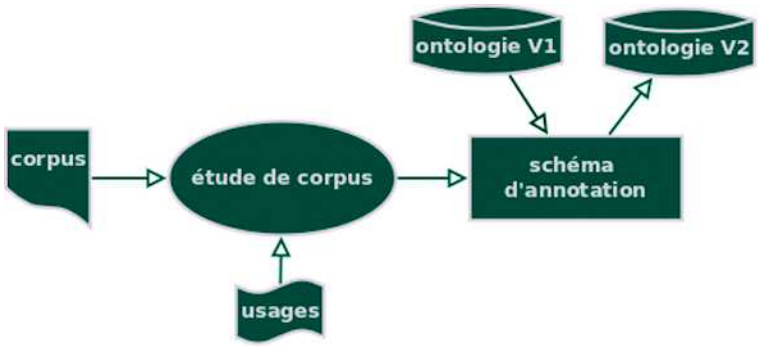


Figure 39 : Élaboration du schéma d'annotation et retour sur l'ontologie

3.1. Choix du format

Le schéma d'annotation que j'ai mis au point aurait pu être développé suivant différents formats, mais nous avons choisi, au sein du projet Eiffel, d'établir une DTD[38] pour le formaliser. Le format XML-Schema[39] a l'avantage d'être déjà en XML et de ne pas nécessiter d'autre langage, mais il est plus complexe que celui des DTD. En effet, il permet de gérer finement des espaces de nom et des types de données complexes. Or nous n'avons pas de tels besoins ; ce que permet une DTD est suffisant pour nos besoins et surtout, les DTD ont l'avantage d'être facilement manipulables et de pouvoir être lues par un humain. La DTD étant un moyen de dialoguer avec les autres partenaires du projet, il était important qu'elle soit lisible. Voir [Bex et al. 2004] pour une comparaison plus poussée des DTD et XML-Schema qui montre que, le plus souvent, les XML-Schema qui sont réellement utilisés pourraient être remplacés par des DTD.

Par ailleurs, le format DTD autorise une grande liberté, ce qui nous a permis de construire le schéma d'annotation pas à pas, voire parfois de laisser des « zones floues » sans être bloqués par le format.

Enfin, ce choix résulte aussi du fait que tous les outils et parseurs XML, anciens ou récents, peuvent vérifier la conformité d'un fichier XML par rapport à une DTD, tandis que seuls les outils récents le font avec XML-Schema.

3.2. Mise au point du schéma

Comme on l'a déjà vu, les informations qu'Adetoa doit annoter se divisent en trois catégories : les expressions temporelles, les expressions spatiales et les objets touristiques.

Sur un plan linguistique, les expressions temporelles sont les plus complexes. Cette complexité se situe à deux niveaux. Tout d'abord, comme le reflète l'étude menée au chapitre 2, les expressions sont très variées : pour une même information, de nombreuses formulations sont possibles, ce qui complique le repérage des informations. De plus, ces informations peuvent être de différents types : horaires d'ouverture ou de fermeture, dates, périodes, exceptions, etc. C'est à ce niveau-là qu'il est important d'avoir un schéma d'annotation bien adapté aux besoins du projet et permettant de tenir compte des données,

38 Document Type Definition – http://www.w3schools.com/DTD/dtd_intro.asp – prévu dans la norme ISO 8879 décrivant le langage SGML.

39 http://www.w3.org/standards/xml/schema

tout en respectant la structure de l'ontologie.

En ce qui concerne les expressions de localisation spatiale, j'ai choisi, en accord avec les autres membres du projet Eiffel, d'annoter uniquement les adresses et non les autres informations de localisation, comme par exemple *l'hôtel est situé près du lac* ou *à dix kilomètres de Bordeaux seulement*. La définition d' « adresse » n'est pas triviale et plusieurs variantes ont été considérées.

Enfin, au sujet de l'annotation des objets, il s'agit en réalité de repérer et d'annoter le type de la ressource touristique dont il est question. L'annotation semble plus basique mais reste néanmoins délicate. En effet, le repérage effectué consiste davantage à identifier des unités lexicales simples que des expressions complexes. Le risque est alors d'annoter de nombreux objets qui ne correspondent pas à l'objet principal de la page. Par exemple, sur la page d'un hôtel, le but est de repérer qu'il s'agit d'un hôtel mais aussi d'annoter son adresse et son éventuelle période de fermeture annuelle. Le type « hôtel » sera probablement marqué, mais si la page mentionne aussi les restaurants voisins et les différentes activités sportives accessibles facilement depuis l'hôtel, ces éléments seront également annotés comme des objets. La difficulté réside alors dans l'identification de l'objet principal.

Au niveau du repérage et de l'annotation, seul le type d'objet est considéré. Nous verrons au chapitre 6 que le nom de la ressource est plus difficile à repérer et qu'un module a été mis au point pour effectuer le nommage des objets touristiques.

De plus, il faut noter que l'annotation des expressions temporelles et celle des objets touristiques sont effectuées simultanément. Au moment où les expressions temporelles sont annotées, il n'est donc pas encore possible de savoir à quel type d'objet elles correspondent. Ces tâches sont donc indépendantes et l'ensemble de balises prévues pour annoter les informations temporelles doit pouvoir convenir à tout type d'objet.

3.2.1. Annotation des expressions temporelles

a. Balises <UT> et <DESCRIPTION>[40]

Ces deux balises sont présentes dans toutes les annotations d'expressions temporelles. La balise <UT> permet d'encadrer l'annotation dans son ensemble. Elle englobe l'expression annotée, ainsi que toutes les balises permettant de décrire les informations temporelles.

La balise <DESCRIPTION> est la dernière de l'expression annotée (et précède immédiatement la balise <UT> fermante). Elle contient l'ensemble du texte annoté, tel qu'il apparaît sur la page : sans balise et sans modification. Elle permet donc de garder l'expression textuelle intacte afin de pouvoir la proposer à un internaute qui aurait éventuellement besoin d'informations complémentaires.

L'exemple suivant illustre l'utilisation de ces balises.

40 Pour plus de lisibilité, les balises seront notées en lettres capitales dans cette partie.

(74) **Horaires d'ouverture : lundi, mercredi 13h30 – 17h30**

```
<UT>
   <periode_ouverture>Horaires d'ouverture :
      <jour> lundi</jour>,
      <jour> mercredi</jour>
      <heure_debut> 13h30</heure_debut> -
      <heure_fin>17h30</heure_fin>
   </periode_ouverture>
   <description>"Horaires d'ouverture : lundi, mercredi 13h30 – 17h30"
   </description>
</UT>
```

b. Balises <PERIODE_OUVERTURE> et <PERIODE_FERMETURE>

Ces deux balises permettent de typer l'information qu'elles encadrent en « ouverture » ou en « fermeture ». Une <UT> peut contenir chacune de ces balises zéro, une ou plusieurs fois. Les données contenues dans ces balises sont encore annotées avec d'autres balises qui les caractérisent (date, horaires, jours).

L'annotation par ces balises est déclenchée par des marqueurs linguistiques du type *ouvert, fermé, ouverture, etc.* L'avantage d'un typage « à haut niveau » en ouverture ou en fermeture est le suivant : les données concrètes en termes de jours et heures sont formulées de la même façon, qu'il s'agisse d'une ouverture ou d'une fermeture, et peuvent donc être repérées et annotées indépendamment du type.

En ce qui concerne l'exemple précédent (74), la balise <PERIODE_OUVERTURE> est déclenchée par la présence de *horaires d'ouverture*, mais l'annotation du contenu de cette balise aurait été identique s'il s'était agi d'une fermeture :

(75) **Fermé : lundi, mercredi 13h30 – 17h30**

```
<UT>
   <periode_fermeture>Fermé :
      <jour> lundi</jour>,
      <jour> mercredi</jour>
      <heure_debut> 13h30</heure_debut> -
      <heure_fin>17h30</heure_fin>
   </periode_fermeture>
   <description>"Fermé : lundi, mercredi 13h30 – 17h30"
   </description>
</UT>
```

c. Balises d'heures, de dates et de jours

Ces balises sont nécessairement contenues dans une balise de période d'ouverture ou de fermeture. Ce sont les balises qui encadrent concrètement les données à stocker dans la base de connaissance.

Les balises <HEURE_DEBUT> et <HEURE_FIN> permettent d'annoter les horaires.

(76) **Horaire d'ouverture :de 12h00 à 14h00**[41]

```
<periode_ouverture> Horaires d'ouverture : de
   <heure_debut> 12h00</heure_debut> à
   <heure_fin> 14h00</heure_fin>
</periode_ouverture>
```

Ces balises permettent aussi d'annoter les parties de la journée (*matin, midi, soir,* etc.). Dans ce cas, il arrive souvent que seule la balise <HEURE_DEBUT> soit renseignée.

Les balises <DATE_DEBUT> et <DATE_FIN> permettent d'annoter les périodes de temps. Elles peuvent encadrer des données de granularités différentes : dates calendaires complètes, dates incomplètes (par exemple jour/mois sans l'année), données plus floues comme *avril* ou encore *début avril*.

(77) **Fermeture annuelle du 15 Février au 20 Mars**

```
Fermeture annuelle
<periode_fermeture> du
   <date_debut> 15 Février</date_debut> au
   <date_fin> 20 Mars</date_fin>
</periode_fermeture>
```

La balise <DATE> permet d'annoter les dates seules, comme dans l'exemple suivant :

(78) **Inauguration du Musée le 7 Juillet 2006**

```
Inauguration du Musée
<periode_ouverture>
   <date> le 7 Juillet 2006 </date>
</periode_ouverture>
```

Il est à noter que les balises de date et heure fonctionnent généralement « en couple ». Si seule la balise « début » est renseignée, la balise « fin » prend la même valeur et est donc omise.

La balise <JOUR> permet de marquer les jours de la semaine. Elle est utilisée quand l'expression mentionne un ou plusieurs jours de la semaine, en tant que collection de jours comme par exemple :

(79) **Fermé les mardi, jeudi, samedi et dimanche**

```
Fermée
<periode_fermeture> les
   <jour> mardi</jour>,
   <jour> jeudi</jour>,
   <jour> samedi</jour> et
   <jour> dimanche</jour>
</periode_fermeture>
```

Les balises <JOUR_DEBUT> et <JOUR_FIN> permettent d'annoter les successions de jours comme *du lundi au vendredi,* dans lesquelles tous les jours ne sont pas énumérés.

41 Par souci de lisibilité, les exemples d'annotations présentés dans cette partie ne sont pas complets (pas de balise <UT>, <DESCRIPTION>, ...) mais ne comprennent que les parties dont il est question.

(80) **Ouvert du lundi que vendredi**

```
<periode_ouverture> Ouvert du
   <jour_debut> lundi</jour_debut> au
   <jour_fin> vendredi</jour_fin>
</periode_ouverture>
```

d. Balises <EXCEPTION> et <INCERTITUDE>

Ces deux balises ont un statut à part : elles peuvent être contenues directement dans la balise <UT> ou bien dans une période d'ouverture ou de fermeture.

La balise <INCERTITUDE> permet d'indiquer que l'information n'est pas fiable. Dans un système tel que celui d'Eiffel, il est important de favoriser la précision, en dépit du rappel. En d'autres termes, mieux vaut un manque d'information qu'une information erronée. Néanmoins, le fait d'annoter uniquement les informations totalement fiables risquait de provoquer une importante perte d'informations. C'est pour cela que la balise <INCERTITUDE> a été définie : elle indique que l'information est probable mais pas certaine et qu'il est préférable de la vérifier. Cela permet au module répondant à l'utilisateur de lui fournir un résultat tout en le mettant en garde. Cette balise est principalement utilisée dans les cas où l'information contenue dans la page Web est floue ou incomplète, comme l'illustrent les exemples suivants :

(81) **Ouverture du camping début février**

(82) **Visites de juin à septembre**

Dans ces deux cas, il est difficile de transformer l'expression en dates précises. En ce qui concerne l'exemple (81), il n'est pas certain que le camping ouvre le 1er février : il peut ouvrir quelques jours plus tôt (par exemple en fonction de la météo) ou quelques jours plus tard, notamment s'il devait par exemple ouvrir pour le week-end et que le 1er est un mardi. Il en est de même pour les mois de juin et septembre de l'exemple (82). Dans ces deux configurations, l'expression est conservée telle quelle et la balise <INCERTITUDE> indique qu'il faut la proposer à l'utilisateur dans son ensemble. Celui-ci saura estimer les chances que cela soit ouvert et vérifiera au besoin. Cette balise n'encadre pas d'information. Sa seule présence indique que l'information globale est incertaine.

La balise <EXCEPTION> permet d'annoter les exceptions formulées dans les horaires, comme dans les exemples suivants :

(83) **Ouvert tous les jours à l'exception du mardi.**

(84) **Ouvert du 1er mars au 30 juin sauf le 14 avril.**

Son utilisation est double. Premièrement, sa seule présence permet d'indiquer que les informations qu'elle englobe consistent en une exception et qu'il faut donc l'indiquer à l'internaute qui interroge la base. Deuxièmement, si l'expression le permet, cette balise peut, à son tour, englober les balises de périodes d'ouverture ou de fermeture, comme dans l'exemple suivant :

(85) Ouvert de juin à septembre, sauf lundi et mardi.

```
<UT>
   <periode_ouverture> Ouvert de
      <date_debut> juin</date_debut> à
      <date_fin> septembre</date_fin>
      <incertitude/>
   </periode_ouverture>,
   <exception> sauf
      <periode_fermeture>
         <jour> lundi</jour> et
         <jour> mardi</jour>
      </periode_fermeture>
   </exception>
   <description>"Ouvert de juin à septembre, sauf lundi et
   mardi"</description>
</UT>
```

Le fonctionnement de la balise <EXCEPTION> découle de la complexité linguistique des expressions qu'elle encadre. En effet, et comme cela a été développé au chapitre 2, les informations exprimées sous forme d'exception sont si variées, aussi bien au niveau même du type d'information qu'au niveau de la formulation, qu'il n'est pas toujours possible de les qualifier finement. Ainsi, la priorité a été donnée au repérage de l'exception dans son ensemble et seulement certaines exceptions peuvent ensuite être balisées plus finement avec des périodes d'ouverture et de fermeture.

e. Transformation dès l'annotation

Comme cela est apparu à travers les exemples précédents, la tâche d'annotation accomplie par Adetoa consiste en un simple ajout de balises dans le texte. Toutefois, dans deux cas, elle modifie également ce texte. Il s'agit des cas des expressions très fréquentes *tous les jours* et *toute l'année*. Ces expressions auraient pu, comme les autres, être annotées par l'ajout de balises simples, mais la transformation directe a semblé plus appropriée et a permis, par la suite, de faciliter le mapping et le stockage dans la base de connaissance. Les annotations générées pour ces expressions sont les suivantes :

(86) Ouvert tous les jours

```
<UT>
   <periode_Ouverture> ouvert tous les jours
      <jour> lundi</jour>
      <jour> mardi </jour>
      <jour> mercredi </jour>
      <jour> jeudi</jour>
      <jour> vendredi </jour>
      <jour> samedi </jour>
      <jour> dimanche </jour>
   </periode_Ouverture>
   <description> "ouvert tous les jours"</description>
</UT>
```

(87) Ouvert toute l'année

```
<UT>
   <periode_Ouverture> ouvert toute l'année
      <date_debut> 01/01 </date_debut>
      <date_fin> 31/12 </date_fin>
   </periode_Ouverture>
   <description> "ouvert toute l'année" </description>
</UT>
```

C'est notamment dans ces cas que la balise <DESCRIPTION> prend tout son sens : elle permet de conserver le texte tel qu'il apparaît sur la page Web, sans qu'il ait subi de modification.

f. Exemples généraux

Les deux exemples suivants permettent de refléter plus concrètement l'application de mon schéma d'annotation.

(88) Ouvert le lundi de 16h30 à 18h et le mardi de 17h30 à 18h30

```
<UT>
   <periode_ouverture>
      <jour> lundi </jour> de
      <heure_debut> 16h30 </heure_debut> à
      <heure_fin> 18h </heure_fin>
   </periode_ouverture> et le
   <periode_ouverture>
      <jour> mardi </jour> de
      <heure_debut> 17h30 </heure_debut> à
      <heure_fin> 18h30 </heure_fin>
   </periode_ouverture>
</UT>
```

Cet exemple montre l'importance des balises <PERIODE_OUVERTURE> et <PERIODE_FERMETURE> : elles permettent d'associer les horaires aux jours correspondants et de grouper les informations qui vont ensemble.

(89) Accueil de juin à septembre, sauf lundi et mardi. Sur rendez-vous le reste du temps

```
<UT>
   <periode_Ouverture>Accueil de
      <date_debut> juin</date_debut> à
      <date_fin> septembre <date_fin>
      <incertitude/>
   </periode_Ouverture> ,
   <Exception> sauf
      <periode_Fermeture>
         <jour> lundi </jour> et
         <jour> mardi </jour>
      </periode_Fermeture>
   </Exception>.
   <Exception> Sur rendez-vous le reste du temps </Exception>
   <description> "Accueil de juin à septembre, sauf lundi et   mardi.
Sur  rendez-vous le reste du temps" </description>
</UT>
```

Cet exemple illustre les deux utilisations possibles de la balise <EXCEPTION>.

3.2.2. Annotation des expressions spatiales

Le schéma d'annotation concernant les expressions de localisation spatiale est beaucoup plus simple. En effet, je ne me suis intéressée qu'aux adresses et aux communes seules. La modélisation était alors assez directe et ne posait pas de questions.

Quatre balises ont été choisies. La balise <LOCALISATION> encadre l'ensemble de l'expression repérée (de la même façon que la balise <UT> encadre les expressions temporelles). La balise <COMMUNE> encadre, comme son nom l'indique, le nom de la commune, qu'il s'agisse d'une ville ou d'un village. La balise <CP> encadre le code postal. Enfin, la balise <ADRESSE> annote le nom de la voie et, éventuellement, le numéro. Seule la balise <COMMUNE> est obligatoire pour annoter une localisation. Trois cas principaux ont été distingués : adresse complète (exemple 90), adresse partielle (exemple 91) et commune seule (exemple 92).

(90) 4, rue Bocquillot 89200 AVALLON

```
<localisation>
   <adresse> 4, rue Bocquillot</adresse>
   <cp> 89200</cp>
   <commune> AVALLON </commune>
</localisation>
```

(91) Camping de l'étang du goulot, 58140 Lormes

```
Camping de l'étang du goulot,
<localisation>
   <cp> 58140 </cp>
   <commune> Lormes </commune>
</localisation>
```

(92) LA VIGNE, le relais gastronomique : Saint-Pierre-le-Moutier – Nièvre

```
LA VIGNE, le relais gastronomique :
<localisation>
   <commune> Saint-Pierre-le-Moutier </commune>
</localisation> – Nièvre
```

Comme cela a déjà été mentionné, nous avons décidé, dans le cadre du projet Eiffel, de ne pas traiter les autres informations de localisation, comme celles des exemples suivants :

(93) L'hôtel est situé près d'un lac.

(94) L'hôtel est à 10 minutes du centre-ville de Nevers.

3.2.3. Annotation des objets

En ce qui concerne les types d'objets touristiques, le schéma d'annotation est directement calqué sur ce qui est modélisé dans l'ontologie. Ainsi, on retrouve les trois classes principales de la partie qui nous intéresse dans l'ontologie : activité, infrastructure et personne. La classe infrastructure se subdivise en hébergement, restaurant, service_local (pour les mairies et offices de tourisme), etc.

Étant donné que la partie du schéma relative aux objets est calquée directement sur la modélisation de l'ontologie et ne dépend pas d'une étude linguistique approfondie, celle-ci ne sera pas détaillée. Les expressions qui sont repérées et annotées sont très courtes et directement liées à un type d'objet. Il n'y a que très peu de problèmes linguistiques qui en résultent.

Un cas particulier a dû être pris en compte. Il s'agit du fait que l'*hôtel de police* n'est pas une ressource touristique et que l'*hôtel de ville* n'est pas un hébergement. Un repérage simple du terme hôtel ne peut donc pas aboutir de façon directe à une annotation en tant qu'hébergement : le contexte est indispensable.

La DTD correspondant à mon schéma d'annotation apparaît dans la figure suivante. Les balises correspondant aux types d'objets y sont représentées.

```
<!ENTITY % periode "(date | date_debut | date_fin | jour | jour_debut
| jour_fin | heure_debut | heure_fin | exception | incertitude )* ">

<!ELEMENT ensemble ( UT | localisation | objet | nom ) * >

<!ELEMENT UT ( incertitude | exception | periode_ouverture |
               periode_fermeture | description) *>

    <!ELEMENT incertitude ( #PCDATA ) >
    <!ELEMENT exception ( #PCDATA | periode_ouverture |
                         periode_fermeture | incertitude ) >
    <!ELEMENT description ( #PCDATA ) >

    <!ELEMENT periode_ouverture ( %periode;) >
    <!ELEMENT periode_fermeture ( %periode;) >

        <!ELEMENT heure_debut ( #PCDATA ) >
        <!ELEMENT heure_fin ( #PCDATA ) >
        <!ELEMENT date_debut ( #PCDATA ) >
        <!ELEMENT date_fin ( #PCDATA ) >
        <!ELEMENT date ( #PCDATA ) >
        <!ELEMENT jour ( #PCDATA ) >
        <!ELEMENT jour_debut ( #PCDATA ) >
        <!ELEMENT jour_fin ( #PCDATA ) >

<!ELEMENT localisation ( adresse?, cp?, commune ) >
    <!ELEMENT adresse ( #PCDATA ) >
    <!ELEMENT cp ( #PCDATA ) >
    <!ELEMENT commune ( #PCDATA ) >

<!ELEMENT objet ( activite | infrastructure | personne ) >
    <!ELEMENT activite ( evenement_manifestation | gastronomie |
                         prestation_assemblee | promenade |
                         sport_loisir | visite_decouverte ) >
        <!ELEMENT evenement_manifestation ( #PCDATA ) >
        <!ELEMENT gastronomie ( #PCDATA ) >
        <!ELEMENT prestation_assemblee ( #PCDATA ) >
        <!ELEMENT promenade ( #PCDATA ) >
        <!ELEMENT sport_loisir ( #PCDATA ) >
        <!ELEMENT visite_decouverte ( #PCDATA ) >

    <!ELEMENT infrastructure ( hebergement | restaurant | equipement |
                               service_local) >
        <!ELEMENT hebergement ( #PCDATA ) >
        <!ELEMENT restaurant ( #PCDATA ) >
        <!ELEMENT equipement ( #PCDATA ) >
        <!ELEMENT service_local ( #PCDATA ) >

    <!ELEMENT personne ( artisan_producteur ) >
        <!ELEMENT artisan_producteur ( #PCDATA ) >
```

Figure 40 : DTD correspondant au schéma d'annotation

3.3. Retour sur l'ontologie

3.3.1. Ontologie – volet temporel V1

Pour développer le volet temporel de l'ontologie, Mondeca s'est basé sur son ontologie du tourisme et a ajouté des propriétés temporelles aux classes des objets touristiques. Dans la première version de l'ontologie Eiffel comprenant des données temporelles, les propriétés de la classe Infrastructure sont les plus complètes. C'est donc sur celles-ci que je me suis appuyée pour développer mon schéma d'annotation. Elles sont représentées dans la figure suivante :

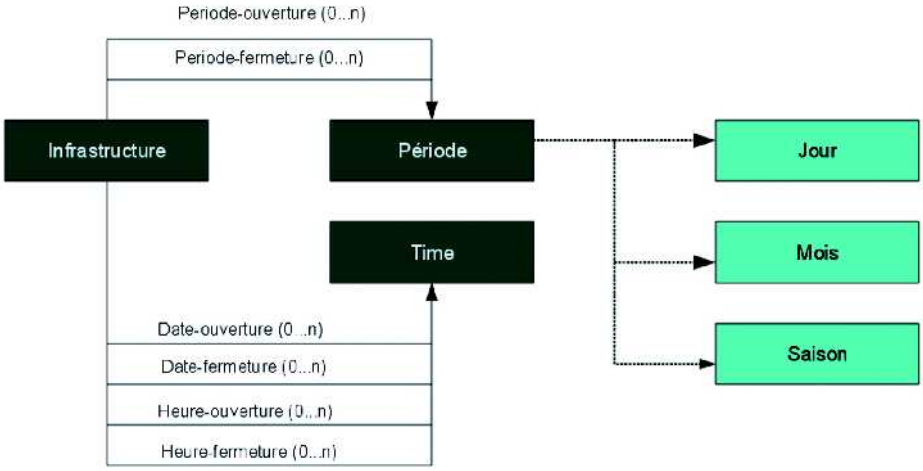


Figure 41 : Propriétés d'Infrastructure dans l'ontologie V1

Dans ce modèle, « Période » est à prendre au sens de périodicité. Les valeurs de période d'ouverture et de fermeture capturent les périodes itératives : jour (*lundi*), mois (*juillet*), saison (*été*) ou période floue (*vacances de Noël*) qui se répètent chaque semaine ou chaque année.

Le schéma d'annotation que j'ai mis au point, ainsi que les données qu'il permet d'annoter dans les pages Web, ont révélé plusieurs failles dans cette modélisation. La plus importante est la suivante : ce modèle ne contraint pas la cardinalité des attributs. Or, pour être utilisable sans ambigüité, il faut qu'il y ait une seule période et une seule plage horaire pour toute la période. Fixer la cardinalité maximale de tous les attributs à 1 aurait permis de lever les ambigüités. Cela ne reflète néanmoins pas les données présentes dans les pages Web de façon optimale. Ce modèle ne permet notamment pas de représenter l'expression suivante :

> (95) **Accueil du 15 avril au 20 mai, le jeudi de 12h à 17h et du 21 mai au 30 septembre, le lundi de 10h30 à 12h30.**

En effet, une fois les dates, jours et horaires instanciés, impossible de faire le lien entre les différentes informations : savoir que l'ouverture du jeudi correspond à la première période, etc.

La deuxième limitation de ce modèle est que la sous-classe Jour permet d'instancier uniquement les jours de la semaine. Or, dans les données, les jours sont très souvent découpés par tranches : *matin, après-midi, soir*, etc. et il semble difficile de les caractériser par des horaires précis (le matin va-t-il jusqu'à 12h, 12h30, 13h ? Le midi correspond-il toujours à

12h ?). Il serait donc dommage de perdre ces données. La version suivante de l'ontologie permet de les modéliser.

3.3.2. Ontologie – volet temporel V2

En conséquence de ces remarques, l'ontologie a été modifiée pour permettre de mieux modéliser les données rencontrées dans les pages Web. La deuxième version du volet temporel de l'ontologie Eiffel peut être représentée comme suit :

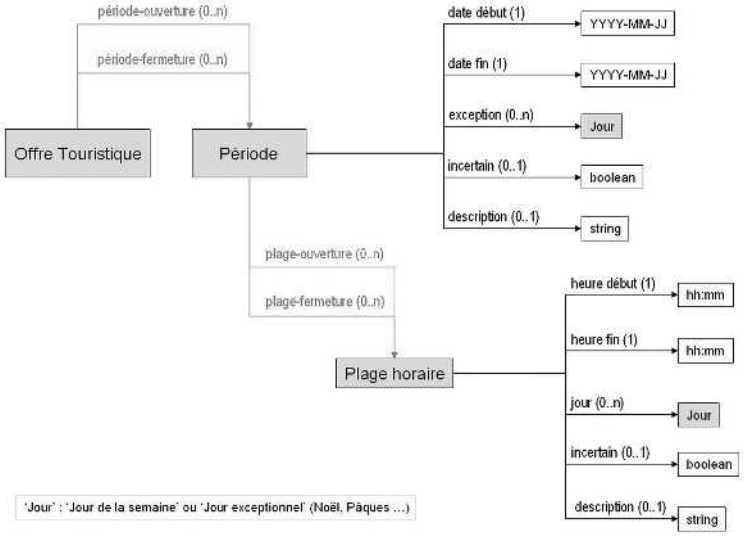


Figure 42 : Propriétés temporelles des offres touristiques dans l'ontologie V2

Ce modèle est plus complet et permet une meilleure modélisation des données réelles. Plus axé sur les données calendaires, il permet de les modéliser selon deux granularités : celle du jour, avec des dates, et celle des heures, avec des horaires.

Plusieurs remarques sont nécessaires pour mieux comprendre ce schéma et les améliorations qu'il apporte par rapport à la première version.

Tout d'abord, le terme « période » n'est pas utilisé dans le même sens que précédemment : en effet il n'a plus un sens de périodicité ou d'itérativité, et désigne désormais un intervalle de temps. Cependant, cet intervalle peut comporter des exceptions, ce qui le rompt et l'empêche d'être considéré comme un intervalle, au sens mathématique du terme. Ici, il semble donc plus approprié de parler « d'agrégat temporel ».

Ce modèle permet de modéliser plusieurs périodes (ou agrégats) ayant des propriétés différentes au niveau des jours ou des horaires. Les données horaires sont liées à une période donnée délimitée par une date de début et une date de fin. Ainsi, le modèle permet de représenter parfaitement l'exemple suivant (déjà donné en 95) :

> **(96) Accueil du 15 avril au 20 mai, le jeudi de 12h à 17h et du 21 mai au 30 septembre, le lundi de 10h30 à 12h30.**

En ce qui concerne les jours, ce modèle est moins contraignant que le précédent : les parties de journée (*lundi matin*) peuvent y être instanciées en tant que Jour.

De plus, la notion d'incertitude a également été intégrée à ce modèle et peut intervenir au niveau de la période ou à celui de la plage horaire.

Mon schéma d'annotation permet d'annoter toutes les expressions qui sont modélisables dans cette ontologie, ainsi que certaines expressions qui ne peuvent pas y être modélisées. Il s'agit des données purement itératives, comme celles énoncées dans l'exemple suivant :

(97) Ouvert le lundi.

Ce type d'information n'est pas modélisable, car dans cette version de l'ontologie, les jours peuvent apparaître dans une plage horaire mais celle-ci doit nécessairement être liée à une période d'ouverture ou de fermeture, décrite par une date de début et une date de fin.

Cela s'explique par le fait que ces données ne correspondent pas aux objectifs premiers du projet Eiffel qui s'est focalisé sur les données calendaires, dans l'optique où ses clients renseigneraient un catalogue annuel avec leurs informations d'ouverture, et que ce catalogue serait mis à jour chaque année.

Néanmoins, l'étude des données contenues dans les pages Web a montré que ces expressions étaient très fréquentes, et c'est pour cela que j'ai décidé de ne pas les exclure de mon système de repérage et donc de les prévoir dans mon schéma d'annotation. Sans avoir été déployés dans Eiffel, des modèles ont été envisagés pour modéliser ces données. Ils sont présentés ci-dessous.

3.3.3. Propositions de modèles

Bernard Vatant (Mondeca), qui a modélisé l'ontologie d'Eiffel, a proposé un nouveau schéma pour pallier ce problème de données itératives non calendaires. Un extrait en est représenté dans la figure suivante ; les informations d'horaires sont également modélisables mais ne sont pas représentées, par souci de clarté.

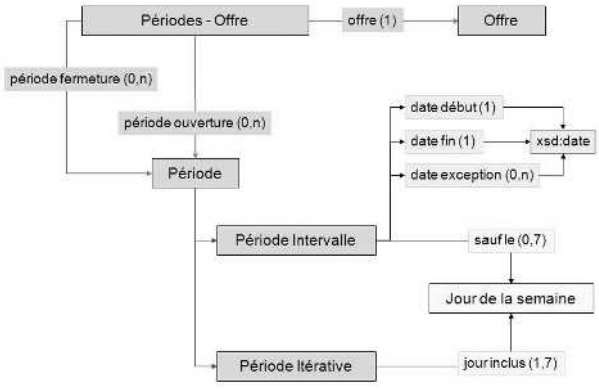


Figure 43 : Proposition de modèle (Bernard Vatant)

Ce modèle permet de modéliser les informations itératives du type *tous les lundi*. Une nouvelle fois, sous « période », se cache un « agrégat temporel » qui peut donc avoir une date de début, une date de fin et des exceptions, et ce, sous forme de date ou de jour itératif. Le modèle permet ainsi de représenter l'expression qui suit.

(98) Ouvert du 15 juin au 30 juillet sauf les mardi et le 14 juillet.

Toutefois, ce modèle reste « bancal ». En effet, il distingue les périodes itératives des intervalles mais une période itérative peut aussi être un intervalle, comme l'illustre l'exemple suivant :

(99) Ouvert du lundi au vendredi.

Ce type d'expression, très fréquent dans mon corpus, n'est toujours pas prévu dans ce modèle et doit être transformé en *ouvert lundi, mardi, mercredi, jeudi, vendredi* pour pouvoir être pris en compte.

Parallèlement à la fin de ma thèse, et pour des travaux sur la modélisation plus générale de données calendaires, Charles Teissèdre (Mondeca) [Teissèdre et al. 2010a] a mis au point un modèle qui pourrait également permettre d'instancier les données annotées par mon schéma d'annotation.

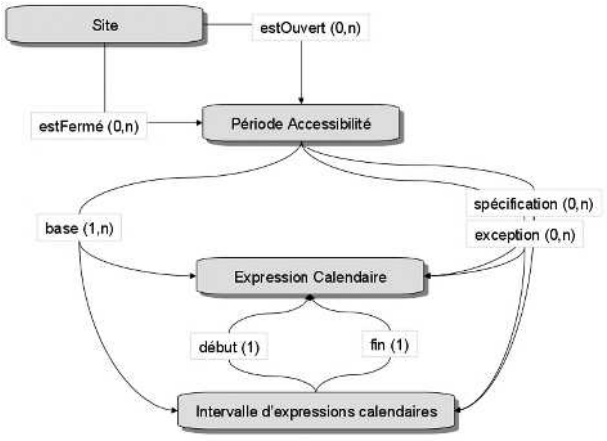


Figure 44 : Proposition de modèle (Charles Teissèdre)

Ce modèle a l'avantage de « coller » au maximum à la façon dont sont exprimées ces données dans la langue. Il modélise les données selon la granularité qui est exprimée, sans la modifier, et permet aussi de représenter les intervalles de données itératives, comme celui de l'exemple (99). Par ailleurs, il introduit les notions de « base » et de « spécification ». Ainsi, une période d'accessibilité contient au moins une « base » (qui pointe vers une expression calendaire ou un intervalle d'expressions calendaires) sur laquelle peuvent s'ajouter des spécifications et des exceptions. Il permet par exemple de traiter l'expression suivante :

(100) Ouvert tous les jours, sauf le lundi, de 9h à 19h.

Pour cette expression, la base est *tous les jours* (soit du lundi au dimanche), l'exception est *lundi*, et la spécification correspond à *de 9h à 19h*.

Conclusion

Après un rapide état de l'art sur les ontologies et la présentation de l'ontologie du projet Eiffel, j'ai présenté, dans ce chapitre, le schéma d'annotation que j'ai développé et qui est suivi par Adetoa.

Ce chapitre a permis de souligner le rôle central que joue l'ontologie dans le cadre d'un projet d'annotation. Néanmoins, j'ai aussi montré l'importance de mettre au point un schéma d'annotation qui permette d'annoter le texte le plus fidèlement possible sur un plan linguistique. C'est l'élaboration de ce schéma d'annotation qui a fait apparaître les lacunes de l'ontologie qui ne permettait pas de modéliser certaines données temporelles touristiques exprimables dans la langue. Ainsi, la mise en place d'un cycle s'est avérée nécessaire : ontologie – schéma d'annotation – amélioration de l'ontologie. Ce cycle est efficace en ce sens qu'il permet de rester proche de la langue en modélisant ce qui est effectivement exprimable, tout en généralisant pour conceptualiser les informations.

Le schéma d'annotation a donc permis de perfectionner l'ontologie, mais il permet malgré tout encore d'annoter des données non modélisables dans l'ontologie. Ces données ne sont effectivement pas très utiles dans le cadre du projet Eiffel. Le schéma d'annotation les prend en compte car elles reflètent des structures existantes dans la langue pour décrire des horaires ou des périodes d'ouverture et de fermeture. Dans d'autres cadres, moins spécifiques que le projet Eiffel, ces données pourraient être exploitables. Par exemple, le modèle proposé dans [Teissèdre et al. 2010a] permet de modéliser les périodes d'accessibilité ; toutes les expressions annotées par Adetoa et prévues dans mon schéma d'annotation sont des périodes d'accessibilité (ou de non-accessibilité). Leur module, capable de convertir les expressions en expressions calendaires, pourrait exploiter mes annotations.

Les chapitres suivants s'intéressent à l'implémentation d'Adetoa et montreront, au chapitre 5, comment le schéma d'annotation a été appliqué, avant d'expliquer, au chapitre 6, comment les données annotées dans Eiffel seront exploitées par la suite.

Troisième Partie

Implémentation et Évaluation

Chapitre 5. Implémentation d'Adetoa

Introduction

Comme cela a déjà été mentionné au fil des chapitres précédents, Adetoa a été conçu pour repérer et annoter, dans des pages Web touristiques, des informations temporelles pratiques, des objets touristiques et des adresses. La question s'est donc posée du choix de l'outil à utiliser pour effectuer de tels traitements.

Comme en témoigne l'existence d'un numéro de la revue TAL : *Plate-formes pour le traitement automatique des langues* [Enjalbert et al. 2008], les outils effectuant des traitements linguistiques se sont multipliés ces dernières années. Chaque outil possède des caractéristiques propres et vise un objectif particulier. Néanmoins, ces plateformes constituent des « *environnements de développement* dédiés au TAL » permettant à des linguistes non informaticiens de développer des outils qui répondent à leurs besoins. Pour ne présenter brièvement que deux exemples, Linguastream est une plateforme permettant de créer visuellement des chaînes de traitements linguistiques pour le TAL. Chaque étape de la chaîne effectue un traitement et annote le document. Les annotations se cumulent alors au fil de la chaîne [Widlöcher & Bilhaut 2008]. Si l'utilisateur peut créer ses propres modules, il peut aussi utiliser les nombreux modules qui sont déjà fournis avec la plateforme. GATE[42] [Cunningham et al. 2002] est un projet plus vaste. Il s'agit d'une infrastructure permettant de développer et de déployer des outils de traitement automatique du langage. Comme pour LinguaStream, de nombreux outils sont fournis avec GATE. L'un des principaux buts de cet outil est de permettre la réutilisation des différents modules.

Des outils plus spécifiques existent également. Il s'agit des systèmes basés sur Intex[43] [Silberztein 1993], [Silberztein 1994], comme Unitex[44] [Paumier 2003], et Outilex[45] [Blanc et al. 2006] qui sont prévus pour faire du traitement de corpus. Si les plateformes comme LinguaStream ou GATE sont très riches et modulables, les systèmes basés sur Intex ont l'avantage de recourir à des ressources linguistiques très importantes. De plus, ces outils ont été développés dans un cadre théorique bien défini, à la suite des travaux de Maurice Gross

42 *General Architecture for Text Engineering.*
43 http://intex.univ-fcomte.fr/
44 http://igm.univ-mlv.fr/~unitex/
45 http://igm.univ-mlv.fr/~mconstan/outilex/

[Gross 1989].

C'est sur Unitex que s'est arrêté mon choix pour effectuer le repérage et l'annotation des différents types de données. Tout en ayant une fonction spécifique, Unitex peut en effet être intégré dans différents projets. Par exemple, GlossaNet[46] [Fairon et al. 2008] qui est basé sur Unitex et sur l'outil Corporator [Fairon 2006], permet de constituer des corpus en ligne et d'y effectuer des concordances. Des détails sur les raisons de ce choix et le fonctionnement d'Unitex sont donnés dans la suite de ce chapitre.

Les contraintes imposées par le projet Eiffel déterminent ce qu'Adetoa prend en entrée et doit fournir en sortie, pour être intégré à la chaîne de traitement du projet qui est développée à Mondeca (voir le chapitre suivant pour plus de détails sur cette chaîne). Le schéma suivant représente l'intégration d'Adetoa à la chaîne.



Figure 45 : Situation d'Adetoa

À partir de ces contraintes, Unitex étant conçu pour traiter un document texte et non une collection de fichiers, j'ai dû développer un environnement se chargeant de gérer les entrées / sorties et de faire appel à Unitex. Au vu du type de données – fichier XML en entrée et document annoté en sortie – j'ai décidé d'utiliser le langage JAVA pour développer cet environnement. En effet, ce langage permet de manipuler facilement les technologies XML (voir figure 46). Par ailleurs, Unitex étant également développé en JAVA, cela rend les appels plus faciles. De plus, un programme JAVA a l'avantage d'être multi-plateforme et de pouvoir être intégré à la chaîne de traitement du projet.

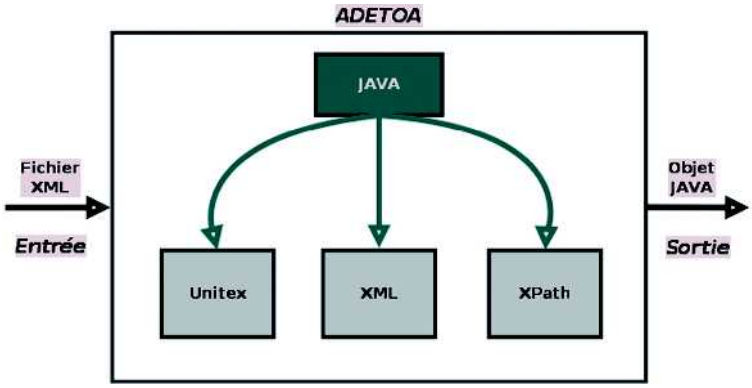


Figure 46 : Adetoa – utilisation des technologies JAVA-XML

Tel que je l'ai développé, Adetoa est donc un programme JAVA prenant en entrée un ou plusieurs documents XML. Il se charge de manipuler les fichiers (gestion du format, zone de texte à analyser) et de faire les appels à Unitex pour annoter les données. Une fois les

46 http://glossa.fltr.ucl.ac.be/

données annotées, il doit effectuer des transformations et construire l'objet à fournir en sortie.

Ce chapitre va donc présenter plus en détail l'outil Unitex et la structure générale du programme JAVA ; les ressources linguistiques que j'ai développées avec Unitex et le module de transformations seront également présentés.

1. Unitex

Unitex étant au cœur d'Adetoa, je vais présenter ici les principales caractéristiques qui m'ont poussée à le choisir pour développer mes ressources linguistiques et effectuer le repérage et l'annotation. La première partie présente les caractéristiques générales, tandis que la seconde indique plus précisément comment il fonctionne et comment s'en servir.

1.1. Principales caractéristiques utiles pour Eiffel et Adetoa

Dans le cadre du projet Eiffel, nous avons décidé d'utiliser Unitex pour effectuer les tâches linguistiques de repérage et d'annotation. En effet, cet outil a l'avantage de permettre un développement direct des ressources linguistiques, sans que l'élaboration préalable d'outils ou de modules informatiques soit nécessaire. Cet outil, qui est décrit plus en détail au paragraphe 1.2, permet de traiter des corpus en y effectuant des recherches de motifs. Les différents critères suivants ont orienté notre choix sur cet outil.

1.1.1. Éditeur visuel – expressivité

Unitex permet de rechercher des motifs dans des textes. L'un de ses principaux avantages est qu'il propose un éditeur visuel permettant de représenter ces motifs sous forme de graphes plutôt que sous la forme d'expressions régulières. Il est ainsi facile à manipuler et suffisamment expressif pour les besoins du projet.

1.1.2. Ressources linguistiques

Les ressources linguistiques fournies avec Unitex sont d'une grande richesse. Il s'agit de dictionnaires très complets, construits par des linguistes du LADL[47] [Courtois & Silberztein 1990]. Ainsi, sont fournis, pour le français, un dictionnaire des mots simples et un dictionnaire des mots composés. Ces deux dictionnaires comprennent également les formes fléchies et permettent donc d'effectuer des recherches sur toutes les formes d'un lemme. Par ailleurs, les catégories associées aux entrées du dictionnaire permettent d'effectuer des recherches sur des catégories de mots : nom, verbe, adjectif, etc., mais aussi verbe à la première personne du singulier, nom singulier, etc.

D'autres ressources plus spécifiques sont également fournies : un dictionnaire des noms propres et un dictionnaire des noms de professions.

Par ailleurs, il est facile de créer de nouvelles ressources linguistiques ; nous verrons ci-dessous le dictionnaire des toponymes que j'ai créé pour les besoins du projet.

1.1.3. Logiciel Libre

Unitex a la particularité d'être totalement gratuit et open source, comme on peut le lire dans

47 Laboratoire d'Automatique Documentaire et Linguistique.

le manuel :

> « Unitex est un logiciel libre. Cela signifie que les sources des programmes sont distribuées avec le logiciel, et que chacun peut les modifier et les redistribuer. Le code des programmes d'Unitex est sous licence LGPL [GNU. Lesser General Public License. http://www.gnu.org/licenses/lgpl.html. 1.1, 10.10.4)], à l'exception de la bibliothèque de manipulation d'expressions régulières TRE de Ville Laurikari [Ville L AURIKARI. TRE home page. http://laurikari.net/tre/. 1.1, 4.7], qui est sous licence GPL [GNU. General Public License. http://www.gnu.org/licenses/gpl.html. 1.1, 10.10.4]. La licence LGPL est plus permissive que la licence GPL, car elle permet d'utiliser du code LGPL dans des logiciels non libres. Du point de vue de l'utilisateur, il n'y a pas de différence, car dans les deux cas, le logiciel peut être librement utilisé et distribué.
>
> Toutes les données linguistiques distribuées avec Unitex sont soumises à la licence LGPLLR. » ([Paumier 2006])

Ainsi, ce logiciel peut très facilement être utilisé dans le cadre d'une application industrielle.

1.1.4. Multi-plateforme

Unitex a l'avantage d'avoir un environnement d'exécution Java et d'être ainsi multi-plateforme :

> « Unitex est composé d'une interface graphique écrite en Java et de programmes externes écrits en C/C++. Ce mélange de langages de programmation permet d'avoir une application rapide et portable sous différents systèmes d'exploitation. » ([Paumier 2006])

Dans le cadre d'un projet mêlant des partenaires issus de différentes structures et ne travaillant pas tous sur le même système, la portabilité de l'outil est primordiale. Unitex fonctionne aussi bien sous MS-Windows que sous Linux ou MacOS.

1.2. Fonctionnement et principes généraux

Unitex permet de traiter des corpus en utilisant des dictionnaires, et ce, au niveau du lexique, de la syntaxe ou de la morphologie. Ces dictionnaires permettent de repérer et d'annoter des structures correspondant à des expressions régulières, représentées par des graphes à états finis. Cet outil permet aussi de créer des dictionnaires et de les appliquer aux textes. Toute l'architecture logicielle est prévue pour que l'utilisateur n'ait besoin de créer que ses ressources linguistiques : graphes ou transducteurs de repérage et d'annotation, et dictionnaires.

1.2.1. Grammaires

Unitex permet de développer des grammaires de sorte à repérer, dans un texte, des motifs complexes. Il est basé sur le formalisme des automates à états finis et permet de manipuler des grammaires algébriques étendues. Celles-ci sont des grammaires algébriques dont la partie droite n'est pas une simple suite de symboles mais une expression régulière. Voici un exemple de règle de réécriture pouvant prendre place dans une grammaire algébrique étendue définissant un groupe nominal pouvant être composé d'un nom suivi ou non d'un ou plusieurs adjectifs.

```
GN -> N (A)*
```

Unitex intègre un outil de représentation de ces grammaires à l'aide de graphes facilement

manipulables par l'utilisateur. La figure 47 représente un graphe exprimant cette règle.

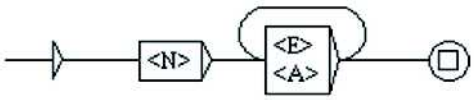


Figure 47 : Graphe « GN »

De plus, dans Unitex, la notion de transduction a été ajoutée à ces grammaires. Ainsi, une grammaire permet non seulement la reconnaissance d'un langage particulier, mais également la production de sorties. On appelle sortie toute modification du texte : en plus d'effectuer un repérage, Unitex peut ajouter des données au texte. Ce sont ces données que l'on appelle des sorties. Le transducteur présenté dans la figure 48 représente la même grammaire que précédemment mais contient en plus des annotations à intégrer : les balises <Nom>, </Nom> et <Adj>, </Adj>.

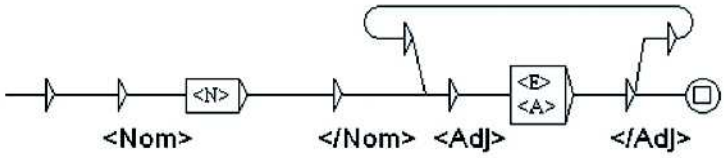


Figure 48 : Transducteur « GN »

1.2.2. Exemples de graphes

Différents types de graphes sont pris en compte et manipulables dans Unitex. Ceux que j'ai développés sont des graphes syntaxiques qui permettent de décrire des motifs syntaxiques et de les rechercher dans des textes. Ils ont un très grand pouvoir expressif étant donné qu'ils permettent de faire référence aux dictionnaires. Par exemple, on peut indiquer que l'on veut rechercher un substantif suivi d'un adjectif – comme dans les grammaires présentées ci-dessus – et toutes les occurrences de ce type seront repérées dans le texte : *chat gentil, chien méchant, voiture rouge, etc.* On peut également rechercher un mot donné avec toutes ses flexions : toutes ses formes pour un verbe, le singulier et le pluriel pour un nom, etc. Les variantes minuscules / majuscules sont autorisées. Les graphes syntaxiques peuvent faire appel à des sous-graphes et gèrent également les sorties et les sorties à variables. Les sous-graphes et les variables sont expliqués à l'aide d'exemples dans la suite de cette partie.

Le graphe représenté dans la figure 49 est en réalité un transducteur puisqu'il comprend des sorties. Il s'agit du principal transducteur permettant de repérer et d'annoter les objets touristiques. La flèche la plus à gauche correspond à l'état initial, tandis que le cercle qui entoure le carré correspond à l'état final. Cette grammaire reconnaît toutes les expressions décrites par des chemins reliant l'état initial à l'état final.

Ce transducteur ne contient aucun marqueur linguistique. Il ne contient que des sous-graphes. Ceux-ci sont représentés par les boîtes grisées. Les sous-graphes contiennent les données linguistiques. Par exemple, le sous-graphe « resto » contient toutes les façons d'indiquer qu'il s'agit un restaurant (*resto*, *restaurant*, *auberge*, etc.). Le but de ce transducteur est

de permettre d'insérer les balises d'annotation. Celles-ci sont représentées par le texte en gras sous des flèches vides. On peut voir sur le schéma que les balises ouvrantes et fermantes sont indiquées. Lors de l'application de ce transducteur au texte, les balises sont insérées aux endroits voulus (balise ouvrante avant le motif repéré, balise fermante après). Ce transducteur constitue un bon exemple car l'annotation y est totalement dissociée du repérage en lui-même qui est effectué dans les sous-graphes.

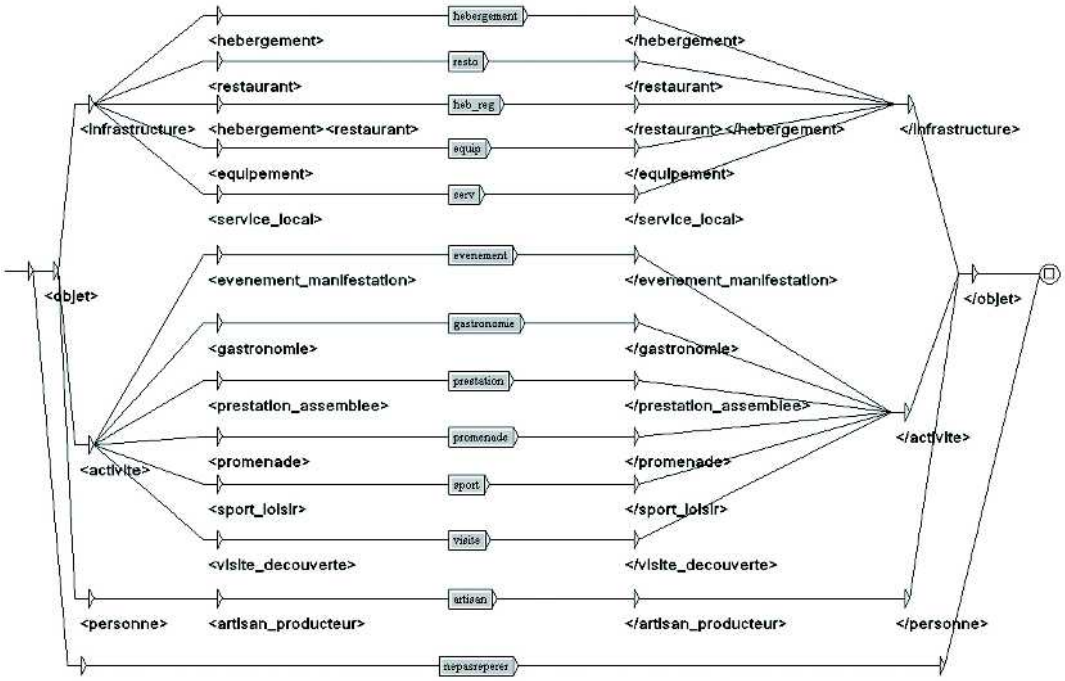


Figure 49 : Transducteur « Objets »

Comme dit plus haut, ce transducteur reconnaît toutes les expressions décrites par des chemins reliant l'état initial à l'état final. Le dernier chemin, au bas de la figure, avec l'appel au sous-graphe « nepasreperer » implique donc aussi un repérage mais pas d'annotation. On peut en effet voir sur ce transducteur que ce chemin ne contient aucune boîte avec annotation. Le but est donc de contrôler certaines expressions sans les annoter. Le sous-graphe « nepasreperer » permet de reconnaître l'expression *hôtel de police* que l'on ne veut pas marquer, contrairement aux autres suites du type <hôtel> + <Prep> + <Nom> qui correspondent à *hôtel de ville* ou *hôtel du lac*, par exemple.

Le graphe représenté dans la figure 50 n'a pas de sortie et contient des données linguistiques. Il permet en effet de reconnaître tous les jours de la semaine, les abréviations des jours et *jour férié*. Les chevrons qui encadrent chaque nom de jour indiquent que ces mots et leurs variantes flexionnelles sont acceptés. Les retours à la ligne dans la boîte correspondent à un « ou » : n'importe quel jour de la semaine peut être repéré. Il n'est pas nécessaire de créer plusieurs boîtes.

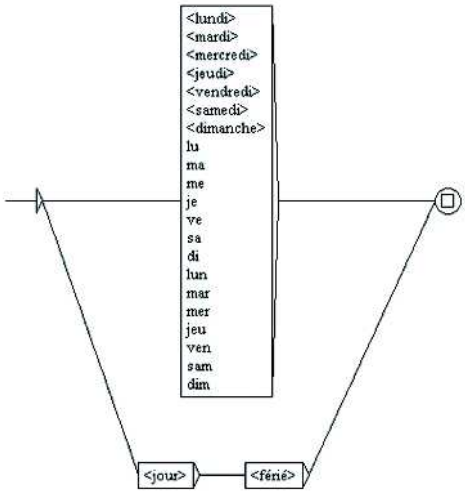


Figure 50 : Graphe « jours »

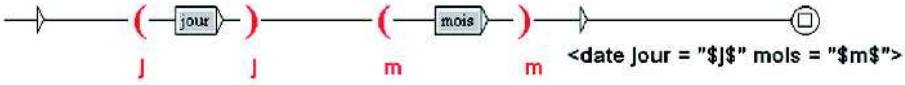


Figure 51 : Transducteur avec variables

Le transducteur de la figure 51 illustre la notion de variable et son utilisation. Les variables permettent de sélectionner une partie du texte reconnu par une grammaire. Dans cet exemple, les variables « j » et « m », illustrées par des parenthèses, permettent de stocker respectivement le jour de la semaine et le mois qui sont repérés et de les replacer dans la sortie. Ainsi, si le transducteur est appliqué en mode « merge », le texte repéré apparaîtra deux fois dans la sortie : une fois dans le texte et une fois dans la balise <date> insérée, à la place des séquences « j » et « m ».

2. Infrastructure JAVA

Adetoa est le nom de l'ensemble du programme JAVA que j'ai développé pour effectuer le repérage, l'annotation et le liage des informations pratiques dans les pages Web. Avant de présenter, au paragraphe suivant, les ressources linguistiques développées, je vais décrire ici la structure générale du programme, ainsi que son fonctionnement. Ne sont présentées ici que les données réellement utiles à la compréhension du fonctionnement d'Adetoa. Sa structure technique, les classes et les méthodes qui le composent figurent en annexe (annexes A et B).

Comme cela est prévu pour l'intégration d'Adetoa à la chaîne plus générale du projet Eiffel, il prend en entrée un ou plusieurs fichiers XML. Ces fichiers sont le résultat de la transformation de pages Web HTML en XML. Leur structure a déjà été présentée au chapitre 3. Le schéma de la figure 52 présente toutes les étapes réalisées par Adetoa.

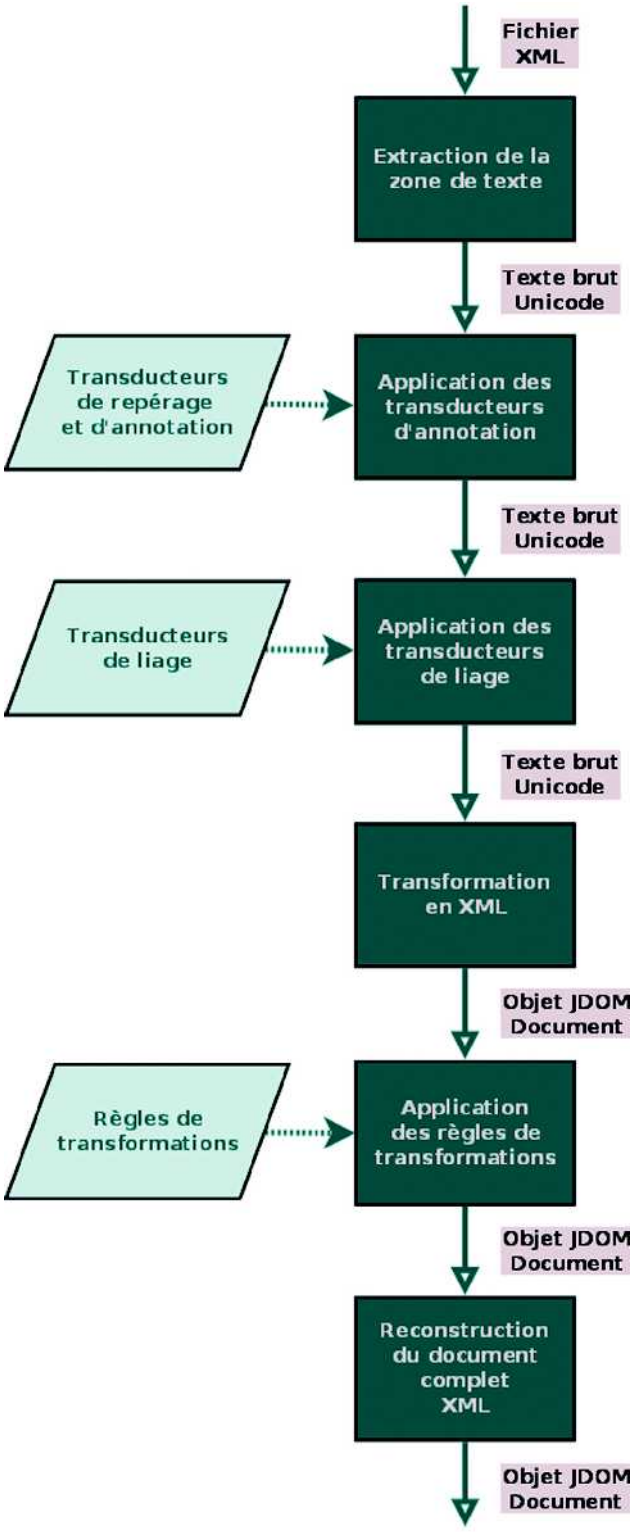


Figure 52 : Structure d'Adetoa

2.1. Extraction des données textuelles

La première étape effectuée par Adetoa est l'extraction, dans le document initial, de la zone de texte sur laquelle va porter l'analyse. Il s'agit de l'ensemble du contenu textuel de la page, après suppression des balises. Ce texte est stocké dans un fichier temporaire qui pourra être analysé par Unitex.

Cette étape s'inscrit dans la continuité du choix que j'ai fait d'analyser uniquement les données textuelles, sans tenir compte de la structuration de la page. L'ordre du texte correspond à celui du fichier XML fourni. Les images sont ignorées.

2.2. Deux appels aux programmes Unitex

L'application des transducteurs Unitex est l'étape suivante et constitue le cœur linguistique de l'implémentation. Un premier appel à Unitex permet d'appliquer les transducteurs de repérage et d'annotation pour les trois types d'information concernés (informations temporelles, adresses, objets touristiques). Ces transducteurs se chargent à la fois du repérage et de l'annotation des données. En sortie, le fichier texte est donc balisé. Les transducteurs de repérage et d'annotation, en tant que ressources développées, sont présentés plus en détail et d'un point de vue plus linguistique au paragraphe 3.

Le deuxième appel à Unitex effectue le liage des données annotées à l'étape précédente. Les transducteurs qui sont appliqués se basent sur les balises déjà insérées pour regrouper les différentes informations. L'algorithme consiste à regrouper ces informations selon l'ordre dans lequel elles apparaissent. Le transducteur principal apparaît dans la figure 53.

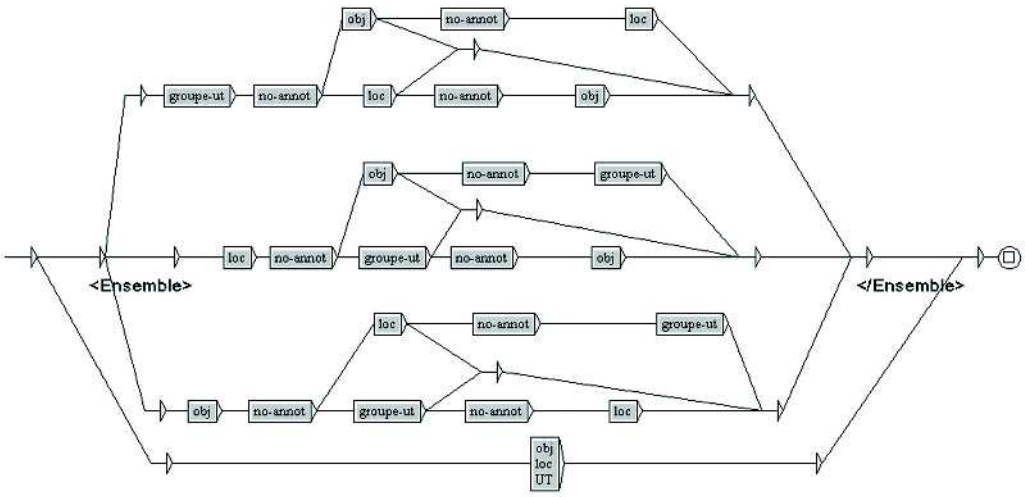


Figure 53 : Transducteur principal de liage

Les ensembles doivent comporter au minimum deux annotations de types différents mais peuvent en comporter trois. Ce nombre peut augmenter si plusieurs annotations d'expressions temporelles se suivent car elles sont alors regroupées (ce qui est représenté par « groupe-ut » dans le transducteur). Peuvent ainsi constituer un ensemble : un objet, une information de localisation suivie ou non d'une ou plusieurs expressions temporelles. L'ordre

des informations n'est pas important, mais, mises à part les informations temporelles successives, il ne peut pas y avoir, dans un ensemble, deux informations du même type. Dans le transducteur, le sous-graphe « no-annot » sert à reconnaître une chaîne de caractères qui ne contient pas d'annotation. Après l'application du transducteur de liage, de nouvelles annotations sont donc insérées dans le fichier texte.

Pour appliquer des transducteurs, Unitex fait appel à de nombreux sous-programmes. Adetoa gère ces appels pour pré-traiter le texte, appliquer les dictionnaires, effectuer le repérage et l'annotation, créer le fichier résultats, etc. Ces programmes et la façon dont ils sont utilisés par Adetoa sont présentés en annexe (annexe C).

2.3. Module de transformations

Le module de transformations permet, une fois le fichier annoté reconverti au format XML, d'appliquer des règles de transformations aux données annotées. Pour ce faire, il se base sur des chemins XPath[48] et peut ainsi explorer la structure XML et effectuer les modifications demandées par les règles. Le chemin XPath permet de localiser l'endroit où la modification doit avoir lieu. Ces modifications ne concernent que les données déjà annotées. Des annotations, et donc des zones de texte, peuvent parfois être supprimées, tandis que d'autres peuvent être créées. Dans tous les cas, le résultat est du code XML valide. Les règles de transformations en elles-mêmes sont présentées en détail au paragraphe 3.3.

2.4. Format des données

Afin d'effectuer ces différentes étapes, Adetoa doit manipuler plusieurs formats de fichiers et de données. Tout d'abord, le fichier donné en entrée est un fichier XML encodé en UTF-8[49]. Adetoa doit, une fois le contenu textuel extrait, le convertir en UTF-16LE, seul encodage accepté par Unitex. Les données restent encodées ainsi pour les deux passages Unitex et sont stockées dans un fichier texte. Pour l'application des règles de transformations, l'encodage repasse en UTF-8 et les données sont stockées dans une structure XML, JDOM Document, conservée jusqu'à la fin du traitement et qui constitue le format de sortie. Le choix de ce format pour l'encodage final est lié à son caractère portable. Il est plus standard et mieux supporté par la plupart des logiciels.

Ces indications très techniques sur le format et l'encodage des données permettent de montrer pourquoi il était nécessaire de créer un environnement en JAVA qui automatise tous les traitements.

3. Développement des ressources

Les ressources que j'ai développées pour Adetoa sont de trois types. Il s'agit des transducteurs Unitex, d'un dictionnaire Unitex et de règles de transformations développées en JAVA. Les transducteurs Unitex constituent le cœur de l'implémentation et sont le résultat de l'étude linguistique (chapitre 2) et de l'établissement du schéma d'annotation (chapitre 4).

Unitex permettant de développer des transducteurs et des dictionnaires, j'ai choisi de m'appuyer essentiellement sur des transducteurs. En effet, l'éditeur d'Unitex permet de visualiser la modélisation au fil du développement des transducteurs, ce qui n'est pas le cas

48 XML Path Language (*langage de chemins XML*) – http://www.w3.org/TR/xpath20/

49 UTF-8 et UTF-16LE sont des formats de codage des caractères, respectivement sur 8 et 16 bits.

pour les dictionnaires qui nécessitent l'édition d'un fichier texte externe et sa compilation par Unitex avant de pouvoir être utilisés. Une modélisation « par types » avec des dictionnaires regroupant les termes propres à chaque type d'information aurait toutefois été possible. Néanmoins, cela aurait nécessité la création de dictionnaires contenant peu d'entrées. De plus, en cas de modification d'un dictionnaire, les conséquences peuvent parfois être difficiles à contrôler. J'ai donc développé de nombreux transducteurs (voir ci-dessous) et, afin d'éviter la duplication d'informations, j'ai privilégié l'utilisation de sous-graphes, ce qui a permis une certaine factorisation.

La création d'un dictionnaire a tout de même été nécessaire pour les noms de villes. En effet, leur nombre élevé justifie tout à fait la création d'un dictionnaire consacré, leur visualisation n'étant, de toute manière, pas aisée. Par ailleurs, si les transducteurs concernant les expressions temporelles et les objets n'ont pas nécessité la création de dictionnaires dédiés, ils font tout de même appel aux dictionnaires DELAF[50] fournis avec Unitex.

Enfin, les transducteurs permettent d'annoter le « texte », et non de manipuler des « informations ». En revanche, une fois le texte annoté en XML, on peut considérer que les « données » peuvent être modifiées et/ou enrichies. Il s'agit précisément du rôle des règles de transformations qui sont présentées à la fin de ce chapitre.

3.1. Transducteurs

Les transducteurs que j'ai développés pour Adetoa sont de deux types : les transducteurs de repérage et d'annotation et les transducteurs de liage. Les seconds correspondent à l'application d'un algorithme et ne contiennent pas de données linguistiques. Ils sont présentés plus haut avec l'implémentation JAVA.

Les transducteurs de repérage et d'annotation constituent en revanche une ressource linguistique. Ils se chargent d'identifier et de baliser les expressions temporelles, les expressions de localisation et les objets touristiques. Étant donnée la richesse des expressions temporelles dans la langue (voir chapitre 2), les transducteurs temporels sont les plus complexes. Ils seront donc présentés en priorité. Tous les transducteurs sont fournis en annexe D. Je ne cherche donc pas ici à les présenter de manière exhaustive mais plutôt à mettre en avant leur richesse et les problématiques qui y sont liées.

3.1.1. Combinatoire

Les expressions que j'ai cherché à modéliser dans les transducteurs ne suivent pas un modèle prédéfini. Elles constituent des énoncés en langage naturel et, comme le permet la langue, chaque information peut être formulée de différentes manières. Les limites de la combinatoire des expressions sont celles de la langue, et celles-ci sont même parfois dépassées dans le cadre, souvent informel, des pages Web, ce qui a pour conséquence de générer des énoncés incomplets ou ambigus. Le nombre de formes possibles pour une information donnée est incalculable. [Culioli 1973] soulève les difficultés posées par le traitement de la paraphrase dans des grammaires formelles :

> « Mais, si l'on prend au sérieux l'assimilation d'une grammaire à un dispositif automatique, comportant une entrée et une sortie, la notion de paraphrase pose un

50 Format de dictionnaire permettant de contenir des lemmes et leurs formes fléchies ainsi que des informations sémantiques et d'inflexions. Voir [Courtois & Silberztein 1990] pour plus de détails sur ces dictionnaires électroniques.

> problème intéressant. Si l'on a autant de représentations métalinguistiques que de phrases équivalentes, comment repère-t-on (et comment note-t-on) la propriété commune aux énoncés d'une famille paraphrastique ? » ([Culioli 1973])

Par ailleurs, les transducteurs permettent de modéliser uniquement des énoncés prévus. Cela permet d'accroître la précision puisque tout repérage aura été décidé et modélisé. Toutefois, la difficulté réside dans le compromis entre la combinatoire des expressions et l'exhaustivité de leur modélisation.

a. Informations temporelles

Parmi les différentes informations considérées par Adetoa, les informations temporelles sont celles pour lesquelles la combinatoire est la plus grande. En effet, ces informations peuvent être de différents types. Chacun de ces types peut alors être exprimé de différentes manières et peut se combiner avec d'autres types. Prenons, par exemple, l'expression suivante :

(101) **Ouvert du lundi au vendredi.**

L'information exprimée dans cet énoncé peut prendre différentes formes, comme :

(102) **Ouvert tous les jours sauf samedi et dimanche.**

(103) **Ouvert le lundi, le mardi, le mercredi, le jeudi et le vendredi.**

(104) **Fermé le samedi et le dimanche.**

Ces exemples ne sont qu'un échantillon des formes possibles. Si l'on y ajoute, par exemple, les abréviations pour les noms de jours (*lun.* ou *lu.*, *mar*, etc.), le nombre de possibilités augmente encore.

De plus, il suffit d'ajouter une information supplémentaire à cet énoncé pour que, de nouveau, la combinatoire augmente :

(105) **Ouvert du lundi au vendredi, de 9h à 19h.**

L'information d'heure peut elle aussi prendre différentes formes, sans compter qu'elle peut se situer avant ou après l'expression initiale :

(106) **... à partir de 9h et jusqu'à 19h.**

(107) **... toute la journée.**

C'est le principe de la combinatoire : si l'on associe deux informations pouvant chacune prendre différentes formes, le nombre total d'expressions possibles est multiplié.

C'est donc un maximum de possibilités d'expressions qu'il a fallu modéliser, le plus efficacement possible, dans les transducteurs. Les appels aux sous-graphes ont été pour cela très utiles. Ils ont effet permis de factoriser les informations. Si un chemin prévoit, par exemple, la suite « jour - heure », il est plus facile de prévoir également l'enchaînement « heure - jour » si « heure » et « jour » sont contenus dans des sous-graphes. Le nombre de transducteurs créés pour modéliser les informations temporelles (voir annexe D) illustre bien cette combinatoire.

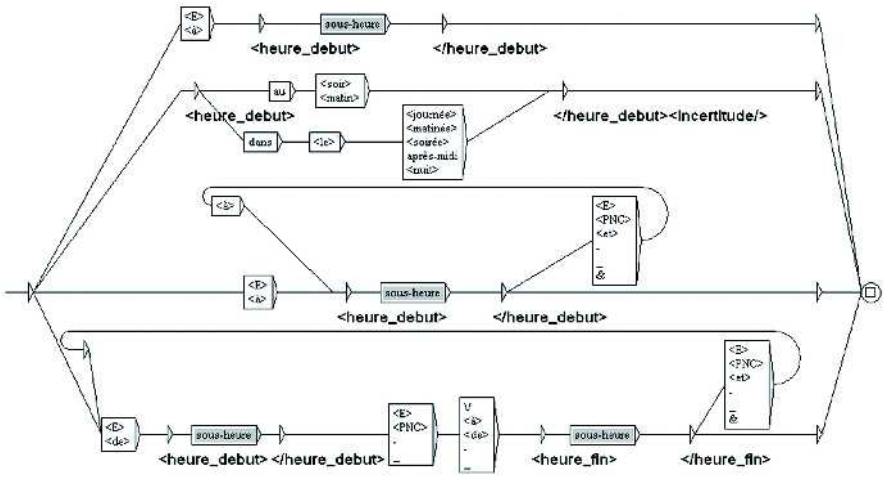


Figure 54 : Transducteur « heure »

Ce transducteur « heure », qui s'intègre à plusieurs endroits dans le graphe « accueil » (éventuellement par l'intermédiaire d'autres sous-graphes), illustre des appels à des sous graphes ainsi que des boucles. Il prévoit quatre principales manières de formuler les heures.

b. Informations de localisation et objets

La question de la combinatoire concerne principalement les informations temporelles car celles-ci sont très riches et difficiles à prédire (heures, dates, périodes, intervalles, jours, etc.) et peuvent chacune prendre plusieurs formes et se combiner. La question est moindre pour les expressions de localisation et les objets.

En effet, les expressions de localisation que j'ai choisi de modéliser pour Eiffel sont en réalité des adresses du type adresses postales (ou adresses postales partielles) pour lesquelles la combinatoire est moins importante. Il n'existe que quelques patrons spécifiques pour exprimer les adresses postales, comme en témoignent les deux transducteurs suivants qui permettent de les repérer. Le transducteur de la figure 56 est le sous-graphe « adresse » appelé dans le transducteur de la figure 55.

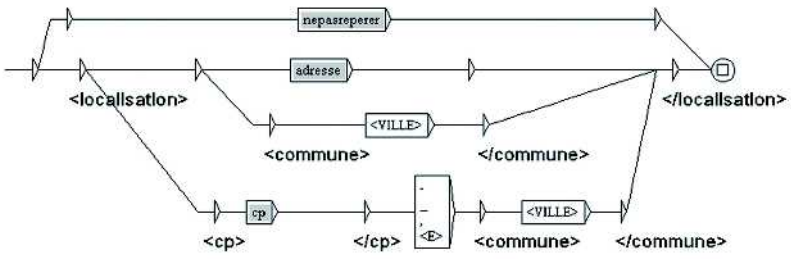


Figure 55 : Transducteur « localisation »

Le transducteur localisation permet de décrire les trois patrons principaux pour le repérage

des adresses : une adresse complète (voir sous-graphe « adresse »), une ville seule, une adresse partielle « code postal - ville ».

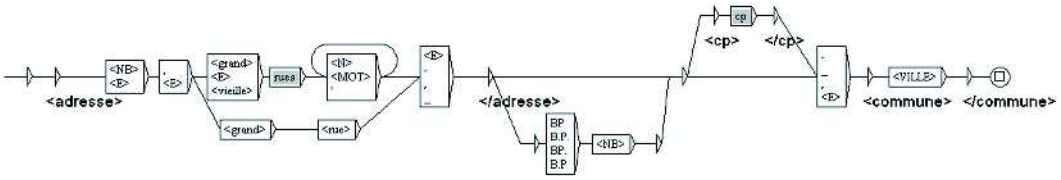


Figure 56 : Transducteur « adresse »

Le sous-graphe adresse donne les différentes possibilités pour les adresses postales complètes : numéro, nom de rue (avec les variantes *rue, avenue, boulevard*, etc., dans le sous-graphe « rues »), code postal, ville. Différentes possibilités existent pour formuler des adresses, mais ces dernières sont plus prédictibles et peuvent donc être formalisées en peu de transducteurs. Par ailleurs, selon l'approche que j'ai adoptée, la présence d'un nom de ville est nécessaire pour qu'une information de localisation soit repérée. Cela réduit donc la combinatoire.

En ce qui concerne la tâche de repérage des objets touristiques, la combinatoire est encore plus faible car très peu d'expressions longues sont à prendre en compte. Il s'agit le plus souvent d'un repérage de mots simples et non pas d'expressions. La complexité dans les transducteurs d'objets réside plutôt au niveau de l'annotation et est présentée au paragraphe suivant.

3.1.2. Typage – annotation

Les transducteurs effectuent le repérage des informations, mais ils doivent aussi les annoter en insérant des balises aux endroits appropriés. C'est là que la richesse de la combinatoire rend la tâche complexe pour les informations temporelles. De plus, il est impératif que les balises insérées constituent du XML valide[51] et bien formé[52].

a. Typage : ouvertures, fermetures, jours, dates, etc.

Un jour de la semaine peut, selon sa situation, être annoté de deux façons différentes :

(108) **Ouvert du lundi au vendredi.**

(109) **Ouvert le lundi et le vendredi.**

Pour le premier exemple, les balises appropriées sont <jour_debut> et <jour_fin>, tandis que pour le second, la balise <jour> suffit. Ainsi, un même terme présent doit pouvoir être annoté de plusieurs façons différentes selon l'expression dans laquelle il se trouve, alors même qu'il peut éventuellement ne se trouver que dans un seul sous-graphe. Il en est de même pour le typage en ouverture ou en fermeture :

(110) **Ouvert le lundi.**

(111) **Fermé le lundi.**

L'exemple (110) doit être annoté avec la balise <periode_ouverture>, tandis que l'exemple

51 XML valide : respectant la DTD associée.
52 XML bien formé : respectant la syntaxe XML.

(111) doit être annoté par une balise <periode_fermeture>.

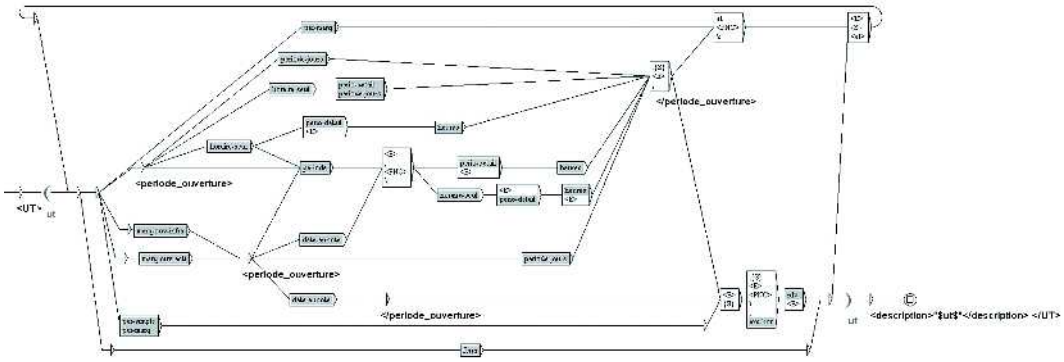


Figure 57 : Transducteur « accueil »

Le transducteur « accueil » présenté ci-dessus permet de visualiser les boucles grâce auxquelles il est possible de combiner plusieurs informations, tout en les annotant correctement. En effet, la boucle se situe à l'extérieur de l'annotation, exceptées les balises <UT> et <description> qui se trouvent en dehors de la boucle.

Ce transducteur ne comprend que des balises <periode_ouverture>, la balise <periode_fermeture> se trouvant dans le sous-graphe « ferm » et les autres balises, plus spécifiques, se trouvant dans les différents sous-graphes auxquels il est fait appel. Néanmoins, ce transducteur permet de visualiser que tout chemin passant par une balise ouvrante doit nécessairement inclure la balise fermante correspondante.

Le transducteur « accueil » est le transducteur général pour les expressions temporelles. Il fait appel à de nombreux sous-graphes, pour couvrir un maximum de cas possibles, sans être surchargé. Par exemple, le transducteur suivant illustre que, pour exprimer une période itérative à l'aide de jours de la semaine, au moins deux types de tournures sont possibles :

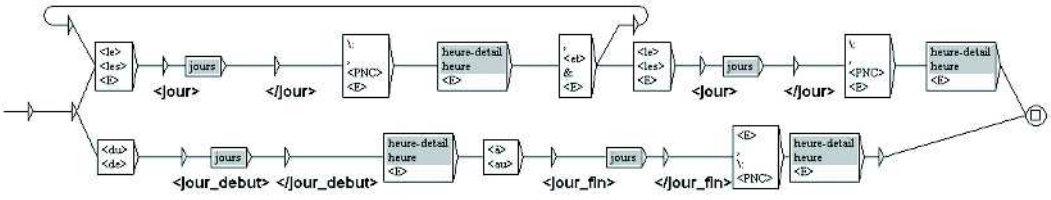


Figure 58 : Transducteur « période-jours »

Le chemin supérieur repère et annote les expressions du même type que celle donnée en (112), tandis que le chemin inférieur permet de repérer et d'annoter des expressions comme celle donnée en (113). Dans les deux cas, il est possible d'ajouter des horaires.

(112) le lundi, le mardi et le vendredi.

(113) du mardi au vendredi.

Le transducteur montre que les balises insérées dépendent du chemin emprunté. L'utilité des sous-graphes qui sont utilisés plusieurs fois se reflète ici.

b. Exceptions

Comme cela a été mentionné au chapitre précédent, le statut de la balise <exception> est particulier : elle peut survenir dans une période d'ouverture ou de fermeture, ou directement dans une UT. De plus, elle peut contenir d'autres balises (période d'ouverture et de fermeture) ou non.

Dans les cas où le contenu d'une exception doit être balisé plus finement, la difficulté réside dans le typage :

(114) **Ouvert du lundi au vendredi sauf le mardi.**

(115) **Fermé du lundi au vendredi sauf le mardi.**

Dans ces deux exemples, l'exception est exprimée de la même manière. Néanmoins, dans le premier exemple, son contenu doit être typé comme une fermeture, tandis que dans le second, il doit être typé comme une ouverture. Pour ce faire, le sous-graphe modélisant les exceptions a dû être dédoublé : dans le cas où il intervient après une balise d'ouverture, il contient des annotations sous forme de <periode_fermeture> et inversement.

c. Incertitude

La balise <incertitude>, quant à elle, doit être insérée dans certains cas précis, si l'expression est floue :

(116) **début mars**

(117) **en mars**

(118) **du 1er au 15 mars**

Ainsi, elle ne doit être ajoutée que pour les exemples (116) et (117) qui ne permettent pas de définir des dates précises de début et de fin. En revanche, l'exemple (118) n'est pas flou et ne nécessite pas l'ajout de la balise.

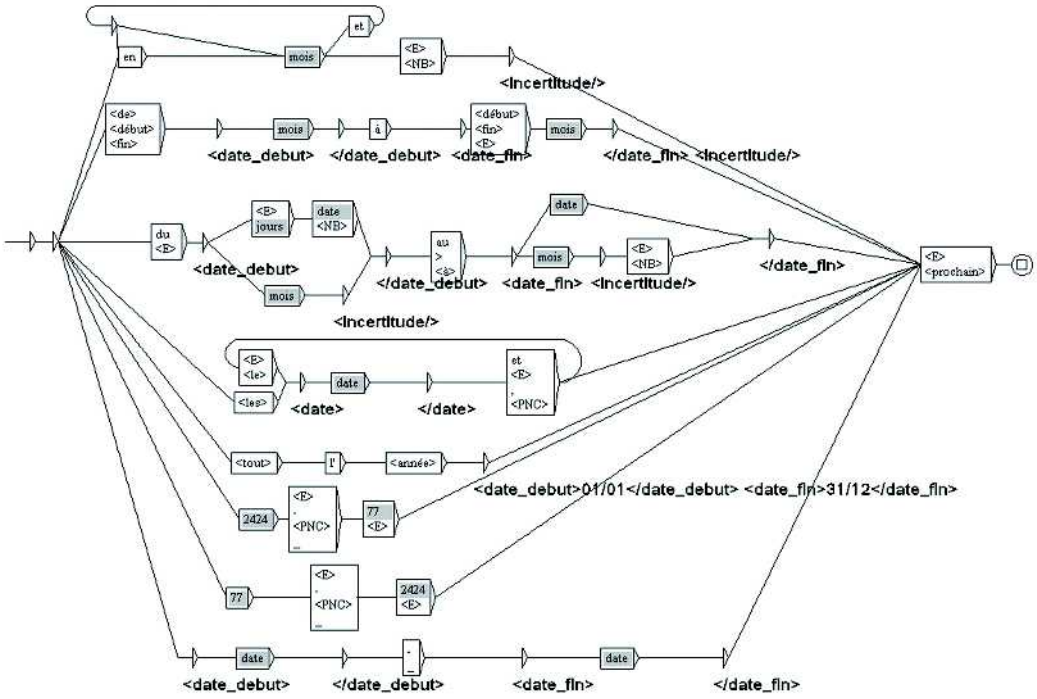


Figure 59 : Graphe « période »

Le transducteur « période » ci-dessus illustre l'insertion de la balise <incertitude> et montre qu'elle a nécessité, pour certains chemins, un dédoublement. Ainsi, elle est insérée si un mois seul est mentionné mais pas si celui-ci fait partie d'une date précise repérée par le sous-graphe « date ».

d. Objets

Le transducteur de repérage des objets (déjà présenté dans la figure 49, page 118) est particulier. Il reproduit la structure de l'ontologie de façon à pouvoir annoter chaque objet repéré par le nom de la classe à laquelle il devra appartenir. Comme cela a déjà été précisé, les objets ne constituent pas d'expressions complexes et chacun des sous-graphes est relativement simple. De plus, ils ne nécessitent pas l'ajout de balises : toutes les annotations sont insérées au niveau du transducteur principal et non dans les sous-graphes, ce qui garantit la validité du XML obtenu.

Un cas particulier est tout de même apparu :

(119) Hôtel de ville

L'*Hôtel de ville* est bien un objet touristique dans la classification du projet Eiffel. En revanche, il ne s'agit pas d'un hébergement et il faut donc le typer correctement comme un service local.

3.1.3. Ne pas repérer

Les transducteurs permettent donc de modéliser les expressions que l'on souhaite repérer et

annoter. En revanche, ils ne permettent pas la négation d'un enchaînement. Ainsi, il n'est pas possible de modéliser des expressions à ne pas repérer.

Toutefois, étant donné qu'Adetoa fonctionne en annotant les données à traiter, il suffit de modéliser les expressions à ne pas repérer, sans y insérer d'annotations.

a. Informations temporelles non touristiques

Certaines informations temporelles qui figurent dans les pages Web ne sont pas directement liées au domaine touristique ou, du moins, ne sont pas des informations pratiques. Elles ne doivent donc pas être repérées.

Il s'agit principalement des dates historiques et des dates de dernière modification du site Web, comme dans les exemples suivants :

(120) **Le Musée a ouvert ses portes en mai 1989.**

(121) **Dernière modification : 12/04/2009**

Pour éviter au maximum les repérages fautifs, les dates seules, c'est-à-dire les dates non introduites par un marqueur d'ouverture ou de fermeture comme *ouvert, fermé, ouverture,* et non complétées par d'autres informations temporelles, ne sont pas repérées. En effet, il est apparu qu'il ne s'agit généralement pas d'informations pratiques.

En ce qui concerne les dates historiques, certaines sont, malgré tout, repérées et annotées. En effet, le patron classique qu'elles suivent généralement peut prendre place dans des expressions d'informations pratiques, et le fait de les ignorer représentait un risque de faire manquer d'autres expressions. En revanche, le moteur d'inférence du projet Eiffel, présenté au chapitre suivant, peut, lors du stockage dans la base de connaissance, appliquer un filtre sur les dates pour éliminer, par exemple, les informations obsolètes.

b. « Fausses adresses » et « faux objets »

Les informations temporelles à ne pas repérer sont à ignorer car elles ne sont pas des informations pratiques touristiques. En revanche, les informations de localisation et les objets touristiques sont des expressions qui peuvent être considérées, à tort, comme des informations utiles.

(122) **Hôtel de police**

Le terme *hôtel* déclenche un repérage et une annotation en tant qu'hébergement (sauf *hôtel de ville* déjà mentionné plus haut). Ici au contraire, il ne signale pas un objet touristique et ne doit pas être annoté. C'est pour cela qu'il figure dans un sous-graphe « nepasreperer » qui ne l'annote pas.

Pour ce qui est des informations de localisation, la difficulté vient du fait que certains noms de communes, qui sont donc présents dans le dictionnaire des noms de villes, sont également des noms communs et peuvent donc apparaître dans les pages Web, sans référer aux communes en question. Les noms correspondant à de très petites communes, qui sont également des noms communs très fréquents, ont simplement été supprimés du dictionnaire. Il s'agit par exemple de *Nuits* et *Chambre*. Les expressions qui risquent par ce fait d'être manquées sont bien moins nombreuses que les expressions qui auraient été repérées à tort si ces termes étaient restés dans le dictionnaire.

3.2. Dictionnaire

Comme cela a déjà été mentionné, le repérage et l'annotation des expressions temporelles, des informations de localisation et des objets touristiques ont finalement nécessité le développement d'un seul dictionnaire, un dictionnaire des noms de commune.

Pour développer ce dictionnaire, je me suis basée sur des données issues du code officiel géographique que l'INSEE publie sur son site Internet[53]. Ces données sont modélisées selon le standard RDF du Web Sémantique. Elles comprennent plusieurs fichiers : l'ontologie en elle-même qui gouverne les données publiées, exprimée en OWL-Lite, et les fichiers contenant les données proprement dites, c'est-à-dire des instances des classes définies dans l'ontologie. Il s'agit d'un fichier « régions », constitué de la liste des régions, avec chefs-lieux et départements, et d'un fichier « département », constitué de la liste des départements avec chefs lieux et arrondissements. Associé à chaque département, un fichier « arrondissement » contient la liste des communes et des cantons, et un fichier « cantons » modélise les liens entre communes et cantons, ainsi que les relations de voisinage entre communes. Tous ces fichiers sont donc très riches et contiennent beaucoup plus d'informations qu'il n'y en a d'exploitables. Une liste des communes en a donc été extraite pour créer un fichier au format des dictionnaires d'Unitex. Ce dictionnaire se veut le plus simple possible et répertorie les villes en leur attribuant la catégorie « ville », comme l'illustre l'extrait suivant :

```
Nanterre,.VILLE
Nantes,.VILLE
```

Cela permet au transducteur de localisation de repérer, dans les textes, des motifs tels que <num> « rue » + <N> + <num> + <VILLE>, afin d'identifier des adresses comme celle-ci :

(123) 5 rue Faidherbe 58000 Nevers

Une fois les noms de villes collectés dans les ressources fournies, un traitement a été effectué sur les noms composés pour que le dictionnaire soit adapté à la façon dont ils sont utilisés dans les pages Web. Ainsi, les noms composés sont acceptés s'ils comportent un (ou des) trait d'union aussi bien que s'ils n'en comportent pas.

Le développement d'un dictionnaire des objets a été envisagé, étant donné que les objets sont souvent exprimés par un terme simple. Toutefois, lorsque ce n'est pas le cas et que l'objet est représenté par une expression de plusieurs mots, la tâche s'avère plus compliquée. De plus, pour pouvoir annoter les objets directement selon les classes de l'ontologie, il aurait fallu modéliser plusieurs types dans le dictionnaire (ou bien plusieurs dictionnaires). Contrairement au dictionnaire de noms de communes, dans lequel toutes les entrées ont le même type et qui s'utilise donc très aisément, l'utilisation d'un dictionnaire des objets n'aurait pas nécessairement simplifié les transducteurs.

3.3. Transformations sur les informations temporelles

Certaines transformations rendent l'exploitation des données annotées plus aisée. Il s'agit principalement des transformations qui complètent l'information, en cas d'ellipse par exemple, ou qui la rendent plus conforme à l'ontologie. Les transformations les plus simples qui ont été mises au point surviennent directement dans les transducteurs Unitex, et ce, dès l'annotation. Un second type de transformations prend la forme de règles implémentées en

53 http://rdf.insee.fr/geo/

JAVA, grâce au langage Xpath.

3.3.1. Transformations dans les transducteurs

Deux cas nécessitant une transformation sont traités dès l'annotation dans les transducteurs Unitex. Il s'agit de deux cas particuliers d'expressions fréquentes :

(124) Ouvert tous les jours.

(125) Ouvert toute l'année.

Le schéma d'annotation ne permet pas d'annoter directement ces expressions : il n'y a ni date de début, ni date de fin, ni jour défini. En outre, ces énoncés ne sont pas ambigus et ne posent pas de problème d'interprétation ; ils peuvent donc être annotés, dès le repérage, à l'aide d'une transformation.

Pour l'exemple (124), l'annotation énumère tous les jours de la semaine :

```
<periode_Ouverture> Ouvert tous les jours
   <jour>lundi</jour>
   <jour>mardi</jour>
   <jour>mercredi</jour>
   <jour>jeudi</jour>
   <jour>vendredi</jour>
   <jour>samedi</jour>
   <jour>dimanche</jour>
</periode_Ouverture>
```

Il faut toutefois noter que cette transformation ne tient compte que des données purement linguistiques contenues dans l'expression de départ. Elle ne prend en effet pas en compte les connaissances métier qui permettraient, par exemple, de savoir que, s'il s'agit d'une mairie, *tous les jours* exclut le dimanche, voire même le samedi. La prise en compte des connaissances métier relève plutôt du module de consolidation des connaissances développé à Mondeca et présenté au chapitre suivant.

Pour l'exemple (125), l'annotation permet d'exprimer directement une date de début et une date de fin :

```
<periode_ouverture> Ouvert toute l'année
   <date_debut>01/01</date_debut>
   <date_fin>31/12</date_fin>
</periode_ouverture>
```

Ainsi, les annotations obtenues sont conformes au schéma d'annotation et proches de l'ontologie et peuvent donc être facilement stockées dans la base de connaissance.

3.3.2. Transformations en JAVA avec XPath

Le module de transformations, réalisé directement en JAVA et utilisant XPath, se base sur les données annotées, et applique trois règles : la première convertit les groupes de jours en énumération, la deuxième convertit certaines fermetures en ouvertures et la troisième permet de compléter certaines dates en cas d'ellipse. Ces trois règles peuvent s'appliquer successivement à la même annotation. Dans tous les cas, le chemin XPath permet de localiser, dans le document XML, l'endroit où la transformation doit avoir lieu. Des règles de transformations utilisant JDOM sont ensuite appliquées.

a. Conversion des groupes de jours en énumérations de jours

Lorsqu'une offre touristique est ouverte plusieurs jours dans la semaine, ces derniers sont généralement groupés comme dans l'exemple (126) et rarement énumérés comme dans l'exemple 127) :

(126) **Ouvert du lundi au vendredi.**

(127) **Ouvert le lundi, le mardi, le mercredi, le jeudi, le vendredi.**

Voici l'annotation fournie par Adetoa pour l'exemple (126) :

```
<UT>
   <periode_Ouverture> ouvert du
      <jour_debut> lundi</jour_debut> au
      <jour_fin> vendredi</jour_fin>
   </periode_Ouverture>
   <description> "Du lundi au vendredi" </description>
</UT>
```

Une transformation de cette annotation en énumération de jours la rend plus facile à stocker dans la base de connaissance que si elle est représentée par un jour de début et un jour de fin. L'annotation voulue, et obtenue après transformation, est la suivante :

```
<UT> <periode_Ouverture>
      <jour> lundi</jour>
      <jour> mardi </jour>
      <jour> mercredi </jour>
      <jour> jeudi </jour>
      <jour> vendredi </jour>
   </periode_Ouverture>
   <description> "Ouvert du lundi au vendredi" </description>
</UT>
```

La règle permettant cette transformation commence par localiser, dans la page Web, une UT comprenant une <periode_ouverture>, qui elle-même contient une balise <jour_debut> et une balise <jour_fin>. À l'aide d'un tableau contenant les jours de la semaine de façon ordonnée, le moteur peut reconstruire les jours d'ouverture. Les variantes d'orthographe des jours de la semaine (majuscule / minuscule, abréviations) sont prises en compte.

Si l'expression comprend également une information d'horaire (exemple 128), cela ne change rien à l'application de la règle de transformation.

(128) **Ouvert du lundi au vendredi, de 9h à 18h.**

b. Conversion des périodes de fermeture en périodes d'ouverture

Si les périodes de fermeture sont modélisables dans l'ontologie, il est tout de même préférable, lorsque c'est possible, de stocker les informations sous forme d'ouverture. Ce choix est motivé par une observation du comportement des internautes qui montre que ces derniers cherchent plus souvent à connaître les moments d'ouverture que les moments de fermeture.

Pour convertir une période de fermeture en période d'ouverture, il faut donc trouver son complémentaire. Or, la période d'ouverture correspondante ne concorde pas toujours avec le complémentaire exact, comme s'il s'agissait d'un monde clos.

(129) **Fermé tous les jours de 12h00 à 14h00.**

Ainsi, si l'on prend le complémentaire exact de l'exemple (129), on obtient que l'offre est ouverte tous les jours de minuit à 12h00 et de 14h00 à minuit. Or une telle expression peut tout à fait concerner un magasin, un musée, une mairie, qui serait ouvert toute la journée, par exemple de 8h00 à 20h00 mais fermé à l'heure du déjeuner.

Toutefois, pour les expressions ayant une granularité journalière, sans information horaire, le complémentaire semble plus facile à définir :

(130) Fermé le mardi.

Une telle expression, sans information supplémentaire, peut, sans trop de risque d'erreur, être interprétée comme « ouvert le lundi, le mercredi, le jeudi, le vendredi, le samedi et le dimanche ».

Une transformation permettant de convertir une information de fermeture en information d'ouverture est donc possible lorsque la granularité de l'expression est celle de jour. De plus, il se trouve que cette granularité est fréquente dans les pages Web du projet Eiffel, notamment lorsqu'il s'agit de donner les jours d'ouverture ou de fermeture. L'exemple (130) est en effet plus simple à formuler que son correspondant en ouverture :

(131) Ouvert le lundi, le mercredi, le jeudi, le vendredi, le samedi et le dimanche.

L'exemple donné en (130) est annoté ainsi par Adetoa :

```
<UT> Fermé
   <periode_fermeture> le
      <jour> mardi </jour>
   </periode_fermeture>
   <description> "Fermé le mardi" </description>
</UT>
```

Une fois la transformation appliquée, l'annotation se rapproche de celle qui aurait été effectuée pour l'exemple (131) et devient la suivante[54] :

```
<UT>
   <periode_ouverture>
      <jour> lundi </jour>
      <jour> mercredi </jour>
      <jour> jeudi </jour>
      <jour> vendredi </jour>
      <jour> samedi </jour>
      <jour> dimanche </jour>
   </periode_ouverture>
   <description> "Fermé le mardi" </description>
</UT>
```

La règle permettant d'effectuer cette transformation commence par localiser dans la page Web, grâce à XPath, une période de fermeture ne contenant que des jours. Le but est ensuite de créer (ou de compléter) une période d'ouverture contenant tous les jours de la semaine qui n'apparaissaient pas dans la période de fermeture.

Cette règle est donc également applicable aux expressions dans lesquelles se trouvent plusieurs jours de fermeture, comme celle de l'exemple (132), équivalent à l'exemple (133).

54 Notons que les annotations établies par Adetoa sont « dans le texte », les mots *fermé* et *le* y apparaissent donc. Une fois la transformation effectuée, ils sont supprimés car le texte en lui-même n'est pas modifié ; seules les balises le sont. Le texte encadré par la balise <description> reste celui qui a été repéré dans la page ; il ne tient pas compte de la transformation.

(132) Fermé le mardi, le mercredi et le jeudi.

(133) Ouvert le lundi, le vendredi, le samedi et le dimanche.

c. Expressions mixtes

Les expressions annotées comprennent parfois des informations d'ouverture et des informations de fermeture, l'ouverture pouvant en plus être formulée sous la forme d'un groupe de jours. Les exemples suivants en témoignent :

(134) Ouvert du lundi au dimanche sauf le mercredi.

(135) Ouvert tous les jours sauf le dimanche.

L'exemple (134) est annoté ainsi par Adetoa :

```
<UT>
   <periode_Ouverture> ouvert du
      <jour_debut> lundi</jour_debut> au
      <jour_fin> dimanche </jour_fin>
   </periode_Ouverture>
   <exception>
      <periode_fermeture> sauf le
         <jour> mercredi </jour>
      </periode_fermeture>
   </exception>
   <description> "Ouvert du lundi au dimanche sauf le mercredi"
   </description>
</UT>
```

Les deux règles présentées plus haut s'appliquent : d'abord, le groupe de jours d'ouverture est converti en une énumération de jours, ensuite, le jour de fermeture est retiré. L'annotation obtenue est donc la suivante :

```
<UT>
   <periode_ouverture>
      <jour> lundi </jour>
      <jour> mardi </jour>
      <jour> jeudi </jour>
      <jour> vendredi </jour>
      <jour> samedi </jour>
      <jour> dimanche </jour>
   </periode_ouverture>
   <description> "Ouvert du lundi au dimanche sauf le mercredi."
</description>
</UT>
```

Pour l'expression donnée en (135), l'annotation fournie par Adetoa est indiquée ci-dessous. Comme cela a déjà été mentionné, pour l'expression *tous les jours*, la conversion en énumération de jours est déjà faite à l'étape d'annotation. Cela provoque toutefois une incohérence puisque dimanche apparaît à la fois comme un jour de fermeture et comme un jour d'ouverture.

```
<UT>
   <periode_ouverture> Ouvert tous les jours
      <jour>lundi</jour>
      <jour>mardi</jour>
      <jour>mercredi</jour>
      <jour>jeudi</jour>
      <jour>vendredi</jour>
      <jour>samedi</jour>
      <jour>dimanche</jour>
   </periode_ouverture>
      <exception>
         <periode_fermeture> sauf le
            <jour>dimanche</jour>
         </periode_fermeture>
      </exception>
   <description> "Ouvert tous les jours sauf le dimanche"
   </description>
</UT>
```

Dans ce cas, priorité est donnée à la fermeture : lorsqu'une période de fermeture est convertie en ouverture, les jours de fermeture sont ôtés de la période d'ouverture, comme dans le cas précédent et l'annotation suivante est obtenue :

```
<UT>
   <periode_ouverture> Ouvert tous les jours
      <jour>lundi</jour>
      <jour>mardi</jour>
      <jour>mercredi</jour>
      <jour>jeudi</jour>
      <jour>vendredi</jour>
      <jour>samedi</jour>
   </periode_ouverture>
   <description> "Ouvert tous les jours sauf le dimanche"
   </description>
</UT>
```

Des cas d'incohérence peuvent aussi apparaître si l'expression de base est ambiguë. Il existe en effet des cas où le rédacteur de la page Web a fait une erreur et a écrit une expression telle que :

(136) Ouvert les lundi, mardi, mercredi et jeudi, fermé le mardi.

On constate que *mardi* apparaît en jour d'ouverture et en jour de fermeture mais il est impossible de trancher pour savoir si c'est ouvert ou fermé. Ici encore, la priorité est donnée à la fermeture, aucune règle spécifique n'est ainsi nécessaire. Toutefois, il s'agit d'un choix arbitraire, étant donnée l'ambigüité de l'expression.

d. Complétion des ellipses dans les intervalles de dates

Cette dernière règle de transformation est spécifique aux expressions comprenant une date de début et une date de fin, mais dont la date de début ne contient que le numéro de jour et non le mois, comme dans l'exemple suivant :

(137) Ouvert du 10 au 20 mars.

Pour cet exemple, Adetoa fournit l'annotation suivante :

```
<UT>
   <periode_ouverture> Ouvert du
      <date_debut> 10 </date_debut> au
      <date_fin> 20 mars </date_fin>
   </periode_ouverture>
   <description> "Ouvert du 10 au 20 mars." </description>
</UT>
```

Une telle date de début est inexploitable et doit donc être complétée pour obtenir l'équivalent de l'expression :

(138) **Ouvert du 10 mars au 20 mars.**

Pour appliquer cette règle, le module de transformations commence par localiser les dates de début. L'utilisation d'expressions régulières lui permet de vérifier si la date de début comprend le mois ou non et, le cas échéant, si celui-ci est complété d'une année. Cette règle fonctionne également pour les périodes de fermeture et peut aussi se combiner avec les autres règles de transformations.

À la différence des règles précédentes, celle-ci permet d'ajouter du texte mais ne modifie pas les balises déjà insérées.

Conclusion

Ce chapitre a permis de présenter Adetoa, l'outil que j'ai mis au point pour effectuer le repérage et l'annotation des données utiles au projet Eiffel. S'il est assez descriptif, puisqu'il décrit le fonctionnement d'Unitex et celui des différents modules d'Adetoa, il a aussi permis de motiver certains choix et de mettre en avant les difficultés rencontrées et la façon dont elles ont été surmontées.

Ainsi, les transducteurs ne permettent d'annoter que le texte en lui-même mais pas de le compléter par des informations extérieures. C'est pourquoi un module de transformations, capable d'insérer de nouvelles données, a été mis au point.

Par ailleurs, l'une des difficultés majeures à laquelle sont confrontés les systèmes de repérage à base de graphes ou transducteurs est qu'il est nécessaire de prédire et de formaliser précisément tous les cas à traiter. Or, la richesse de la langue permet de générer un nombre incalculable d'énoncés et il est impossible de prédire toutes les expressions pouvant être rencontrées dans un texte en langage naturel, même en se concentrant sur des informations précises. Les transducteurs nécessitent alors un compromis, de façon à pouvoir rendre compte au maximum de la combinatoire, tout en restant manipulables.

Le chapitre suivant a pour but de montrer comment sont exploitées, dans le projet Eiffel, les données produites par Adetoa.

Chapitre 6. Exploitation des Données Annotées dans une Chaîne de Traitement

Introduction

Ce chapitre a pour but de montrer comment les données qui ont été repérées puis annotées et liées par Adetoa peuvent ensuite être utilisées. Ces données sont exploitées dans le cadre du projet Eiffel, mais les questions soulevées dans ce chapitre peuvent s'appliquer à d'autres projets, que ce soit sur le plan théorique ou sur le plan pratique lors des implémentations.

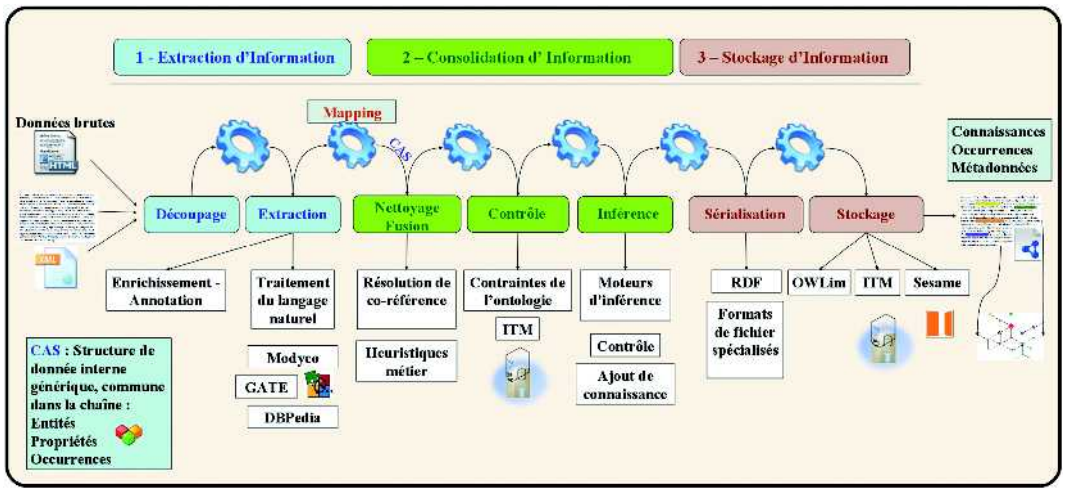


Figure 60 : Chaîne d'annotation sémantique – Content-Augmentation Manager [Coste 2009]

La chaîne de traitement du projet Eiffel, telle qu'elle est implémentée à Mondeca, est représentée dans le schéma ci-dessus. Celle-ci est également présentée dans [Weiser et al. 2009] et des chaînes similaires sont présentées dans [Jilani & Amardeilh 2009] et [Amardeilh & Damljavonic 2009]. Cette chaîne, mise au point dans le Content Augmentation Manager

(CA Manager[55]), permet de récupérer les données annotées, de les transformer en RDF[56], format exploitable par le CA Manager, de les consolider et enfin de les stocker dans la base de connaissance du projet. Le CA Manager est basé sur l'infrastructure UIMA[57] [Ferrucci & Lally 2004].

La première étape « Extraction d'information » correspond à l'appel à Adetoa. Les étapes suivantes « Consolidation d'information » et « Stockage d'information » permettent de vérifier la qualité des données et de les stocker dans la base de connaissance. Ces trois étapes correspondent à la structure de ce chapitre.

Dans un premier temps, je vais donc présenter le mapping qui se charge de récupérer les données annotées par Adetoa. Pour être stockées dans une base de connaissance, les données doivent être associées à un identifiant unique. Par ailleurs un libellé doit aussi être associé à chaque ressource, de façon à pouvoir effectuer des recherches dans la base, présenter des résultats à l'utilisateur, etc. Nous verrons que c'est le nom de la ressource touristique qui a été choisi pour servir de libellé, mais il n'est pas si simple à définir ni à trouver dans la page Web. Je présenterai donc les règles d'acquisition de connaissance qui ont été élaborées dans le cadre d'Eiffel, pour repérer ces libellés dans les pages Web ou les créer automatiquement.

Dans un second temps nous verrons comment tirer le plus d'information possible des données temporelles annotées. Les expressions contenues dans les pages Web peuvent faire l'objet de transformations afin de pouvoir être stockées et interrogées le mieux possible et parfois complétées à l'aide de connaissances du monde.

Enfin, dans un troisième temps, seront abordées les questions du stockage à proprement parler des données et de l'interrogation du système pour répondre à l'utilisateur de la plateforme. J'aborderai alors les questions de saturation de la base, la fiabilité des informations et l'interface permettant à l'utilisateur d'effectuer des requêtes.

1. Mapping

La première étape du schéma de la figure 60 correspond à Adetoa. Des fichiers annotés, organisés selon une arborescence (XML), en constituent la sortie. Ces fichiers doivent faire l'objet d'un mapping avant que les données qu'ils contiennent puissent être consolidées puis stockées dans la base de connaissance. Le résultat de cette étape de mappping est un graphe, au format RDF.

Rappelons qu'avant d'en arriver à l'étape de mapping, Adetoa a donc annoté les documents et effectué certaines transformations, au niveau textuel, de façon à ce que les données soient le plus proches possible de l'ontologie, et cela, grâce au schéma d'annotation. Néanmoins, comme on l'a vu au chapitre précédent, le schéma d'annotation n'est pas exactement conforme à l'ontologie et des règles de mapping sont donc nécessaires.

Le format utilisé dans Eiffel permet d'exprimer des relations sous forme de graphes. Suivant les annotations XML fournies par Adetoa, un graphe sera créé pour chaque balise

55 CA Manager, nouvelle version de la plateforme OntoPop [Amardeilh 2007] est développé dans le cadre du projet de recherche européen TAO - Transitioning Applications to Ontologies dont fait partie Mondeca [Amardeilh & Damljavonic 2009].

56 Resource Description Framework - http://www.w3.org/RDF/

57 *Unstructured Information Management applications* – plateforme qui permet d'analyser de grands volumes d'information non structurée.

<ensemble> du fichier. Mais cet ensemble doit d'abord être associé à un libellé.

Ainsi, il faut donc, pour chaque ressource touristique à prendre en compte, associer un libellé à l'ensemble des différentes données repérées. Par exemple, dans une page Web, Adetoa aura repéré que le type de l'objet est « mairie », ses horaires d'ouverture (*du lundi au vendredi de 9h à 18h*) et son adresse (*1, place de l'hôtel de ville, Nevers*) et aura déjà effectué le liage des données. Il faudra alors associer un nom à cet ensemble, par exemple *Mairie Nevers*, pour pouvoir stocker l'ensemble des données dans la base de connaissance.

Dans cette partie, nous verrons donc dans un premier temps comment choisir le nom qui servira de libellé « préféré » (voir les « prefered terms » en terminologie) dans la base de connaissance. Dans un second temps, nous verrons, sur un plan plus technique, les règles d'acquisition des connaissances qui ont été adoptées pour le projet Eiffel.

1.1. Quel libellé choisir pour les différentes ressources touristiques ?

Comme on l'a vu plus haut, c'est le « nom de la ressource » qui a été choisi pour servir de libellé. Si ce choix a été effectué spontanément, c'est bien parce que, dans la langue, les noms propres ont pour rôle de désigner des référents uniques. C'est en tout cas les définitions que l'on peut trouver dans [Charolles 2002] :

> « D'un point de vue référentiel, les Np [noms propres] peuvent référer à différentes sortes d'entités particulières » (p.53)

> « L'usage d'un Np [nom propre] ne signale aucune autre intention chez le locuteur que de **viser un être unique** » (p.54)

[Leroy 2004] confirme que cette définition du nom propre en tant qu'unité permettant de désigner un référent unique est en effet la plus répandue dans la littérature. Néanmoins, elle signale qu'elle peut être remise en question :

> « Ce critère de l'unicité référentielle [en tant que définitoire du nom propre] est donc lui aussi discutable, bien qu'en partie fondé. S'il correspond bien au fonctionnement du nom propre, il ne peut suffire à le définir ni à en délimiter la catégorie, ne serait-ce que parce que le nom commun y répond également dans certains cas, et que le nom propre y échappe dans d'autres cas. » (p.24)

Cela ne remet pas pour autant en cause le statut de nom propre. Tout en restant prudente, je garde la propriété d'unicité référentielle du nom propre en langue pour la rapprocher de l'informatique où des identifiants uniques sont également nécessaires pour faire référence aux objets que l'on traite. Dans son sens le plus fréquent, le nom a donc, en langue tout comme en informatique, pour vocation d'être unique et de ne permettre ainsi de référer qu'à un seul objet, ou du moins comme le mentionne Charolles en citant les travaux de Searle, il permet aux interlocuteurs concernés de se mettre d'accord sur l'objet dont ils parlent :

> « Les Np [noms propres] ne sont en réalité, conclut J. Searle, que des **outils commodes nous permettant de désigner des êtres particuliers sans avoir à nous engager et à nous mettre d'accord avec nos interlocuteurs sur ce qui en fait précisément des êtres singuliers**, mais il n'en demeure pas moins que, visant des particuliers en tant que particuliers, ils doivent leur être attachés par une relation descriptive (substantielle) quelconque. » ([Charolles 2002] p.57)

Mais peut-on dire de tous les objets touristiques qu'ils ont un nom ? Certains types de ressources linguistiques comme les hôtels, restaurants et musées portent habituellement un nom propre. Ce nom n'a qu'une seule forme et ne varie pas, je parlerai donc de nom propre

singulier. Il n'en est généralement pas de même en ce qui concerne les mairies, offices de tourisme et syndicats d'initiative qui n'ont pas de nom propre prédéfini et invariable et pour lesquels d'autres procédés de référence sont mis en œuvre.

J'ai donc choisi d'analyser les ressources touristiques et de leur associer un libellé en fonction de leur type, en distinguant les ressources qui portent un nom propre singulier et celles qui n'en ont pas. En effet, la démarche diffère puisque dans le premier cas il s'agit de « trouver » le nom dans la page Web, tandis que dans le second, il s'agit d'en « créer » un pouvant servir de libellé.

1.1.1. Ressources touristiques ayant un nom propre singulier

Parmi les différents types de ressources touristiques qui nous intéressent dans le cadre du projet Eiffel, les principaux à posséder un nom propre singulier sont les hôtels, les restaurants et les musées. Comme on l'a vu ci-dessus, ce nom permet de les identifier de façon unique.

Sur une page Web, ce nom apparaît donc en général clairement, ce qui permet à l'internaute de savoir directement de quoi il est question. Il est souvent mis en avant, la plupart du temps à l'aide de procédés visuels extra-linguistiques : couleurs, caractères plus grands, images etc.

Sur le plan linguistique, les noms des ressources touristiques sont des noms propres. Comme le mentionne Charolles, ils peuvent être très divers :

> « Les noms propres sont tout aussi **arbitraires** que les noms communs dont ils sont du reste souvent tirés, mais comme ils sont appelés à désigner des particuliers, ils sont ouverts à une créativité sans limite. » ([Charolles 2002] p.54)

Il peut donc s'agir de noms communs utilisés en noms propres, de mots d'autres catégories lexicales, également alors utilisés en noms propres, ou encore de mots fabriqués et qui n'existaient pas auparavant.

De plus, les noms des ressources touristiques peuvent être constitués d'un ou plusieurs mots et prendre des formes très diverses. En ce qui concerne la nature des mots composant ces noms, tout est possible : groupe nominal introduit par une préposition, substantif avec ou sans déterminant, adverbe, groupe verbal, etc. Voici quelques exemples de noms de musées :

(139) **Musée du Louvre**

(140) **Musée du Quai Branly**

(141) **Musée Magritte**

(142) **Musée des Arts et Métiers**

(143) **Fondation Cartier**

(144) **Palais de la Porte Dorée**

(145) **Cité des Sciences et de l'Industrie**

Nous pouvons ainsi constater la diversité des noms de musées, et ces noms se veulent pourtant explicites, il y a donc une moins grande variété que dans les noms d'autres types d'objets. L'avantage des quatre premiers exemples (139 à 142) est que le terme *Musée* est contenu dans le nom. En revanche, ce qui suit ce terme prend des formes diverses : trois fois sur les quatre, c'est une proposition qui introduit un groupe nominal, avec un nom propre pour *Louvre*, un nom commun et un nom propre pour *quai Branly* et deux noms communs

coordonnés pour *Arts et Métiers*. Pour *Magritte*, seul un nom propre compose le nom, il n'est pas introduit. Pour les exemples (143), (144) et (145), le terme *musée* n'apparaît pas dans le nom, il est remplacé par *fondation*, *palais* et *cité*. Le problème de ces termes est qu'ils sont ambigus : ils ne font pas toujours référence à des musées : le *palais de justice*, la *cité des Ulis* ou la *fondation pour l'enfance* ne sont en effet pas des noms de musées. Une nouvelle fois, ce qui suit prend des formes variées : un nom propre pour *Cartier*, un groupe prépositionnel pour *Palais de la Porte Dorée*, composé d'une préposition, d'un déterminant, d'un nom commun et d'un adjectif, et pour C*ité des Sciences et de l'Industrie* : deux groupes prépositionnels coordonnés. On peut remarquer que parmi les exemples proposés, chacun a sa forme propre, même si certains peuvent sembler similaires : les exemples (142) et (145) sont proches puisqu'ils se composent tous les deux de deux groupes prépositionnels coordonnés, mais dans le second la préposition et le déterminant sont répétés, contrairement au premier.

Rappelons que, parmi les noms propres qui forment les noms des ressources touristiques, un grand nombre sont issus de noms communs et autres catégories grammaticales de la langue. Sur ces sept exemples, trois ne sont pas formés à partir de noms propres préexistants. Ayant été choisis au hasard, ces exemples n'ont pas la prétention de refléter exactement la réalité et encore moins de permettre d'en tirer des pourcentages ; néanmoins, la présence de trois noms sans noms propres – et qui n'ont pas été sélectionnés pour cette caractéristique en particulier – suffit à avancer qu'une simple analyse des noms propres connus dans la langue n'est pas une solution suffisante pour déterminer les noms des ressources touristiques. De plus, selon la norme, les termes composant des noms d'offres touristiques et devenant alors noms propres doivent commencer par une lettre majuscule – même si ce ne sont pas à la base des noms propres mais que ce sont des substantifs, adjectifs ou autres – mais j'ai déjà pu montrer que la norme est rarement respectée sur Internet et le critère basé sur l'utilisation de lettres capitales n'y est donc pas fiable.

Il est donc impossible d'établir, pour déterminer le libellé d'une ressource, des règles du type : « le mot *musée* et le mot qui suit », « le mot *musée* et les deux mots qui suivent si c'est une préposition et un substantif » car le nombre de possibilités est trop grand. De plus, le risque d'erreur est bien trop grand : le mot « musée » peut très bien apparaître sans être suivi de son nom et l'application de telles règles mènerait trop souvent à des troncations du nom, ou au contraire à considérer des termes comme faisant partie du nom alors qu'ils apparaissent simplement dans le discours.

Comme cela a déjà été dit ci-dessus, les musées constituent l'un des types de ressources touristiques pour lesquels le nom est le plus explicite. En effet, le nom a vraiment pour vocation d'informer le visiteur. Il contient donc en général, une information de localisation (*Palais de la Porte Dorée*, *musée du Quai Branly*) ou une information sur le type de musée dont il est question : artiste ou thème (*Musée Magritte, Cité des Sciences et de l'Industrie, Musée des Arts et Métiers*). Pour les autres types de ressources touristiques, comme les restaurants, bars ou hôtels, le nom est souvent plus imagé, plus poétique. Il se veut souvent original, ce qui permet également de lui conférer son caractère unique. De plus, les noms des offres touristiques sont souvent constitués de créations lexicales : de mots inventés et n'existant pas dans la langue.

Les noms suivants sont des noms de bars ou restaurants parisiens :

(146) Les Marcheurs de Planète

(147) Sanz Sans

(148) **Les Abeilles**

(149) **Le Requin Chagrin**

(150) **Chez Gladines**

Une nouvelle fois, ces exemples permettent de refléter la grande diversité que les noms peuvent présenter. À part les lettres majuscules qui permettent de se douter qu'il s'agit de noms – et encore nous avons vu qu'elles sont rarement utilisées sur Internet – rien ne permet de savoir qu'il s'agit de noms de restaurants ou de bars.

Les noms de bars ou restaurants sont donc extrêmement variés et impossibles à prédire. Sans contexte, il est difficile, voire impossible, de les identifier (même pour un humain). En ce qui concerne les noms d'hôtels, ceux-ci combinent les caractéristiques des noms de musées et des noms de restaurants. En effet, ils comprennent très souvent le terme « hôtel », ou « auberge » et le reste du nom donne parfois une information sur la localisation, comme pour les musées, mais ils sont néanmoins souvent très imagés, comme pour les noms de restaurants.

(151) **Hôtel Molière**

(152) **Hôtel de Verdun**

(153) **Hôtel Restaurant Du Parc**

(154) **Hôtel Au Vieux Morvan**

(155) **Hôtel la Buissonnière**

Les noms donnés en (153) en (154) donnent une indication géographique : le premier est vraisemblablement situé près d'un parc tandis que le second est dans le Morvan. En revanche, ces indications peuvent être trompeuses, puisque l'hôtel de l'exemple (152) est situé à Nevers et non à Verdun.

Tous ces exemples permettent donc de montrer que les noms des ressources touristiques (musées, restaurants, hôtels) sont très variés. De plus, ceux-ci ne sont pas prévisibles et sont souvent difficiles à identifier. Ils peuvent se composer de mots qui n'existent pas dans la langue, ou alors, au contraire, comprendre des mots très courants qui pourraient apparaître dans n'importe quel texte. Ils peuvent également contenir des noms propres (noms de lieux, d'artistes, etc.).

1.1.2. Ressources touristiques n'ayant pas de nom propre

Comme cela a été présenté ci-dessus, les ressources touristiques ont été séparées en deux catégories. La première comprend les ressources portant un nom propre singulier, et la deuxième les ressources n'en ayant, à première vue, pas. En effet, une mairie porte-t-elle un nom ? Il est bien sûr possible d'y référer, mais ce sont d'autres procédés de références qui permettent d'identifier chaque mairie de façon unique, et non l'utilisation d'un nom propre. Il s'agit plutôt d'un usage, pour faire référence aux mairies, qui consiste à dire « la mairie de la ville de x ». Étant donné qu'il ne s'agit pas d'un nom propre prédéfini, il peut y avoir des variantes d'expressions. Ainsi, pour une même ville, on peut faire référence à la mairie de différentes manières :

(156) **Mairie de Nevers**

(157) **Mairie de la ville de Nevers**

(158) Hôtel de ville de Nevers

Les trois exemples ci-dessus peuvent tous être considérés comme des noms propres, en ce sens qu'ils font référence à un objet unique. De plus, ils comprennent un toponyme, l'un des prototypes du nom propre [Leroy 2004]. Ce qui les différencie des exemples de la catégorie précédente est qu'ils ne sont pas figés, plusieurs noms font ainsi référence au même objet. Contrairement aux objets ayant un nom propre singulier, ces noms peuvent varier et prendre plusieurs formes. Il est également possible d'en créer d'autres ayant encore le même référent.

Le cas des offices de tourisme et syndicats d'initiative est le même. On ne se sert pas du nom propre de la ressource pour y faire référence, mais d'un libellé fabriqué à l'aide du type de la ressource (*syndicat d'initiative* par exemple) et du nom de la ville.

Ces manipulations et cet usage pour faire référence à des ressources touristiques sont rendus possibles pour les mairies et offices de tourisme car ces types de ressources ont la caractéristique d'être uniques pour une ville donnée. Ainsi, sauf l'exception des grandes villes ayant une mairie par arrondissement, il n'y a qu'une mairie par ville et qu'un seul office de tourisme.

Il existe également d'autres types de ressources touristiques, qui n'ont pas de nom propre. Il s'agit par exemple de concerts, spectacles, ou autres manifestations. Prenons plus précisément l'exemple des concerts. On ne peut pas dire qu'ils aient un nom propre, mais il est facile de s'y référer en donnant le nom de l'artiste et la date du concert, ou bien en donnant, toujours le nom de l'artiste, mais cette fois avec le nom de la salle (en admettant que l'artiste n'y fasse qu'un seul concert), ou encore en donnant le nom de la salle et la date.

1.2. Règles d'acquisition des connaissances

Les règles d'acquisition des connaissances (RAC) sont les règles permettant le mapping : il s'agit d'apparier les données qui ont été annotées avec la structure de l'ontologie et donc de la base de connaissance. Les règles sont de trois types. Les premières permettent de créer, dans la base de connaissance, les instances des offres touristiques repérées dans les pages Web. Les deuxièmes permettent d'enrichir ces instances avec toutes les informations qui y sont associées (horaires, adresse, etc.). Les troisièmes permettent d'effectuer les associations entre les périodes et les offres.

Ces règles sont modélisées en langage OPAL (Ontology Population and Annotation Language) dont la grammaire est définie dans [Amardeilh 2007]. Les règles présentées ci-dessous en langage naturel ont été développées à Mondeca et figurent, en annexe, en langage OPAL (annexe E).

1.2.1. Création des instances d'offres touristiques à partir des annotations

La création des instances d'offres touristiques nécessite d'associer un « nom » à chaque offre. Les règles établies pour ce faire tiennent compte du classement effectué ci-dessus, selon lequel les offres peuvent, ou non, avoir un nom propre.

Si la ressource touristique dont il est question fait partie de la catégorie des offres qui « n'ont pas de nom propre » alors, le système doit lui associer un libellé et ce sont les règles d'acquisition qui s'en chargent, en se basant sur les informations données par les annotations ou qui se trouvent ailleurs dans la page Web.

En revanche, s'il s'agit d'une offre « ayant un nom propre », l'idéal aurait été de repérer ce

nom dans la page pour l'associer à la ressource. Toutefois, et comme cela a déjà été mentionné, si le nom propre de la ressource apparaît en général sur la page Web et y est même mis en valeur, il n'est pas pour autant facile à repérer automatiquement. En effet, les procédés permettant de le mettre en valeur font appel aux capacités cognitives de l'internaute mais ne sont que rarement linguistiques. Le développement de transducteurs pour le repérage des noms des ressources ne semble donc pas une solution possible. En effet, les marques linguistiques permettant d'indiquer le nom de la ressource sont peu fréquentes : le nom apparaît souvent seul, sans être introduit par un marqueur du type *bienvenue au restaurant x*. Le type de l'objet n'apparaît même pas nécessairement à proximité de son nom. Par ailleurs, il aurait été possible d'utiliser un système de repérage des entités nommées et de repérer ainsi les noms propres dans la page. Mais le problème qui se serait alors posé est que, dans une même page, plusieurs entités nommées auraient pu être repérées, et comment alors, savoir laquelle correspond au nom de l'objet touristique que l'on traite ?

Pour effectuer cette tâche de nommage des offres touristiques, ce sont des règles d'acquisition des connaissances qui ont été mises au point et non un traitement linguistique.

a. Services locaux

Selon mon schéma d'annotation, et l'ontologie, « Service Local » peut correspondre, entre autres, à une mairie, un office de tourisme ou un syndicat d'initiative. Il s'agit, selon la classification donnée ci-dessus, d'offres « n'ayant pas de nom propre ». Le mapping doit donc créer le libellé à associer à ce type de ressource touristique. Le choix a été fait de combiner le type d'objet et le nom de la commune pour nommer les services locaux. En effet, sauf pour certaines grandes villes ayant par exemple une mairie par arrondissement, chaque commune n'accueille en général qu'une mairie, qu'un office de tourisme et qu'un syndicat d'initiative, le libellé obtenu est donc unique. La règle établie pour instancier les services locaux est la suivante :

```
Règle : si la balise <service local> contient l'une des trois valeurs
« mairie », « office de tourisme » ou « syndicat d'initiative », le
nom de l'instance touristique est créé à partir de la valeur de la
balise <service Local> et de la valeur de la balise <Commune>.

S'il n'y a pas de balise <Commune> comprise dans la balise parente
<Ensemble>, alors s'il y a au plus une balise <Commune> dans la page
(balise faisant partie d'un ensemble dont le type est <service local>
et a la même valeur, ou balise n'ayant pas de balise parente), sa
valeur peut être concaténée à la valeur de la balise <Service Local>.

Sinon prendre la valeur de la balise <title> de la page analysée.
```

Il s'agit en réalité d'un ensemble de trois règles permettant de couvrir plusieurs cas. Ainsi, si les deux premières règles ne peuvent être appliquées alors, le contenu de la balise <title> est utilisé.

En langage OPAL, la règle correspondant à la première partie de la règle ci-dessus, pour la classe « service local », est la suivante :



Figure 61 : Règle OPAL pour les services locaux

Sans entrer dans les détails de la syntaxe du langage OPAL, voici quelques éléments permettant de comprendre cette règle [Amardeilh 2007]. Les autres règles implémentées en OPAL sont données en annexe E (règles R2, R3 et R4 pour les services locaux).

- La partie *Action* n'est exécutée que si les conditions sont vérifiées. Elle indique l'élément à instancier dans l'ontologie utilisée par l'application.
- La partie *déclenchement* contient le nom du nœud qui, lorsqu'il est rencontré, active la règle.
- La partie *conditions*, qui est optionnelle, permet de donner des conditions nécessaires au déclenchement de la règle, elles concernent la position du nœud déclencheur dans l'arbre XML.
- La partie *options* permet de compléter les informations instanciées avec la valeur générée par la règle, la position de l'information dans le document source et le niveau de confiance associé aux données obtenues.

Pour cette règle, il s'agit donc de trouver dans l'arbre XML, un nœud ayant l'étiquette <service_local>. Celui-ci doit être le fils d'un nœud <infrastructure> et avoir un nœud <ensemble> dans ses ancêtres. Cet <ensemble> doit contenir un nœud <commune>. Si ces conditions sont réunies alors la règle peut être appliquée et elle crée le nom de la ressource en concaténant la valeur du nœud <service_local> et celle du nœud <commune>.

b. Concerts

Pour les concerts, tout comme pour les musées ou les restaurants, il est impossible, dans le cadre du projet Eiffel, de repérer le nom de l'artiste, ce qui, combiné à la date, constituerait un bon libellé. En revanche, dans la page Web, en plus du type « concert », sont aussi repérées sa date et son adresse. Les cas de lieux de concerts accueillant plusieurs concerts par jour n'étant pas si fréquents, voici donc la règle qui est appliquée pour instancier les données

concernant un concert :

```
Règle : le nom d'un objet de type <événement> ayant pour valeur
« concert » est constitué de toutes les informations contenues dans
la balise <localisation> et de la date.(Annexe E, règle R5)
```

c. Hébergements et restaurants

Les hébergements et restaurants font partie des offres touristiques « ayant un nom propre », toutefois, comme cela a été mentionné ci-dessus, Adetoa ne repère pas ce nom dans la page Web. Par ailleurs, les restaurants et hôtels font souvent l'objet d'une page sans que d'autres objets y soient mentionnés. Le contenu de la balise <title> de la page Web en HTML correspond alors souvent au nom de la ressource. La règle suivante a été établie pour pouvoir associer le nom à la ressource :

```
Règle : prendre chaque balise <ensemble> comprenant les horaires et
la localisation de l'objet et créer le nom de l'instance à partir de
la balise <titre> de la page. (règle R1)

Exception : si l'objet comprend à la fois la description d'un
hébergement et d'un restaurant, alors la classe de l'instance sera
par inférence "Hotel-Restaurant".
```

1.2.2. Création des instances de périodes

Le mapping des périodes est assez simple, il correspond soit à des périodes d'ouverture soit à des périodes de fermeture. Deux règles ont donc été créées pour les instances de périodes (annexe E, règles R6 et R7). Des règles pour les propriétés des périodes existent également : date_début (règle R8), date_fin (règle R9), jour (règle R10), heure_debut (règle R11), heure_fin (règle R12), description (règle R13), exception (règle R14).

Ces règles consistent simplement, pour chaque propriété, à trouver la période d'ouverture ou de fermeture à laquelle elle correspond.

1.2.3. Création des associations reliant les périodes aux offres touristiques

La création des associations reliant les périodes aux offres touristiques dépend du liage qui a été effectué. Pour chaque période d'ouverture ou de fermeture, il s'agit de regarder si une offre a été repérée dans le même ensemble d'annotation, auquel cas, une association est créée. Ce sont les règles R15 et R16 données en annexe E qui effectuent ces associations.

2. Consolidation

L'étape de consolidation consiste à vérifier que toutes les données présentes dans les graphes RDF générés à partir de l'étape de mapping sont bien conformes à l'ontologie et peuvent être stockées. Plus les annotations ont été faites en respectant l'ontologie, plus cette étape sera donc aisée. Un module de consolidation similaire est décrit dans [Amardeilh 2007]. La consolidation comporte trois étapes : la première se subdivise en une tâche de fusionnage et une tâche de dédoublonnage et consiste à nettoyer les données. La seconde consiste à contrôler la qualité des données présentes et enfin la troisième consiste à faire des inférences pour enrichir les données.

2.1. Fusionner – dédoublonner

La tâche de fusion consiste à regrouper toutes les données concernant un même objet. Par exemple, prenons une page Web présentant un hôtel. Le type « hôtel » est répété plusieurs fois et il est donc annoté plusieurs fois. Une de ses occurrences est liée à une adresse tandis qu'une autre est liée à une période de fermeture. La fusion permettra de ne créer qu'un graphe reprenant à la fois la période de fermeture et l'adresse.

La tâche de dédoublonnage quant à elle, sert à vérifier que les données ne figurent pas déjà dans la base de connaissance. L'interrogation dans la base de connaissance peut être faite sur la base du libellé de l'objet dans un premier temps et si besoin à partir des différentes informations contenues dans le graphe et représentant l'objet en question (horaires, localisation, etc.) par une requête multi-critères. Pour chaque graphe RDF, si l'objet est déjà présent dans la base, on lui attribue la même URI[58], de façon à pouvoir simplement compléter les données existantes sans créer de nouvelle instance de l'objet. Si son libellé est différent, cela permet aussi de lui rattacher un autre libellé, alors synonyme du « préféré » appelé « alternatif » ou « alias ».

Si l'objet n'y est pas encore stocké alors, doit-il être créé ? Si oui, une URI lui est attribuée, si non, il est « abandonné » ou plutôt stocké dans un fichier temporaire, nommé le « tampon ».

Ces deux tâches permettent de simplifier les graphes RDF en ne conservant que les données à stocker, sans doublon.

2.2. Contrôle-qualité

L'étape de contrôle-qualité est chargée de vérifier la conformité des données avec l'ontologie. Pour chaque future instance, elle vérifie donc que la classe existe dans l'ontologie. Elle vérifie aussi que les propriétés sont bien des propriétés possibles pour cette classe et que la cardinalité prévue est respectée. Le contrôle-qualité s'intéresse aussi aux relations pour vérifier qu'une relation concerne au moins deux objets et que ceux-ci sont liés par le bon lien. Si ces conditions ne sont pas vérifiées, alors l'instance ne sera pas créée dans la base de connaissance mais là encore conservée dans le fichier tampon pour être ultérieurement présentée à l'utilisateur chargé de l'administration et de la qualité de la base de connaissance.

Par exemple, l'ontologie peut introduire des restrictions du type « un événement ne peut avoir qu'une seule date de début ». Si, dans l'annotation fournie par Adetoa, un événement est associé à deux dates de début alors les informations ne pourront pas être stockées telles quelles dans l'ontologie. Il en est de même pour les adresses : chaque objet touristique ne peut avoir qu'une seule adresse. En revanche, d'autres informations n'ont pas de cardinalité imposée. Par exemple, d'autres types de ressources touristiques comme les hôtels, les musées ou les mairies peuvent avoir un nombre variable de périodes d'ouverture et de fermeture.

2.3. Inférence

Les inférences qui peuvent être faites au niveau du CA Manager sont distinctes des transformations effectuées dans Adetoa au moment de l'annotation. Les inférences dont il est question ici permettent d'enrichir les données à l'aide de connaissances du domaine ou de raisonnements. Cette étape se base également sur l'ontologie dans la mesure où celle-ci peut fournir des connaissances du domaine. De plus, les données une fois enrichies par des

58 URI *Uniform Resource Identifier* – « identifiant uniforme de ressource ».

raisonnements ou inférences doivent toujours être conformes à l'ontologie pour pouvoir être ensuite stockées dans la base de connaissance.

Je présente ici deux pistes pour effectuer de telles inférences. La première, à base de règles, permet d'enrichir les données en se basant sur des horaires prototypiques. La seconde, fondée sur des méthodes possibilistes, permet de gérer les connaissances floues.

2.3.1. Enrichissement des données à l'aide de connaissances du domaine

Les « connaissances du domaine » sont les connaissances générales d'un domaine particulier, ici celui du tourisme. Celles-ci sont généralement partagées par une grande communauté de personnes. Pour le domaine du tourisme, elles sont très vastes. Il peut s'agir des informations d'ouverture ou de fermeture, par exemple, une mairie est généralement fermée le dimanche. Les informations temporelles étant au cœur du projet Eiffel, les inférences proposées ici les concernent.

Les règles que je propose sont basées sur des horaires prototypiques[59]. Ces inférences sont à manier avec précaution : lorsqu'elles sont appliquées, il faut spécifier la fiabilité de l'information comme plus basse. Un marqueur de fiabilité doit être prévu dès l'application des règles d'acquisition des connaissances et peut donc être modifié ici en cas d'inférence, de manière à pouvoir mettre en garde l'internaute. Les horaires prototypiques sont utiles mais ne sont pas universels et il existe de nombreuses offres touristiques dont les horaires d'ouverture ne correspondent pas au prototype. Néanmoins, on peut s'attendre à ce que les horaires soient précisés clairement sur la page Web lorsqu'ils ne sont pas standards et ils ne seraient alors pas traités par le moteur d'inférences. Toutefois, l'intérêt de ces inférences est certain car, en complétant les informations, elles permettraient à un plus grand nombre d'offres d'être fournies en résultat de recherche.

Je me suis intéressée plus particulièrement à un nombre limité de types d'offres touristiques : les restaurants, les musées, les hébergements et les services locaux. Voici, pour ces types d'offres, les règles à implémenter dans le module d'inférence du CA Manager pour enrichir les données fournies par Adetoa.

a. Restaurants

Si certains restaurants sont ouverts 24 heures sur 24, d'autres en journée continue, etc., des horaires prototypiques sont tout de même attribuables à un grand nombre de restaurants.

Sachant que les restaurants ont en général au moins un ou deux créneaux de fermeture dans la semaine et qu'il s'agit souvent du dimanche soir et/ou du lundi, mais qu'ils sont toutefois impossibles à prévoir, j'ai décidé de ne pas tenir compte de cette fermeture dans les horaires prototypiques que je propose. Cette décision s'appuie sur le fait que les fermetures particulières sont le plus souvent mentionnées dans les pages Web.

Le prototype suivant, pour les horaires d'ouverture des restaurants, est donc principalement prévu pour les sites dans lesquels aucune information temporelle n'est donnée :

```
Prototype : restaurants ouverts tous les jours, midi et soir.
```

Ce prototype est annoté ainsi selon le schéma d'annotation d'Adetoa :

59 « Prototypique » au sens de meilleur élément d'une catégorie, élément le plus représentatif, proposé dans [Kleiber 1990] et déjà cité au point 2.1. du chapitre 2.

```
<UT> <periode_Ouverture>
      <jour> lundi midi </jour>
      <jour> lundi soir </jour>
      <jour> mardi midi </jour>
      <jour> mardi soir </jour>
      <jour> mercredi midi </jour>
      <jour> mercredi soir </jour>
      <jour> jeudi midi </jour>
      <jour> jeudi soir </jour>
      <jour> vendredi midi</jour>
      <jour> vendredi soir</jour>
      <jour> samedi midi </jour>
      <jour> samedi soir </jour>
      <jour> dimanche midi</jour>
      <jour> dimanche soir</jour>
   </periode_Ouverture>
</UT>
```

Lorsque des informations temporelles apparaissent dans la page Web, soit aucune inférence n'est nécessaire car les informations sont suffisantes, soit les informations sont données sous forme de fermeture comme dans l'exemple suivant et le prototype peut alors être complété.

(159) Restaurant fermé le dimanche soir.

L'expression de l'exemple (159) est annotée ainsi par Adetoa :

```
<UT>
   <periode_fermeture> restaurant fermé le
      <jour> dimanche soir </jour>
   </periode_fermeture>
   <description> "Restaurant fermé le dimanche soir." </description>
</UT>
```

Pour exprimer cette expression en ouverture, et tenir compte du prototype, les deux annotations doivent être croisées et mener à l'annotation suivante :

```
<UT> <periode_Ouverture>
      <jour> lundi midi </jour>
      <jour> lundi soir </jour>
      <jour> mardi midi </jour>
      <jour> mardi soir </jour>
      <jour> mercredi midi </jour>
      <jour> mercredi soir </jour>
      <jour> jeudi midi </jour>
      <jour> jeudi soir </jour>
      <jour> vendredi midi</jour>
      <jour> vendredi soir</jour>
      <jour> samedi midi </jour>
      <jour> samedi soir </jour>
      <jour> dimanche midi</jour>
   </periode_Ouverture>
</UT>
```

La règle à appliquer pour effectuer ce croisement peut être formulée ainsi :

```
Partir du prototype :
   Pour chaque JOUR appartenant à <periode_fermeture>
      Supprimer <jour>JOUR</jour> de <periode_ouverture>
```

b. Musées

Il est difficile d'établir un prototype en termes d'horaires pour les musées. En effet, les heures d'ouverture et de fermeture sont variables, le musée peut avoir des horaires réservés aux groupes scolaires ou encore une « nocturne » un jour par semaine. En revanche, avec une granularité « jour » et des « parties de journée », il est possible de proposer le prototype suivant pour les musées[60] :

```
Musées ouverts les lundi, mercredi, jeudi, vendredi, samedi et
dimanche, le matin, le midi et l'après-midi.
```

Lorsqu'aucune information temporelle n'est fournie dans la page Web, le moteur d'inférence peut associer ces horaires au musée dont il est question. Lorsqu'une information temporelle apparaît dans la page, il faut par contre en tenir compte, et trois cas sont alors possibles. Dans le premier cas, l'information est donnée en ouverture et semble complète. Aucune inférence n'est alors à effectuer, l'annotation fournie par Adetoa est celle à conserver. Dans le deuxième cas, l'information est également donnée en ouverture mais est partielle, comme dans l'exemple suivant :

(160) Ouvert le jeudi soir.

Une information comme celle-ci n'enlève rien à l'information prototypique, il faut simplement lui ajouter le « jeudi soir ». Une règle comme la suivante peut compléter l'annotation prototypique :

```
partir de l'horaire prototypique :
   Pour chaque JOUR appartenant à <periode_ouverture> et n'appartenant
   pas à l'horaire prototypique
      Ajouter <jour>JOUR</jour> dans <periode_ouverture>
```

Enfin dans le troisième cas, l'information temporelle apparaissant dans la page est formulée en fermeture comme celle de l'exemple suivant :

(161) Fermé le jeudi.

Une telle expression modifie les horaires prototypiques à deux niveaux. Premièrement, et c'est évident, le jeudi doit être annoté comme fermé. Mais en plus, le mardi doit alors être considéré comme ouvert. En effet, si une information de fermeture est formulée sur la page, l'hypothèse peut être faite qu'elle est complète. Il n'est alors pas nécessaire de recourir aux horaires prototypiques, il suffit de convertir les données obtenues par Adetoa en une annotation en ouverture, de la même manière que le fait déjà le module de transformations d'Adetoa.

Il est ainsi possible de compléter les informations fournies dans la page Web à l'aide d'informations propres au domaine du tourisme : sauf si le contraire est mentionné dans la page, un musée est fermé le mardi. De plus, il est en général fermé la nuit, etc.

c. Hébergements

En ce qui concerne les hébergements, les horaires prototypiques dépendent du type d'hébergement dont il est question : un hôtel dans une station de ski ou un camping n'ont certainement pas les mêmes périodes d'ouverture.

60 Une distinction plus fine pourrait distinguer les musées d'art habituellement fermés le mardi et les musées scientifiques habituellement fermés le lundi.

Pour ce qui est des hôtels n'étant pas situés dans une zone ayant une saison touristique marquée, ils sont souvent ouverts toute l'année, tous les jours et à toutes les heures (l'hôtel peut toutefois avoir une fermeture annuelle). Souvent, aucune information temporelle n'est alors précisée sur la page Web. Le prototype suivant peut donc être proposé :

```
Hôtels ouverts toute l'année, tous les jours.
```

Pour certains types d'offres d'hébergements, comme les campings, une information temporelle est toutefois souvent fournie car l'offre peut ne pas être ouverte toute l'année, mais seulement par exemple à la belle saison. On peut ainsi trouver l'expression suivante :

(162) **Camping ouvert de mars à octobre.**

La règle d'inférence doit modifier les dates de début et de fin en fonction de celles données dans l'annotation d'Adetoa mais conserver les jours d'ouverture du prototype.

Le plus souvent, les jours ne sont pas pertinents pour les hébergements, sauf éventuellement pour des gîtes qui pourraient n'être ouverts que le week-end. Dans ces cas-là, l'expression repérée et annotée dans la page Web est en général assez complète et c'est celle-ci qu'il faut garder, sans tenter de la modifier à l'aide d'inférences.

d. Services locaux

Sur les sites de services locaux (mairies, offices de tourisme, syndicats d'initiative), les horaires d'ouverture sont en général clairement annoncés. Il n'est alors pas nécessaire d'avoir recours aux connaissances du domaine.

Néanmoins, si ceux-ci ne sont pas mentionnés, les horaires prototypiques suivants peuvent être appliqués :

```
Services locaux ouverts du lundi au vendredi le matin et l'après-midi
et le samedi matin.
```

2.3.2. Inférences à l'aide d'une modélisation possibiliste

Les expressions temporelles repérées dans les pages Web touristiques suscitent d'autres types d'inférences de la part des internautes qui les consultent. Par exemple, pour l'expression suivante, l'internaute va naturellement évaluer les chances que le camping soit ouvert à la période où il pense y aller :

(163) **Fermeture du camping début octobre.**

Ainsi, plus le mois d'octobre sera avancé, plus les chances de fermeture seront grandes. À l'inverse, les derniers jours de septembre sont incertains, mais pas exclus, tout comme les premiers jours d'octobre. En effet, l'interprétation de cette expression, qui peut être utilisée plusieurs années de suite, peut dépendre de plusieurs critères. Par exemple, le camping peut décider de rester ouvert jusqu'au 15 octobre si la météo est excellente. Le jour précis de fermeture peut aussi être décidé en fonction du jour de la semaine : inutile de rester ouvert jusqu'à un jeudi si la plus forte affluence est prévue pour le week-end, etc. Ainsi, la date de fermeture peut varier.

Nous avons, dans [Fortin et al. 2009], proposé une modélisation possibiliste de ce genre d'expressions. Ainsi, au lieu de fixer arbitrairement *début octobre* au « 1er octobre », on peut modéliser l'ouverture de l'établissement par les distributions de possibilités.

« Le principe est d'affecter à une ressource pour chaque date, un degré de possibilité

> d'ouverture μO et un degré de possibilité de fermeture μF , dont les valeurs peuvent aller de 0 à 1 selon qu'il est respectivement impossible ou possible que la ressource touristique soit ouverte ou fermée. » ([Fortin et al. 2009])

Pour cette expression, les possibilités d'ouverture diminuent dès la fin du mois de septembre : du 25 au 30 septembre, il est de moins en moins pensable que l'établissement soit ouvert, le degré de possibilité d'ouverture diminue donc progressivement de 1 à 0 entre le 25 et le 30 septembre. Inversement, du 1er au 15 octobre, il est de plus en plus pensable que l'établissement soit fermé.

Dans cet article, nous proposons également une modélisation des expressions ambiguës comme celles présentées au chapitre 2 (exemple 67).

> « Soit l'expression *Ouvert les après-midi sauf lundi et mardi*. Dans cette expression, il est certain que l'établissement est ouvert les après-midi du mercredi au dimanche (degrés de possibilité d'ouverture=1 et de fermeture=0). On est sûr qu'il est fermé les lundi après-midi et mardi après-midi (degrés de possibilité d'ouverture=0 et de fermeture=1), et l'ouverture est incertaine pour le reste du temps puisqu'assujettie à l'interprétation de l'expression. On peut modéliser cette incertitude en affectant un degré de possibilité d'ouverture 0,5 aux matinées (avec un degré de possibilité de fermeture de 1). » ([Fortin et al. 2009])

Une approche comme celle-ci a le mérite de se rapprocher un peu plus des phénomènes en jeu dans la langue et plus précisément dans l'élaboration de ce genre d'expressions. Elle permet d'éviter, dans une certaine mesure, les décisions arbitraires parfois nécessaires en informatique mais qui ne reflètent pas le système d'interprétation du langage.

3. Stockage

Le stockage dans la base de connaissance concerne les données qui ont été consolidées à l'étape précédente. Il s'agit donc de créer de nouvelles instances ou de compléter les instances qui existent déjà dans la base.

3.1. Saturation ?

Les données étant, après la phase de consolidation, conformes à l'ontologie, le stockage dans la base ne devrait pas poser de problème. Néanmoins, certaines questions se posent quant à la manière de stocker les informations. Une première décision a été prise, au sein du projet Eiffel, qui consiste à toujours stocker l'information telle qu'elle a été repérée dans la page Web (contenu de la balise <description>). D'un autre côté, pour être exploitables et surtout interrogeables, les informations doivent être stockées d'une certaine manière. Par exemple, en ce qui concerne les dates, sous quel format doivent-elles être entrées ? Pour les périodes, doit-on simplement stocker les dates de début et les dates de fin, ou doit-on les développer ?

Par exemple, pour *ouvert du 20 au 27 janvier*, il est possible de stocker simplement la date de début et la date de fin, sans faire de saturation, ou au contraire de stocker en tant qu'ouverture chacune des dates comprises dans cette période. Pour reprendre le vocabulaire des mathématiques, cette action de saturation concerne les expressions qui sont « factorisées » et peuvent donc être « distribuées ». Il s'agit principalement des intervalles et des groupes de jours, données itératives.

Deux approches sont donc possibles selon que la base doit être saturée ou non. La première

consiste à saturer la base, c'est-à-dire que les informations à stocker sont développées au maximum, mais un calcul doit donc avoir lieu. La seconde consiste à l'inverse à ne pas la saturer en y entrant le moins d'informations possible, c'est-à-dire en gardant des informations factorisées ou des périodes, sans faire de calcul.

La principale différence entre ces deux approches réside dans le moment auquel les calculs doivent être effectués : au préalable, avant le stockage dans la base pour l'approche avec saturation ou à la volée au moment de l'interrogation par l'internaute pour l'approche sans saturation. Les implications de ce choix dépendent du type d'expression dont il est question ; la portée de la saturation peut en effet varier beaucoup selon que la période couverte est longue ou courte.

3.1.1. Intervalles

Par « intervalle », j'entends ici les informations d'ouverture données sous forme de date de début et de date de fin, pour former une période de plusieurs jours ou mois d'ouverture, comme dans les exemples suivants :

(164) Ouvert du 15 mai au 31 juillet.

(165) Ouvert de juin à septembre.

Si la saturation n'est pas prévue, l'opération consiste alors à stocker dans la base (qui prévoit ces champs) la date de début et la date de fin, telles qu'elles sont annotées par Adetoa. Un calcul ultérieur sera alors nécessaire pour indiquer, lors de l'interrogation, si l'objet est ouvert par exemple le 15 juin.

Avec saturation de la base, au contraire, le calcul doit être effectué avant le stockage, afin de convertir l'expression en une suite de dates pouvant être stockées de manière calendaire.

Dans le premier cas, la réponse à l'interrogation de l'utilisateur peut être ralentie si la période couverte est longue et que le calcul prend alors du temps. Dans le second, c'est une question de place de stockage. Si la période concernée est longue un grand nombre de dates est alors à stocker. De plus, si aucun utilisateur interroge les périodes d'ouverture de l'objet, ces calculs auront été faits inutilement.

3.1.2. Groupes de jours

Je reprends ici les groupes de jours déjà présentés au chapitre précédent. Comme nous l'avons vu, ceux-ci font déjà l'objet d'une transformation au niveau d'Adetoa et sont ainsi convertis en énumération de jours.

(166) Ouvert du lundi au vendredi.

Ainsi, pour l'exemple ci-dessus, une fois converti en un ensemble de jours d'ouverture, la question de la saturation se pose et concerne son caractère itératif. Deux possibilités sont en effet de nouveau possibles, selon que l'on souhaite ou non, saturer la base. On peut rentrer les informations pour « une semaine type » sans date calendaire ou bien renseigner les jours d'ouverture pour toutes les semaines du calendrier et ainsi saturer la base. Pour la première option, il faut un moyen de préciser, dans la base, le caractère itératif de l'information qui est alors valable pour toutes les semaines de l'année, sauf indication contraire.

Le problème qui se pose, et qui distingue ce cas du cas précédent, est qu'il n'y pas de limite dans le temps. Il faut donc fixer une limite arbitrairement. L'idée d'une année glissante à

partir du moment où le calcul est effectué semble être la plus réaliste : l'information couvre une année entière et pas de risque de se voir donner une réponse concernant uniquement l'année en cours si on est déjà en décembre par exemple.

D'autres expressions itératives existent, sans être du type « groupe de jours » :

(167) **Ouvert les dimanches.**

Une expression comme celle-ci se rapproche beaucoup de l'expression précédente, en ce qui concerne le calcul et le stockage. En effet, l'itérativité est la même puisqu'elle concerne les jours de la semaine, la seule différence est qu'il n'y a ici qu'un jour.

3.1.3. Solutions adoptées dans Eiffel

Les expressions totalement itératives, comme celle de l'exemple (167) sont calculées à la volée au moment de l'interrogation du système par un utilisateur.

Les expressions calendaires, même sous forme d'intervalles, comme celles des exemples (164) et (165) sont stockées dans la base (saturation).

Les expressions mixtes, comme celle de l'exemple suivant, sont également stockées dans la base.

(168) **Ouvert le dimanche en juillet et août.**

Ainsi, les expressions calendaires sont interprétées afin que chaque date puisse être stockée dans la base tandis que les expressions itératives sont marquées comme itératives et c'est à la demande de l'utilisateur que le calcul est effectué pour vérifier si l'objet est ouvert ou fermé au moment voulu.

Ces calculs peuvent être effectués par le moteur de raisonnement et d'inférences du CA Manager. Ce moteur prépare donc les informations à être stockées.

3.2. Sérialisation

La dernière étape effectuée par le CA Manager est la sérialisation, qui consiste à mettre les données au format accepté par la base de connaissance et à les y stocker. Ainsi, selon les besoins le format peut être différent. Dans le cadre d'Eiffel, le format obtenu après la sérialisation est du XTM, format propre à la base ITM[61].

Ce module permet aussi de générer des données au format RDF ou au format XML, ce qui peut notamment être utile lors de phases de tests.

4. Interface

Dans cette partie seront présentées brièvement les quelques facettes de l'interface utilisateur du projet Eiffel qui se rapprochent le plus de mon travail. Il s'agit de celles qui concernent les recherches en fonction de critères temporels et de la façon dont les informations sont transmises à l'utilisateur.

61 ITM - *Intelligent Topic Manager*, solution développée à Mondeca pour la gestion de connaissance. http://www.mondeca.com/index.php/fr/technology/technical_specifications

4.1. Critères de recherche

Les interfaces qui découlent du projet Eiffel comportent plusieurs facettes, permettant aux utilisateurs de visualiser les résultats selon différents critères et selon différents points de vue. La structure de l'ontologie est pour cela exploitée. Ainsi, un utilisateur prévoyant un voyage peut par exemple rechercher les musées ouverts pendant ses vacances dans la ville de son choix. Le système sélectionne alors, pour la période donnée, les objets qui sont ouverts pendant toute la période, mais aussi les objets qui sont ouverts seulement une partie du temps de cette période.

L'utilisateur peut également faire des requêtes en multipliant les critères : le prix, l'accessibilité (famille, enfants, animaux), la proximité d'autres lieux, etc. Le système lui présente alors les résultats sous la forme de vignettes qu'il peut sélectionner pour avoir toutes les informations stockées dans la base. Selon la recherche effectuée, ces vignettes sont organisées par type d'objet ou selon les critères sélectionnés. L'interface intègre aussi un système de suggestions.

En ce qui concerne les expressions temporelles stockées dans la base et que l'utilisateur peut utiliser comme critère de recherche, deux modes de recherche sont proposés. L'utilisateur peut choisir sur un semainier les jours qui l'intéressent et alors dans la base sont consultées les informations itératives. Il peut aussi choisir des dates calendaires sur un calendrier et alors seulement celles-ci sont considérées. Un problème peut néanmoins se poser : si l'utilisateur cherche des informations sur des objets ouverts le mardi, par exemple, le système doit aussi consulter si dans les dates renseignées, il y a des mardi. La solution est de consulter les informations itératives du type « ouvert le mardi » et les dates des trois ou quatre prochains mardi : cela évite de tomber sur « ouvert le mardi en août » quand on est en janvier.

4.2. Degré de certitude et donnée textuelle

Une indication stockée dans la base de connaissance concerne la fiabilité des informations. En effet, une information sur laquelle un calcul a été effectué devient nécessairement moins fiable qu'une information qui n'a pas été transformée depuis son repérage dans la page Web. Il est donc important d'attribuer un degré de certitude à chaque information, de façon à pouvoir l'indiquer à l'utilisateur.

Le degré de certitude baisse selon le nombre et le type de traitements que l'information aura subis. Ainsi, l'information la plus fiable est celle qui n'aura subi aucun traitement autre que son repérage et son annotation. Les informations qui ont nécessité une transformation et celles qui ont fait l'objet d'un calcul calendaire au moment du stockage ou de l'interrogation de la base viennent ensuite.

Le degré de fiabilité qui est ainsi stocké dans la base est indiqué à l'utilisateur. De plus, l'expression telle qu'elle a été repérée dans la page Web est également fournie à l'utilisateur. Celui-ci peut ainsi apprécier la fiabilité globale de l'information et, en cas de doute ou d'ambigüité, il peut chercher à la vérifier par lui-même.

Conclusion

Ce chapitre a permis de montrer comment mon travail peut être intégré dans une chaîne de traitement plus globale et comment il l'a été, pour le projet Eiffel, dans le CA Manager. La première étape, le mapping, sert à transformer les annotations XML en un format exploitable par le CA Manager. Pour ce faire, un libellé « préféré » doit être associé aux

données annotées qui vont ensuite être stockées dans la base de connaissance. Dans la base du projet Eiffel, cet identifiant étant un nom, j'ai étudié, la façon dont les objets touristiques sont nommés dans les pages Web. Sans que cela soit le sujet central de ce chapitre, j'ai abordé les problèmes posés par la définition du nom propre qui pourraient être approfondis et éventuellement mener à un système de repérage d'entités nommées spécialisées dans le domaine du tourisme. Les règles de mapping qui ont été créées avec Mondeca pour attribuer cet identifiant et transformer les données en graphes sont ensuite détaillées.

J'ai ensuite présenté l'étape de consolidation des données en insistant plus particulièrement sur le moteur d'inférences qui peut enrichir les données à l'aide de connaissances du domaine. Des questions théoriques sont également soulevées, concernant des informations temporelles prototypiques.

Enfin, j'ai présenté des considérations plus techniques pour le stockage dans la base et son interrogation : faut-il saturer la base en développant toutes les informations temporelles ou ne rentrer, par exemple, que les dates de début et les dates de fin ainsi que des indications de périodicité ? L'interface utilisateur est également présentée sommairement : quels critères de recherche sont proposés, et quelles informations sont fournies.

Chapitre 7. Expérimentation et Évaluation

Introduction

Le grand nombre de publications traitant des méthodes d'évaluation en TAL montre l'importance de ce sujet. Un numéro de la revue TAL [Paroubek et al. 2007] est entièrement dédié aux problématiques liées à l'évaluation des systèmes de TAL. La conférence LREC[62] s'intéresse également aux techniques d'évaluation. Les conférences MUC[63] ont de plus mené à des campagnes d'évaluation spécialisées dans l'extraction d'information.

Des méthodes d'évaluation sont utilisées dans la plupart des disciplines du TAL (extraction d'information, traitement de la parole, traduction automatique, etc.). Néanmoins elles sont souvent sujettes à controverse, comme le mentionne [Popescu-Belis 2007] :

> « Alors que certains prennent position contre l'évaluation et ses effets sur le monde de la recherche, d'autres déplorent les limites des évaluations actuelles et souhaitent leur généralisation et leur perfectionnement. »

Cet auteur montre bien l'importance de l'évaluation dans les systèmes de TAL et le rôle qu'elle doit jouer. Il mentionne également qu'il est aussi possible d'évaluer la vitesse, l'interopérabilité (des annotations produites, de l'ontologie, ou même du système en lui-même), la facilité d'installation et d'utilisation, ou encore la réutilisabilité du système, comme cela est prévu dans les normes ISO/IEC[64], habituellement utilisées pour évaluer les logiciels informatiques[65]. Toutefois, ces normes ne semblent pas répondre aux attentes des concepteurs de systèmes de TAL qui souhaitent plutôt évaluer la qualité des résultats obtenus à un niveau linguistique. Si des métriques existent pour remplir ces objectifs, aucune n'est encore universellement acceptée et c'est là que réside l'une des difficultés de l'évaluation des systèmes de TAL.

Ce chapitre s'intéresse donc à l'évaluation d'Adetoa, dans le but de vérifier si le système

62 Language Resources and Evaluation Conference - http://www.lrec-conf.org/

63 Message Understanding Conferences.

64 http://www.standardsinfo.net

65 Il existe entre autres la norme SquaRE : Software product Quality Requirements and Evaluation (spécifications et évaluation de la qualité des logiciels).

fonctionne effectivement comme il devrait fonctionner. Après une revue des méthodes d'évaluation les plus fréquemment utilisées pour ce type de systèmes, il présente l'expérimentation que j'ai menée, le protocole d'évaluation que j'ai mis au point et les résultats qui en découlent.

1. Principales méthodes d'évaluation

Les mesures d'évaluation les plus fréquemment utilisées pour les systèmes de TAL sont le rappel et la précision. Toutefois, comme je vais le montrer ci-dessous, ces mesures ne suffisent pas toujours à refléter la qualité du système et ne sont pas toujours simples à mettre en œuvre.

1.1. Rappel et précision

Les mesures de rappel et précision sont issues du domaine de la recherche d'information et se sont largement répandues dans l'évaluation des systèmes d'extraction d'information. [Maynard et al. 2002] montrent en quoi l'évaluation des systèmes d'extraction est différente de celle des systèmes de recherche d'information.

De manière générale, le rappel mesure le silence du système, c'est-à-dire les informations pertinentes qui n'ont pas été trouvées, tandis que la précision en mesure le bruit, c'est-à-dire les informations non pertinentes trouvées. Les formules[66] présentées dans les figures suivantes indiquent comment calculer ces mesures :

$$\text{Rappel} = \frac{\text{Nombre d'informations bien repérées}}{\text{Nombre d'informations à repérer}}$$

Figure 62 : Calcul du rappel

$$\text{Précision} = \frac{\text{Nombre d'informations bien repérées}}{\text{Nombre d'informations repérées}}$$

Figure 63 : Calcul de la précision

Plus le taux de rappel est haut, moins il y a de silence, plus la précision est élevée, moins il y a de bruit.

Une troisième mesure, la F-mesure, permet de combiner le rappel et la précision pour n'attribuer qu'une seule note d'évaluation au système. Elle se calcule selon la formule donnée en figure 64.

66 Ces formules sont présentées ici pour une activité de repérage d'information. Les mêmes s'appliquent pour d'autres tâches comme l'annotation, le typage, etc. Il suffit alors de changer, dans la formule « repérées » et « à repérer » par les termes correspondants.

$$F_\alpha = \frac{(1+\alpha^2) * \text{précision} * \text{rappel}}{(\alpha^2 * \text{précision}) + \text{rappel}}$$

Figure 64 : Calcul de la F-mesure – cas général

$$F_1 = \frac{2 * \text{précision} * \text{rappel}}{\text{précision} + \text{rappel}}$$

Figure 65 : Calcul de la F-mesure - F_1

Le α (figure 64) permet de pondérer cette mesure en donnant plus ou moins de poids à la précision, et au rappel. Dans la plupart des cas, il est fixé à 1 pour que les deux aient la même importance (figure 65).

Parmi les travaux cités dans les chapitres précédents, et plus généralement dans les travaux d'extraction d'information, ces mesures sont celles utilisées pour l'évaluation [Wang & Hu 2002], [Tengli et al. 2004], [Amardeilh 2007]. Ce sont également les mesures utilisées dans le cadre des conférences MUC.

Ces taux de rappel et précision sont habituellement utilisés pour juger une application dans son ensemble. Toutefois, certains systèmes comportent plusieurs modules effectuant chacun une tâche précise et il n'est alors pas toujours pertinent d'évaluer le système dans son ensemble. De nombreux auteurs ont ainsi choisi d'évaluer indépendamment les différentes étapes de leurs systèmes : [Gatterbauer et al. 2007] évaluent d'une part le repérage des tableaux et d'autre part l'interprétation de ces tableaux, ces deux évaluations étant mesurées par le rappel et la précision. [Nagy et al. 2009] font de même en évaluant d'un côté le repérage, de l'autre le typage. [Hong et al. 2009] évaluent le repérage et le parsing de références bibliographiques, mais l'évaluation du parsing ne se fait que sur un extrait du corpus d'évaluation du repérage. Adetoa étant composé de plusieurs « briques » en charge des différentes étapes du processus (repérage, annotation, transformations, liage), je me suis plutôt rapprochée de ces derniers travaux.

Toutefois, les mesures de rappel et précision ne sont pas parfaites. Tout d'abord, elles ne sont pas toujours significatives, selon le corpus sur lequel le système est évalué : certains corpus ne comportent pas (ou presque pas) d'informations non pertinentes qui auraient pu être repérées à tort. La mesure de précision n'a alors pas réellement de valeur. Certains auteurs ne se basent que sur l'une des deux mesures pour évaluer leur système ; [Jacques & Aussenac-Gilles 2006] par exemple, n'évaluent que la précision, mais sur différents corpus.

De plus, et c'est surtout en cela que ces mesures ne me semblent pas tout à fait appropriées, elles ne permettent pas de rendre compte de résultats imparfaits. Ainsi, une donnée repérée est considérée comme bonne ou mauvaise ; elle ne peut pas être partiellement bonne.

1.2. Autres mesures

L'un des reproches que l'on peut donc faire aux mesures de rappel et de précision, est qu'elles ne permettent pas de rendre compte de résultats imparfaits. [Lavelli et al. 2004] se sont intéressés à cette question :

> « One issue specific to IE evaluation is how leniently to assess inexact identification of filler boundaries. (Freitag 1998) proposes three different criteria for matching reference instances and extracted instances: exact, overlap, contains. »

> L'une des difficultés propres à l'évaluation des systèmes d'extraction d'information est liée à la souplesse à avoir face aux expressions dont les frontières sont mal identifiées. (Freitag 1998) propose trois critères pour comparer les résultats voulus aux résultats obtenus : exact, contenu, imbriqué. [Ma traduction]

L'approche de [Freitag 1998] permet ainsi de juger des résultats partiels, tout en les catégorisant : les résultats peuvent être exacts, ou bien le résultat attendu peut être contenu dans le résultat obtenu, ou encore les deux peuvent se chevaucher. Ce type d'approche est utile pour l'évaluation des tâches de repérage d'information où le problème se pose très fréquemment : si seule une partie de l'expression est repérée, il est dommage de considérer ce repérage comme totalement mauvais, surtout que la partie manquante peut parfois être minime. [De Sitter & Daelemans 2003] évaluent leur système en calculant plusieurs taux de rappel, selon que les résultats partiels sont pris en compte ou non. C'est de ce type d'approche que je me suis inspirée pour mettre au point mon protocole d'évaluation, présenté ci-dessous.

Par ailleurs, lors de résultats partiels, il est dommage de ne pas prendre en compte la nature de l'erreur : peut-on considérer comme équivalents les cas où, pour une expression repérée partiellement, la partie qui manque est primordiale pour la compréhension globale, des cas où elle ne l'est pas ? De plus, comment évaluer les résultats qui sont manqués ou mal considérés par le système lorsque le problème vient de la page de base : par exemple des fautes d'orthographe ou de syntaxe empêchent le repérage. La mise au point d'un système d'évaluation permettant de rendre compte de tous ces critères semble difficile. Néanmoins, ce sont des problèmes très fréquents en TAL et il serait dommage de les ignorer totalement.

Dans [Maynard 2005], l'auteur s'est intéressée aux problèmes que pose l'évaluation des systèmes basés sur des ontologies. Elle signale que les mesures de rappel et de précision ne sont pas adaptées et propose la mesure « augmented precision and recall » (précision et rappel augmentés). Cette mesure prend directement en compte la structure de l'ontologie liée au système qu'elle évalue et ne peut donc pas être étendue à des tâches autres que l'annotation.

[Maynard et al. 2002] montrent pourquoi le rappel et la précision, utilisés en recherche d'information, ne sont pas appropriés pour l'extraction d'information.

> « Typically, in IR, people want to know how many relevant documents are to be found in the top N percent of the ranking. This is reflected well by the precision metric. In IE, however, people typically want to know for each entity type how many entities have been correctly recognised and classified. In IE therefore, the proportion of entities belonging to each type has an impact on the outcome of the evaluation, in a way that the proportion of relevant documents in the collections does not in IR. Evaluation mechanisms in IE can also be affected by the notion of *relative document richness*, i.e. the relative number of entities of each type to be found in a set of documents. For this reason, error rate is sometimes preferred in the IE field, because, unlike precision, it is not dependant on relative richness. »
>
> Traditionnellement, en recherche d'information, le but est de connaître le nombre de documents pertinents dans les N premiers pour-cent du classement, ce que reflète bien la mesure de la précision. En revanche, en extraction d'information, le but est de connaître, pour chaque type d'élément, combien ont été bien reconnus et classés. Par conséquence, en extraction d'information, la proportion d'éléments de chaque type a un impact sur le résultat de l'évaluation, d'une manière différente de la proportion de documents pertinents en recherche d'information. Les mécanismes d'évaluation en extraction d'information peuvent aussi être influencés par la notion de *richesse relative du*

> *document*, c'est-à-dire le nombre relatif d'éléments de chaque type dans un ensemble de documents. Ainsi, le taux d'erreur est parfois préféré dans le domaine de l'extraction d'information, car, contrairement à la précision, il ne dépend pas de la richesse relative. [Ma traduction]

Ces auteurs présentent alors une autre méthode : « cost-based evaluation », évaluation basée sur le coût. Ce type d'évaluation, utilisé dans le cadre d'ACE[67], caractérise le coût des erreurs : par exemple, si le repérage d'un nom de personne est plus important que de trouver un événement alors, le coût associé à cette donnée est plus grand. L'attribution des coûts aux erreurs se fait sur base statistique.

Les questions que pose l'évaluation des systèmes de TAL sont donc variées et difficiles. S'il n'existe pas réellement de mesure standard pour l'évaluation des systèmes d'extraction et d'annotation, les mesures de rappel et de précision sont tout de même les plus communément utilisées. Elles ne permettent pas de refléter toutes les facettes d'un système mais donnent toutefois une indication de sa qualité. J'ai basé mon évaluation sur ces mesures mais en introduisant des variantes, qui seront présentées dans la suite de ce chapitre, afin de rendre compte de résultats imparfaits.

2. Protocole d'évaluation

Adetoa constitue l'une des « briques » du projet Eiffel. L'évaluation pourrait donc concerner l'ensemble du projet et faire intervenir l'utilisateur final. Toutefois, une telle évaluation ne permettrait pas de déterminer en détail la qualité d'Adetoa. En effet, la qualité du projet dans son ensemble dépend de toutes les étapes, de l'aspiration des pages Web par l'outil de crawling à l'interface permettant à l'utilisateur final d'interroger la base de connaissance. L'évaluer permettrait donc de rendre compte des interactions entre les différents modules du projet. Néanmoins, si une information erronée parvenait à l'utilisateur final, il serait alors difficile de déterminer à quel niveau l'erreur s'est produite. Une méthodologie spécifique serait alors nécessaire afin de coordonner l'évaluation sur l'ensemble des modules.

Mes responsabilités étant liées au développement d'Adetoa, j'ai choisi d'évaluer en détail ses performances propres, ce qui ne remplace ni n'exclue la possibilité d'une évaluation plus globale, menée au niveau du projet.

Par conséquent, il n'est pas possible d'évaluer Adetoa par l'interface utilisateur. J'ai donc dû mettre au point un protocole d'évaluation strict et faire appel à un évaluateur extérieur. Adetoa a ainsi été évalué de manière indépendante mais tout en gardant la vision globale du projet : ses résultats ont été évalués en fonction des besoins du projet Eiffel.

Afin d'être le plus impartial possible, le protocole d'évaluation commence dès le choix des pages sur lesquelles Adetoa doit être testé. Je présente donc dans un premier temps comment le corpus de test a été constitué. Je détaille ensuite précisément ce qui a été évalué et comment cette évaluation a été menée, selon quels critères.

67 *Automatic Content Extraction* – Campagnes d'évaluation en extraction automatique d'information, depuis 1999.

2.1. Mise au point du corpus

Pour constituer le corpus sur lequel évaluer Adetoa, et afin de rendre l'évaluation la plus significative possible, j'ai mis au point et respecté plusieurs critères.

Tout d'abord, Adetoa étant amené à fonctionner sur des pages Web crawlées par Antidot, c'est sur ces mêmes pages qu'il doit être évalué. Les pages sur lesquelles l'évaluation a eu lieu font donc partie du corpus que m'a fourni Antidot. Une fonction aléatoire m'a permis de sélectionner des pages pour constituer le corpus d'évaluation.

Néanmoins, un certain nombre des pages tirées ne devraient pas faire partie du corpus d'évaluation, comme des pages non touristiques. La présence de ces pages est due à l'outil de crawling et Adetoa n'est pas conçu pour les traiter correctement. C'est pourquoi j'ai choisi de les supprimer manuellement du corpus d'évaluation.

Les pages qui ont été enlevées sont de plusieurs natures :

- Pages vides : pages ne contenant pas, ou presque pas (moins d'une dizaine de mots) de contenu textuel. Résultat par exemple d'une page Web exclusivement constituée d'images.
- Pages qui ne sont pas en français : un filtre de langue est appliqué lors du crawl des pages mais des pages multilingues ou contenant peu de texte peuvent y échapper.
- Pages non touristiques : le but de l'outil de crawling d'Antidot est de ne sélectionner que des pages touristiques mais des pages non touristiques sont parfois mal classifiées.

Ces trois critères permettent d'exclure les pages dont la présence résulte de quelques défauts de l'outil de crawling.

Par ailleurs, et comme je l'ai montré au chapitre 3, certains types de pages ne sont finalement pas pris en compte dans Adetoa, car ils nécessiteraient un traitement trop lourd. Ces pages ont également été supprimées car elles ne font donc pas partie du champ défini ; il s'agit des pages suivantes :

- Pages-agenda : ces pages contiennent souvent un grand nombre de dates et leur présence dans le corpus d'évaluation générerait beaucoup d'erreurs de repérage.
- Pages avec tableaux structurants : les informations présentes dans ce type de tableaux ne sont pas interprétées par Adetoa et généreraient des erreurs de repérage et d'annotation.

De plus, les pages ne contenant aucune information pratique ont également été retirées du corpus d'évaluation car elles alourdiraient le travail de l'évaluateur « à la recherche de l'information manquée » et ne reflètent en rien la qualité du système.

Ces filtres ont été appliqués aux pages tirées aléatoirement jusqu'à ce qu'un corpus d'évaluation de 100 pages soit obtenu. Adetoa a été conçu pour traiter ce type de pages et ce corpus permet donc de refléter sa qualité.

2.2. Quoi évaluer ?

Plutôt que d'évaluer Adetoa comme un « tout » ayant un résultat unique, j'ai choisi d'évaluer chaque tâche indépendamment. Adetoa comprend quatre tâches principales pour le

traitement des pages :

- Le repérage – consiste à bien détecter, dans les pages Web, les informations voulues.
- L'annotation – correspond au typage des informations repérées.
- Les transformations – concernent certaines expressions temporelles qu'elles modifient.
- Le liage – consiste à bien grouper les informations concernant un même objet.

Toutes les tâches dépendent du repérage puisque seules les informations repérées sont annotées et peuvent faire l'objet d'un liage. Les transformations dépendent de l'annotation. C'est pour cela qu'il est utile d'évaluer les différentes tâches de façon indépendante. Ainsi, les erreurs d'une des tâches ne se propagent pas dans l'évaluation des autres.

Si le repérage et l'annotation sont tout de même très proches puisque ce sont les mêmes transducteurs qui effectuent les deux tâches, l'annotation peut être modifiée sans modifier le repérage et les graphes de repérage peuvent être enrichis sans influencer l'annotation. De plus, le module de transformations et celui de liage sont indépendants (voir le chapitre 5 pour plus de détails sur l'architecture générale d'Adetoa).

Par ailleurs, j'ai fait le choix de n'évaluer les tâches de repérage et d'annotation que sur les expressions temporelles et les expressions de localisation et pas sur les objets touristiques. En effet, les transducteurs d'objets touristiques sont basés sur des données lexicales et permettent de repérer de simples termes et de les annoter de façon directe en fonction de la façon dont ils sont modélisés dans l'ontologie. Aucune expression plus complexe n'est repérée ; la tâche de repérage et celle d'annotation ne sont donc pas très intéressantes. De plus, le traitement des objets est tout de même évalué indirectement au sein de la tâche de liage. Les transformations, quant à elles, ne concernent que les données temporelles d'un certain type (voir chapitre 5) ; elles ne seront donc évaluées que sur les informations de ce type.

Les parties suivantes présentent plus en détail la méthodologie d'évaluation et les critères permettant de juger chacune de ces tâches. Rappelons que le but est d'évaluer la qualité des informations fournies par le système par rapport à leur utilité dans le cadre d'Eiffel.

2.3. Méthodologie d'évaluation

La méthode la plus répandue pour évaluer des systèmes de repérage ou d'annotation se base sur un corpus annoté manuellement par un, ou mieux plusieurs, annotateurs humains (cela est fait par exemple dans [Handschuh 2005]). Ainsi, les annotateurs annotent un ensemble de documents. Leurs annotations sont comparées et en cas de différence, soit ils se mettent d'accord et choisissent la meilleure annotation, soit un calcul de distance est effectué. Ces mêmes pages sont données au système à évaluer et les résultats peuvent ainsi être comparés avec les résultats « humains » : chaque résultat peut être considéré bon (si identique) ou mauvais (si différent), ou bien, un calcul de distance peut être effectué et le résultat considéré bon si la distance ne dépasse pas un seuil fixé.

Je n'ai pas suivi une telle approche pour l'évaluation d'Adetoa et n'ai donc pas constitué de corpus de référence annoté manuellement. Plusieurs raisons sont en cause. Tout d'abord, les transducteurs permettant d'effectuer le repérage sont d'une grande complexité et le nombre

de cas possibles est très grand ; il est donc difficile d'apprendre à un humain à effectuer cette tâche (dans un temps raisonnable et pour un nombre suffisant de pages). J'ai donc choisi de fournir à un évaluateur[68] des pages annotées automatiquement et de lui demander de juger les différentes tâches (repérage, annotation, liage, transformations). L'évaluateur doit juger aussi bien ce qui a été annoté que ce qui ne l'a pas été, de façon à pouvoir mesurer le rappel.

Si de nombreux travaux se basent sur un corpus de référence annoté par des humains, c'est que cela permet ensuite de comparer les résultats obtenus automatiquement aux résultats manuels, sans que l'évaluateur puisse être influencé par la solution proposée par le système. Toutefois, dans le cas d'Adetoa, il me semble que l'évaluateur a la capacité de juger les résultats fournis par le système de manière impartiale. En effet, il peut pour ce faire se mettre à la place d'une personne utilisant le système et juger les informations fournies selon les critères présentés dans ce chapitre. Le fait que le domaine est celui du tourisme et que les informations traitées sont des informations pratiques facilite le travail de l'évaluateur.

Par exemple, en ce qui concerne le repérage, une expression peut être repérée en entier ou partiellement. D'une part l'évaluateur doit donc juger si les frontières de l'expression sont les bonnes. D'autre part, en cas de repérage partiel, il doit juger si la partie manquante entraîne ou non une perte d'information. Pour une expression temporelle, il peut effectuer ce jugement en cherchant s'il existe des dates, jours ou heures précis, pour lesquels la présence ou l'absence du segment en question change son interprétation.

Pour résumer, l'évaluateur peut se demander si les expressions annotées lui donnent bien toutes les informations pratiques qui concernent l'objet qui l'intéresse. Si tel n'est pas le cas, il doit alors se demander si l'information est ailleurs dans la page et n'a pas été trouvée par le système ou si elle est simplement absente de la page d'origine.

Un évaluateur a donc été chargé de juger chacune des tâches effectuées par Adetoa. Son but est de juger la qualité du système selon l'utilité des informations qu'il fournit. Chaque tâche étant évaluée individuellement et ayant des caractéristiques propres, les critères d'évaluation dépendent également de la tâche. Les critères d'évaluation, pour chaque tâche, sont donnés ci-dessous.

Le guide d'évaluation détaillant la méthodologie d'évaluation sur lequel s'est basé l'évaluateur est donné en annexe F.

2.3.1. Comment évaluer le repérage ?

Rappelons que ne sont évaluées pour le repérage, que les expressions temporelles et les expressions de localisation. Pour chaque page à juger, l'évaluateur doit compter, à l'aide des critères associés, les expressions appartenant à chacune des catégories présentées ci-dessous. Rappelons également que les informations à repérer sont des informations pratiques. En ce qui concerne les expressions temporelles, il s'agit de jours et horaires d'ouverture ou de fermeture, dates de début ou de fin, exceptions, etc. Pour les informations de localisation, elles comprennent les adresses des lieux touristiques : adresse complète de type « adresse postale », adresse incomplète (par exemple, rue et ville sans code postal ni numéro) ou encore nom de ville seul si aucune adresse plus précise n'est mentionnée.

a. Expressions repérées complètement

Les expressions repérées complètement sont celles dont les frontières ont bien été délimitées.

68 Merci à Julie Glikman d'avoir bien voulu jouer le rôle d'évaluateur.

Que ce soit pour les expressions temporelles ou les expressions de localisation, cela signifie qu'aucune partie de l'information pratique n'a pas été repérée.

Le cas où l'expression repérée dépasserait l'expression à repérer semble très improbable, étant donné le fonctionnement d'Adetoa (voir chapitre 5). Toutefois, un tel cas de figure ne serait pas considéré comme fautif : le principal est d'avoir l'information, l'annotation se charge ensuite de la rendre exploitable.

En ce qui concerne les expressions temporelles, il est possible qu'une expression soit repérée « en deux fois » : sous forme de deux <UT>. Si les deux couvrent tout de même l'expression en entier alors elle va bien dans cette catégorie. Si une partie de l'expression n'est pas repérée, alors elle ira dans l'une des catégories suivantes.

(169) en juin : du lundi au vendredi 16h-19h en juillet et août : tous les jours 11h-19h

L'exemple ci-dessus constitue une expression temporelle mais est repéré sous la forme de deux <UT>, comprenant chacune une période d'ouverture : *en juin : du lundi au vendredi 16h-19h* et *en juillet et août : tous les jours 11h-19h*. Aucune partie n'étant manquée, cette expression est à considérer comme repérée complètement.

b. Expressions repérées partiellement sans perte d'information

Entrent dans cette catégorie, les expressions qui ne sont pas repérées en entier mais dont la partie manquante n'est pas gênante pour la compréhension et n'entraîne pas de perte d'information. Par exemple, *ouvert du 15 juin au 30 juillet, tous les jours*. Si *tous les jours* n'est pas repéré aucune information n'est perdue car, sans cette précision supplémentaire, l'interprétation est que, pour la période donnée, l'offre touristique est ouverte tous les jours. Il en est de même pour un traitement automatique : sans autre précision, un intervalle donné sous forme de date de début et date de fin est interprété comme comprenant tous les jours de la période.

c. Expressions repérées partiellement avec perte d'information

Cette catégorie comprend le deuxième type d'expressions repérées partiellement. Contrairement aux expressions de la catégorie précédente, la partie non repérée nuit cette fois à la compréhension globale ou entraîne une perte d'information. Par exemple, *ouvert du 15 juin au 30 juillet, les mardi et vendredi* : ici, si *les mardi et vendredi* n'est pas repéré alors la perte d'information est évidente puisqu'une interprétation en « ouvert tous les jours » serait erronée.

d. Expressions manquées

Les expressions manquées sont celles qui auraient dû être repérées par Adetoa mais ne l'ont pas été. Il s'agit toujours d'informations pratiques (temporelles ou de localisation). Les expressions qui sont dans cette catégorie sont celles qui n'ont pas été repérées du tout, sinon elles seraient dans l'une des catégories précédentes.

e. Expressions repérées à tort

Les expressions repérées à tort peuvent être des expressions temporelles ou de localisation mais qui ne sont pas des informations pratiques. Par exemple des dates historiques, la date de modification du site, le nom des villes voisines. Pour trancher, l'évaluateur doit donc se demander si l'information lui permettra d'accéder plus facilement à l'objet touristique dont il

est question.

Les expressions qui entrent dans cette catégorie constituent des « faux positifs » faussant les calculs de rappel et précision. La mesure du « fallout » décrite dans la catégorie suivante permet de mesurer la résistance du système face aux faux positifs.

f. Expressions non pertinentes

La catégorie des expressions non pertinentes est moins classique dans les mesures d'évaluation ; elle comprend les informations non repérées à raison. Toutefois, il est difficile de caractériser précisément les expressions non pertinentes, c'est-à-dire « les expressions qui auraient pu être repérées à tort » et j'ai donc choisi de ne considérer pour cela que les expressions temporelles : dates historiques, dates de modification du site, heure actuelle, etc. Ces expressions, si elles avaient été repérées, auraient constitué des faux positifs. Le rapport entre les expressions repérées à tort et l'ensemble des expressions non pertinentes (repérées ou non) est parfois appelé « fallout » [Freitag 1998], que l'on peut traduire par « retombées » ou « répercussions ». Je ne vais utiliser cette mesure que sur le périmètre restreint des expressions temporelles.

Ces critères d'évaluation permettent donc de catégoriser les résultats de façon chiffrée et de faire des calculs de rappel et précision. Toutefois, ces mesures, si elles ont l'avantage d'être souvent utilisées et donc d'être familières à la communauté, ne reflètent pas toutes les facettes du système. Ainsi, étudier plus précisément les repérages partiels et les repérages fautifs permet de caractériser la nature des erreurs et éventuellement de classer ces erreurs selon une échelle de gravité. Ainsi, une expression manquée car elle comporte des fautes d'orthographe et n'est donc pas repérée par le système est-elle aussi grave qu'une expression manquée car son type n'a pas été prévu ? Une telle étude menée de manière exhaustive sur l'ensemble du corpus est très coûteuse en temps. Sans mettre réellement au point cette échelle de jugement selon le type d'erreur, je présente tout de même, dans ce chapitre, un inventaire des erreurs les plus fréquentes.

Les tableaux suivants permettent de récapituler les catégories dans lesquelles doivent être classées les expressions comprises dans les pages. Rappelons-le, il s'agit de compter, dans chaque page Web, les expressions correspondant à chacune des catégories présentées. Il faut à la fois évaluer ce qui a été repéré (tableau 2 – le texte en caractères gras correspond à ce qui a été repéré) et ce qui ne l'est pas – pour une bonne ou mauvaise raison (tableau 3).

Catégorie	Caractéristiques	Exemple
Repérées complètement	Toute l'expression est repérée, elle ne continue pas en dehors de la balise encadrante.	*L'hôtel est* ***ouvert toute l'année.*** *Il se trouve près d'un lac*
Repérées partiellement sans perte	Une partie de l'expression n'est pas repérée, mais cela n'entraîne pas de perte d'information.	***ouvert du 1er au 15 août****, tous les jours.*
Repérées partiellement avec perte	Une partie de l'expression n'est pas repérée, et cela entraîne une perte d'information.	***Ouvert du 1er au 15 août*** *et du 2 au 7 septembre*
À tort	L'expression n'aurait pas dû être repérée car elle ne constitue pas une information temporelle pratique ou ne concerne pas la localisation du lieu touristique dont il est question.	***En mai 1968***

Tableau 2 : Classification des expressions repérées

Catégorie	Caractéristiques	Exemple
Manquées	Expression à repérer (donc information temporelle pratique ou adresse d'un objet touristique) mais qui n'a pas été repérée par Adetoa.	*Les vendredis matin*
Non pertinentes (temporelles)	Informations temporelles mais qui ne sont pas des informations pratiques touristiques et qui ne doivent pas être repérées. Date de modification du site, heure actuelle, date historique...	*Ouvert depuis 1987*

Tableau 3 : Classification des expressions non repérées

2.3.2. Comment évaluer l'annotation ?

L'évaluation de l'annotation ne concerne également que les expressions temporelles et les expressions de localisation. Comme pour le repérage, l'évaluateur est chargé de classer, en suivant des critères prédéfinis, les annotations dans les catégories présentées ci-dessous. Par rapport à l'évaluation du repérage, il ne s'agit plus de compter les membres de chaque catégorie mais de classer chaque annotation dans une des catégories. Afin que les erreurs de repérage ne se propagent pas à l'évaluation de l'annotation, ne seront évaluées que les annotations d'expressions « bien repérées » : expressions repérées en entier ou partiellement. Les expressions repérées à tort ne sont pas prises en compte ici ; les expressions manquées, n'ayant donc pas donné lieu à une annotation, ne le sont pas non plus.

Plus que de simples catégories, l'évaluateur devra plutôt juger les annotations selon l'échelle

présentée ci-dessous.

a. Très bien

L'expression est parfaitement bien annotée, avec la granularité la plus fine. Toutes les informations présentes dans l'expression sont correctement annotées. Les deux exemples suivants sont parfaitement annotés et viendraient dans cette catégorie : chacune des informations présentes dans ces expressions est encadrée par une balise qui la type correctement.

(170) 6, Grande rue Chauchien 71400 Autun

```
<localisation>
   <adresse>6, Grande rue Chauchien</adresse>
   <cp> 71400</cp>
   <commune> Autun </commune>
</localisation>
```

(171) Ouvert du mardi au samedi, de 10h30 à 12h30

```
<UT>
   <periode_ouverture> Ouvert du
      <jour_debut> mardi</jour_debut> au
      <jour_fin> samedi</jour_fin>, de
      <heure_debut> 10h30</heure_debut> à
      <heure_fin> 12h30</heure_fin>
   </periode_ouverture>
   <description>"Ouvert du mardi au samedi, de 10h30 à
   12h30"</description>
</UT>
```

b. Granularité à affiner (ou problème d'incertitude)

Vont dans cette catégorie les expressions qui sont bien annotées, c'est-à-dire bien typées, mais dont la granularité des annotations n'est pas la plus fine possible. Pour le même exemple que le précédent, l'annotation suivante le placerait dans cette catégorie (les heures ne sont pas annotées) :

```
<UT>
   <periode_ouverture> Ouvert du
      <jour_debut> mardi</jour_debut> au
      <jour_fin> samedi</jour_fin>, de 10h30 à 12h30
   </periode_ouverture>
   <description>"Ouvert du mardi au samedi, de 10h30 à
   12h30"</description>
</UT>
```

Plus précisément, les annotations qui appartiennent à cette catégorie sont celles dont la granularité pourrait être affinée.

De plus, si la balise <Incertitude> est là à tort, ou si elle n'est pas là alors qu'elle devrait l'être, l'annotation vient aussi dans cette catégorie, à condition que le reste des informations soit bien typé, avec au maximum la granularité la plus fine non atteinte.

Entrent aussi dans cette catégorie les cas où les frontières de l'expression ont été mal définies lors du repérage, si l'annotation n'a pas réussi à s'en affranchir.

```
Course d'Orientation BP
<localisation>
   <cp>220 75967</cp>
   <commune>PARIS</commune>
</localisation>
```

Dans cet exemple, le repérage est incomplet puisque *BP* ne fait pas partie de l'expression repérée. Au niveau de l'annotation, le code postal ne devrait pas contenir en plus le numéro de la boîte postale. L'annotation suivante serait considérée comme correcte, malgré l'erreur de repérage :

```
Course d'Orientation BP
<localisation> 220
   <cp>75967</cp>
   <commune>PARIS</commune>
</localisation>
```

c. Annotation incomplète

Par rapport au cas précédent, cette catégorie comprend les expressions annotées partiellement, mais dont les balises manquantes ne concernent pas uniquement la granularité la plus fine : si les heures sont bien annotées mais les jours non, alors l'expression appartient à cette catégorie ; comme l'annotation suivante, toujours pour le même exemple :

```
<UT>
   <periode_ouverture> Ouvert du mardi au samedi, de
      <heure_debut> 10h30</heure_debut> à
      <heure_fin> 12h30</heure_fin>
   </periode_ouverture>
   <description>"Ouvert du mardi au samedi, de 10h30 à
   12h30"</description>
</UT>
```

Les balises qui figurent dans les expressions de cette catégorie typent correctement les informations.

Ces deux catégories sont distinctes car, dans le premier cas, l'information aurait pu être affinée mais elle est tout de même exploitable. Dans le second cas en revanche, les informations annotées, n'étant pas celles avec la granularité la plus large, deviennent difficiles à exploiter.

Les exceptions sont un cas particulier. Les expressions en contenant étant souvent complexes, le choix a été fait dans Eiffel d'annoter la partie textuelle comprenant une exception, avec une balise <exception>. Ce qui se trouve dans cette balise peut être annoté plus finement, avec une période d'ouverture ou de fermeture mais cela n'est pas toujours le cas, le principal étant de savoir que le segment textuel correspond à une exception. Les expressions formant ainsi des exceptions et n'étant pas plus finement qualifiées ne constituent pas des « annotations incomplètes » mais sont considérées comme bien annotées (si tout le reste est bon).

d. Mauvais typage

Les annotations de cette catégorie sont celles qui typent mal les informations qu'elles annotent. Il s'agit principalement des cas de mauvaise catégorisation entre ouverture et fermeture. L'annotation suivante de l'expression précédente en est un exemple :

```
<UT>
   <periode_fermeture> Ouvert du
      <jour_debut> mardi</jour_debut> au
      <jour_fin> samedi</jour_fin>, de
      <heure_debut> 10h30</heure_debut> à
      <heure_fin> 12h30</heure_fin>
   </periode_fermeture>
   <description>"Ouvert du mardi au samedi, de 10h30 à
   12h30"</description>
</UT>
```

De telles erreurs d'annotation produisent en effet des données totalement erronées et les expressions sont placées dans cette catégorie même si une partie des annotations est bonne.

Comme cela a été vu plus haut, les expressions contenant des exceptions qui ne sont pas qualifiées finement ne sont pas pénalisantes dans l'évaluation d'Adetoa. Toutefois, si une exception est qualifiée à l'aide des balises <periode_ouverture> ou <periode_fermeture> mais que le typage est mauvais, alors l'annotation dans son ensemble est considérée comme mauvaise.

Le tableau qui suit récapitule les catégories dans lesquelles doivent entrer chacune des expressions annotées (à l'exception des expressions repérées à tort).

Catégorie	Caractéristiques	Exemple
Parfaitement annotée	L'expression est parfaitement bien annotée, avec la granularité la plus fine. Toutes les informations présentes dans l'expression sont correctement annotées. Cas particulier : si dans la balise exception, les informations ne sont pas balisées, l'annotation peut tout de même être considérée comme parfaitement annotée.	*Du 16 Septembre au 11 Novembre :* <UT> <periode_ouverture> du <date_debut>16 Septembre </date_debut> au <date_fin>11 Novembre </date_fin> </periode_ouverture> </UT>
Granularité à affiner	L'expression est bien annotée, c'est-à-dire bien typée, mais la granularité des annotations n'est pas la plus fine possible. Cas particulier : si la balise <incertitude> manque ou est présente à tort, l'annotation est à placer dans cette catégorie.	*Lundi de 9h à 18h :* <UT> <periode_ouverture> <jour>lundi</jour> </periode_ouverture> de 9h à 18h </UT>
Annotation incomplète	L'expression est annotée partiellement, et les balises manquantes ne concernent pas uniquement la granularité la plus fine.	Catégorie non représentée dans le corpus d'évaluation
Mal annotée	Au moins une balise de l'expression ne donne pas le bon type à l'information. (ouverture au lieu de fermeture, etc.).	*Fermeture Dimanche soir et lundi sauf juillet août Et jours fériés :* <UT>fermeture <periode_fermeture> <jour>Dimanche</jour> <heure_debut>soir </heure_debut> et <jour>lundi</jour> </periode_fermeture> <exception> sauf <periode_fermeture> juillet août <incertitude /> </periode_fermeture> </exception> Et <periode_fermeture> <jour>jours fériés </jour> </periode_fermeture> </UT>

Tableau 4 : Classification des annotations

2.3.3. Comment évaluer le module de transformations ?

Le module de transformations que j'ai réalisé (voir chapitre 5) ne concerne qu'un nombre restreint d'expressions temporelles. L'évaluateur doit, pour chaque expression temporelle annotée, se demander si une transformation aurait dû avoir lieu ou non, en se reportant aux règles énoncées au chapitre 5, et classer les expressions dans les catégories suivantes.

Les expressions ne nécessitant pas de transformation et n'en ayant pas subie n'apparaissent

nulle part dans l'évaluation du module de transformations. Pour en faciliter l'évaluation, lorsqu'une transformation a lieu, elle est marquée par une balise[69].

a. Transformation bien faite

Vont dans cette catégorie les expressions qui correspondent à l'une des règles de transformations lorsque cette transformation a bien été effectuée, c'est-à-dire lorsque l'interprétation des données reste la même mais est plus simple. Par exemple, l'expression suivante est dans un premier temps annotée ainsi :

(172) Ouvert du lundi au vendredi

```
<UT>
   <periode_ouverture> Ouvert du
      <jour_debut> lundi </jour_debut> au
      <jour_fin> vendredi</jour_fin>
   </periode_ouverture>
   <description>"Ouvert du lundi au vendredi"</description>
</UT>
```

Voici l'annotation après transformation :

```
<UT>
   <periode_ouverture> Ouvert du
      <jour> lundi </jour>
      <jour> mardi </jour>
      <jour> mercredi </jour>
      <jour> jeudi </jour>
      <jour> vendredi</jour>
   </periode_ouverture>
   <inference/>
   <description>"Ouvert du lundi au vendredi"</description>
</UT>
```

Toutes les informations sont conservées, la collection de jours est plus simple à traiter qu'une période. Le contenu de la balise <description> doit rester le même.

b. Transformation mal faite

Les expressions de cette catégorie sont celles qui nécessitent une transformation mais dont celle-ci modifie l'interprétation. Par exemple, si un jour de fermeture devient, après transformation, un jour d'ouverture alors la transformation est mal faite.

(173) Ouvert du lundi au vendredi sauf le mardi

Voici le résultat obtenu pour cet exemple quand le module de transformations n'est pas activé :

69 Cette balise (<inference>) n'apparaît pas dans la DTD car elle n'est utilisée que pour l'évaluation.

```
<UT>
   <periode_ouverture> Ouvert du
      <jour_debut> lundi </jour_debut> au
      <jour_fin> vendredi</jour_fin>
   </periode_ouverture>
   <exception>
      <periode_fermeture> sauf le
         <jour> mardi </jour>
      </periode_fermeture>
   </exception>
   <inference/>
   <description>"Ouvert du lundi au vendredi sauf le
   mardi"</description>
</UT>
```

En revanche, si la transformation produisait le résultat suivant alors celle-ci serait considérée comme mal faite :

```
<UT>
   <periode_ouverture> Ouvert du
      <jour> lundi </jour>
      <jour> mardi </jour>
      <jour> mercredi </jour>
      <jour> jeudi </jour>
      <jour> vendredi</jour>
   </periode_ouverture>
   <inference/>
   <description>"Ouvert du lundi au vendredi sauf le
   mardi"</description>
</UT>
```

En effet, après une telle transformation, l'annotation contiendrait une information erronée.

c. Transformation non nécessaire

Les transformations non nécessaires sont repérables grâce à la balise <inference>. Il s'agit des cas où la transformation n'apporte rien, ne rend pas l'expression plus claire, voire même ne change rien à l'annotation.

d. Transformation manquante

Les expressions qui vont dans cette catégorie sont celles qui correspondent bien à l'une des règles de transformations mais qui n'ont pas été repérées par le système et où la transformation n'a donc pas eu lieu. Elles ne contiennent donc pas la balise <inference>. L'évaluateur doit se référer à la liste des cas pour lesquels une règle de transformation a été définie au chapitre 5.

2.3.4. Comment évaluer le liage ?

La tâche de liage est la dernière à être évaluée. Comme cela a déjà été présenté (chapitre 5), cette tâche consiste à regrouper les informations qui, dans une même page, concernent le même objet. L'évaluateur doit donc juger si les liages, représentés par les balises <ensemble> sont effectués correctement. Une nouvelle fois (comme pour le repérage) il doit donc tenir compte de la page entière et classer les ensembles (ou absences d'ensemble) dans les catégories présentées ci-dessous ; les ensembles sont donc comptés et ce sont ces chiffres qui permettent d'évaluer le module de liage. Chaque ensemble sera évalué en fonction de

« l'objet » auquel il se rapporte : toutes les informations qu'il regroupe doivent se rapporter au même objet.

L'évaluation du liage est à deux niveaux. En effet, le repérage des objets n'ayant pas été évalué, c'est ici qu'il se répercute. Le liage a donc été évalué selon deux axes. Le premier, purement algorithmique ne s'intéresse qu'aux données déjà annotées et vérifie si elles ont été liées conformément à l'algorithme – on parlera alors de liage selon balises. Le second, prend en compte le repérage des objets : si un objet est manqué, le liage peut être bon d'un point de vue algorithmique mais incohérent quant aux données qu'il contient réellement. Lorsque le liage est bien fait du point de vue du sens, on parlera alors de liage exact.

(174) Piscine [...] Patinoire ouverte tous les jours. 4 Rue de Londres 89470 Monéteau

```
<ensemble>
   <objet>Piscine</objet> [...]
   Patinoire
   <UT> ouverte tous les jours</UT>
   <Localisation> 4 Rue de Londres 89470 Monéteau<Localisation>
</ensemble>
```

Dans cet exemple[70], l'objet *patinoire* n'est pas repéré. L'ensemble est donc incohérent. Toutefois, l'algorithme semble fonctionner : un seul objet peut faire partie d'un ensemble à la fois. Si *patinoire* était repéré, il ferait donc bien partie du bon ensemble. Cet exemple constitue donc un bon liage selon balises mais pas un liage exact.

a. Ensembles cohérents et complets (évaluation du liage exact)

La première catégorie comprend les ensembles cohérents et complets. Un ensemble est « cohérent » si toutes les informations qu'il comprend se rapportent bien à un seul objet. Ce même ensemble est dit « complet » s'il contient toutes les informations qui se rapportent à cet objet et qui sont repérées dans la page.

b. Ensembles cohérents mais incomplets (évaluation du liage exact)

Cette catégorie regroupe les ensembles cohérents mais incomplets : au moins une information concernant l'objet en question est présente dans la page et est repérée correctement mais ne figure pas dans l'ensemble.

c. Ensembles incohérents (évaluation du liage exact)

Par opposition, les ensembles incohérents regroupent des informations qui ne concernent pas le même objet touristique. Il suffit qu'une information d'un ensemble se rapporte à un autre objet pour que l'ensemble soit considéré comme incohérent.

d. Ensembles manquants

Si, dans une page, plusieurs informations concernant le même objet sont repérées, mais qu'aucun ensemble n'est créé alors il manque un ensemble. Ces ensembles manqués sont comptabilisés dans cette catégorie.

e. Ensembles cohérents selon balises

Cette dernière catégorie permet de rendre compte des cas où le repérage est insuffisant

70 Exemple simplifié et « fabriqué » pour l'illustration du phénomène.

(objet non repéré par exemple). Si un objet est manqué, il ne peut pas faire partie d'un ensemble et la page a alors toutes les chances de contenir un ensemble incomplet ou même incohérent. Cela permet de distinguer les erreurs de repérage des erreurs dues à l'algorithme de liage en lui-même. Les ensembles qui entrent dans cette catégorie font aussi partie d'une des catégories précédentes, indiquant si l'ensemble est exact au niveau du sens.

3. Résultats

Les tâches d'Adetoa ayant été évaluées individuellement, la présentation des résultats de l'évaluation suit la même organisation.

3.1. L'évaluation du repérage

Pour l'évaluation du repérage, j'ai choisi de présenter les résultats selon deux axes. Le premier consiste à évaluer le repérage des expressions individuellement pour chacune. Le second en revanche consiste à évaluer la qualité du repérage par page Web (pour l'ensemble des expressions qui s'y trouvent).

3.1.1. Évaluation expression par expression

Adetoa analyse chaque page Web individuellement. Cependant, pour le repérage, chaque expression est identifiée, indépendamment des autres et plusieurs expressions (y compris de même type) peuvent être repérées dans une même page.

Ainsi, l'évaluation s'est concentrée sur le jugement de la qualité du repérage pour chaque expression. Les résultats selon le nombre d'expressions sont présentés ci-dessous. Le diagramme suivant montre, sur les 331 expressions à repérer dans le corpus, la répartition entre expressions bien repérées, expressions repérées partiellement (avec ou sans perte) et expressions manquées.

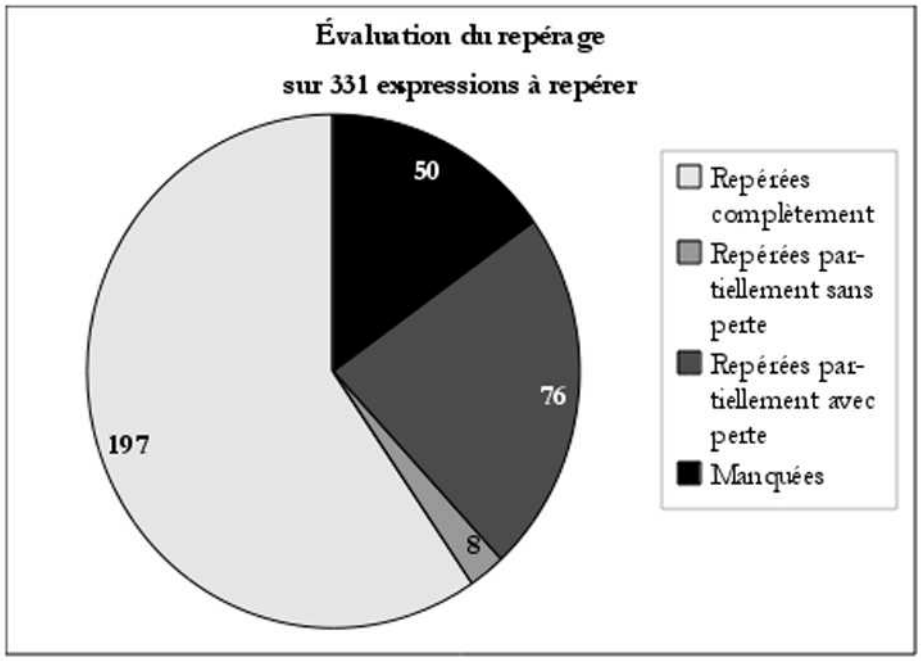


Figure 66 : Diagramme représentant les résultats, par expression, de l'évaluation du repérage

Ce diagramme ne représente que les expressions à repérer (au nombre de 331) et non l'ensemble des expressions repérées par Adetoa (au nombre de 338). Or, sur ce jeu d'évaluation, 57 expressions ont également été repérées à tort. Pour permettre une meilleure interprétation des résultats, les taux de rappel et précision ainsi que le fallout ont été calculés et sont présentés ci-dessous. Par ailleurs, une analyse des résultats imparfaits est proposée ensuite.

a. Rappel et précision

Je l'ai mentionné au début de ce chapitre, les mesures de rappel et précision sont parfois controversées et elles ne sont pas toujours représentatives des résultats obtenus par un système. Dans le cas d'Adetoa, ces mesures semblent tout de même permettre de rendre compte du module de repérage. Elles peuvent être calculées car les données qu'elles mettent en rapport sont effectivement disponibles : il s'agit du nombre d'expressions à repérer (comptées par l'évaluateur), du nombre d'expressions repérées et du nombre d'expressions bien repérées. L'évaluation permet justement de compter les expressions selon ces catégories.

Toutefois, afin de ne pas devoir considérer le repérage d'une expression comme « bon » ou « mauvais », j'ai introduit une certaine flexibilité sous la forme d'une graduation. Lors de l'évaluation, les expressions sont classées en quatre catégories (présentées plus en détail ci-dessus) :

- Expressions parfaitement repérées
- Expressions repérées partiellement sans perte d'information
- Expressions repérées partiellement avec perte d'information
- Expressions repérées à tort

En ce qui concerne les expressions repérées, je propose de calculer les taux de rappel et précision selon la graduation suivante (du plus strict au moins strict) :

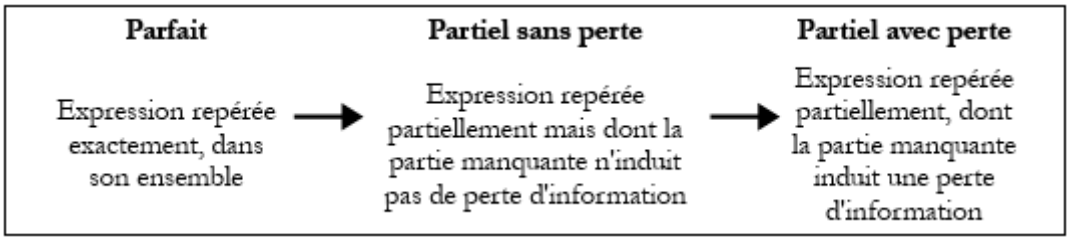


Tableau 5 : Graduation des résultats

Les taux de rappel et précision peuvent donc être calculés pour trois ensembles de résultats : le premier, le plus strict, ne comprenant que les résultats parfaits, le deuxième comprenant, en plus de ces résultats parfaits, les résultats partiels sans perte et enfin le troisième comprenant en plus les résultats partiels avec perte. Si cette troisième catégorie fait donc entrer dans les calculs de rappel et précision des résultats qui peuvent être considérés comme « moins bons », puisqu'une partie de l'information est perdue, ils ne prennent néanmoins pas en compte des résultats « mauvais » au sens où aucune information n'est « fausse ». Les résultats obtenus selon cette graduation à trois niveaux sont présentés dans le tableau 6, tandis que le tableau 7 donne la f-mesure pour chacun des trois niveaux.

<table>
<tr><th colspan="4">Rappel</th><th colspan="4">Précision</th></tr>
<tr><td rowspan="3">Parfait
+ Sans perte
+ Avec perte</td><td colspan="3">281/331 – 84,9%</td><td rowspan="3">Parfait
+ Sans perte
+ Avec perte</td><td colspan="3">281/338 – 83,1%</td></tr>
<tr><td rowspan="2">Parfait
+ Sans perte</td><td colspan="2">205/331 – 61,9%</td><td rowspan="2">Parfait
+ Sans perte</td><td colspan="2">205/338 – 60,7%</td></tr>
<tr><td>Parfait</td><td>197/331 – 59,5%</td><td>Parfait</td><td>197/338 – 58,3%</td></tr>
</table>

Tableau 6 : Taux de rappel et précision obtenus par Adetoa pour la tâche de repérage

F-mesure	
Parfait + Sans perte + Avec perte	84%
Parfait + Sans perte	61,3%
Parfait	58,9%

Tableau 7 : F-mesures obtenues par Adetoa pour la tâche de repérage avec $\alpha = 1$

Il est difficile de relativiser ces résultats dans la mesure où il n'existe pas de « résultats de référence » permettant d'effectuer une comparaison. En effet, la tâche effectuée par Adetoa est très spécifique aussi bien au niveau du type d'information à extraire que parce que celles-ci se trouvent dans des pages Web. Aucun outil équivalent n'existe, les résultats ne peuvent donc pas être directement comparés. Toutefois, les résultats obtenus par d'autres tâches

d'extraction d'information, dans des pages Web, peuvent donner une indication de la qualité des résultats.

	Rappel	**Précision**	**F-mesure** (calculée avec α=1)
[Gatterbauer et al. 2007] Repérage et interprétation de tableaux	Repérage		
	81%	68%	73,9%
	Interprétation		
	57%	48%	52,1%
[Hong et al. 2009] Repérage et annotation de références bibliographiques	Couples de résultats selon la configuration du système		
	96% 96,6% 96,3% 98,4% 99,2%	57,5% 53,6% 51,6% 40,9% 16,1%	71,9% 68,9% 67,2% 57,8% 27,8%
[Tengli et al. 2004] Extraction de tableaux	87,84%	95,31%	91,42%
[Nagy et al. 2009] Repérage des affiliations de chercheurs	62,88%	78,73%	69,92%

Tableau 8 : Résultats obtenus par d'autres systèmes d'extraction d'information

Le tableau 8 présente les résultats obtenus par différents systèmes pour des tâches d'extraction d'information plus ou moins proches de celle effectuée par Adetoa. Les travaux présentant ces systèmes ont déjà été cités au début de ce chapitre. Les méthodes utilisées par ces outils étant toutes différentes, les résultats ne sont pas directement comparables, ni entre eux, ni avec ceux obtenus par Adetoa. Je les donne donc à titre indicatif. Il est toutefois possible de remarquer que, chez [Hong et al. 2009], le rappel est très haut mais la précision beaucoup plus basse, tandis que dans les autres cas les deux taux sont assez proches mais également moins hauts. Les résultats obtenus par Adetoa pour le repérage se situent dans cette seconde catégorie. Le fait que les taux de rappel et précision soient proches montre un certain équilibre entre le nombre d'informations manquées et le nombre d'informations repérées à tort. Si l'on augmente la sensibilité du système pour réduire le nombre d'expressions manquées, cela risquerait de faire augmenter le nombre d'expressions repérées à tort, et inversement.

b. Fallout

Cela a été précisé précédemment, je ne me suis intéressée au fallout que pour les expressions temporelles. Le fallout permet de représenter la résistance du système aux faux positifs en mesurant le rapport entre les faux positifs repérés et les faux positifs potentiels.

Adetoa repère peu d'expressions temporelles à tort, malgré le nombre relativement élevé d'expressions temporelles à ne pas repérer dans les pages. Le taux de fallout obtenu est le suivant :

6 expressions repérées à tort	**40** expressions ignorées à raison	6/46 – 13%

Tableau 9 : Taux de fallout

Parmi les informations qui auraient pu être repérées à tort se trouvent principalement des dates historiques (exemple 175), des dates de dernière modification du site (exemple 176) et quelques expressions diverses (exemple 177).

(175) En 1990

(176) MISE A JOUR LE 04/04/2007

(177) À 1h30 de Paris

3.1.2. Évaluation par page Web

Adetoa repérant chaque expression individuellement, ce sont sur celles-ci que le repérage a été évalué. Toutefois, une page Web pouvant contenir plusieurs informations à repérer, il est également intéressant de présenter les résultats par page. Ce nouvel angle permet de qualifier le repérage dans les pages sans tenir compte du nombre d'expressions qui s'y trouve. En effet, si, de par sa construction, une page contient un grand nombre d'expressions manquées ou, à l'inverse, si une page bien construite, sans erreur de syntaxe, contient un grand nombre d'expressions bien repérées, ce nombre ne pondère pas le résultat obtenu par l'évaluation du repérage dans la page.

Le diagramme suivant présente une catégorisation des pages évaluées. Trois catégories sont présentées :

- Pages sans erreur de repérage : toutes les expressions contenues dans la page sont correctement repérées – aucune information n'est repérée à tort.
- Pages avec perte d'information : certaines informations contenues dans la page ne sont pas repérées (expressions manquées ou partiellement repérées) – aucune information n'est repérée à tort. Une partie des informations peut aussi être bien repérée.
- Pages avec fausses informations : certaines informations sont repérées à tort (ce qui n'empêche pas que d'autres soient bien repérées dans la page).

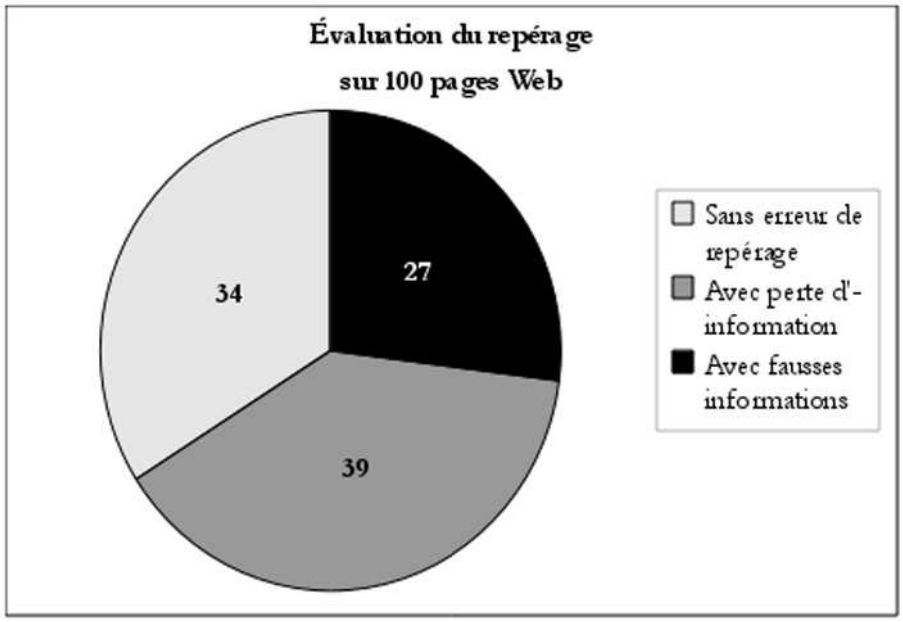


Figure 67 : Diagramme représentant les résultats, par page, de l'évaluation du repérage

Cela permet de remarquer que des expressions repérées à tort sont présentes dans 27% des pages. Lorsque ces pages sont traitées, cela entraîne donc le stockage d'informations fausses dans la base de connaissance.

En revanche, si le repérage est incomplet dans 39% des pages (informations manquantes ou expressions partiellement repérées), les expressions qui y sont tout de même repérées ont l'avantage de ne pas contenir de fausses informations. De plus, sur les 123 expressions à repérer dans ces 39 pages, 60 sont bien repérées (sans perte d'information), 31 expressions sont repérées avec perte d'information et 32 expressions sont manquées. Ces pages ne contenant pas d'informations repérées à tort, toutes les expressions repérées qui s'y trouvent sont exploitables.

En additionnant les pages ne comprenant que des informations bien repérées et les pages comprenant des informations manquées, 73% des pages évaluées permettent d'enrichir, sans fausse information, la base de connaissance.

3.1.3. Relevé d'erreurs

Les erreurs que peut commettre Adetoa au niveau du repérage sont de deux types. Soit une information est manquée – ou partiellement manquée – ce qui induit une perte d'information, soit une information est repérée à tort, ce qui induit des fausses informations. Pour ces deux types d'erreurs, les causes sont assez peu nombreuses et sont présentées ci-dessous.

a. Expressions manquées ou partiellement repérées

En ce qui concerne les informations temporelles, les expressions manquées ou partiellement repérées le sont car leur structure n'a pas été prévue dans les transducteurs.

(178) **Ouverture: de Pâques au 15 Octobre**

(179) **Fermeture hebdomadaire : dimanche soir hors saison**

(180) **Ouvert du premier avril jusqu'au mi-novembre, mais ces dates sont surtout fonctions de la présence de campeurs sur le terrain, nous pouvons ouvrir avant et après. S'il fait beau ou sur réservation les dates d'ouverture peuvent être un peu plus amples.**

(181) **de 1 juillet jusqu'au 2 septembre**

(182) **<UT>Horaires : de 9h à 19 h</UT> et nocturne le vendredi 9 mars jusqu'à 23h**

(183) **Mardi <heure_debut>9h 17</heure_debut>h**

Les expressions des exemples (178) à (181) sont manquées. Pour les exemples (178) et (179), cela vient réellement de leur structure qui n'a pas été modélisée : pour le premier, les fêtes (*Pâques*) ne sont pas prévues dans les transducteurs, pour le second, c'est les termes exprimant une périodicité (*hebdomadaire*) qui ne le sont pas. La première partie de l'exemple (180) n'est pas repérée à cause de *premier* écrit en toutes lettres. La suite n'est pas repérée car elle n'est pas formalisable et encore moins généralisable. Quant à l'exemple (181), s'il n'est pas pris en compte, c'est parce qu'il comprend une faute de syntaxe : cela devrait être *du* au lieu de *de*[71].

Les exemples (182) et (183)[72] sont partiellement repérés, comme le montre l'encadrement par la balise <UT>. La seconde partie de l'exemple (182) n'est pas repérée car la tournure avec *nocturne* n'est pas prévue. L'exemple (183) est plus problématique. Il pose la question de la limite entre le repérage et l'annotation. Cet exemple donne une heure de début et une heure de fin comme *de 9h à 17h* mais seules les heures sont mentionnées, sans séparateur : *mardi 9h 17h*. Du coup, le dernier *h* n'est pas repéré et le *17* n'est pas annoté comme une heure de fin mais comme faisant partie de l'heure de début. Toutefois, cet exemple est à considérer comme une erreur de repérage car, si la structure sans séparateur avait été prévue, alors l'ensemble serait repéré et probablement du coup bien annoté.

Les informations de localisation qui sont manquées ou partiellement repérées relèvent principalement d'un problème de nom de ville composé. Un travail sur ces noms de villes composés a pourtant été fait, dans le dictionnaire, pour permettre de les reconnaître aussi bien quand ils contiennent des espaces que quand ils contiennent des traits d'union. Les expressions qui sont, malgré cela, manquées ou partiellement repérées, adoptent une orthographe hybride comme dans l'expression suivante :

(184) **<localisation> 58140 Saint-Martin </localisation> du Puy**

Un trait d'union est présent entre les deux premiers mots du nom de ville mais pas entre les suivants. Les deux cas suivants auraient été bien repérés :

(185) **58140 Saint-Martin-du-Puy**

(186) **58140 Saint Martin du Puy**

71 Il est à noter que la page a manifestement été écrite par un non francophone et comprend de nombreuses fautes de français.

72 Ce ne sont pas les annotations complètes qui sont données ici, mais seulement les balises utiles à l'analyse.

Ce problème est souvent rencontré pour les noms composés longs comme *Saint-Honoré-les-Bains*. Il résulte la plupart du temps d'un manque de rigueur de la part du rédacteur de la page qui ne respecte pas l'orthographe du nom.

b. Expressions repérées à tort

La majorité des expressions repérées à tort (51/57 – 89,5%) sont des expressions de localisation. Ces expressions se répartissent dans deux catégories. La première comprend les expressions qui n'expriment pas une localisation mais qui sont repérées comme telle à cause d'un phénomène d'homographie (exemple 187). La seconde comprend des expressions qui contiennent effectivement un nom de ville mais qui ne constituent pas une information de localisation au sens d'adresse (exemples 188 et 189). La seconde catégorie est la plus fréquente.

(187) <localisation><CP>65</CP><commune> Salle</commune> </localisation>

Cet exemple est repéré comme une localisation car *Salle* fait partie du dictionnaire des noms de villes. Comme ce terme est précédé d'un nombre, celui-ci est considéré comme un code postal et un repérage fautif a ainsi lieu[73]. Les cas de ce type sont rares (5/51 – 9,8%). Pour les éviter, il faudrait par exemple faire un travail sur le dictionnaire, identifier les noms de villes qui ont des homographes dans la langue et définir des règles particulières pour déterminer, à l'aide du contexte, s'il s'agit du nom de la ville ou de son homographe.

En revanche, les exemples suivants constituent la majorité des localisations repérées à tort (46/51 – 90,2%).

(188) <localisation><adresse>25, route de</adresse> <commune>Paris</commune> </localisation>

(189) 66 Nevers

L'exemple (188) est considéré comme fautif car il ne contient pas le nom de la ville. Or, comme cela a été vu au chapitre 5, la présence du nom de la ville est nécessaire pour identifier une localisation. Ici, *Paris* est considéré à tort comme le nom de la ville alors qu'il fait en réalité partie du nom de la rue et que le nom de la ville n'est pas mentionné (il apparaît ailleurs dans la page, mais pas au niveau de l'adresse). Pour éliminer ce type de mauvais repérage, il faudrait étudier plus précisément la composition des noms de rues et prévoir des repérages d'adresses sans nom de ville directement associé.

Dans l'exemple (189), *Nevers* est bien un nom de ville. Toutefois, il ne s'agit pas d'une localisation, mais d'une indication géographique : *66* est la distance en kilomètres entre Nevers et l'objet dont il est question (tournure du type : *notre hôtel se situe à 66 km de Nevers...*). Ce type d'expressions repérées à tort est difficile à éviter. Néanmoins, les règles de mapping pourraient éviter de stocker ces fausses informations dans la base de connaissance en vérifiant la longueur du code postal.

Parmi les six expressions temporelles repérées à tort se trouvent une date historique (exemple 190), deux indications de prix selon la saison (exemple 191), et trois cas particuliers.

(190) en mai 1968

73 Il n'est pas possible, dans Unitex, de définir la taille d'un nombre pour préciser qu'il doit être composé de cinq chiffres exactement (longueur du code postal en France).

(191) <UT>en juillet août 55</UT> €

Pour ce qui est des dates historiques, une règle de mapping peut vérifier que la date est pertinente pour éviter de stocker des informations trop anciennes. Pour les autres cas, les transducteurs devraient être affinés et le contexte mieux pris en compte.

3.2. L'évaluation de l'annotation

Pour évaluer l'annotation, comme cela a déjà été présenté, chaque expression repérée (à raison) et par conséquent évaluée, a été catégorisée comme bien annotée, partiellement annotée, granularité à affiner ou mal annotée. Étant donné que certaines expressions sont repérées en plusieurs fragments, le nombre d'annotations n'est pas le même que le nombre d'expressions repérées (à raison). Cette partie présente à la fois les résultats chiffrés de l'évaluation de l'annotation et une analyse des erreurs les plus fréquentes.

3.2.1. Quelques chiffres

Le nombre total d'expressions annotées (une fois retirées les expressions repérées à tort et dont il est donc inutile de juger l'annotation) est de 296. Le diagramme ci-dessous en donne la répartition dans les catégories proposées pour l'évaluation. Avec 93,6% de ces expressions parfaitement annotées et seulement 2% d'expressions mal annotées, l'annotation n'introduit donc que très peu de fausses informations dans la base de connaissance. Il faut aussi noter que la catégorie « annotation incomplète » n'est pas représentée dans le corpus d'évaluation.

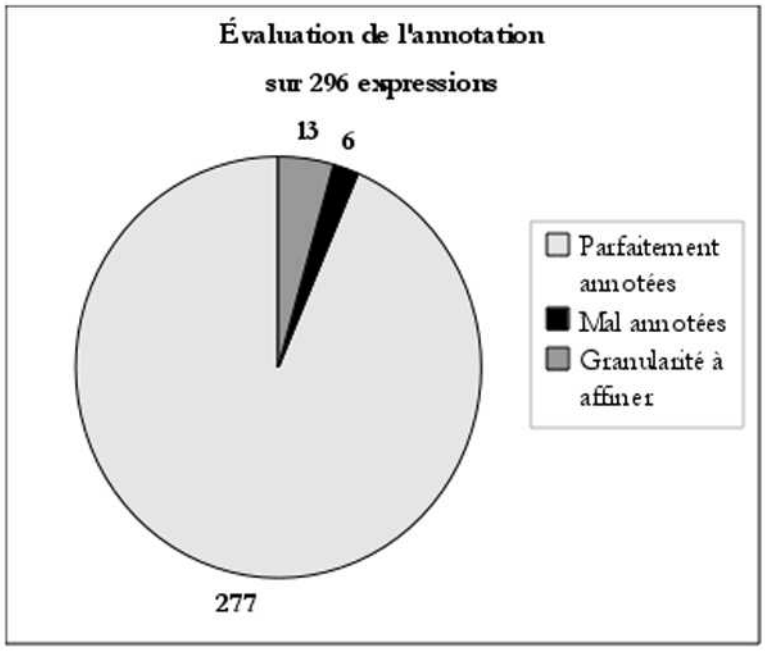


Figure 68 : Diagramme représentant, par expression, l'évaluation de l'annotation

Comme cela a été fait pour l'évaluation du repérage, il est aussi intéressant de présenter les résultats de l'évaluation de l'annotation par page Web. Le diagramme suivant montre le nombre de pages dans lesquelles toutes les annotations sont bien effectuées, le nombre de

pages dans lesquelles certaines annotations sont à affiner et le nombre de pages dans lesquelles certaines expressions sont mal annotées. Les expressions mal annotées (qui sont au nombre de 6) se regroupent dans 5% des pages, tandis que 85% des pages ont l'ensemble de leurs expressions bien annotées.

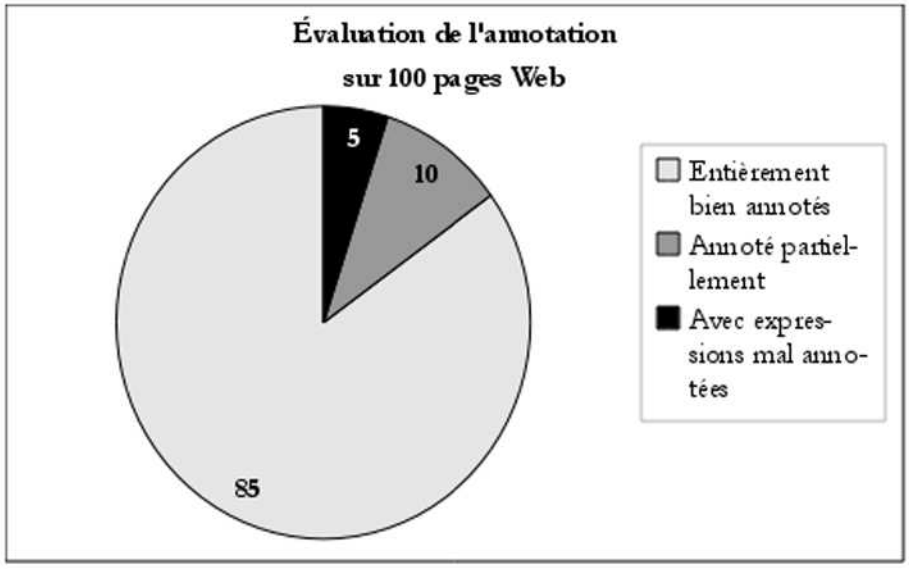


Figure 69 : Diagramme représentant, par page, l'évaluation de l'annotation

3.2.2. Relevé d'erreurs

Les chiffres ci-dessus l'ont montré, la plupart des expressions repérées par Adetoa ont été bien annotées.

La principale source d'erreur pour l'annotation des expressions de localisation provient du code postal. En effet, s'il est précédé d'un autre nombre, celui-ci sera annoté comme faisant partie du code postal :

(192) **<cp>220 75967</cp> <commune>PARIS</commune>**

En ce qui concerne les expressions temporelles, les erreurs sont rares et découlent le plus souvent d'une erreur de repérage :

(193) **Horaires : 24h**/24

Dans cet exemple, seule la partie en gras a été repérée, l'annotation a donc considéré *24h* comme l'heure de début.

(194) **09/07/07 à 13 h au 14/07/07 à 10h**

Cette expression a été repérée en deux fois, le *au* central ayant été manqué. Les deux parties sont alors annotées individuellement et donc en ouverture, tandis qu'il s'agit en réalité d'une information comportant une date de début et une date de fin. L'annotation est par conséquent fautive pour la deuxième partie puisque celle-ci est une fermeture mais qu'elle est annotée comme une ouverture.

Une seule expression est correctement repérée mais mal annotée :

(195) fermeture Dimanche soir et lundi sauf juillet août Et jours fériés

En effet, comme le montre l'annotation obtenue qui figure ci-dessous, seules des périodes de fermeture sont annotées dans cette expression alors que l'exception (*sauf juillet et août*) ne constitue pas une fermeture mais plutôt une ouverture.

```
<UT>fermeture
   <periode_fermeture>
      <jour>Dimanche</jour>
      <heure_debut>soir</heure_debut> et
      <jour>lundi</jour>
   </periode_fermeture>
   <exception> sauf
      <periode_fermeture> juillet août <incertitude />
      </periode_fermeture>
   </exception> Et
   <periode_fermeture>
      <jour>jours fériés</jour>
   </periode_fermeture>
   <description>"fermeture Dimanche soir et lundi sauf juillet août Et
   jours fériés"</description>
</UT>
```

Pour résumer, une seule expression bien repérée est mal typée, tandis que les autres cas de mauvais typages en ouverture ou en fermeture sont liés à un repérage imparfait.

3.3. L'évaluation du liage

Comme cela a été vu plus haut, l'évaluation du liage se fait selon deux axes : liage exact du point de vue du sens et liage selon les balises. De façon à évaluer effectivement l'algorithme de liage, c'est cette deuxième approche qui est principalement présentée ici ; le liage n'est ainsi pas pénalisé par des repérages manquants ou de mauvaises annotations.

3.3.1. Quelques chiffres

L'évaluation du liage a pour visée principale de montrer la capacité de l'algorithme à effectuer le traitement voulu sur les pages annotées. Le diagramme suivant montre donc les résultats obtenus avec l'approche du liage selon balises. Il représente la totalité des ensembles qui auraient dû être créés et les distingue selon les catégories présentées au début de ce chapitre.

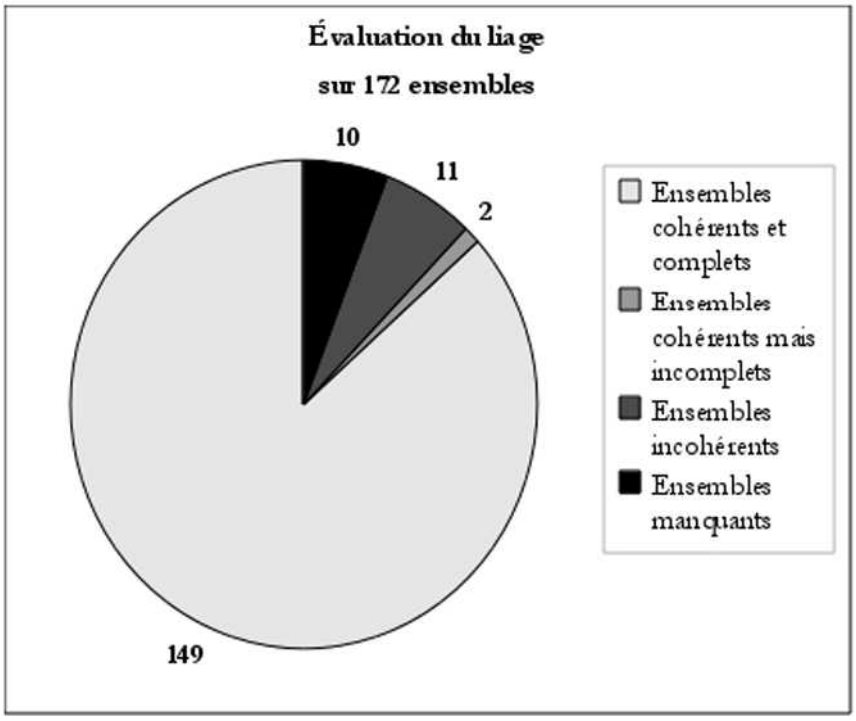


Figure 70 : Diagramme représentant l'évaluation du liage selon les balises

Selon la seconde approche, 57 ensembles ne peuvent pas être considérés comme des liages exacts. Par rapport aux 172 ensembles à lier, cela équivaut à un taux de 33,1%. Dans la majorité des cas, les ensembles inexacts sont incomplets ou incohérents car l'objet n'est pas repéré. Si l'on considère que l'algorithme utilisé est simpliste, puisque le seul critère qu'il prend en compte est la proximité, ces résultats peuvent être qualifiés de corrects.

Par ailleurs, une amélioration et un enrichissement des transducteurs de repérage et d'annotation des objets amélioreraient donc sensiblement le score obtenu par la tâche de liage, en ce qui concerne effectivement le liage exact.

3.3.2. Relevé d'erreurs

Les erreurs rencontrées lors de l'évaluation du liage sont de trois types, qui correspondent aux catégories qui ont été définies pour évaluer ce liage : ensembles incohérents (comprenant des informations concernant plusieurs objets), ensembles incomplets et ensembles manquants.

a. Ensembles incohérents

La plupart des ensembles incohérents sont causés par un objet mentionné, et donc repéré dans la page, mais qui n'est lié à aucune autre information. Il peut alors causer des liages incohérents s'il est placé, dans la page, à l'intérieur de la description d'un autre objet, comme dans l'exemple suivant :

(196) Notre hôtel, situé en face de la mairie, est ouvert toute l'année.

Pour une telle expression, deux objets (*hôtel* et *mairie*) et une information temporelle sont repérés. Le liage, appliquant l'algorithme de proximité, crée un ensemble englobant *mairie* et *ouvert toute l'année*.

Pour contourner ce type d'erreur, un algorithme plus élaboré doit être mis au point. Celui-ci pourrait se baser sur une analyse des prépositions. Par exemple *en face de la mairie* indique bien que *mairie* n'est pas l'objet principal dont il est question.

Ce problème des objets liés à des informations qui ne les concernent pas se généralise si tous les objets de la page ne sont pas repérés. En effet, si un objet n'est pas repéré, il ne peut donc pas faire partie d'un ensemble et un liage incohérent a toutes les chances d'être effectué. Cela correspond à l'exemple (174) donné précédemment (p. 175).

Pour pallier ce problème, les transducteurs permettant de repérer les objets devraient être enrichis de façon à ce que les objets manqués soient réduits au minimum. Un tel travail se rapprocherait de la constitution d'un dictionnaire, mais devrait tout de même garder la structure de l'ontologie afin de garantir la bonne annotation des objets.

b. Ensembles incomplets

Les ensembles incomplets, qui sont peu nombreux dans le corpus d'évaluation (2/172 – 1,2%), sont causés par la répétition du type d'objet à plusieurs endroits dans la page. En effet, l'algorithme prévoit de n'intégrer qu'un seul nom d'objet touristique par ensemble. Une structure comme la suivante ne peut donc pas faire l'objet d'un liage correct :

(197) L'hôtel est ouvert toute l'année. L'hôtel est situé 4 rue de Lourdes, 58000 Nevers.

Le type *hôtel* y est repéré deux fois, et deux ensembles sont donc créés alors que toutes les informations concernent un seul et même objet.

Deux types de solution peuvent être envisagés pour remédier à ce problème. La première solution consisterait à regrouper les ensembles qui se suivent et dont le type d'objet est le même. Toutefois cela risquerait de générer des ensembles incohérents si plusieurs objets de même type sont présentés dans une page. La seconde solution, plus élaborée sur un plan linguistique, consisterait à analyser le texte séparant les ensembles pour trouver des marqueurs de continuité ou au contraire de rupture permettant d'indiquer si les informations concernent le même objet ou non.

c. Ensembles manquants

Dans le corpus d'évaluation, les ensembles manquants (10/172 – 5,8%) sont dus à la façon dont l'algorithme de liage est implémenté dans Adetoa, à l'aide de transducteurs Unitex.

En effet, si un transducteur reconnaît plusieurs chemins dans une page, et que deux d'entre eux sont imbriqués, il annotera uniquement le premier, le second ensemble est alors manqué.

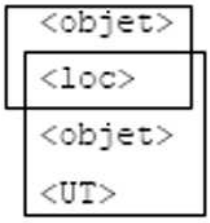


Figure 71 : Exemples de liage

La figure 71 représente une suite de repérages possibles dans une page. Deux ensembles sont possibles et sont représentés par les rectangles noirs. Le « bon » ensemble, conformément à l'algorithme défini, est le second, qui comprend une localisation, un objet et une information temporelle. Pourtant Unitex s'arrête au premier ensemble possible, qu'il annote. Le second ensemble est bien repéré mais ne peut pas être annoté pour éviter toute imbrication ; il est donc ignoré.

3.4. L'évaluation des transformations

Les transformations ne sont que très peu représentées dans le corpus d'évaluation.

16 cas ont nécessité une transformation et celle-ci a eu lieu, avec succès dans tous les cas sauf un. Aucune transformation n'a été réalisée à tort et dans 15 cas sur 16, les transformations étaient souhaitées et ont correctement enrichi l'information.

Ainsi, le petit nombre de transformations dans le corpus ne permet pas de considérer le score obtenu comme représentatif pour le module de transformations. Toutefois, sur les 16 transformations évaluées, les résultats sont bons. En effet, seule une transformation est manquée et aucune n'introduit de fausse information.

Voici un exemple de transformation réalisée :

(198) **Ouverture Tous les jours de 10h00 à 12h30 et de 15h00 à 19h00 Sauf le mercredi et le jeudi**

Sans l'intervention du module de transformations, cet exemple est annoté ainsi :

```
<UT>
   <periode_ouverture>Ouverture Tous les jours de
      <jour>lundi</jour>
      <jour>mardi</jour>
      <jour>mercredi</jour>
      <jour>jeudi</jour>
      <jour>vendredi</jour>
      <jour>samedi</jour>
      <jour>dimanche</jour>
      <heure_debut>10h00</heure_debut>à
      <heure_fin>12h30</heure_fin>et de
      <heure_debut>15h00</heure_debut>à
      <heure_fin>19h00</heure_fin>
   </periode_ouverture>
   <exception>Sauf
      <periode_fermeture>le
         <jour>mercredi</jour>et le
         <jour>jeudi</jour>
      </periode_fermeture>
   </exception>
   <description>"Ouverture Tous les jours de 10h00 à 12h30 et de 15h00
   à 19h00 Sauf le mercredi et le jeudi"
   </description>
</UT>
```

Si aucun raisonnement n'intervient, cette annotation peut être considérée comme fausse car, si mercredi et jeudi sont annotés comme une fermeture dans l'exception, ils sont également annotés comme une ouverture. Sans raisonnement, le mapping ne peut donc pas trancher et l'information devient inexploitable.

En revanche, après la transformation effectuée par Adetoa, l'annotation est la suivante :

```
<UT>
   <periode_ouverture>Ouverture Tous les jours de
      <jour>lundi</jour>
      <jour>mardi</jour>
      <jour>vendredi</jour>
      <jour>samedi</jour>
      <jour>dimanche</jour>
      <heure_debut>10h00</heure_debut>à
      <heure_fin>12h30</heure_fin>et de
      <heure_debut>15h00</heure_debut>à
      <heure_fin>19h00</heure_fin>
   </periode_ouverture>
   <description>"Ouverture Tous les jours de 10h00 à 12h30 et de 15h00
   à 19h00 Sauf le mercredi et le jeudi"
   </description>
</UT>
```

Cette annotation permet tout à fait de rendre toute l'information contenue dans l'expression.

L'expression suivante, qui est plus longue, a elle aussi été bien traitée par le module de transformations d'Adetoa :

> **(199) toute l'année Le lundi de 15h à 18h Du mardi au vendredi de 10h à 12h et de 14h à 18h. Le samedi de 10h à 12h.**

Une fois l'expression annotée, et après transformation, trois périodes d'ouverture sont marquées et les jours mentionnés sont énumérés.

L'expression dans laquelle une transformation aurait pu avoir lieu mais n'a pas été effectuée est la suivante, pour laquelle l'annotation qui suit est obtenue :

(200) **Du lundi au samedi : 8h-12h30 et 13h30_19h Dimanche : 8h-12h30 et 14h-17h Fermé le mercredi après-midi**

```
<UT>
   <periode_ouverture> Du
      <jour_debut>lundi</jour_debut> au
      <jour_fin>samedi</jour_fin> :
      <heure_debut>8h</heure_debut> _
      <heure_fin>12h30</heure_fin> et
      <heure_debut>13h30</heure_debut> _
      <heure_fin>19h</heure_fin>
   </periode_ouverture>
   <periode_ouverture>
      <jour>Dimanche</jour> :
      <heure_debut>8h</heure_debut> _
      <heure_fin>12h30</heure_fin> et
      <heure_debut>14h</heure_debut> _
      <heure_fin>17h</heure_fin>
   </periode_ouverture> Fermé
   <periode_fermeture> le
      <jour>mercredi</jour>
      <heure_debut>après-midi</heure_debut>
   </periode_fermeture>
   <description>"Du lundi au samedi : 8h-12h30 et 13h30_19h Dimanche :
   8h-12h30 et 14h-17h Fermé le mercredi après-midi"</description>
</UT>
```

La transformation aurait pu convertir le groupe de jours *du lundi en samedi* en énumération. Si cette transformation n'a pas été déclenchée cela est dû à la granularité de la période de fermeture qui contient des horaires : la transformation agit avec une granularité « jour » et peut donc, une fois un groupe de jours converti en énumération, en retirer les jours de fermeture, mais la règle n'est pas conçue à un niveau de granularité plus fin et ne peut donc pas en retirer une partie de journée (*mercredi après-midi*) comme cela serait nécessaire ici. La transformation n'a donc pas lieu pour ne pas risquer d'introduire de fausses informations.

4. Évaluation complémentaire – RMM2

Afin d'estimer l'adaptabilité des transducteurs de repérage et d'annotation d'expressions temporelles développés pour Adetoa, j'ai eu recours à un corpus d'expressions temporelles indiquant des périodes d'accessibilité (terme emprunté à [Teissèdre et al. 2010a]). Ce corpus, établi dans le cadre du projet RMM2[74], contient 513 expressions, stockées dans un tableur (voir aussi [Teissèdre et al. 2010b] qui présentent des travaux menés dans ce projet). Chaque expression est donc indépendante des autres et fournie sans contexte ; elle décrit spécifiquement une période d'accessibilité. Les trois exemples suivants sont issus du corpus :

(201) **Vendredi, samedi et dimanche de 15h à 19h.**

(202) **Du lundi au vendredi, de 10h à 19h.**

(203) **De 14h à 19h.**

74 RMM2 – RelaxMultiMedias2 – financé par l'ANR « Contenus et Interactions » (2009- 20011).

Les transducteurs que j'ai développés pour Adetoa étant prévus pour typer les expressions en ouvertures ou fermetures, le repérage est déclenché par un marqueur du type *ouvert, fermé,* ou encore *accès*. Telles quelles, les expressions ne sont donc en grande majorité pas reconnues par les transducteurs d'Adetoa.

Dans le but de mener une évaluation plus pertinente, j'ai donc ajouté le marqueur *ouvert :* devant chaque expression. J'ai ensuite évalué les résultats obtenus selon un critère de frontière : l'expression est-elle reconnue en entier ou partiellement ? Étant donné que ce corpus se rapproche plus d'un jeu de test que d'un véritable corpus, et que les expressions sont données sans contexte, il ne m'a pas semblé pertinent de calculer les taux de rappel et précision. Les résultats seront donc donnés en pourcentages simples selon le critère de frontière.

Une évaluation de l'annotation semble moins pertinente dans la mesure où le marqueur *ouvert* a été ajouté ; la difficulté de typer l'expression en ouverture ou fermeture devient inappropriée. Une expression a cependant été mal typée, mais l'ajout du marqueur *ouvert* l'a rendue incohérente :

(204) **Relâche dimanche et lundi.**

Les résultats obtenus montrent que les transducteurs développés pour Adetoa pourraient facilement être généralisables pour des périodes d'accessibilité propres à d'autres domaines que celui du tourisme et ne se trouvant pas nécessairement sur des pages Web.

Parmi les expressions du corpus, mis à part quelques tournures originales ou cas particuliers, deux cas ont attiré mon attention car ils n'étaient pas bien repérés.

(205) **Lundi au dimanche de 13h à 19h.**

(206) **Tous les jours à partir de 12h.**

Le premier concerne les expressions du type de celle de l'exemple (205) : le début de ces expressions n'était pas bien repéré car le premier jour de la semaine n'est pas introduit, alors qu'une tournure comme *du lundi au vendredi* est plutôt attendue. Les expressions commençant ainsi par un intervalle de jours de la semaine sont très fréquentes, puisqu'elles sont au nombre de 252 (sur 513 expressions) dans le corpus. Afin qu'elles ne fassent pas chuter les résultats, j'ai choisi d'ajouter le marqueur *de* ou *du* avant le premier jour de la semaine dans ces cas, en plus du marqueur *ouvert*, ce qui donne donc des expressions comme celle-ci :

(207) **Ouvert : du Lundi au dimanche de 13h à 19h.**

Une fois cette modification effectuée, 247 (sur les 252) expressions sont entièrement repérées.

Le deuxième cas concerne les expressions qui débutent par *tous les jours*, comme celle de l'exemple (206). Si cette expression est prévue dans les transducteurs, le cas où elle se trouve en début d'expression et suivie d'horaires ne l'est pas. Ces expressions ne sont alors pas bien repérées mais rien ne peut être fait au niveau des expressions pour changer le résultat. Or, cette évaluation a pour visée de juger les transducteurs créés pour Adetoa. Une modification de ces transducteurs permettrait de bien traiter ces expressions mais cela introduirait également un biais dans l'évaluation. Je n'ai donc pas modifié les transducteurs et ces expressions, qui sont au nombre de 94 ne sont pas repérées en entier.

Le tableau 10 synthétise les résultats obtenus par les transducteurs d'Adetoa sur l'ensemble des expressions du corpus RMM2. Étant donné le nombre tout de même important

d'expressions qui débutent par *tous les jours* et sont donc mal repérées, j'ai décidé de les écarter du corpus. En effet, leur présence nuit beaucoup aux résultats alors qu'un simple chemin ajouté dans les transducteurs permettrait de les reconnaître. Étant devenue incohérente, j'ai aussi choisi d'écarter l'expression de l'exemple (204) qui est mal typée. Une fois ces filtres appliqués au corpus, les résultats présentés dans le tableau 11 sont obtenus.

Catégorie	**Exemples**	**Nombre**	**Pourcentage dans le corpus**
Sans perte d'information	• Ouvert : Dimanche à 16h30. Lundi à 20h30. • Ouvert : Le mardi de 12h à 21h, du mercredi au vendredi de 12h à 19h, les samedi et dimanche de 10h à 19h.	88 repérées parfaitement 12 repérées partiellement sans perte d'information	19,5%
Intervalle de jours (avec ajout du *du*) repérés parfaitement	• Ouvert : du Lundi au dimanche de 13h à 19h. • Mardi à vendredi de 14h à 18h. Samedi de 14h à 19h. Fermé du 20 décembre au 4 janvier.	247 repérées parfaitement (sur 252 intervalles)	48,2%
Repérées partiellement	• <UT>Ouvert : de Jeudi et dimanche à 18h30,</UT> sauf 24 et 31 décembre. • <UT>Ouvert : de Lundi à samedi de 13h à 19h.</UT> Jeudi jusqu'à 22h.	59 repérées partiellement	11,5%
Cas particuliers	• Ouvert : Dès 8h pour les détenteurs du billet privilège "Floor right" A partir de 10h pour les détenteurs du billet classique. • Elles s'allument à la tombée de la nuit.	12 non repérées	2,3%
Tous les jours	Tous les jours à partir de 12h.	94 non repérées	18,3%
Mauvais typage	Relâche dimanche et lundi.	1 mal typée	0,2%

Tableau 10 : Résultats de l'évaluation sur les 513 expressions du corpus RMM2

Repérées sans perte d'information	**Repérées avec perte d'information**	**Non repérées**
347/418 – 83%	59/418 – 14,1%	12/418 – 2,9%

Tableau 11 : Résultats de l'évaluation sur corpus partiel

Moins de 3% des expressions ne font pas l'objet d'un repérage. Parmi le reste des expressions, 83% sont repérées parfaitement et 14,1% sont repérées partiellement. Dans la plupart de ces expressions partielles, la partie manquante constitue une information complémentaire, comme dans l'exemple suivant :

(208) **<UT>Mardi au vendredi de 11h à 18h, samedi et dimanche de 10h à 18h</UT>, nocturne jusqu'à 21h jeudi.**

La perte est difficilement calculable. Toutefois, il est possible d'avancer que les expressions (ou parties d'expressions) qui font l'objet d'un repérage n'introduisent pas de fausses informations.

Conclusion

L'évaluation d'Adetoa ainsi que la présentation des résultats ont considéré chaque tâche indépendamment, de façon à ce que les résultats de l'une impacte le moins possible sur les résultats des autres.

Les tâches de repérage et d'annotation sont celles qui ont demandé l'étude linguistique la plus avancée et c'est donc l'évaluation de celles-ci qui est présentée avec le plus de précision. Il en est ressorti qu'un repérage bien mené est la quasi garantie d'une annotation bien faite : la majorité des erreurs d'annotation surviennent en effet dans des expressions qui n'ont été que partiellement repérées.

Par ailleurs, en ce qui concerne le repérage en lui-même, l'analyse des erreurs a permis de montrer que celles-ci sont regroupables en un petit nombre de cas. Un travail supplémentaire sur ces cas précis permettrait donc d'améliorer sensiblement les résultats obtenus.

Pour le liage et l'application des règles de transformations, l'évaluation a mis en valeur la qualité des algorithmes développés. Ceux-ci obtiennent en effet de bons résultats. Le liage des données dans les pages est souvent bien effectué et lorsque cela n'est pas le cas, cela est le plus souvent dû à un repérage manquant. Une amélioration des graphes de repérage, notamment en ce qui concerne les objets, permettrait donc d'améliorer la qualité du liage. Pour ce qui est des transformations, les règles sont appliquées à bon escient et permettent effectivement d'enrichir les annotations. Elles ne causent aucune information erronée. La seule façon de les améliorer serait donc de développer de nouvelles règles, plus riches, permettant de compléter les informations obtenues à d'autres niveaux.

Pour rendre compte de la façon dont les résultats d'Adetoa sont réellement exploitables, par exemple dans le cadre de la plateforme Eiffel, il faudrait mettre en relation les pages Web de départ et les informations stockées dans la base de connaissance. En effet, les pages annotées produites par Adetoa ne constituent qu'une étape intermédiaire, qui ne prend sens qu'une fois complétée par l'application des règles de mapping.

Une évaluation complémentaire des transducteurs de repérage et d'annotation des expressions temporelles a été menée sur un corpus de 513 expressions fourni par le projet RMM2. Les résultats obtenus montrent que les transducteurs développés pour Adetoa pourraient être adaptés pour d'autres applications.

CONCLUSION

Cette thèse a présenté le développement d'un système – Adetoa – dédié au repérage et à l'annotation sémantique automatique d'expressions temporelles dans des pages Web pour une application de e-tourisme.

Les expressions temporelles que j'ai étudiées ont deux caractéristiques principales : elles se trouvent sur des pages Web et sont propres au domaine du tourisme. L'étude linguistique que j'ai menée dans les premiers chapitres a mis en évidence le fait que ces expressions présentent une plus grande variété dans des pages touristiques que dans des guides touristiques papier.

Comme c'est le cas dans les affiches publicitaires, la présentation visuelle joue un rôle important dans les pages Web. De plus, le placement des expressions n'est pas régulier et, de manière plus générale, les pages Web ne sont pas structurées de manière homogène. Une étude sémiotique m'a cependant permis de montrer que les pages sont constituées d'unités de sens et qu'elles restent souvent interprétables quelle que soit la disposition de ces unités.

En conséquence, pour le développement d'Adetoa, je me suis appuyée uniquement sur le contenu textuel des pages Web. L'étude sémiotique ne donne par directement lieu à une implémentation ; toutefois, la structure du texte présent dans le code source laisse souvent intact le texte des unités de sens. J'ai choisi, pour traiter ces contenus, d'utiliser des transducteurs (Unitex). Je me suis appuyée sur l'étude linguistique pour en constituer un ensemble important, reflétant les formulations variées et la combinatoire. Ces transducteurs repèrent et annotent les informations temporelles ainsi que les objets du tourisme et les adresses puis lient toutes les informations concernant le même objet.

Dans le cadre du projet Eiffel, une ontologie du tourisme m'a été fournie pour annoter les données dans les pages Web. Toutefois, le schéma d'annotation qui en résulte s'est avéré être insuffisant pour modéliser toutes les données temporelles. Je l'ai donc enrichi et modifié de façon à annoter plus finement les expressions et à respecter précisément leur structure linguistique. Grâce à ces propositions de modification, le projet étant encore en développement, l'ontologie a ensuite pu être adaptée en conséquence. De plus, les règles de transformations que j'ai élaborées permettent de rendre les données plus conformes à cette nouvelle ontologie, pour pouvoir les stocker dans la base de connaissance qui lui correspond. Pour une chaîne de traitement comme celle d'Eiffel, qui se place dans le cadre du Web Sémantique, l'ontologie joue donc un rôle central et critique. Il est pourtant nécessaire, outre la modélisation du domaine, de tenir compte de la façon dont les données peuvent être formulées dans la langue. Je constate que l'interaction, comme celle que nous avons mise en

œuvre dans le projet Eiffel, entre le développement des ressources linguistiques et celui de l'ontologie est un facteur de succès pour les projets de ce type.

Par la suite, les différentes tâches effectuées par Adetoa ont été évaluées et j'ai pour cela adapté les méthodologies d'évaluation souvent utilisées pour les systèmes de TAL. Ainsi, j'ai proposé d'utiliser les mesures classiques de rappel et précision tout en introduisant une graduation permettant d'intégrer des résultats partiels. D'une certaine manière, cette graduation s'appuie sur la qualité des services rendus à l'utilisateur : s'il est inacceptable de lui fournir des informations fausses, il peut néanmoins parfois se satisfaire de résultats légèrement imparfaits.

Ces études théoriques, aussi bien que ces développements techniques, ont mené aux résultats suivants.

L'étude sémiotique a montré que, dans un cadre comme celui du projet Eiffel, où les pages Web à analyser sont d'une telle variété, il n'est pas possible de faire appel à une technique basée sur des wrappers. En effet, ces pages, qui ne sont pas narratives, ne présentent aucune régularité au niveau de leur structure. La structure étant ainsi inexploitable, la solution que j'ai proposée est de se concentrer sur de petites unités plutôt que sur la page dans son ensemble. C'est pourquoi les transducteurs sont construits de façon à ne repérer que des informations pertinentes, sous la forme d'expressions correspondant à une unité de sens, en excluant au maximum le contexte.

Par ailleurs, les transducteurs que j'ai développés présentant une grande combinatoire, ils permettent de couvrir de manière satisfaisante les expressions temporelles qui se trouvent dans les pages Web étudiées pour le projet Eiffel. Cela se reflète dans l'évaluation présentée au chapitre 7. Pourtant, la variabilité de la langue étant infinie, il est impossible d'en avoir une grammaire finie. Même pour un type d'expressions précis et restreint, cette combinatoire reste importante. C'est là que se trouve la limite du développement de transducteurs. Ceux-ci permettent en effet de repérer uniquement les expressions qui ont été prévues.

Les contenus textuels étudiés ont leurs caractéristiques propres, liées au fait qu'ils apparaissent sur un certain type de pages Web. Les informations qu'Adetoa cherche à extraire se trouvent rarement dans des portions de texte rédigées, il y a peu de prédications et la grammaire ainsi que la syntaxe du français ne sont pas toujours respectées. Toutefois, l'analyse plus fine des expressions elles-mêmes, en tant que portions de textes autonomes, s'est révélée satisfaisante.

De plus, l'avantage qui en découle est que, comme je l'ai montré dans l'évaluation, les transducteurs réalisés sont ainsi plus généralisables que s'ils tenaient largement compte du contexte. D'une part, ils peuvent être généralisés aux informations temporelles touristiques se trouvant sur d'autres supports que des pages Web : guides papier, brochures, plaquettes, affiches, etc. D'autre part, ils peuvent s'étendre aux informations non touristiques : toute période d'accessibilité donnée textuellement peut être interprétée. Par exemple, dans le domaine médical, ils pourraient servir à annoter les horaires des pharmacies ou médecins de garde, ce qui est utile en ce sens que ces informations changent très régulièrement (roulement hebdomadaire ou mensuel).

L'analyse de pages Web étant au centre de cette thèse, il m'est impossible d'ignorer ici les

limites que l'évolution des technologies qu'elles emploient impose. En effet, si les pages étaient à l'origine constituées exclusivement de texte, elles comportent maintenant aussi des images et pictogrammes, des contenus dynamiques ou encore des animations. Les technologies comme AJAX [Garret 2005] et Flash, qui sont de plus en plus répandues, ont tendance à « masquer » les contenus textuels (voir sur ce point [Garron et al. à par.]). Les transducteurs ne pouvant traiter que du texte, leur limite peut sembler atteinte ici. Cependant, cette évolution est accompagnée de nouveaux outils grâce auxquels ces contenus peuvent être reconvertis en texte et être ainsi de nouveau exploitables par les transducteurs : reconnaissance de caractères dans les images, de formes dans les pictogrammes ou bien encore des Services Web permettant de récupérer du contenu dynamique.

Parallèlement à cela se développe le Web Sémantique [Laublet 2007] et l'intégration de méta-données dans les pages. Ces méta-données doivent donc être créées et cette tâche est fastidieuse pour un traitement manuel. Dans ce cadre, un outil comme Adetoa servirait à générer ces méta-données automatiquement lors de la création et de la mise à jour des pages.

L'étude linguistique l'a montré, certaines des expressions temporelles contenues dans les pages Web sont floues, ambiguës ou font référence aux connaissances du monde, ce qui complique le traitement automatique. Elle peut ainsi servir de base à un « guide de bonne pratique » destiné aux professionnels du tourisme. Ce guide leur fournirait des conseils les aidant à mieux formuler les données temporelles. Suivre ces recommandations, de la même manière qu'ils en suivent pour que leurs pages soient bien référencés dans les moteurs de recherche, améliorerait ainsi leur visibilité et permettrait d'éviter les expressions difficilement interprétables, comme la suivante :

(209) **Ouverture du camping En principe le camping est ouvert du premier avril jusqu'au mi-novembre, mais ces dates sont surtout fonctions de la présence de campeurs sur le terrain, nous pouvons ouvrir avant et après. S'il fait beau ou sur réservation les dates d'ouverture peuvent être un peu plus amples.**

BIBLIOGRAPHIE

[Amardeilh & Damljavonic 2009] Amardeilh F., Damljanovic D. (2009) « Du texte à la connaissance : annotation sémantique et peuplement d'ontologie appliqués à des artefacts logiciels ». Actes de *IC'09 - 20èmes journées francophones d'Ingénierie des Connaissances*, Hammamet, Tunisia.

[Amardeilh & Francart 2006] Amardeilh F., Francart T. (2006) « Enrichissement de bases de connaissances par l'annotation sémantique », In *Ingénierie des Systèmes d'Information*, 11(2), pp. 53-70.

[Amardeilh 2007] Amardeilh F. (2007) *Web Sémantique et Informatique Linguistique : propositions méthodologiques et réalisation d'une plateforme logicielle*. Thèse de doctorat, Université Paris X.

[Amardeilh 2009] Amardeilh F. (2009) « Filtrer sémantiquement les textes dans le cadre du web sémantique », In Minel J.-L. : *Filtrage sémantique : de l'annotation à la navigation textuelle*, Traité IC2, série Informatique et Systèmes d'Information, Hermès, pp. 189-224.

[Amardeilh et al. 2005] Amardeilh F., Laublet P., Minel J.-L. (2005) « Document Annotation and Ontology Population from Linguistic Extractions ». Actes de *KCap, the Third International Conference on Knowledge Capture*, Banff, Canada.

[Battistelli et al. 2006] Battistelli D., Minel J.-L., Schwer S. (2006) « Représentation des expressions calendaires dans les textes : vers une application à la lecture assistée de biographies », In *TAL*, 47/2, pp. 11-37.

[Bex et al. 2004] Bex G. J., Neven F., Van den Bussche J. (2004) « DTDs versus XML schema: a practical study ». Actes de *WebDB 2007, the 7th International Workshop on the Web and Databases*, ACM, Paris, pp. 13-18.

[Blanc et al. 2006] Blanc O., Constant M., Laporte É. (2006) « Outilex, plate-forme logicielle de traitement de textes écrits ». Actes de *TALN'06, 13ème conférence sur le traitement automatique des langues naturelles*, UCL, Presses Universitaires de Louvain, Leuven, Belgique, pp. 83-92.

[Bourigault et al. 2004] Bourigault D., Aussenac-Gilles N., Charlet J. (2004) « Construction de ressources terminologiques ou ontologiques à partir de textes : un cadre unificateur pour trois études de cas », In Pierrel J.-M., Slodzian M : *Techniques Informatiques et Structuration de Terminologies*, Revue d'Intelligence Artificielle, 18(1), Numéro spécial, Hermès, pp. 87-110.

[Bry et al. 2003] Bry F., Lorenz B., Ohlbach H. J., Spranger S. (2003) « On Reasoning on Time and Location on the Web ». Actes de *PPSWR 2009, Workshop on Principles and Practice of Semantic Web Reasoning*, Springer-Verlag, Mumbai, India, pp. 69-83.

[Chang & Lui 2001] Chang C.-H., Lui S.-C. (2001) « IEPAD: information extraction based on pattern discovery ». Actes de *WWW 2001, the 10th international conference on World Wide Web*, Elsevier, Hong Kong, pp. 681-688.

[Charlet 2002] Charlet J. (2002) *L'Ingénierie des connaissances : développements, résultats et perspectives pour la gestion des connaissances médicales*. Mémoire d'habilitation à diriger des recherches, Université Paris VI.

[Charlet et al. 2004] Charlet J., Bachimont B., Troncy R. (2004) « Ontologies pour le Web sémantique », In *I3 (Information, Intéraction, Intelligence)*, Numéro hors série*, Web sémantique*, pp. 43-63.

[Charolles & Péry-Woodley 2005] « Les adverbiaux cadratifs », *Langue Française*, 148 (2005), Direction : Charolles M., Péry-Woodley M.-P.

[Charolles 2002] Charolles M. (2002) *La référence et les expressions référentielles en français*, Paris, Ophrys.

[Cohen et al. 2002] Cohen W. W., Hurst M., Jensen L. S. (2002) « A flexible learning system for wrapping tables and lists in HTML documents ». Actes de *WWW 2002, the 11th international conference on World Wide Web*, ACM, Hawai, USA, pp. 232-241.

[Coste 2009] Coste M. (2009) *Projet de Chaîne d'Annotation Sémantique et Extraction automatique de Connaissances – Annotation sémantique*. Rapport de stage en Génie Informatique, Université de Technologie de Compiègne.

[Courtois & Silberztein 1990] « Les dictionnaires électroniques du français », *Langue française*, 87 (1990), Direction : Courtois B., Silberztein M.

[Crescenzi & Mecca 2004] Crescenzi V., Mecca G. (2004) « Automatic Information Extraction from Large Web Sites », In *ACM*, 51(5), pp. 731-779.

[Culioli 1973] Culioli A. (1973) « Sur quelques contradictions en linguistique », In *Communications*, 20, *Le sociologique et le linguistique*, pp. 83-91.

[Cunningham et al. 2002] Cunningham H., Maynard D., Bontcheva K., Tablan V. (2002) « GATE : A Framework and Graphical Development Environment for Robust NLP Tools and Applications ». Actes de *ACL-02, the 40th Anniversary Meeting of the Association for Computational Linguistics*, ACL, Philadelphia, PA, USA, pp. 168-175.

[Davallon à par.] Davallon J. (à par.) *Économies de l'écriture sur le web, Volume 1 : les traces d'usage dans un corpus de sites de tourisme*, Hermès.

[Davidson 1969] Davidson D. (1969) « L'Individuation des Evénements », In : *Actions et événements (1993)*, PUF, pp. 219-243.

[De Sitter & Daelemans 2003] De Sitter A., Daelemans W. (2003) « Information extraction via double classification ». Actes de *ATEM 2003, Workshop on Adaptive Text Extraction and Mining*, Cavtat-Dubrovnik, Croatia.

[Eiffel 2009] Lacroix F. (2009) *Rapport final Eiffel* - Compte-rendu de fin de projet.

[Enjalbert et al. 2008] « Plate-formes pour le traitement automatique des langues », *Traitement Automatique des Langues*, 49/2 (2008), Direction : Enjalbert P., Bontcheva K., Habert B.

[Fairon 2006] Fairon C. (2006) « Corporator: A tool for creating RSS-based specialized corpora ». Actes de *EACL 2006, Workshop Web as corpus*, Trento, Italie.

[Fairon et al. 2008] Fairon C., Macé K., Naets H. (2008) « GlossaNet 2: a linguistic search engine for RSS-based corpora ». Actes de *LREC 2008, the sixth international conference on Language Resources and Evaluation*, Marrakech, Maroc.

[Ferrucci & Lally 2004] Ferrucci D., Lally A. (2004) « UIMA: an Architectural Approach to Unstructured Information Processing in the Corporate Research Environment », In *Natural Language Engineering*, 10, pp. 327-348.

[Fortin et al. 2009] Fortin J., Carloni O., Leclère M., Weiser S. (2009) « Extraction et exploitation de données temporelles pour un portail d'e-tourisme ». Actes de *EGC 2009, atelier Fouille de données temporelles et analyse de flux de données*, Strasbourg, France.

[Freitag 1998] Freitag D. (1998) *Machine Learning for Information Extraction in Informal Domains*. Thèse de doctorat, Université Carnegie Mellon.

[Fresnault-Deruelle 1997] Fresnault-Deruelle P. (1997) *L'image placardée*, Paris, Nathan.

[Garret 2005] Garret J. J. (2005) *Ajax: A New Approach to Web Applications*, http://www.adaptivepath.com/ideas/essays/archives/000385.php.

[Garron et al. à par.] Garron I., Minel J.-L., Couto J., Weiser S. (à par.) « Plan technique et plan sémiotique dans l'analyse des sites web « participatifs » », In Davallon J. : *Économies de l'écriture sur le web*, Hermès.

[Gatterbauer et al. 2007] Gatterbauer G., Bohunsky P., Herzog M., Krüpl B., Pollak B. (2007) « Towards Domain-Independent Information Extraction from Web Tables ». Actes de *WWW 2007, the 16th International World Wide Web Conference*, ACM Press, Banff, Canada, pp. 71-80.

[Gayral & Grandemange 1992] Gayral F., Grandemange P. (1992) « Une ontologie du temps pour le langage naturel ». Actes de *Coling 1992, the 14th conference on Computational Linguistics*, ACL, Nantes, pp. 295-302.

[Gross 1989] Gross M. (1989) « The Use of Finite Automata in the Lexical Representation of Natural Language », In *Electronic Dictionaries and Automata in Computational Linguistics, Lecture Notes in Computer Science*, 377, pp. 34-50.

[Gruber 2008] Gruber T. (2008) *Ontology* in « the Encyclopedia of Database Systems », Liu L., Özsu M. T. (éds.).

[Guarino 1998] Guarino N. (1998) *Formal Ontology in Information System*, Amsterdam, IOS Press.

[Habegger & Qualafou 2004] Habegger B., Qualafou M. (2004) « Building web information extraction tasks ». Actes de *WI'04, the International Conference on Web Intelligence*, IEEE Computer Society, Beijing, China, pp. 349-355.

[Hamon et al. 2007] Hamon T., Nazarenko A., Poibeau T., Aubin S., Derivière J. (2007) « A Robust Linguistic Platform for Efficient and Domain specific Web Content Analysis ». Actes de *RIAO 2007, the 8th Conference on Large-Scale Semantic Access to Content (Text, Image, Video and Sound)*, Pittsburgh, USA.

[Handschuh 2005] Handschuh S. (2005) *Creating Ontology-based Metadata by Annotation for the Semantic Web*. Thèse de doctorat, Université de Karlsruhe.

[Heijst et al. 1997] van Heijst G., Schreiber A., Wielinga B. (1997) « Using Explicit Ontologies in KBS Development », In *International Journal of Human Computer Studies*, 46, pp. 183-292.

[Ho-Dac & Péry-Woodley 2009] Ho-Dac L.-M., Péry-Woodley M.-P. (2009) « A data-driven study of temporal adverbials as discourse segmentation markers », In *Discours*, 4, revue électronique.

[Hobbs & Pan 2004] Hobbs J. R., Pan F. (2004) « An ontology of time for the semantic web », In *ACM Transactions on Asian Language Information Processing*, 3/1, pp. 66-85.

[Hobbs & Pustejovsky 2003] Hobbs J. R., Pustejovsky J. (2003) « Annotating and Reasoning about Time and Events ». Actes de *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, Stanford, California.

[Hong et al. 2009] Hong C. H. A., Gozali J. P., Kan M.-Y. (2009) « FireCite: Lightweight real-time reference string extraction from webpages ». Actes de *NLPIR4DL, the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, ACL, Singapour, pp. 71-79.

[Jacques & Aussenac-Gilles 2006] Jacques M.-P., Aussenac-Gilles N. (2006) « Variabilité des performances des outils de TAL et genre textuel. Cas des patrons lexico-syntaxiques », In *TAL*, 47/1, pp. 11-32.

[Jilani & Amardeilh 2009] Jilani I., Amardeilh F. (2009) « Enrichissement automatique d'une base de connaissances biologiques à l'aide des outils du Web sémantique ». Actes de *IC 2009, 20èmes Journées francophones d'Ingénierie des Connaissances*, Hammamet, Tunisie.

[Joly 1993] Joly M. (1993) *Introduction à l'analyse de l'image*, Paris, Nathan.

[Kleiber 1986] Kleiber G. (1986) « Déictiques, embrayeurs, etc. Comment les définir ? », In *L'information grammaticale*, 30, pp. 3-22.

[Kleiber 1990] Kleiber G. (1990) *La sémantique du prototype. Catégories et sens lexical*, Paris, Presses universitaires de France.

[Laender et al. 2002] Laender A. H. F., Ribeiro-Neto B. A., da Silva A. S., Teixeira J. S. (2002) « A brief survey of web data extraction tools », In *Sigmod Record*, 31/2, pp. 84-93.

[Laublet 2007] Laublet P. (2007) « Web sémantique et ontologies », In Brossaud C., Reber B. : *Humanités numériques. Nouvelles technologies cognitives et concepts des sciences sociales*, Hermès, pp. 231-244.

[Laublet et al. 2009] Laublet P., Aussenac-Gilles N., Camps V., Glize P., Hernandez N., Maurel H., Mbarki M., Mothe J., Ralalason B. J. V., Reymonet A., Rothenburger B., Sellami Z., Thomas J., Tissaoui A. (2009) *Projet ANR DYNAMO : Etat de l'art - Livrable lot 2*, Rapport de contrat.

[Lavelli et al. 2004] Lavelli A., Califf M. E., Ciravegna F., Freitag D., Giuliano C., Kushmerick N., Romano L. (2004) « IE evaluation: Criticisms and recommendations ». Actes de *ATEM 2004, the AAAI 2004 Workshop on Adaptive Text Extraction and Mining*, San Jose, California.

[Le Draoulec & Péry-Woodley 2005] Le Draoulec A., Péry-Woodley M.-P. (2005) « Encadrement temporel et relations de discours », In *Langue Française*, 148, pp. 45-60.

[Leroy 2004] Leroy S. (2004) « Le nom propre en français », In , , revue électronique.

[Mani & Wilson 2000] Mani I., Wilson G. (2000) « Robust temporal processing of news ». Actes de *the 38th Annual Meeting on Association for Computational Linguistics*, Morgan Kaufmann Publishers, Hong Kong, pp. 69-76.

[Mani 2004] Mani I. (2004) « Recent Developments in Temporal Information Extraction ». Actes de *RANLP'03*, John Benjamins, Borovets, Bulgaria, pp. 45-60.

[Maynard 2005] Maynard D. (2005) « Benchmarking ontologybased annotation tools for the Semantic Web ». Actes de *AHM 2005, the Workshop Text Mining, eResearch and Gridenabled Language Technology*, Nottingham, UK.

[Maynard et al. 2002] Maynard D., Cunningham H., Bontcheva K., Dimitrov M. (2002) « Adapting A Robust Multi-Genre NE System for Automatic Content Extraction ». Actes de *AIMSA 2002, the Tenth International Conference on Artificial Intelligence: Methodology, Systems, Applications*, Varna, Bulgaria.

[Moeschler 1993] Moeschler J. (1993) « Aspects pragmatiques de la référence temporelle : indétermination, ordre temporel et inférence », In *Langages*, 112*, Temps, référence et inférence*, pp. 39-54.

[MUC-6 1995] *Proceedings of the Sixth Message Understanding Conference (DARPA)* (1995), San Franciso, USA, Morgan Kaufmann Publishers.

[Nagy et al. 2009] Nagy I., Farkas R., Jelasity M. (2009) « Researcher affiliation extraction from homepages ». Actes de *NLPIR4DL, the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, ACL, Singapour, pp. 1-9.

[Nédellec & Nazarenko 2005] Nédellec C., Nazarenko A. (2005) « Ontology and Information Extraction: A Necessary Symbiosis », In Buitelaar P., Cimiano P., Magnini B. : *Ontology Learning from Text: Methods, Evaluation and Applications*, IOS Press, pp. 155-170.

[Nguyen 2006] Nguyen T. D. (2006) *Extraction d'information à partir de documents Web multilingues : une approche d'analyses structurelles*. Thèse de doctorat, Université de Caen.

[Noël et al. 2008] Noël L., Carloni O., Moreau N., Weiser S. (2008) « Designing a knowledge-based tourism information system », In *International Journal of Digital Culture and Electronic Tourism*, 1(1)*, Special Issue on National Tourism Organisations and Exploitation of Information Technologies*, pp. 1-17.

[Novotný et al. 2009] Novotný R., Vojtás P., Maruscák D. (2009) « Information Extraction from Web Pages ». Actes de *WI'09, the International Conference on Web Intelligence*, IEEE Computer Society Press, Milan, Italie, pp. 121-124.

[Pan & Hobbs 2005] Pan F., Hobbs J. (2005) « Temporal Aggregates in OWL-Time ». Actes de *FLAIRS18, the Florida Artificial Intelligence Research Society Conference*, AAAI Press, Florida, USA, pp. 560-565.

[Paroubek et al. 2007] « Principes de l'évaluation en Traitement Automatique des Langues », *Traitement Automatique des Langues*, 48/1 (2007), Direction : Paroubek P., Chaudiron S., Hirschman L.

[Paumier 2003] Paumier S. (2003) *De la reconnaissance de formes linguistiques à l'analyse syntaxique*. Thèse de doctorat, Université de Marne-la-Vallée.

[Paumier 2006] Paumier S. (2006) *Unitex 1.2 Manuel d'utilisation*.

[Peirce 1978] Peirce C. S. (1978) *Ecrits sur le signe*, Paris, Seuil.

[Poibeau 2003] Poibeau T. (2003) *Extraction automatique d'information. Du texte brut au web sémantique*, Paris, Hermès.

[Poibeau 2008] Poibeau T. (2008) *Des mots aux textes. Analyse sémantique pour l'accès à l'information*. Mémoire d'habilitation à diriger des recherches, Université Paris Nord, LIPN.

[Popescu-Belis 2007] Popescu-Belis A. (2007) « Le rôle des métriques d'évaluation dans le processus de recherche en TAL », In *TAL*, 48/1, pp. 67-91.

[Pustejovsky et al. 2003] Pustejovsky J., Castaño J., Ingria R., Saurí R., Gaizauskas R., Setzer A., Katz G. (2003) « TimeML: Robust Specification of Event and Temporal Expressions in Text ». Actes de *IWCS-5, the Fifth International Workshop on Computational Semantics*, Tilburg, Netherlands.

[Routard Paris 2007] *Le guide du routard, PARIS balades* (2007), Hachette.

[Saussure 1974] de Saussure F. (1974) *Cours de linguistique générale*, Lausanne-Paris, Payot, 1906-1911, Payot.

[Silberztein 1993] Silberztein M. (1993) *Dictionnaires électroniques et analyse automatique de textes : le système INTEX*, Paris, Masson.

[Silberztein 1994] Silberztein M. (1994) « INTEX: a corpus processing system ». Actes de *Coling 1994, the 15th conference on Computational Linguistics*, ACL, Kyoto, Japan, pp. 579-583.

[Studer et al. 1998] Studer R., Benjamins V. R., Fensel D. (1998) « Knowledge engineering: principles and methods », In *IEEE Transactions on Data and Knowledge Engineering*, 25(1&2), pp. 161-197.

[Teissèdre et al. 2010a] Teissèdre C., Battistelli D., Minel J.-L. (2010) « Du texte au portail sémantique : cas d'utilisation lié à des données temporelles ». Actes de *IC 2010, 21èmes Journées francophones d'Ingénierie des Connaissances*, Nîmes.

[Teissèdre et al. 2010b] Teissèdre C., Battistelli D., Minel J.-L. (2010) « Resources for Calendar Expressions Semantic Tagging and Temporal Navigation through Texts ». Actes de *LREC 2010, the seventh international conference on Language Resources and Evaluation*, Malte.

[Tengli et al. 2004] Tengli A., Yang Y., Ma N. (2004) « Learning Table Extraction from Examples ». Actes de *Coling 2004, the 20th International Conference in Computational Linguistics*, Genève, Suisse.

[Tenier et al. 2006a] Tenier S., Toussaint Y., Napoli A, Polanco X. (2006) « Instantiation of relations for semantic annotation ». Actes de *WT'06, the International Conference on Web Intelligence*, IEEE Computer Society, Hong Kong, China, pp. 463-472.

[Tenier et al. 2006b] Tenier S., Napoli A., Polanco X., Toussaint X. (2006) « Annotation sémantique de pages web ». Actes de *EGC 2006, Extraction et Gestion des Connaissances*, Lille, France.

[TramedWeb 2006] *Traces d'usage et médiations éditoriales dans les grands corpus du Web – TramedWeb* (2006), Projet scientifique.

[Uren et al. 2006] Uren V., Cimiano P., Iria J., Handschuh S., Vargas-Vera M., Motta E., Ciravegna F. (2006) « Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art », In *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 4(1), pp. 14-26.

[Vaillant 1999] Vaillant P. (1999) *Sémiotique des langages d'icônes*, Paris, Honoré Champion.

[Valette 2004] Valette M. (2004) « Sémantique interprétative appliquée à la détection automatique de documents racistes et xénophobes sur Internet », In Enjalbert P., Gaio M. : *Approches Sémantiques du Document Numérique, Actes du 7e Colloque International sur le Document Electronique*, Europia Productions, pp. 215-230.

[Vatant 2006] Vatant B. (2006) *Représentation du temps dans les ontologies du tourisme*, document interne.

[Vatant 2008] Vatant B. (2008) *EIFFEL Ontologie Version 2*, document interne.

[Wang & Hu 2002] Wang Y., Hu J. (2002) « A Machine Learning Based Approach for Table Detection on The Web ». Actes de *WWW 2002, the Eleventh International World Web Conference*, ACM, Hawaii, USA, pp. 242-250.

[Weiser 2008] Weiser S. (2008) « Informations spatio-temporelles et objets touristiques dans des pages web : repérage et annotation ». Actes de *Recital 2008, 10ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*, Avignon.

[Weiser et al. 2008] Weiser S., Laublet P., Minel J.-L. (2008) « Automatic identification of temporal information in tourism web pages ». Actes de *LREC 2008, the sixth international conference on Language Resources and Evaluation*, Marrakech, Maroc.

[Weiser et al. 2009] Weiser S., Coste M., Amardeilh F. (2009) « Chaîne de traitement linguistique : du repérage d'expressions temporelles au peuplement d'une ontologie de tourisme ». Actes de *TALN'09, 16ème conférence sur le Traitement Automatique des Langues Naturelles*, Senlis.

[Widlöcher & Bilhaut 2008] Widlöcher A., Bilhaut F. (2008) « Articulation des traitements en TAL », In *TAL*, 49/2, pp. 73-101.

ANNEXES

A. Documentation technique

Dans cette section, je présente toutes les informations utiles pour l'installation et l'utilisation d'Adetoa. Les informations relatives à sa structure et à son fonctionnement interne sont données dans l'annexe B.

1. Paramètres du module Adetoa

Adetoa attend cinq paramètres. Le premier est le chemin d'un fichier à analyser ou du dossier dans lequel on aura placé les fichiers XML à analyser : si un seul fichier est donné en paramètre, il sera analysé directement ; s'il s'agit d'un dossier contenant plusieurs fichiers, Adetoa compte le nombre de fichiers et lance le traitement uniquement sur les fichiers dont l'extension est « .xml ». Le second paramètre indique l'emplacement du logiciel Unitex, tandis que le troisième précise son espace de travail. Le quatrième paramètre contient le nom du graphe à appliquer. Enfin, le dernier contient le chemin du dossier temporaire dans lequel des fichiers intermédiaires seront stockés.

Les programmes d'Unitex étant utilisés comme des outils externes, en ligne de commande, ces paramètres permettent de préciser où ils se trouvent et facilitent donc la portabilité du logiciel : rien n'est imposé, chacun est libre de mettre ces programmes où il le souhaite sur son ordinateur. Le format des chemins d'accès dépend du système utilisé (Windows, Linux...) mais JAVA ayant l'avantage d'être multi-plateforme, il aurait été dommage que cela empêche d'en profiter. Le fait que les chemins puissent avoir des formats différents a donc été pris en compte.

2. Utilisation en ligne de commande

Pour appeler Adetoa en ligne de commande, il faut l'exécuter comme un programme JAVA au format JAR, avec des paramètres. Voici un exemple, sous Linux, d'appel à Adetoa. Dans le dossier temporaire (dernier paramètre) seront stockés les fichiers XML contenant les résultats.

```
java -jar '/home/steph/Bureau/adetoa.jar'
   "/home/steph/Bureau/test1"
   "/home/steph/Programmes/Unitex_1.2"  "/home/steph/Bureau/Unitex/French"
   "/home/steph/Bureau/Unitex/French/Graphs/lieux/localisation"
   "/tmp/adetoa/"
```

Sous Windows, la commande est la même, seul le format des chemins est différent.

3. Utilisation comme un objet dans un programme JAVA

Pour faire appel à Adetoa dans le cadre d'un programme JAVA, il faut instancier la classe *Lanceur* puis appeler la méthode *lancer* avec en argument, les paramètres. Voici un extrait du code. Les résultats du traitement sont renvoyés dans un tableau d'objets de type « Document » appelé *resultats*.

```
Lanceur l = new adetoa.Lanceur();
Document[] resultats = l.lancer(new String[]
   {"/home/steph/Bureau/test1",
   "/home/steph/Programmes/Unitex_1.2",
   "/home/steph/Bureau/Unitex/French",
   "/home/steph/Bureau/Unitex/French/Graphs/lieux/localisation",
   "/tmp/adetoa/"});
```

4. Encodage des caractères

Les fichiers fournis en entrée par le projet Eiffel sont des fichiers XML. Ils sont encodés au format UTF-8. Unitex manipule des textes au format Unicode Little-Endian codé sur 16 bits. Lorsqu'Adetoa crée le fichier texte à fournir à Unitex, il l'encode donc à ce format. En sortie d'Unitex, les fichiers sont toujours au format Unicode Little-Endian. Le résultat étant souhaité au format UTF-8, la conversion inverse est réalisée. Cette conversion est nécessaire pour tenir compte, par exemple, des caractères accentués.

Se pose aussi un problème d'échappement des caractères protégés : Unitex insère les balises comme du texte. Certains caractères déjà présents dans le texte peuvent donc poser problème lors de la conversion de ce texte en XML. Par exemple, les caractères « < » ou « > » sont interprétés par le convertisseur XML. Ces caractères sont donc échappés avant le traitement par Unitex pour qu'ils ne se confondent pas avec les annotations insérées par les transducteurs et qui doivent ensuite être interprétées comme des balises XML.

5. Installation

Unitex doit être téléchargé sur le site http://www-igm.univ-mlv.fr/~unitex/download.html. C'est la version 1.2 du logiciel qui est utilisée dans le projet Eiffel. Lors de la première exécution d'Unitex, celui-ci permet de définir un dossier personnel (sous Windows, l'utilisateur choisit ce dossier, sous Linux, Unitex indique simplement le dossier créé) dans lequel seront stockées toutes les données personnelles (graphes, dictionnaires, etc.).

Les graphes et transducteurs doivent tous être stockés dans un seul dossier, à l'intérieur du dossier personnel d'Unitex, en respectant, pour la langue donnée, l'architecture des dossiers prévue (espace de travail > French > Graphs).

Le dictionnaire de communes créé pour Adetoa (communes.dic) doit être copié dans le dossier d'Unitex (Unitex 1.2 > French > Dela).

6. Sortie

Lorsqu'Adetoa est utilisé par un programme JAVA, la sortie est, comme on l'a vu plus haut, constituée d'un objet JAVA. Cet objet est un tableau de « Document » contenant autant d'éléments que de fichiers analysés. Il peut donc ne contenir qu'un seul élément, lorsque seul un fichier a été fourni en argument, ou plusieurs éléments lorsque c'est un dossier contenant plusieurs fichiers XML qui est fourni.

Lorsqu'il est utilisé en ligne de commande, Adetoa reconstruit autant de fichiers XML que de fichiers se trouvant dans le tableau d'objets « Document ». Cela a l'avantage de permettre de consulter manuellement les résultats facilement. Ces fichiers sont nommés par un numéro qui correspond à leur position dans le tableau de « Document ».

B. Structure d'Adetoa

Adetoa est composé de quatre classes qui sont présentées ici.

1. Lanceur

Lanceur est la classe principale d'Adetoa. Elle contient la méthode *main* qui permet de l'appeler en ligne de commande ainsi que la méthode *lancer* qui permet d'utiliser Adetoa dans un autre programme JAVA.

Selon les paramètres qui sont donnés, elle permet de rechercher les fichiers à analyser dans un dossier ou d'analyser un fichier donné, et lance le traitement en créant une instance de la classe *analyseur_structurel*.

2. Analyseur_structurel

Analyseur_structurel est la classe qui contient les méthodes chargées des transformations au niveau des formats de fichiers et d'encodage. Son rôle est d'extraire le contenu textuel des fichiers donnés en entrée puis de les traiter à l'aide des classes *Unitex* et *Transformeur*. Au retour, elle récupère les données annotées et reconstruit un document XML valide.

3. Unitex

La classe *Unitex* permet de gérer les appels aux différents programmes d'Unitex. L'annexe C décrit en détail les programmes utilisés.

4. Transformeur

La classe *Transformeur* correspond au module d'application des règles de transformations. Ses méthodes utilisent XPath pour retrouver les éléments qui nécessitent une transformation et appliquent les règles présentées au chapitre 5, paragraphe 3.3.

C. Description de l'ensemble des programmes d'Unitex utilisés

Unitex se compose de nombreux programmes effectuant chacun une tâche précise. Cette annexe présente les programmes utilisés par Adetoa. Pour appliquer un transducteur à un texte afin d'effectuer un repérage, Unitex a besoin que plusieurs tâches soient accomplies. Elles se décomposent en deux phases. La première est le pré-traitement du texte, la seconde est l'application de la grammaire.

1. Le pré-traitement

Le pré-traitement est chargé de la normalisation des séparateurs, du découpage en unités lexicales, de la normalisation de formes non ambiguës, du découpage en phrases et de l'application des dictionnaires :

Normalize effectue une normalisation des séparateurs (l'espace, la tabulation et le retour à la ligne) du texte. Toute suite de séparateurs est remplacée par un seul séparateur ; la distinction entre retours à la ligne et espaces est conservée.

Fst2Txt est un programme qui applique un transducteur au texte. Il est utilisé une première fois pour le découpage du texte en phrases. Cette étape permet de définir les unités de traitement linguistique et ne consiste pas simplement à trouver les marques de ponctuation : en effet une analyse plus poussée est nécessaire pour définir par exemple si le point marque la fin de la phrase ou s'il se trouve à la fin d'une abréviation. Une seconde fois, ce programme est utilisé pour normaliser certaines formes non ambiguës, lorsqu'une même unité peut avoir plusieurs formes, comme par exemple « l'on» et « on ».

Tokenize découpe le texte en unités lexicales. Ce programme tient compte de la langue du texte.

Dico applique des dictionnaires au texte afin de construire le dictionnaire des formes présentes dans le texte. Le texte doit au préalable avoir été découpé en unités lexicales par le programme Tokenize.

Ces quatre programmes constituent le pré-traitement du texte. Une fois celui-ci effectué, les transducteurs de repérage et d'annotation peuvent être appliqués au texte ainsi obtenu.

2. L'application des transducteurs

Grf2Fst2 compile la grammaire pour obtenir un fichier .fst2, utilisable par Unitex. Ce programme vérifie que la grammaire ne contient pas d'erreur ou de boucle infinie et compile également les sous-graphes quand il y en a.

Locate permet d'appliquer une grammaire au texte. Ce programme construit un fichier référençant les occurrences ainsi trouvées dans le texte. L'un de ses paramètres permet à l'utilisateur de choisir entre les modes « longest matches », « shortest matches » ou « all matches ». Le premier donne la priorité aux séquences les plus longues, le deuxième aux plus courtes et le troisième donne toutes les séquences reconnues. Sauf mention contraire, tous les graphes que j'ai créés sont conçus pour fonctionner en mode « longest matches ». Le choix du mode est très important car il influe énormément sur les résultats obtenus. Par exemple, si l'on souhaite repérer le texte *sur rendez-vous le reste du temps* et que l'on applique pour ce faire le transducteur présenté dans la figure 72 mais en mode « shortest match », on repérera simplement *sur rendez-vous* et non la séquence complète.

C'est aussi ce programme qui permet de choisir si l'on veut tenir compte ou non des sorties : celles-ci peuvent être ignorées, les séquences produites peuvent remplacer les séquences repérées ou les séquences produites peuvent être insérées dans le texte. C'est ce dernier mode qui est utilisé dans Adetoa.

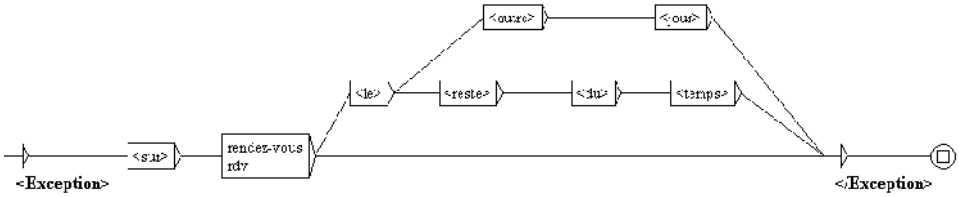


Figure 72 : Transducteur « rdv »

Concord prend en paramètre le fichier produit par le programme Locate et crée une concordance tenant compte des transductions associées aux occurrences. Il peut produire un fichier .txt ou .html. D'autres paramètres permettent de définir la taille de contexte, l'apparence souhaitée, l'ordre des résultats, etc.

Ces différents programmes utilisent de nombreux fichiers temporaires. Souvent la sortie de l'un constitue l'entrée du suivant. Ces fichiers sont stockés dans le répertoire du texte traité. L'un des paramètres d'Adetoa permet de définir le répertoire où stocker tous ces fichiers temporaires.

F. Guide pour l'évaluation (fourni à l'évaluateur)

Évaluation de quatre tâches : repérage des informations temporelles et de localisation, annotation de ces informations, transformation sur quelques expressions temporelles, liage des données.

Dans la page, ne s'intéresser qu'au contenu de la balise <Text>.

Méthode globale :

- Ouvrir la page dont le numéro est dans la première colonne du tableau de repérage et de liage
- Évaluer le repérage et le liage
- Évaluer l'annotation et les transformations

1. Le repérage

Quoi repérer ?

Informations pratiques :

- dates touristiques : *concert le 15 mars*
- ouvertures ou fermetures : *ouvert de juin à septembre, du lundi au vendredi, de 8h30 à 19h sauf le 14 juillet*
  - heures
  - dates
  - jours
  - intervalles de jours
  - exceptions

Localisations :

Les informations de localisation contiennent au minimum un nom de ville

- adresses
  - adresse complète (postale) *5, rue de l'église, 58000 Nevers*
  - adresse incomplète (pas de code postal, pas de nom de rue...) *rue de l'église Nevers*
  - nom de ville *Nevers*

Ne pas repérer :

Informations non touristiques / non pratiques :

- dates de modification
- dates historiques
- heure actuelle

Note : « **une expression** » = ensemble des informations temporelles présentées de façon continue. Une expression peut être très longue et composée de plusieurs phrases. Elle peut présenter plusieurs périodes. Mais tant qu'il est question d'informations temporelles pratiques alors c'est considéré comme une seule expression : *Ouvert de juin à septembre, du lundi au vendredi, de 8h30 à 19h sauf le 14 juillet* = 1 expression.

Ouvert de juin à septembre. *Nous accueillons les enfants* ***en juillet*** = 2 expressions.

Démarche d'évaluation :

1) lire la page et chercher ce qui aurait dû être repéré
2) comparer avec ce qui a été repéré

Compter le nombre d'expressions appartenant à chacune des catégories prédéfinies.

Les expressions repérées sont encadrées par la balise <UT> pour les informations temporelles et par la

balise <localisation> pour les informations de localisation.

Repérage							
Repérées complètement	Repérées partiellement sans perte	Repérées partiellement avec perte	Manquées	A tort	Non pertinentes (temporelles)	Repérées en plusieurs fragments	Fragments

Repérées complètement :

Toute l'expression (voir note ci-dessus) est repérée, elle ne continue pas en dehors de la balise encadrante.

Repérées partiellement sans perte :

Une partie de l'expression n'est pas repérée, mais cela n'entraîne pas de perte d'information. Exemple *ouvert du 1er au 15 août, tous les jours* : si *tous les jours* n'est pas repéré, on ne perd pas d'information puisque sans, l'interprétation est la même.

Repérées partiellement avec perte :

Une partie de l'expression n'est pas repérée, et cela entraîne une perte d'information. *Ouvert du 1er au 15 août et du 2 au 7 septembre*. Si *du 2 au 7 septembre* n'est pas repéré : perte d'information.

Manquées :

Expression à repérer (donc information temporelle pratique ou information sur la localisation de l'objet touristique dont il est question) mais qui n'a pas été repérée par Adetoa.

Note : si une expression est repérée partiellement, la partie manquante ne constitue pas une expression manquée.

Note : Si une expression est repérée en plusieurs UT alors, compter 1 dans l'une des trois premières catégories et remplir les cases gris clair : 1 dans repérées en plusieurs fragments et le nombre d'UT dans la colonne fragments.

À tort

L'expression n'aurait pas dû être repérée car elle ne constitue pas une information temporelle pratique ou ne concerne pas la localisation du lieu touristique dont il est question.

Compter séparément les expressions temporelles et les expressions de localisation.

Note : pour faciliter l'évaluation de l'annotation, à chaque expression repérée à tort rencontrée, la marquer dans le tableau d'évaluation de l'annotation.

Non pertinentes

Ne concerne que les informations temporelles. Informations temporelles mais qui ne sont pas des informations pratiques touristiques. Date de modification du site, heure actuelle, date historique...

Note : il s'agit d'informations non repérées (les non pertinentes repérées à tort sont dans la catégorie précédente).

2. Le liage

Quoi lier ?

Le liage sert à regrouper les différentes informations qui concernent un même objet touristique.

La balise <ensemble> encadre les différentes informations liées. Elle peut contenir plusieurs UT.

Dans l'idéal, elle contient une UT, une localisation et un objet. Mais elle peut ne pas contenir d'informations d'une des catégories.

Démarche d'évaluation :

Même démarche que pour le repérage : regarder l'ensemble de la page. Compter, dans chaque page les différents ensembles possibles et les classer dans les catégories prédéfinies. Ne considérer que les informations qui sont repérées.

Liage				
Liage exact				
Ensembles cohérents et complets	Ensembles cohérents mais incomplets	Ensembles incohérents	Ensembles cohérents selon balises	Ensembles manquants

Cohérents et complets

Toutes les informations comprises dans l'ensemble se rapportent bien à un seul objet. Cet ensemble contient toutes les informations qui se rapportent à cet objet et qui sont repérées dans la page.

Cohérents mais incomplets

Toutes les informations comprises dans l'ensemble se rapportent bien à un seul objet. Au moins une autre information de la page se rapporte au même objet mais ne fait pas partie de l'ensemble.

Incohérents

L'ensemble comprend des informations concernant des objets différents.

Ensembles cohérents selon balises

L'ensemble est incohérent ou incomplet selon le liage exact, néanmoins, si on ne regarde que les données annotées, l'algorithme de liage englobe correctement les données balisées. Les ensembles qui entrent dans cette catégorie font aussi partie d'une des catégories précédentes, indiquant si l'ensemble est exact au niveau du sens.

Manquants

Au moins deux informations concernant un objet sont repérées mais aucun ensemble n'est créé.

3. L'annotation

Quoi annoter ?

Adetoa annote les données qui ont été repérées.

Pour les expressions temporelles : période d'ouverture et de fermeture, exceptions, incertitude. Dans les périodes : jour de début et de fin, jour, date, date de début et de fin, heure de début et de fin, exception, incertitude. Les exceptions peuvent contenir des périodes ou simplement du texte brut non balisé.

La balise incertitude marque les informations floues (dates imprécises).

Pour les localisations : adresse (rue avec ou sans numéro), code postal, ville.

Démarche d'évaluation :

Pour chaque expression (temporelle ou de localisation) repérée complètement ou partiellement, juger la qualité de l'annotation en cochant l'une des cases du tableau.

Repérée à tort	Annotation			
A enlever	Parfaitement annotée	Granularité à affiner	Annotation incomplète	Mal annotée

La première colonne, expression, est pré-remplie : elle contient toutes les UT et localisations du corpus. Il faut marquer les expressions repérées à tort (éventuellement au moment de l'évaluation du repérage) : cocher la colonne *repérée à tort*.

Il est en principe inutile de consulter la page Web, toutes les informations sont déjà dans le tableau.

Parfaitement annotée

L'expression est parfaitement bien annotée, avec la granularité la plus fine. Toutes les informations présentes dans l'expression sont correctement annotées.

Cas particulier – **exceptions** : si dans la balise exception, les informations ne sont pas balisées, l'annotation peut quand même être considérée comme parfaitement annotée.

Granularité à affiner

(ou problème d'incertitude – ou frontières mal définies)

L'expression est bien annotée, c'est-à-dire bien typée, mais la granularité des annotations n'est pas la plus fine possible. La granularité pourrait être affinée : les jours sont repérés mais pas les heures...

Si la balise <Incertitude> est là à tort, ou si elle n'est pas là alors qu'elle devrait l'être, cette case est à cocher, à condition que le reste des informations soit bien typé, avec au maximum la granularité la plus fine non atteinte.

Si le repérage n'avait pas les bonnes frontières et que l'annotation ne le « corrige » pas : par exemple une adresse où la boîte postale et code postal sont dans <cp>.

Annotation incomplète

L'expression est annotée partiellement, mais les balises manquantes ne concernent pas uniquement la granularité la plus fine : si les heures sont bien annotées mais les jours non, alors l'expression appartient à cette catégorie.

Les balises qui figurent dans l'expression typent correctement les informations.

Cas particulier – **exceptions** : si dans la balise exception, les informations ne sont pas balisées, l'annotation n'est pas considérée comme incomplète.

Mal annotée

L'expression est mal annotée : au moins une balise ne donne pas le bon type à l'information. (ouverture au lieu de fermeture, etc.). Si le mauvais typage survient dans la balise <exception>, l'expression est aussi considérée comme mal annotée.

Note : pour chaque expression, cocher l'une des cases.

4. Les transformations

Sur quoi faire des transformations ?

L'évaluation des transformations ne concerne qu'un nombre réduit d'expressions temporelles :

- les groupes de jours doivent être convertis en collection de jours, balise <jour>. Il ne doit plus y avoir de balises <jour_debut> et <jour_fin>.
- sans horaire dans l'UT, les jours de fermeture doivent être convertis en périodes d'ouverture (fermé le mardi => ouvert lundi, mercredi, etc.), ou enlevés de la période d'ouverture existante.
- les deux règles peuvent se combiner.

Une balise <inference> indique quand le système a procédé à une transformation et contient l'annotation d'origine.

L'expression *tous les jours* est également transformée en collection de jours, mais dès l'annotation, et non pas par le module de transformations, elle n'est pas à prendre en compte.

Démarche d'évaluation :

Même démarche que pour l'évaluation de l'annotation : ne regarder que les expressions pré-remplies dans le tableau.

Identifier les cas où une transformation aurait dû avoir lieu et juger son apport.

Transformation			
Transformation bien faite	**Transformation mal faite**	**Transformation à tort**	**Transformation manquante**

Transformation bien faite

L'expression correspond à une de celles pour lesquelles il existe une règle de transformation et après la transformation, toutes les informations présentes dans l'expression sont conservées (bien annotées)

Transformation mal faite

L'expression correspond à une de celles pour lesquelles il existe une règle de transformation mais, après la transformation, le sens n'est plus le même (information manquante – information erronée, une ouverture devient une fermeture ou il manque des jours par exemple).

Transformation à tort

Aucune transformation n'était attendue et pourtant une a lieu.

Transformation manquante

Une transformation aurait dû avoir lieu mais elle n'a pas eu lieu.

Note : pour chaque cas où une transformation est nécessaire et / ou a eu lieu, cocher l'une des quatre cases du tableau.

Repérage et typage d'expressions temporelles pour l'annotation sémantique automatique de pages Web

Application au e-tourisme

Stéphanie Weiser

Université Paris Ouest Nanterre La Défense
École doctorale 139 – Connaissance, langage, modélisation
Laboratoire MoDyCo – UMR 7114

Résumé : Cette thèse présente Adetoa, système dédié au repérage et à l'annotation sémantique automatique d'expressions temporelles dans des pages Web pour une application de e-tourisme. Une étude linguistique détaillée a permis de mettre en avant les caractéristiques et la complexité de l'expression de la temporalité dans les pages Web touristiques. Une étude sémiotique de ce type de pages a montré que les données y étaient organisées de manière fort variée, ne présentant aucune régularité, ce qui rend difficile voire parfois impossible l'automatisation de leur analyse.

Ces analyses ont mené à l'élaboration d'un ensemble important de transducteurs (avec Unitex) pour les tâches de repérage et d'annotation des expressions temporelles, ce qui constitue une ressource pouvant être généralisée. De plus, d'autres informations du domaine touristique sont repérées : les objets du tourisme et les adresses. Des transducteurs de liage permettent de grouper toutes les informations concernant une même offre touristique.

Pour l'annotation et l'intégration d'Adetoa à la chaîne de traitement du projet Eiffel, un schéma d'annotation et des règles de transformations ont été mis au point. Sans en être un calque direct, le schéma d'annotation suit une ontologie du tourisme. Il permet ainsi de rester au plus près des expressions linguistiques de manière à les caractériser finement. L'ontologie a ensuite pu être adaptée en conséquence, pour un meilleur stockage des données dans la base de connaissance qui lui correspond.

L'évaluation d'Adetoa, présentée dans cette thèse, a montré des résultats satisfaisants aussi bien d'un point de vue théorique que pour cette application industrielle.

Mots-clés : Extraction d'information automatique, ontologie, schéma d'annotation, expressions temporelles, e-tourisme, transducteurs.

Abstract: *This thesis presents Adetoa, a system designed to automatically locate temporal expressions in Web pages and tag them with semantic annotations, in the field of e-tourism. A detailed linguistic study has revealed that the expression of temporal information in Web tourism pages is complex and has specific properties. A semiotic study of these pages has pointed out that data are organised in various ways, without any regularity. An automatic analysis of their structure is therefore difficult or even sometimes impossible.*

These analyses have led to the development of a large number of transducers (under Unitex) for the extraction and mark-up tasks. They can be regarded as a generally applicable resource. Other tourist information is also extracted, such as tourist objects and addresses. Linking transducers have been developed to group all the information concerning one tourist destination.

An annotation scheme and transformation rules have been developed in order to mark the annotations and to integrate Adetoa in the processing chain of the Eiffel project. The annotation scheme is based on a tourism ontology but is not a direct replica, thus enabling the expressions to be accurately characterized on a linguistic level. The ontology has then been adapted accordingly, so that the information can more easily be included in the corresponding knowledge base.

The evaluation of Adetoa, which is detailed in the last chapter, showed satisfying results, both on a theoretical level and for industrial purposes.

Keywords: *Automatic information extraction, ontology, annotation scheme, temporal expressions, e-tourism, transducers.*