Portage linguistique d'applications de gestion de contenu

Najeh Hajlaoui, Christian Boitet

Portage linguistique d'applications de gestion de contenu

Najeh HAJLAOUI, Christian BOITET
GETALP, laboratoire LIG, Université Joseph Fourier, CNRS, INPG, INRIA

385 rue de la Bibliothèque, BP 53
38041 Grenoble, cedex 9, France
Najeh.Hajlaoui@imag.fr
Christian.Boitet@imag.fr

Résumé

Nous nous intéressons à la multilinguïsation, ou "portage linguistique" (plus simple que la localisation) des services de gestion de contenu traitant des énoncés spontanés en langue naturelle, souvent bruitée mais contrainte par la situation. Tout service de ce type (soit `App`) est muni d'un extracteur de contenu `(EC-App)` produisant une forme interne spécifique `(CRL-App)` à partir de la langue "native" L1. Trois stratégies de portage vers une langue L2 ont été étudiées : (1) traduction des énoncés de L2 vers L1 ; (2) localisation "interne", i.e. adaptation à L2 de l'EC, donnant `EC-App-L2` ; (3) localisation "externe", i.e. adaptation d'un EC existant pour L2 au domaine et à la représentation de contenu de App `(EC-X-L2-App)`. Le choix de la stratégie est contraint par la situation traductionnelle : types et niveau d'accès possibles, ressources disponibles, compétences langagières et linguistiques des intervenants pour la multilinguïsation des applications. Les stratégies (2) et 3) ont été expérimentées sur le portage d'arabe en français de la partie de CATS concernant l'occasion automobile. CATS est une application de e-commerce construite par D. Daoud et déployée en Jordanie sur le réseau FastLink. Elle traite des petites annonces envoyées par SMS et concernant l'occasion automobile (Cars), l'immobilier à Amman (RealEstate), l'emploi (Jobs), et autres (Misc). En localisation interne, la partie grammaticale a été très faiblement modifiée, ce qui prouve que, malgré la grande distance entre l'arabe et le français, ces deux sous-langages sont très proches l'un de l'autres, une nouvelle illustration de l'analyse de R. Kittredge.

Introduction

La multilinguïsation des services de e-commerce traitant des énoncés spontanés en langue naturelle est un problème difficile, et de ce fait très peu de services le font. Des facteurs principaux dépendant de la situation traductionnelle interviennent :

- le niveau d'accès aux ressources des applications, avec quatre cas possibles: accès complet au code source, accès limité à la représentation interne, accès limité au dictionnaire, et aucun accès.
- le niveau de compétence langagière et linguistique des intervenants dans la « portage » vers une nouvelle langue : connaissance des deux langues, source et cible, et compétences en TALN.

La multilinguïsation ou le « portage linguistique » dont nous parlons n'est pas une « localisation », qui implique une adaptation à un autre contexte culturel. Il s'agit uniquement de permettre l'accès dans plusieurs langues à un service de e-commerce, tel qu'il est et où il est.

Nous présentons d'abord le besoin en localisation d'application d'extraction de contenu suivi d'une analyse des méthodes possibles. Ensuite, nous illustrons notre étude par le portage linguistique (arabe vers français) d'une application d'e-commerce déployée, pour laquelle les facteurs présentés ci-dessus sont assurés, et plusieurs stratégies de

multilinguïsation sont alors possibles. Nous présentons deux stratégies de localisation, dites « interne » et « externe » et nous évaluons leurs résultats.

1. Contexte du problème

1.1 Application munies d'un extracteur de contenu

Les applications qui nous intéressent sont celles qui donnent de la valeur ajoutée par le traitement du contenu de messages spontanés en langue naturelle. Les principaux types d'applications et de services susceptibles de le faire sont : la catégorisation de documents divers (Bélanger 2003) (dépêches de l'AFP « Agence France-Presse », brèves des différentes bourses, messages de clients à un serveur de SAV), l'extraction d'informations pour nourrir ou consulter une base de données (petites annonces ciblées, FAQ intelligentes, indexation ciblée à un domaine/métier), les hotlines automatisées, et plus généralement l'interaction en langage naturel avec des bases de données (Califf 1998).

Ces applications reposent en général sur un « extracteur de contenu » (EC) (Cardie 1997) plus ou moins puissant, produisant une représentation formelle du contenu extrait. Il peut s'agir d'une liste de propriétés (couples attribut-valeur), ou d'une forme logique, ou d'une forme arborescente plus ou moins « plate » (IF de CSTAR/Nespole !) (Besacier, Blanchon et al. 2001), etc.

1.2 Importance croissante de la multilinguïsation des services

La plupart des services à destination d'utilisateurs finals, et dans une seule langue, sont en anglais. Par exemple, CISCO (http://www.cisco.fr/) ne distribuait sa documentation qu'en anglais jusqu'à un passé récent. En Asie (Chine, Corée, Japon), bien que tous les utilisateurs visés aient étudié l'anglais 8 ou 9 ans, l'anglais technique n'est pas du tout bien compris, et les centres d'appel, qui coûtent très cher, étaient débordés. La production de traductions automatiques (Systran) a permis de diminuer notablement le recours aux centres d'appel, malgré leur assez mauvaise qualité.

En ce qui nous concernent, nous visons non pas à créer des services de traduction comme dans cet exemple, mais à multilinguïser l'accès à des services comme pourrait l'être un centre d'appel ou un service de SAV automatisé.

Notre exemple principal sera un système de petites annonces déployé à Amman en arabe (Daoud 2005) : notre but sera alors de le rendre accessible en français à des francophones résidant à Amman, puis dans d'autres langues pour les locuteurs de ces langues.

La nécessité de services multilingues sur place est très claire dans des pays multilingues (Canada, Inde, USA), mais elle apparaît aussi dans des pays monolingues (France) à cause du tourisme et de la nouvelle mobilité.

1.3 Malgré l'intérêt des énoncés spontanés, peu de services les traitent

L'interaction avec un service au moyen de formulaires présente des limites :

- l'interaction n'est pas naturelle ;
- surtout, les formulaires et les menus, à caractère modal et figé, ne permettent pas aux utilisateurs d'exprimer ce qu'ils veulent, comme par exemple de décrire le contexte du dysfonctionnement d'un logiciel ou d'un graveur de DVD.

En e-commerce, la navigation par mots-clés pilotée par des menus, telle qu'on la trouve dans la plupart des sites commerciaux, tend à accabler et frustrer les utilisateurs avec des interactions prolongées et rigides (Ritchie 1995).

L'intérêt de l'utilisateur pour un site particulier diminue exponentiellement avec l'augmentation du nombre de clics de souris (Huberman, Pirolli et al. 1998). Par

conséquent, le raccourcissement du chemin d'interaction pour fournir des informations utiles devient important.

Beaucoup de sites de e-commerce essayent de résoudre ce problème en fournissant des possibilités de recherche par mots-clés. Cependant, il s'agit du grand public, et outre le défaut d'ergonomie signalé plus haut, il y a un problème de compétence, car il faut que les utilisateurs connaissent le jargon spécifique du domaine.

Peu de services traitent des énoncés spontanés et cela, même en contexte monolingue. En interrogeant plusieurs moteurs de recherche (Google, Altavista, Tiscali…) sur les applications traitant des énoncés spontanés en langues naturelles, avec des requêtes variées[1], nous avons obtenu très peu de résultats positifs, et très peu de renseignements sur le fonctionnement interne de tels services et leur multilinguïsation, quand on en trouve. Il semble qu'il y en a encore très peu ! Nous avons cependant trouvé :

- Pertinence Summarizer (Lehman 1996), un logiciel de résumé automatique de textes multilingues ;
- Amilcare (Ciravegna 2001), un système adaptatif d'extraction d'information ;
- NLSA « Naturel Language Sales Assistant », un système basé sur le dialogue à travers le Web déployé par IBM) ;
- CATS « Classifieds Ads Through SMS » (Daoud 2006), un système d'achat et de vente de voitures d'occasion et d'immobilier basé sur l'utilisation des SMS en arabe.

Notre hypothèse quant aux raisons possibles qui font que peu de services traitent les ESLN (Enoncés Spontanés en Langues Naturelles) est que ce travail présente des difficultés inhérentes et que la multilinguïsation est perçue comme un gros problème. On rencontre en effet des problèmes un peu analogues à ceux de l'oral : grammaire « non standard » (plus ou moins proche de l'oral), abondance d'erreurs (fautes de frappe, d'orthographe), utilisation de conventions typographiques propres au contexte (abréviations propres aux SMS et à la langue "tchatée", utilisation d'émotions pour noter les émotions et l'affect).

Assez souvent, on est dans un sous-langage relativement éloigné de la langue générale, comme les petites annonces ou des alarmes/avertissements (trafic routier, catastrophes naturelles). On ne peut pas utiliser des outils faits pour du langage écrit général et « propre ». De plus, il faut « traduire » les ESLN dans un formalisme de représentation de contenu (Content Representation Language ou CRL), et chaque application possède son propre formalisme.

1.4 Nécessité d'une approche spécifique à chaque sous-langage et inefficacité des outils faits pour les langues générales

Les applications traitant les ESLN utilisent en général une représentation du contenu. On trouve plusieurs formes de représentation de contenu : listes <attribut, valeur(s)>, structures de traits typés, expressions logiques (Prolog), expressions logico-fonctionnelles, objets (classes (méthodes, attributs), instances).

Par exemple, le système CATS utilise une représentation de type `Propriété=couleur{objet=saloon, valeur=bleu}`, dans le domaine de l'occasion automobile, pour exprimer que la couleur d'une voiture (saloon) est bleue.

---

[1] Requêtes : localizing natural language message processors, localization NLP free text, localization NLP interfaces, multilingual customer message processing, multilingual customer messages tools, multilingual, customer relationship processing, multilingual NLP e-commerce, multilingual online sales customer support, multilingual online sales NLP customer support, categorizing natural language messages, handling natural language messages in business, Natural Language Conversational Interface in Online Sales….

Souvent, une application possède son propre formalisme et l'adaptation ou le portage d'un « extracteur de contenu » d'une application à l'autre, même pour la même langue, est difficile car il faut pouvoir garantir un niveau minimum de qualité, (exactitude et complétude) de l'extraction de contenu, de pertinence des réponses produites (traitement et gestion) et de l'adéquation linguistique de ces réponses.

La donnée primaire à fournir pour un portage linguistique d'une application traitant les ESLN est un corpus d'ESLN relatif à la même tâche et dans la langue cible, ce qui n'est pas toujours facile à trouver. Il faut travailler le plus souvent par adaptation, simulation et imagination. Par exemple, il nous faudra construire un corpus, au départ nécessairement imaginaire, de SMS supposés écrits par des francophones désirant acheter ou vendre de l'occasion automobile en Jordanie.

Généralement, le « sous-langage » (Sekine 1994) des énoncés spontanés en langue naturelle associés à un service donné est relativement éloigné de la « langue générale ». Par conséquent, les outils existants faits pour la langue générale ne marchent pas, qu'il s'agisse d'outils de TA (Traduction Automatique) ou d'EC (Extraction de Contenu). Une approche spécifique à chaque sous-langage (Slocum 1986) s'impose alors.

2. Approches possibles

2.1 Traduction automatique des énoncés vers la langue originale

Une première idée consiste à traduire les ESLN de la « nouvelle langue » L2 vers la langue originale L1 de l'application à localiser. L2 est « cible » du portage, mais « source », pour la traduction !

Quelle que soit l'approche linguistique choisie pour cette TA (Traduction Automatique), il faut créer un système spécialisé, et donc disposer d'un corpus parallèle L2//L1. On peut l'obtenir en traduisant le corpus des ESLN, disponible par hypothèse en L1, sachant bien que le corpus parallèle obtenu sera « à l'envers » et donc nettement moins représentatif qu'un corpus L2//L1. Mais enfin c'est un début.

La question qui se pose ensuite est la taille du corpus nécessaire. Si l'on utilise une approche calculatoire de la TA « fondée sur des corpus » (TA statistique (Koehn 2004), TA par analogie (Lepage 2006)), on sait qu'il faut d'énormes corpus s'il s'agit de langue générale (entre 50 et 200 millions de mots d'après K. Knight et Ph. Koehn), bien plus grands que ceux disponibles après deux ou trois ans de fonctionnement d'un e-service. Il est possible que, dans le cas de sous-langages restreints, des corpus beaucoup plus petits suffisent, mais ce n'est qu'une hypothèse, et nous n'avons trouvé aucune étude sur le sujet. Nous avons commencé à travailler sur ce point, mais n'avons pas encore de résultat.

Si l'on utilise une approche calculatoire « par règles », il faut disposer de linguistes computationnels, ce qui est rare.

En résumé, le portage par réalisation d'un système de TA L2→L1 est possible en théorie, mais nous ne sommes pas encore en mesure de déterminer si on peut le faire, sans linguistes qualifiés, par des méthodes d'apprentissage automatique.

2.2 Réalisation d'un nouvel extracteur de contenu (EC) pour chaque langue visée

La réalisation d'un nouvel EC peut se faire par plusieurs méthodes.

La première solution est d'adapter l'EC existant de L1 à L2, mais cela n'est viable, que si

- les développeurs acceptent d'ouvrir leur code ou leur boîte à outils (BàO) à des collaborateurs nécessairement éphémères ;
- ce code ou cette BàO est assez facile à maîtriser ;

- les ressources ne sont pas trop lourdes à créer (en particulier le dictionnaire s'il existe) ;
- la maintenance peut ensuite se faire à coût faible, par des collaborateurs épisodiques.

Cette méthode d'adaptation interne de l'EC natif nécessite bien sûr une formation de l'équipe de localisation aux outils et aux méthodes utilisées.

La deuxième solution consisterait, pour une société voulant offrir des services de multilinguïsation/portage, à implémenter un EC générique et à l'adapter à chaque situation (langue, sous-langage, domaine, CRL — « Content Representation Language », tâche, contraintes). On verra ci-dessous que cela semble très difficile à envisager actuellement.

Dans beaucoup de contextes, les « multilinguïseurs » n'auront donc pas accès à l'EC de l'application, ni à un EC « universel ». Une troisième solution pourrait alors être de rechercher et d'adapter un EC existant et disponible, soit pour la langue L2, pour un domaine et/ou une tâche différents, soit pour le même domaine et la même tâche, pour une autre langue (différente de L2).

Dans ce qui suit, nous illustrons la première et la troisième méthode par le cas du portage d'arabe en français du système CATS. Cette expérience vise à permettre à des francophones vivant en Jordanie (à Amman) et disposant d'un mobile d'envoyer des SMS pour vendre et acheter des voitures d'occasion.

3. Illustration : localisation du système CATS

3.1 Présentation de CATS

CATS est un système d'achat et de vente basé sur l'utilisation des SMS en arabe (Daoud 2006). Il est déployé en Jordanie par la société FastLink, le plus gros opérateur local de téléphonie mobile. Bien qu'il n'y ait pas de transaction directe, CATS aide les utilisateurs à vendre et acheter sans avoir à se déplacer, en les mettant en contact.

Les SMS sont envoyés à un numéro spécial unique[2]. Leur contenu est extrait dans le langage CRL-CATS, puis transformé en requêtes SQL. Une réponse est envoyée automatiquement à l'expéditeur du SMS en cas de correspondance de la demande avec l'une des propositions. Si rien n'est trouvé, le système le dit dans sa réponse, et réessaie plus tard quand la base de données change.

CATS a deux principaux composants : un EC (Extracteur de Contenu) et un gestionnaire de requêtes QM « Query Manager ».

Voici un exemple de SMS et de sa représentation CRL-CATS produite automatiquement.

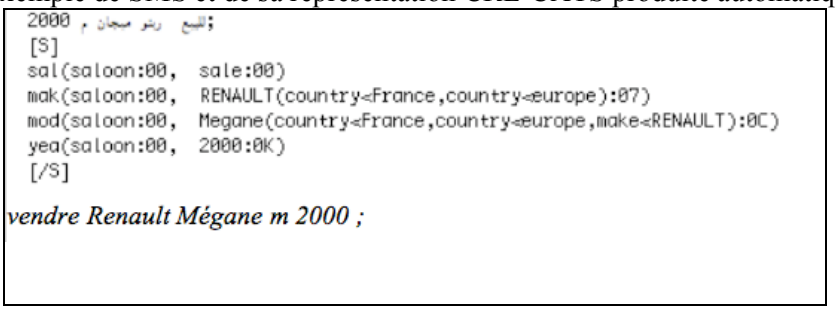

Figure 1 : exemple de représentation CRL-CATS

Dans cet exemple, la propriété est le type `(make)`, l'objet est une voiture `(saloon)` et la valeur est égale à `(RENAULT(country<France, country<europe))`. Pour la propriété

[2] Ils sont enregistrés, ce qui nous fournit un corpus d'ESLN en arabe pour CATS.

modèle `(mod)`, l'objet est une voiture `(saloon)` et la valeur est égale à `(Megane(country<France, country<europe, make<RENAULT))`. Pour la propriété année `(yea)`, l'objet est une voiture `(saloon)` et la valeur est `(2000)`.

Le QM permet de convertir la représentation CRL-CATS vers un texte SQL (requête de sélection pour l'achat et/ou requête la d'insertion pour la vente). Il traite aussi les situations dans lesquelles aucune réponse n'a été trouvée.

3.2 Besoin d'un corpus de démarrage

Pour toutes les méthodes de localisation de CATS vers le français, la première chose à faire est de constituer un « corpus de démarrage » en français, analogue à celui utilisé par D. Daoud au départ de son projet pour l'arabe. Cela est évidemment nécessaire pour étudier la forme syntaxique des SMS à traiter en français, et aussi pour voir à quelles variantes lexicales il faut s'attendre.

Une première idée pour fabriquer un corpus français de démarrage est de partir du corpus CATS en arabe et de le traduire en « SMS français spontanés », supposés être envoyés (en Jordanie) par des francophones.

Une traduction « brute » produite par un non Français est généralement très différente d'une traduction naturelle et fonctionnelle produite par un Français, c'est-à-dire de ce que dirait un Français d'une façon spontanée dans la même situation. Nous avons évalué cette différence entre traduction brute (ou littérale) et naturelle (ou fonctionnelle) en calculant la distance d'édition entre les deux traductions. La mesure de distance moyenne trouvée est de 21,88 (Hajlaoui 2006), sachant que les SMS ne sont pas très longs (moins de 100 caractères en moyenne). La distance moyenne trouvée est le nombre minimal de suppressions, insertions ou remplacements de lettres nécessaires pour transformer une traduction brute (ou littérale) en une traduction naturelle (ou fonctionnelle).

Nous avons montré dans un article antérieur (Hajlaoui 2006) que les traductions directes d'un corpus réel constitué de phrases naturelles ne donnent pas de résultats naturels en français. Tenant compte de ce résultat, nous avons essayé de produire un petit corpus français fonctionnellement équivalent au corpus arabe initial. Afin de développer ce corpus, nous avons adopté la technique suivante : à partir d'un ensemble de 50 SMS révisés et jugées fonctionnels, nous avons construit un ensemble plus grand en formant des combinaisons différentes des arguments utilisés (type, modèle, année, couleur, prix…). Par exemple, on remplace une année par une autre (*je cherche une voiture modèle 98*) → (*je cherche une voiture modèle 99*) ou une marque par une autre, une couleur par une autre (*A vendre BMW rouge*) → (*A vendre PEUGEOT noire*), etc.

3.3 Adaptation interne

L'extracteur de contenu de CATS est écrit avec l'outil EnCo, un LSPL[3] développé par H. Uchida dans le cadre du projet UNL (Uchida, Zhu et al. 2005-2006) pour écrire des « enconvertisseurs » vers le langage pivot UNL.

Cet outil a été utilisé par D. Daoud pour produire une représentation syntaxiquement semblable à UNL, mais qui ne correspond pas du tout à la représentation UNL (Uchida and Zhu 2003) standard, liée à une expression linguistique en anglais (même si elle en est une représentation profonde). En effet, CRL-CATS est une représentation de type `Propriété {objet, valeur}`, et pas un graphe représentant l'analyse sémantique d'un énoncé.

[3] LSPL :Langage Spécialisé pour la Programmation Linguistique.

3.3.1 Le langage spécialisé EnCo et l'extracteur de contenu de l'arabe

EnCo (Uchida and Zhu 1999) attend en entrée :

- un dictionnaire et une grammaire (linguiciel).
- un texte découpé en phrases.

Il compile le linguiciel, puis traite successivement chaque phrase. Les structures de données manipulées par EnCo sont :

- une liste de nœuds avec deux têtes de lecture/écriture placées sur deux nœuds successifs (LW « Left Windows », RW « Right Windows ») et deux têtes de lecture (LC, RC) pour les contextes gauche et droit.
- un graphe de nœuds, initialement vide, pouvant contenir des nœuds de la liste, et dont les arcs portent des « relations » identifiées par des symboles à trois caractères alphabétiques.

Au départ, la liste comporte trois nœuds : la limite gauche, le nœud courant et la limite droite. Le nœud courant contient comme chaîne la phrase à traiter.

De façon générale, un nœud peut contenir quatre éléments : une chaîne, un ensemble d'attributs « de chaîne » (initialisés lors des appels au dictionnaire), une UW (référence lexicale, venant du dictionnaire ou créée par une règle), et un ensemble d'attributs « de graphe » (préfixés par « `.@` »). Les attributs sont booléens, et ne sont pas déclarés. Seul « `.@entry` » a un rôle spécial.

La syntaxe des règles à appliquer est la suivante (Uchida, Zhu et al. 2005-2006) :

```
<TYPE>...(<PRE2>)(<PRE1>){<LNODE>} {<RNODE>} (<SUF1>) (<SUF2>)… P<PRI>;
```

avec

```
<LNODE>:="{“ [<COND1>] ":" [<ACTION1>] ":" [<RELATION1>]":" [<ROLE1>] "}"
<RNODE>:="{“ [<COND2>] ":" [<ACTION2>] ":" [<RELATION2>]":" [<ROLE2>] "}"
```

Une règle peut s'appliquer si sous la fenêtre d'analyse gauche `(LW)` se trouve un nœud qui satisfait la condition `<COND1>` et sous la fenêtre d'analyse droite se trouve un nœud qui satisfait la condition `<COND2>`. Quand les nœuds à gauche et à droite de la fenêtre d'analyse répondent aux conditions trouvées dans `<PRE>` et `<SUF>`, les propriétés grammaticales dans la fenêtre d'analyse sont réécrites selon les actions `<ACTION1>` et `<ACTION2>`.

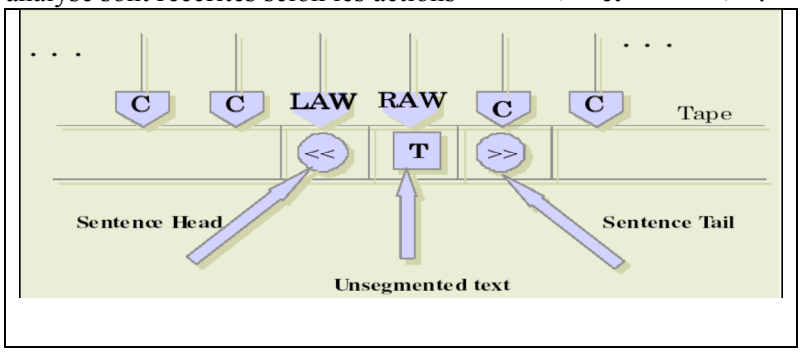

Figure 2 : Configuration initiale d'EnCo (Daoud 2006)

Voici un petit exemple. Initialement, `LW` contient le symbole '`<<`' appelé `SHEAD` et `RW` contient le premier mot de la phrase qui est « recherche »:

- SMS en entrée : *recherche voiture*
- Articles du dictionnaire utilisés :

```
[chaîne]   {}  "UW"     (traits)  <priorité>
[recherche]{}  "wanted" (want)    <F,1,1>;
[voiture]  {}  "saloon" (vech)    <A,1,1>;
```

- Plusieurs règles sont appliquées, dont la première est : `R{SHEAD:::}{wanted:::}P20;`

Cette règle fait un "shift right" désigné par `R` car sous la fenêtre gauche il y a `SHEAD`, et sous la fenêtre de droite, il y a le mot « recherche » qui est mis en correspondance avec l'UW `"wanted"` et le trait `wan` dans le dictionnaire. `P20` indique la priorité affectée à cette règle par rapport aux autres.

- Le résultat final est :

```
;========== UNL =================
;Recherche voiture.
[S]
wan(saloon:0A,      wanted:00)
[/S]
;================================
```

Le dictionnaire utilisé dans la version arabe a environ 30.000 entrées, dont 20.000 ont été générées automatiquement (grâce à un répertoire de variantes et de fautes d'orthographe fréquentes). Il relie les mots et les expressions arabes des domaines de CATS (Cars, RealEstate, Jobs, Misc) aux concepts de CRL-CATS en précisant les propriétés sémantiques, syntaxiques et morphologiques utilisées dans l'analyse des SMS arabes. La structure des entrées inclut des abréviations, différentes écritures pour la même entrée, différentes formes orthographiques et d'autres formes de jargon utilisées dans le sous-langage en question.

Les 710 règles EnCo utilisées dans le système CATS extraient les informations utiles, et ne font pas l'analyse linguistique au sens classique. Elles affectent des valeurs à des objets préfinis dans le dictionnaire pour construire des relations de type `Propriété{objet, valeur}`. L'ensemble de ces relations forme la représentation CRL-CATS.

3.3.2 Adaptation au français de l'extracteur de contenu écrit en EnCo

Contrairement à la difficulté de trouver un corpus fonctionnel en français, la bonne surprise de ce travail a été que nous n'avons modifié que légèrement les règles fabriquées initialement pour la version arabe, et que l'EC obtenu fonctionne bien pour le sous-langage correspondant du français, celui des SMS spontanés pour l'achat et la vente de voitures d'occasion.

Cela confirme la théorie linguistique de (Kittredge and Lehrberger 1982) selon laquelle deux sous-langages équivalents dans deux langues différentes sont proches (très proches ici) l'un de l'autre, même si leurs deux langues mères sont éloignées.

Le dictionnaire fabriqué est destiné à tout type d'utilisateurs, de tous niveaux. Il faut donc s'attendre à recevoir des erreurs de frappe, des abréviations étranges, des mots étrangers, et des fautes. Par exemple, quelqu'un écrira « Alfa Roméo » au lieu de « Alpha Roméo ». De plus, le dictionnaire doit évoluer suivant l'usage.

L'exemple ci-dessous montre qu'il faut tenir compte dans le dictionnaire du sous-langage français : un Français peut dire « *je cherche une A3…* » au lieu de dire « *je cherche une voiture AUDI A3…* » (comme on le dit de préférence en arabe). Ainsi, on doit ajouter les entrées suivantes qui doivent converger vers le même concept CRL-CATS.

```
[AUDI]{}"AUDI(country>germany,country>europe)"(make,car) <A,3,3>;
[A3]  {}"AUDI(country>germany,country>europe)"(make,car) <A,3,3>;
[A4]  {}"AUDI(country>germany,country>europe)"(make,car) <A,3,3>;
[A6]  {}"AUDI(country>germany,country>europe)"(make,car) <A,3,3>;
```

Figure 3 : Convergence de plusieurs entrées dictionnairiques vers une même entrée

Il se trouve que l'outil EnCo fait la différence entre les minuscules et les majuscules. Pour l'arabe, le problème ne s'est pas posé vu qu'il n'y a pas cette distinction. Le nombre d'entrées dans certains cas est beaucoup plus réduit en français qu'en arabe, car une seule entrée peut être écrite en arabe de plusieurs façons, avec ou sans ECHAKEL, avec ou sans ELHAMZA et avec ou sans voyelles diacritiques. Dans d'autres cas, le nombre d'entrées doit augmenter, car il faut tenir compte de la casse et des abréviations utilisées dans le sous-langage des SMS en français. Par exemple, « *cse départ* » à la place de « *cause départ* ».

Nous sommes partis d'un ensemble de 638 d'entrées de base en arabe pour le domaine *Cars,* avec un coefficient d'expansion égal à 3, dû aux erreurs, aux formes diacritiques, et aux transcriptions multiples de noms étrangers etc., ce qui fait un total de 1914 lexèmes. Nous avons étendu les 638 entrées de base correspondantes en français à 1761 lexèmes français, pour la prise en compte d'erreurs, d'alternance masculin/féminin, singulier/pluriel, minuscule/majuscule, et de transcriptions multiples de noms étrangers. Cela donne un coefficient d'expansion de 2,7, presque égal à celui de l'arabe (3).

3.4 Adaptation externe

Il s'agit d'adapter un extracteur de contenu d'un autre système, destiné à la même langue et à un autre domaine. Nous avons ainsi adapté à CATS l'extracteur du français développé pour le projet Nespole! par H. Blanchon en Tcl/tk, en utilisant des transducteurs réguliers. Cela représente un peu moins de 32.000 lignes de code (Blanchon 2004).

La représentation de contenu obtenue est en IF (« Interchange Format »), un pivot sémantico-pragmatique utilisé pour des domaines restreints.

La figure ci-dessous montre les composants d'une représentation en IF : actes de parole, concepts et arguments.

Au début de ce travail, nous disposions du code (celui du second démonstrateur de Nespole!) et de la version papier et électronique de la spécification de l'IF (version du 18/08/2002).

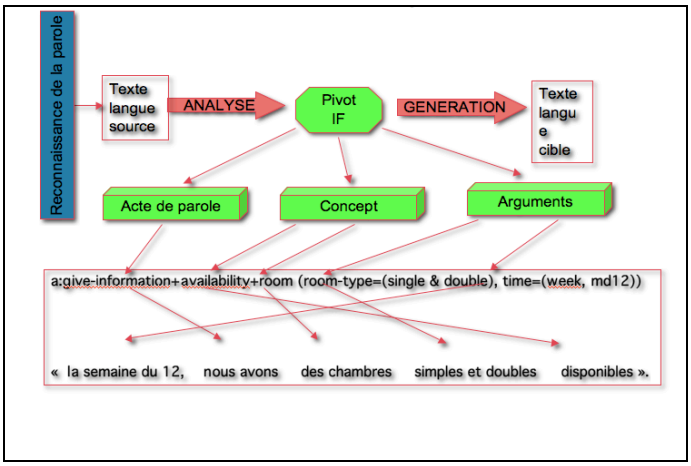

Figure 4 : extracteur de contenu pour français dans le domaine du tourisme

3.4.1 Méthode d'extraction de contenu dans Nespole !

Comme le montre le schéma ci-dessous, la méthode utilisée pour l'analyse du français vers l'IF est composée des étapes suivantes :

- Segmentation des SDU (Unités Sémantiques de Dialogue).
- Détection du domaine.

- Construction d'un préfixe de l'acte de dialogue et instanciation des arguments liés.
- Instanciation des arguments liés au domaine et gestion des subordinations.
- Complémentation de l'acte de dialogue

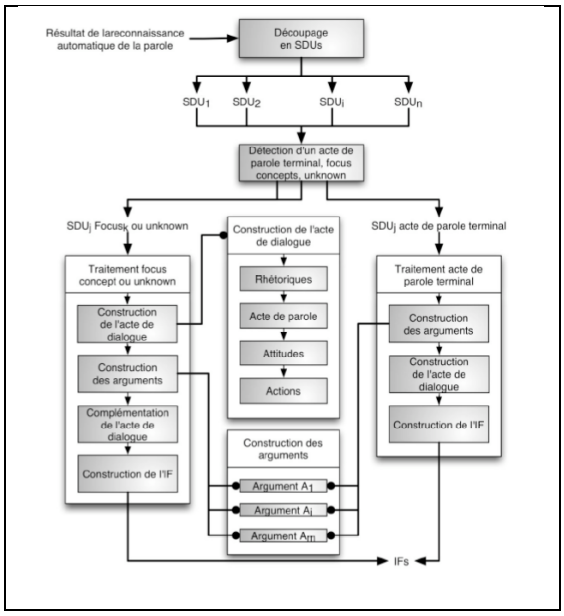

Figure 5 : Architecture du module d'analyse français vers IF du second démonstrateur NESPOLE! (Blanchon 2004).

3.4.2 Méthode pour transformer l'IF en IF-CATS

Nous avons adapté la spécification de l'IF au domaine de l'automobile, et l'avons enrichie en ajoutant de nouveaux arguments comme `vehicle-motor-type, vehicle-hand…` et de nouvelles actions, essentiellement l'action d'achat `e-buy` et l'action de vente `e-sell`. Nous avons utilisé les mêmes étapes que celles du second démonstrateur pour adapter la spécification IF au domaine de l'automobile, en essayant d'éliminer les instructions qui posent problème et/ou qui ne sont pas nécessaires, pour réduire le temps de calcul, et d'ajouter de nouvelles instructions…

Un travail important a été fait dans l'étape d'instanciation des arguments liés au domaine des véhicules (`vehicle`) : on instancie essentiellement la spécification d'un véhicule « *vehicle-spec* », ainsi que d'autres arguments moins intéressants tels que : `theDistance, theLocation, theDuration, theDestination, theTime, thePrice`... La nouvelle fonction `VehicleSpec2If` permet la recherche et la construction des arguments liés au « focus concept » `vehicle` : le seul argument qui existe et qui était programmé dans le code du second démonstrateur est `frenchvehicle`, qui peut avoir comme valeur `voiture, ski, camion, bus, train, avion`… D'autres arguments existent dans la spécification, mais qui ne sont pas programmés tels que : `makevehicle, modelvehicle, sizevehicle, frenchcolor, agevehicle, pricevehicle`...

Afin d'adapter la spécification IF au domaine de l'automobile, nous avons ajouté d'autres arguments liés à ce domaine, tels que `motortypevehicle`, `handvehicle`, `conditionvehicle`... De la même façon, des fonctions `Argument2if` construisent les valeurs IF associées. La figure suivante est un exemple du résultat obtenu après adaptation.

Entrée 1 = je veux vendre une grande voiture française BM 325 4 portes diesel bleue TBE première main assurance complète avec CT sans climatisation TB prix dernier mod

Sortie1 = {c:give-information+disposition+vehicle(disposition=(desire, who=i), action=e_sell, vehicle-spec=(car, vehicle-make=BMW, vehicle-model=325, vehicle-size=4 door, vehicle-shape=big, vehicle-motor-type=diesel, vehicle-hand=first_hand, vehicle-color=blue, vehicle-condition=good, vehicle-assurance=inssured, vehicle-controle=total_check, vehicle-air-condition=no_air_condition, vehicle-nationality=french, age-vehicle=new_mod, price-vehicle=good_price))}

Figure 6 : extracteur de contenu pour le français pour l'occasion automobile

Nous avons appelé le résultat obtenu IF-CATS (`sortie1` dans l'exemple précédent).

3.4.3 Compilateur IF-CATS_CRL-CATS

Nous avons construit un compilateur qui analyse la sortie IF-CATS et la transforme dans le format CRL-CATS en utilisant un dictionnaire IF-CRL lié à cette structure qui permet la substitution des arguments. La figure suivante montre qu'en passant par cette transformation, on arrive à la même sortie que celle donnée par l'outil EnCo, à l'exception des symboles `00, 0J, 0R` ajoutés par ledit outil.

```
;======================== CRLcats ======================
; je veux vendre une Renault clio mod 1998
;{a:give-information+disposition+vehicle(disposition=(desire, who=i),
;action=e_sell, vehicle-spec=(, vehicle-make=RENAULT, vehicle-model=clio,
;vehicle-size=, vehicle-shape=, vehicle-motor-type=, vehicle-hand=,
;vehicle-color=, vehicle-condition=, vehicle-assurance=,
;vehicle-controle=, vehicle-air-condition=, vehicle-nationality=,
;vehicle-age=1998, vehicle-price=, vehicle-mileage=))}
S
sal(saloon, sale)
mak(saloon, RENAULT(country>France,country>europe))
mod(saloon, clio)
yea(saloon,  1998)
/S
;================================================
```

Figure 7 : exemple de sortie du compilateur IF-CATS

3.5 Résultats et évaluation par rapport à la version originale

3.5.1 Méthode d'évaluation

Nous avons traduit manuellement le corpus d'évaluation utilisé pour l'évaluation de la version arabe (originale) du système. C'est un corpus constitué de 200 SMS réels (100 SMS d'achat + 100 SMS de vente) envoyés par des utilisateurs réels en Jordanie. Nous avons mis 289 mn pour traduire les 200 SMS arabes (2082 mots équivaut à 10 mots/SMS, environ 8 pages standard[4]) de l'arabe vers une traduction "brute" (littérale), soit 35 mn par page. Nous avons mis 10 mn par page standard pour passer d'une traduction brute à une traduction fonctionnelle. Nous avons obtenu 200 SMS français jugés fonctionnels (1361 mots, soit 6,8 mots/SMS, environ 5 pages standard).

Pour évaluer les résultats d'extraction, nous avons calculé le rappel R, la précision P et la F-mesure F pour chacune des propriétés les plus importantes (action de vente ou d'achat, marque, modèle, année, prix) définis comme suit :

P = Nombre d'entités correctes identifiées par le système / Nombre total d'entités identifiées par le système ;

R = Nombre d'entités correctes identifiées par le système / Nombre d'entités identifiées par l'humain ;

F = 2*P*R / P+R

[4] Une page standard contient 250 mots.

3.5.2 Résultats

Nous avons fait des évaluations pour les propriétés les plus importantes. Les pourcentages de portage par adaptation interne (par rapport à la version originale) varient entre 95% et 100%, avec une moyenne de 98 %. Les pourcentages de portage par adaptation externe (par rapport à la version originale) varient entre 46% et 99%, avec une moyenne de 77 %. Notons que ce sont les propriétés traitant les chiffres comme prix et années qui rendent faible la valeur du pourcentage du portage par adaptation externe, mais son avantage c'est qu'elle ne nécessite qu'un simple accès à la représentation interne de l'application.

	EnCoAR			EnCoFR (adapatation interne)				RegExpFR (adapatation externe)			
Propriété	Précision	Rappel	F-mesure (EnCoAR)	Précision	Rappel	F-mesure (EnCoFR)	% portage	Précision	Rappel	F-mesure (RegExpFR)	% portage
Achat/vente	0,956	0,970	0,963	0,994	0,855	0,919	95	1,000	0,835	0,910	95
Année	0,817	0,960	0,883	0,879	0,819	0,848	96	0,828	0,271	0,409	46
Prix	0,800	0,822	0,811	0,789	0,822	0,805	99	0,955	0,288	0,442	55
Marque	0,978	0,963	0,970	0,961	0,961	0,961	99	0,994	0,928	0,960	99
Modèle	0,901	0,837	0,868	0,842	0,903	0,872	100	0,965	0,661	0,785	90
Moyenne	0,890	0,910	0,899	0,893	0,872	0,881	98	0,948	0,597	0,701	77

Tableau 1 : Comparaison entre les résultats d'EC

La figure suivante permet de mieux visualiser la comparaison entre les valeurs de F-mesure trouvées pour chacune des versions du système.

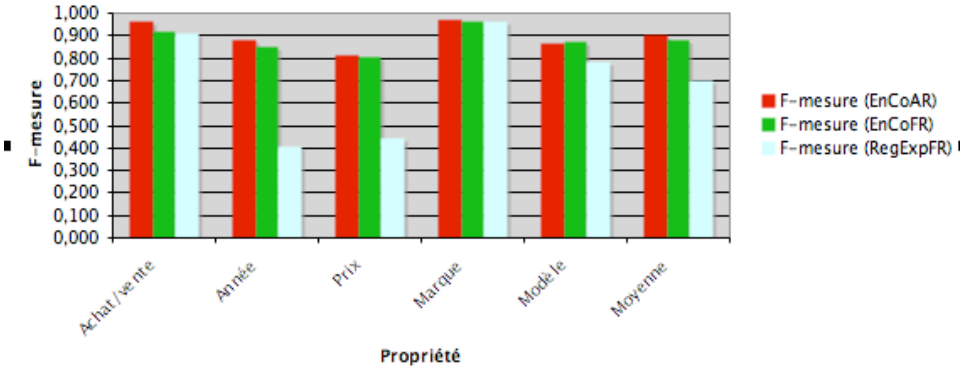

Figure 8 : comparaison entre F-mesures

Conclusion

Nous avons choisi CATS comme application à localiser car c'est une plate-forme qui traite des ESLN et qu'on a accès à toutes ses ressources. Nous avons présenté une première méthode de localisation « interne » qui nécessite un accès total au code source et aux ressources linguistiques de l'application. Malgré la grande distance qui existe entre le français et l'arabe, cette méthode donne de bons résultats à cause de la proximité des sous-langages. Nous avons présenté une deuxième méthode, dite « externe », qui demande un simple accès à la représentation interne de l'application, et consiste à adapter un EC existant pour la langue "cible". Enfin, nous sommes en train de faire une expérience pour déterminer si on pourrait construire un système de TA statistique, de qualité suffisante non pas pour la compréhension, mais pour l'extraction de contenu, à partir de corpus beaucoup plus petits que dans le cas général, en profitant du fait qu'il s'agit de sous-langages restreints, et qu'on peut "injecter" des dictionnaires spécifiques relativement faciles à construire.

Bibliographie

Besacier, L., H. Blanchon, et al. (2001). Speech Translation for French in the NESPOLE! European Project. Eurospeech, Aalborg, Denmark.

Blanchon, H. (2004). Comment définir, mesurer et améliorer la qualité, l'utilisabilité et l'utilité des systèmes de TAO de l'écrit et de l'oral. Une bataille contre le bruit, l'ambiguïté, et le manque de contexte. Grenoble, UJF. HDR: 380. Grenoble, Université Joseph Fourier: 380.

Bélanger, L. (2003). Le traitement automatisé des courriels pour les services aux investisseurs: une approche par la question-réponse, Université de Montéréal: 48.

Califf, M. E. (1998). Relational learning techniques for natural language information extraction. Artificial Intelligence Laboratory, The university of Texas at Austin: A198-276.

Cardie, C. (1997). "Empirical methods in information extraction." AI Journal 18, 4: 65-79.

Ciravegna, F. (2001). Adaptive information extraction from text by rule induction and generalisation. 17th International Joint Conference on Artificial Intelligence (IJCAI), Seattle.

Daoud, D. M. (2005). "Building SMS-based System using Information Extraction Technology". ACIDCA-ICMI, Tozeur, Tunisia.

Daoud, D. M. (2006). It is necessary and possible to build (multilingual) NL-based restricted e-commerce systems with mixed sublanguage and contend-oriented methods. GETA - CLIPS. Grenoble, Université Joseph Fourier: 296 pages.

Hajlaoui, N. (2006). Recherche et production de corpus de messages pour la multilinguïsation de sites de e-commerce en SMS, initialement en arabe. 6th international IBIMA (International Business Information Association) Conference, Bonn, Allemagne.

Huberman, B., P. Pirolli, et al. (1998). Strong Regularities in World Wide Web Surfing. Science.

Kittredge, R. and J. Lehrberger (1982). Sublanguage: study of language in restricted semantic domain.

Koehn, P. (2004). Pharaoh: a Beam Search Decoder for Phrase-Based SMT. 6th AMTA, Washington.

Lehman, A. (1996). Construction d'un système de résumé automatique de textes de type scientifique et technique. RECITAL (Rencontre des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues), Paris.

Lepage, Y. (2006). Traduction par analogie. MTS, Pukhet, IAMT, ed.

Ritchie, I. A. G. D. (1995). "Natural Language Interfaces to Databases D an Introduction. Natural Language Engineering." Cambridge University Press: 29-81.

Sekine, S. (1994). A new direction for sublanguage NLP. International Conference on New Methods in Language Processing.

Slocum, J. (1986). How one might automatically identify and adapt to a sublanguage. Analyzing language in restricted domains.

Uchida, H. and M. Zhu (1999). Enconverter Specifications, UNU/IAS UNL Center.

Uchida, H. and M. Zhu (2003). The Universal Networking Language specification, UNL Center UNDL Foundation.

Uchida, H., M. Zhu, et al. (2005-2006). Universal Networking Language.