TTR
Traduction, terminologie, rédaction

Multilinguïsation des systèmes traitant des sous-langages

Najeh Hajlaoui

Volume 26, numéro 1, 1er semestre 2013

Traduction et contact multilingue
Translation and Multilingual

URI : https://id.erudit.org/iderudit/1036952ar
DOI : https://doi.org/10.7202/1036952ar

Aller au sommaire du numéro

Éditeur(s)

Association canadienne de traductologie

ISSN

0835-8443 (imprimé)
1708-2188 (numérique)

Découvrir la revue

Citer cet article

Hajlaoui, N. (2013). Multilinguïsation des systèmes traitant des sous-langages. *TTR*, *26*(1), 123–151. https://doi.org/10.7202/1036952ar

Résumé de l'article

Dans le cadre de nos travaux sur la multilinguïsation ou « portage linguistique » des services de gestion de contenu traitant des énoncés spontanés en langue naturelle, nous avons dégagé trois méthodes de portage possibles d'une langue L1 vers une nouvelle langue L2, et les avons appliquées sur des cas de systèmes de e-commerce. Le portage par traduction statistique, une de ces trois méthodes, a donné de très bonnes performances, et ce, avec un corpus d'apprentissage très petit (moins de 10 000 mots). Cela prouve que, dans le cas de sous-langages très petits, la traduction statistique peut être de qualité suffisante en partant de corpus 100 à 500 fois moins grands que pour de la langue générale.

Multilinguïsation des systèmes traitant des sous-langages

Najeh Hajlaoui
Université Joseph Fourier

Résumé
Dans le cadre de nos travaux sur la multilinguïsation ou « portage linguistique » des services de gestion de contenu traitant des énoncés spontanés en langue naturelle, nous avons dégagé trois méthodes de portage possibles d'une langue L1 vers une nouvelle langue L2, et les avons appliquées sur des cas de systèmes de e-commerce. Le portage par traduction statistique, une de ces trois méthodes, a donné de très bonnes performances, et ce, avec un corpus d'apprentissage très petit (moins de 10 000 mots). Cela prouve que, dans le cas de sous-langages très petits, la traduction statistique peut être de qualité suffisante en partant de corpus 100 à 500 fois moins grands que pour de la langue générale.

Abstract
This article focuses on our work on multilinguization, or "linguistic porting," and content management services. These systems handle spontaneous, natural-language utterances. Within this framework, we developed three methods for porting language L1 to a new language, L2, and have applied them to e-commerce. Statistical translation porting is one of these methods and performed very well with a very small training corpus (less than 10,000 words). This proves that, in the case of very small sub-languages, statistical translation may be of sufficient quality when working from a corpus 100 to 500 times smaller than for general language.

Mots-clés: portage linguistique, sous-langage, langue générale, énoncés spontanés et bruités, traduction statistique, extraction de contenu

Keywords: linguistic porting, sub-language, general language, spontaneous and noisy utterances, statistical translation, content extraction

Introduction

Une conséquence majeure de la mondialisation est l'importance croissante du multilinguisme, accentuée par l'utilisation d'Internet et le multimédia. Or, les développeurs d'applications informatiques ne sont toujours pas en mesure de fournir des applications utilisables dans la langue maternelle de tous les utilisateurs. Certains services et applications sont rarement offerts dans de nombreuses langues et/ou ne peuvent traiter les textes rédigés spontanément et pouvant contenir des erreurs, des abréviations, différentes formes typographiques, etc.

Dans ce cadre, nous nous intéressons à un problème précis, celui de la multilinguïsation des applications de e-commerce traitant des énoncés spontanés en langue naturelle (LN). Ces applications extraient le contenu pertinent des énoncés, les représentent dans un langage approprié appelé Content Representation Language (CRL) et traitent ensuite les objets obtenus.

1. Systèmes traitant des sous-langages

L'extraction de contenu est rarement fondée sur une analyse complète des énoncés : on utilise le plus souvent des grammaires locales et un dictionnaire mettant en relation des termes du domaine et les symboles (concepts, attributs, relations) du CRL. L'architecture des applications qui nous intéressent se présente sur le modèle illustré à la Figure 1.

La multilinguïsation de tels systèmes est un problème difficile, ce qui explique que très peu de services soient multilingualisés. La difficulté dépend de deux facteurs relatifs à la situation traductionnelle :

- le niveau d'accès aux ressources des applications, pour lequel quatre situations se présentent : accès complet au code source ; accès à la représentation interne uniquement ; accès aux dictionnaires uniquement ; aucun accès ;

- le niveau de compétence langagière des intervenants, qui peut être défini par rapport à la langue source ou par rapport aux compétences linguistiques de l'équipe qui veut faire la localisation.

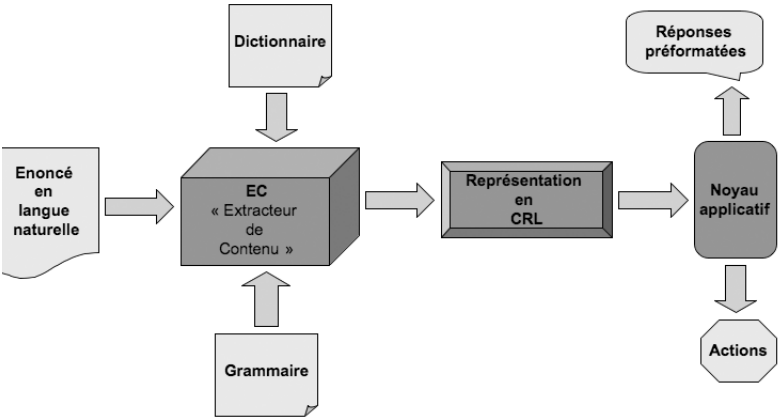

Figure 1. Architecture générale des systèmes de traitement de contenu

La multilinguïsation, ou *portage linguistique,* n'est pas nécessairement une localisation (Hajlaoui, 2008). Une localisation implique une adaptation à un autre contexte, en tenant compte des aspects linguistiques, culturels, géographiques. Par contre, un portage linguistique doit seulement permettre l'accès dans une autre langue à un service de e-commerce, tel qu'il est et où il est (en restant dans le même contexte). Notre but est de trouver, pour multilingualiser de telles applications, des solutions simples, peu coûteuses en ressources et en temps, efficaces, applicables sur le terrain, adaptées à la situation traductionnelle, en réutilisant ce qui est disponible comme outils, ressources et capacités langagières et linguistiques.

Un service de gestion de contenu utilise une représentation interne spécifique sur laquelle travaille le noyau fonctionnel[1]. Le plus souvent, cette représentation est produite à partir de la langue « native » L1 par un extracteur de contenu. Nous avons dégagé trois approches de portage et les avons illustrées par le portage en français d'une partie de CATS[2] (Daoud, 2006), un système de traitement de petites annonces en SMS (en arabe) déployé à

1. Par exemple, le noyau fonctionnel dans CATS est un ensemble de programmes autour d'une base de données.
2. Pour Classified Ads Through SMS.

Amman, ainsi que sur IMRS[3] (Kumamoto, 2007), un système de recherche de pièces musicales dont l'interface native est en japonais et dont seule la représentation de contenu est accessible. Le choix de la stratégie est contraint par la situation traductionnelle : types et niveau d'accès possibles, ressources disponibles (dictionnaires, corpus), compétences langagières et linguistiques des personnes intervenant dans la multilinguïsation des applications.

Généralement, les énoncés traités par le genre d'applications auxquelles on s'intéresse constituent un sous-langage plus ou moins restreint. La notion de sous-langage soulève toutefois des questions. Un sous-langage est-il un sous-ensemble de la langue standard ? Peut-on définir un sous-langage comme étant l'ensemble d'énoncés sur lequel un système de traduction statistique peut s'entraîner rapidement en se basant sur un petit corpus ?

Kittredge et d'autres ont montré l'importance de la notion de sous-langage dans le traitement des textes d'un langage naturel modifié ou simplifié par l'utilisation de restrictions lexicales, syntaxiques ou sémantiques spécifiques (Kittredge et Lehrberger, 1982 ; Grishman et Kittredge, 1986 ; Slocum, 1986 ; Biber, 1993 ; Sekine, 1994). La notion de sous-langage a donné lieu à différents critères d'identification et définitions[4]. Selon Grishman *et al.* (1986) et Deville (1989), les sous-langages correspondent à des formes spécialisées d'une langue naturelle employées dans un domaine particulier. Pour notre part, nous appellerons *langue standard* l'ensemble des énoncés d'une communauté linguistique formés d'une façon « correcte » par rapport à la grammaire et au vocabulaire usuels. Par ailleurs, nous appellerons *langue générale* l'union d'une langue standard et de toutes ses variantes (jargons[5], langues de spécialité, parlers régionaux, langages « techniques » et langages « sécrétés » par des contextes socioprofessionnels).

3. Pour Impression-based Music-Retrieval System

4. La première définition a été proposée par Harris (1968). Une autre définition a été proposée par Bross *et al.* (1972) et utilisée dans le projet TAUM-METEO (1972-1973 ; Chandioux, 1988), puis pour des manuels de maintenance d'avions dans le cadre du projet TAUM-AVIATION (1974-1981) et du PN-TAO (Projet National de TAO, 1982-1987) en France.

5. Nous entendons par « jargon » le parler propre aux représentants d'une profession ou d'une activité.

2. Portage interne

En ayant accès au code de l'application, nous pouvons appliquer une première approche nommée *portage interne* (Hajlaoui *et al.*, 2007). Comme le montre la Figure 2, une telle approche consiste à adapter à L2 l'extracteur de contenu de l'application. Cette approche nécessite un corpus et un dictionnaire fonctionnellement équivalents dans la langue d'arrivée (L2).

2.1 Extracteur de contenu pour les SMS en arabe

EnCo est un langage spécialisé pour la programmation linguistique (LSPL) développé dans le cadre du projet Universal Networking Language (UNL) (Uchida, Zhu *et al.*, 2005-2006) pour écrire des enconvertisseurs vers le langage pivot UNL. EnCo a été utilisé dans CATS pour produire une représentation syntaxiquement semblable à UNL, appelée CRL-CATS. EnCo attend en entrée : un dictionnaire et une grammaire (linguiciel) ; un texte découpé en phrases ; éventuellement une base de connaissances (les probabilités des triplets relations, UNL, UWL) non utilisée par CATS.

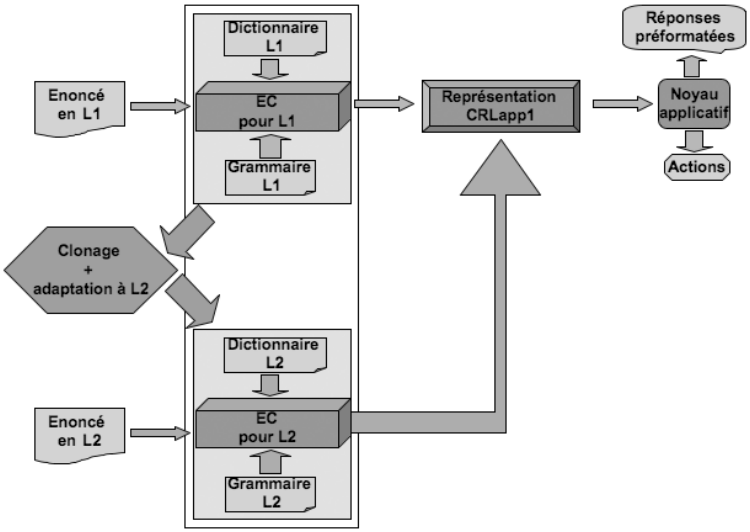

Figure 2. Méthode de portage interne

Le linguiciel est compilé, puis chaque phrase est traitée successivement. Les structures de données manipulées par EnCo sont : une liste de nœuds avec deux têtes de lecture/écriture placées sur deux nœuds successifs (LAW et RAW) et deux têtes de lecture (LCW et RCW)[6] 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), un Universal Word (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 Figure 3 montre la structure d'EnCo.

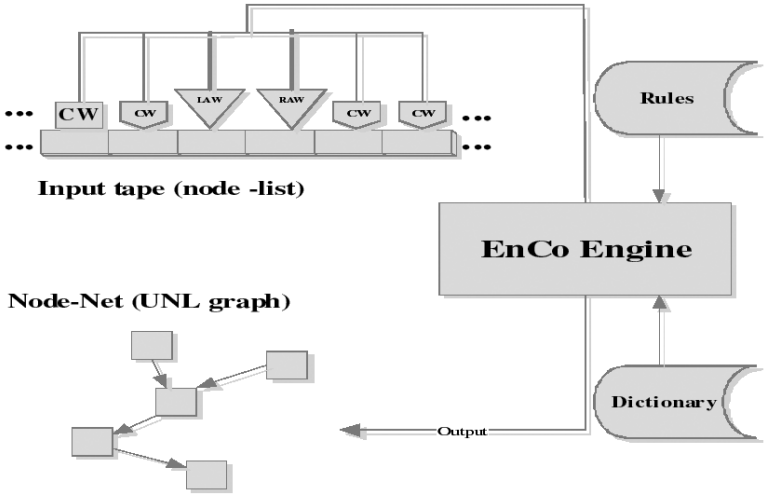

Figure 3. Fonctionnement d'EnCo

6. Les abréviations utilisées correspondent respectivement à Left Analysis Windows, Right Analysis Windows, Left Conditions Windows et Right Conditions Windows.

EnCo utilise les fenêtres de condition pour tester si les nœuds voisins des deux côtés des fenêtres d'analyse remplissent les conditions pour l'application d'une règle d'analyse. Les fenêtres d'analyse sont utilisées pour vérifier l'existence de deux nœuds adjacents afin d'appliquer une règle d'analyse. S'il existe une règle applicable aux nœuds courants, EnCo ajoute ou supprime les propriétés grammaticales indiquées par la règle appliquée et/ou insère un nouveau nœud dans le graphe, ainsi qu'une relation sémantique, selon le type de la règle.

2.1.1 Règles

La syntaxe des règles d'EnCo (Uchida, Zhu *et al.* 2005-2006) est illustrée ci-dessous. Notons que, pour une règle d'analyse donnée, il est possible d'insérer (respectivement de supprimer) un seul nœud dans la liste des nœuds. Ici, LNODE fait référence au nœud sous la fenêtre d'analyse gauche (LAW) et RNODE fait référence au nœud sous la fenêtre d'analyse droite (RAW).

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

avec

<LNODE>:=»{" [<COND1>]«:»[<ACTION1>]«:»[<RELATION1>]»:»[<ROLE1>]«}»

<RNODE>:=»{" [<COND2>]«:»[<ACTION2>]«:»[<RELATION2>]»:»[<ROLE2>]«}»
```

Figure 4. Syntaxe des règles d'ENCO

L'interprétation générale de la syntaxe d'une règle en EnCo est comme suit : une règle peut s'appliquer si sous la fenêtre d'analyse gauche (LAW) se trouve un nœud qui satisfait la condition <COND1> et sous la fenêtre d'analyse droite (RAW) se trouve un nœud qui satisfait la condition <COND2>. Quand il y a des nœuds qui remplissent les conditions trouvées dans <PRE1> et <SUF1> du côté gauche (respectivement dans <PRE2> et <SUF2> du côté droit) des fenêtres d'analyse, les propriétés grammaticales dans les fenêtres d'analyse sont réécrites selon les actions <ACTION1> (respectivement <ACTION2>).

- Le champ <TYPE> porte le type de la règle à appliquer, qui indique l'opération qui doit être effectuée dans la liste des nœuds (p. ex. une opération d'insertion, de suppression, etc.).

- <COND1> et <COND2> sont des conjonctions de conditions élémentaires testant la présence (ATTR) ou l'absence (^ATTR) de certains attributs grammaticaux.
- <ACTION1> et <ACTION2> contiennent des attributs grammaticaux à insérer ou à supprimer dans des nœuds sous les fenêtres d'analyse.
- Les champs <RELATION1> et <RELATION2> sont utilisés pour créer des relations UNL entre les nœuds sous les fenêtres d'analyse.
- <ROLE1> et <ROLE2> sont des attributs de la base de connaissances qui sont optionnels et qui ne sont pas utilisés dans CATS.
- <PRI> indique la valeur de priorité de la règle, qui doit être comprise entre 0 et 255. Une règle dont la valeur de priorité n'est pas indiquée est considérée comme une règle de priorité 0.

Les 710 règles utilisées dans CATS réalisent l'extraction des informations utiles, et non pas l'analyse linguistique au sens classique. Elles affectent des valeurs à des objets préfinis dans le dictionnaire pour construire des relations semblables au type *Propriété (objet, valeur)* en les représentant dans le graphe construit par *objet — propriété→ valeur*. L'ensemble de ces relations forme la représentation CRL-CATS.

2.1.2 Dictionnaire

EnCo recherche dans le dictionnaire tous les lexèmes candidats à partir du premier caractère. Il leur affecte ensuite des priorités basées sur la fréquence d'apparition et la longueur du lexème. Il accorde la priorité la plus élevée à l'entrée qui possède la fréquence d'apparition la plus élevée. Si plusieurs candidats ont la même fréquence la plus élevée, il choisit le lexème le plus long. En cas d'échec ultérieur ou d'activation directe d'un retour arrière, le système effectue un retour arrière, et le lexème suivant dans la liste est alors découpé, puis le traitement reprend sur cette nouvelle «branche» du calcul. L'extracteur de contenu de CATS est

construit pour utiliser cette possibilité le plus rarement possible. On peut voir une entrée du dictionnaire à la Figure 5[7].

```
[HW]"UW" (ATTR1, ATTR2, …) <FLG, FRE, PRI>;
HW: (Head Word) est un lexème qui peut être un candidat
UW: (Universal Word) est une référence lexicale
ATTR1, ATTR2 sont des attributs
FLG: indique la langue
FRE: indique la fréquence
PRI: indique la priorité
```

Figure 5. Syntaxe d'une entrée dictionnairique

Le dictionnaire utilisé dans la version arabe et pour l'ensemble des domaines (automobile, immobilier, divers, etc.) compte environ 10 000 CW et 30 000 lexèmes, dont 20 000 ont été générés automatiquement.

2.2 Adaptation de l'extracteur de contenu

Pour porter l'extracteur de contenu vers le français, nous avons effectué essentiellement un travail dictionnairique et un travail de modification de règles que nous illustrons par un exemple. L'architecture générale est restée la même.

2.2.1 Exemple de traitement

Afin de mieux comprendre le fonctionnement de l'outil EnCo, nous en montrons le détail sur un petit exemple de SMS en français « recherche voiture » (Figure 6, page suivante).

La fenêtre d'analyse gauche (LAW) contient le symbole << (appelé SHEAD) ; la fenêtre d'analyse droite (RAW) contient le texte non segmenté, c'est-à-dire « recherche voiture ». EnCo consulte le dictionnaire pour chercher les préfixes de cette chaîne. Un seul article de dictionnaire est trouvé :

```
[recherche]{} "wanted" (want) <F,1,1>;
```

7. Pour un blanc, et dans le code ASCII, EnCo crée automatiquement l'entrée suivante dont l'UW est la chaîne vide : `[ ]{} "" (BLK) <., 0, 0>;`

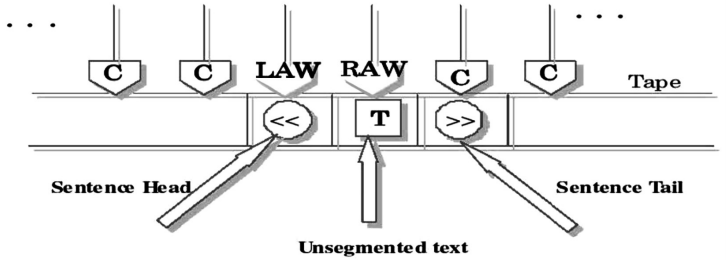

Figure 6. Configuration initiale d'EnCo

La Figure 7 ci-dessous montre le résultat de segmentation de « recherche voiture ». RAW pointe sur un nœud contenant la chaîne « recherche », l'UW « wanted » et l'attribut dictionnairique « want », et aucun attribut de graphe. Par suite de la segmentation lexicale, le nœud initial a été divisé en deux nœuds, le deuxième contenant seulement la chaîne restant à analyser [voiture]. À ce niveau, EnCo cherche à trouver une règle qui satisfait la configuration actuelle : LAW pointe sur le symbole << et RAW pointe sur le nœud [recherche]{}« wanted » (want).

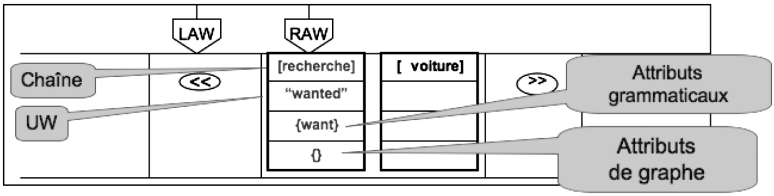

Figure 7. Segmentation de « recherche voiture »

La première règle qui peut être appliquée est :

```
R{SHEAD:::}{want:::}P20;
```

Cette règle fait un *shift right* désigné par «» (TYPE = R). P20 indique la priorité affectée à cette règle. Elle est appliquée sans aucune condition (car COND1 et COND2 sont vides) et sans aucun changement de propriétés grammaticales (car ACTION1 et ACTION2 sont vides). Après l'application de cette règle, LAW pointe sur « recherche » et RAW pointe sur un blanc « BLK » (Figure 8). En effet, le *shift* provoque aussi la segmentation d'un préfixe, ici le blanc.

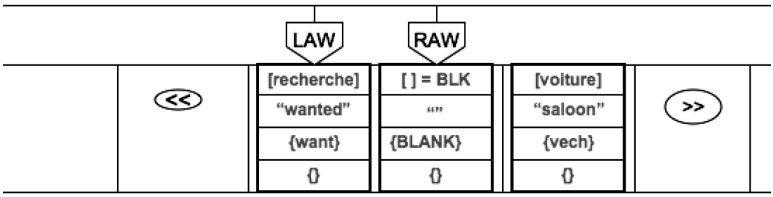

Figure 8. Configuration d'EnCo après un « shift right »

À ce moment, la première règle applicable est :

```
+{:+BLANK::}{BLK:::}P255;
```

- Type de l'opération = + qui signifie la combinaison du nœud droit avec le nœud gauche (autrement dit, ajouter l'attribut de chaîne BLANK au nœud à gauche).
- Cond1 = rien
- Cond2 = BLK (présence de l'attribut BLK)
- Action1 = +BLANK (ajout du symbole BLANK à la liste des attributs du nœud à gauche).
- Action2 = rien
- P255 = Priorité 255 (élevée)

Suite à l'exécution de cette règle, LAW pointe sur le symbole SHEAD et RAW pointe sur le nœud « recherche ». EnCo fait ensuite un *shift right* en appliquant la règle suivante :

```
R{:::}{:::}()P1;
```

La Figure 9 (page suivante) montre l'état d'EnCo après l'exécution des deux précédentes règles.

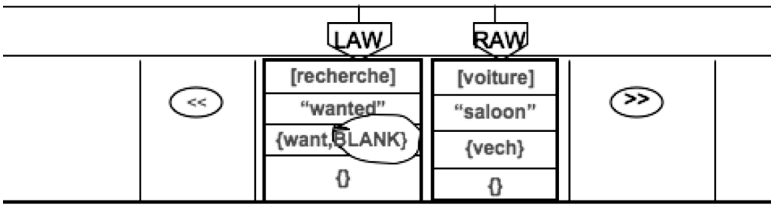

Figure 9. Suppression du BLANK et shift right

LAW pointe sur le nœud contenant la chaîne [recherche] et RAW pointe sur le dernier nœud, contenant (après segmentation) les informations du dictionnaire sur la chaîne [voiture]. EnCo n'a pas créé de nouveau nœud, car la chaîne restante est vide. À ce niveau, nous avons besoin d'une règle qui crée, dans le graphe des nœuds, une relation [wan] qui part du nœud contenant [voiture] et arrive au nœud contenant [recherche], puisque LAW pointe sur un nœud contenant {want, BLANK} et RAW pointe sur un `vech`. Cette règle est la suivante :

```
>{want,BLANK:-want:wan:}{vech,^want_
        add:want_add::}()P70;
```

- Type d'opération = modification à droite désignée par > (le nœud gauche est supprimé de la liste des nœuds, le nœud à droite devient l'origine de l'arc inséré, portant le symbole wan, et allant vers l'ancien nœud gauche.
- Cond1 = want, BLANK (le nœud sous LAW doit contenir l'attribut want et un attribut BLANK).
- Cond2 = vech, ^want_add (le nœud sous RAW doit contenir l'attribut *vech* et ne doit pas contenir l'attribut want_add).
- Action1 = -want (supprimer l'attribut want au nœud gauche (placé sous LAW)).
- Action2 = want_add (ajouter l'attribut want_add au nœud droit (placé sous RAW)).
- RELATION1 = wan (la relation wan va du nœud droit au nœud gauche).

La Figure 10 (page suivante) montre le résultat de l'application de cette dernière règle. Le résultat final produit par EnCo est :

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

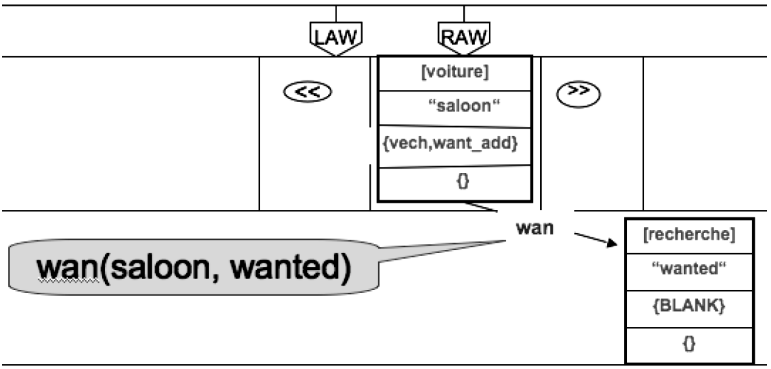

Figure 10. Création d'une relation dans le graphe des nœuds

2.2.2 Modifications apportées

La bonne surprise de ce travail est que nous n'avons dû modifier que légèrement les règles fabriquées initialement pour la version arabe, et que l'extracteur de contenu obtenu fonctionne bien pour le sous-langage correspondant du français, celui des SMS spontanés pour l'achat et la vente des voitures d'occasion. Cela confirme les théories linguistiques (Kittredge et Lehrberger, 1982) selon lesquelles 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 (voir Figure 11). Par portage 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'autre, une nouvelle illustration de l'analyse de Kittredge.

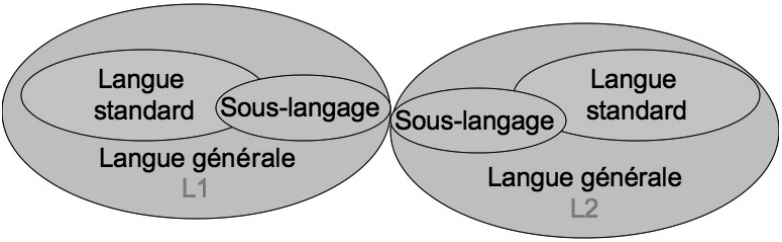

Figure 11. Proximité des sous-langages

Le Tableau 1 montre la répartition de l'effort pour le portage interne en termes de temps de travail et de pourcentage du code modifié ou ajouté.

Tableau 1. Répartition de l'effort pour le portage interne

Adaptation de EC-CATS	Dictionnaire	Règles
Temps de travail (H)	100	45
% du code modifié	90	5

3. Portage externe

Si l'on a uniquement un simple accès à la représentation interne de l'application, nous pouvons appliquer une deuxième approche nommée *portage externe* (Hajlaoui *et al.*, 2008), qui consiste à adapter un extracteur de contenu existant pour L2 au domaine et/ou à la tâche. Cette approche prévoit de traduire le résultat obtenu vers la représentation interne de l'extracteur de contenu original. Elle nécessite aussi un corpus et un dictionnaire fonctionnellement équivalents dans la langue d'arrivée (L2) (Figure 12).

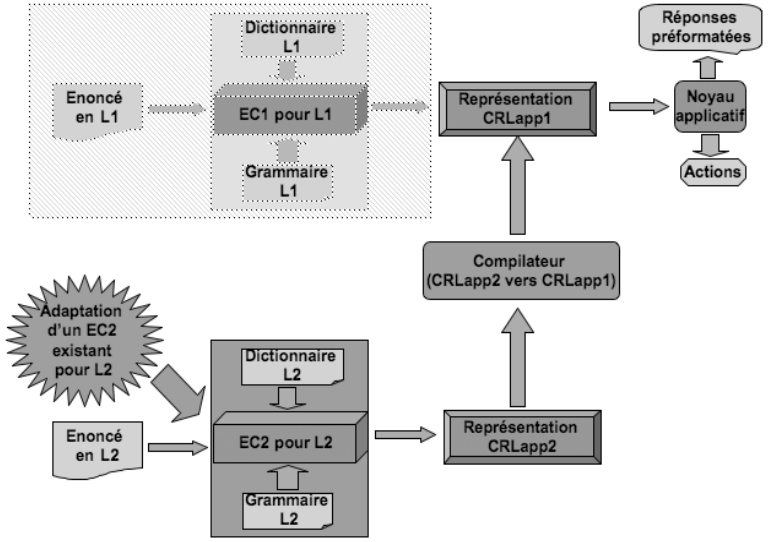

Figure 12. Méthode de portage externe

Le portage externe a été expérimenté sur CATS, mais aussi sur une deuxième application de recherche de musique (IMRS) (Kumamoto, 2007) qui traite des énoncés spontanés en japonais en adaptant le même extracteur de contenu du français construit initialement pour le domaine du tourisme (Blanchon, 2004), en restant dans la même langue, puis en changeant de langue (anglais et arabe).

3.1 Extracteur de contenu existant

Nous avons utilisé un extracteur de contenu pour le français construit dans le cadre des projets de traduction de parole C-STAR (Projet CSTAR) et Nespole ! (Blanchon, 2004). L'objet du projet Nespole ! est de permettre à un agent touristique italophone situé à Trente en Italie de dialoguer avec un client parlant l'anglais, le français ou l'allemand, chacun disposant d'un simple PC. Nous avons utilisé le code du second démonstrateur pour l'adaptation du système CATS au français. La traduction se fait *via* un pivot sémantique appelé Interchange Forma (IF). L'IF a été proposé par Hans-Ulrich Block de Siemens. C'est un pivot sémantico-pragmatique utilisé pour des domaines restreints. Dans le projet Nespole !, le passage par le pivot IF utilise une méthode basée sur les automates de reconnaissance de séquences pertinentes.

L'IF est fondé sur des actes de dialogue (DA, *Dialogue Act*) auxquels sont adjoints des arguments. Un acte de dialogue est constitué d'un acte de parole (SA, *Speech Act*) complété par des concepts. Les actes de dialogue décrivent les intentions, les besoins de celui qui parle (*give-information, introduce-self,* etc.). Les concepts précisent à propos de quoi l'acte de dialogue est exprimé (*price, room, activity,* etc.). Les arguments permettent d'instancier les valeurs des variables du discours (*room-spec, time, price,* etc.). Voici un exemple d'extraction vers l'IF :

> « La semaine du 12, nous avons des chambres simples et doubles disponibles ».

Il s'agit d'une réponse de l'agent touristique qui correspond à l'IF suivante :

> a:give-information+availability+room (room-type=(single & double), time=(week, md12))

3.2 Adaptation à CATS

La spécification IF telle quelle ne couvre pas le domaine de l'automobile. Nous avons donc dû enrichir cette spécification en ajoutant d'autres informations liées à ce domaine, comme celle qui précise l'action d'achat ou de vente ou d'autres informations liées à l'état du véhicule, le moteur, etc. Nous notons IF-CATS la spécification IF adaptée au domaine de l'automobile. La transformation se compose de deux opérations essentielles :

- Adaptation et enrichissement de l'IF : opération du passage au domaine de la vente et de l'achat de voitures d'occasion, qui consiste à ajouter de nouveaux arguments tels que `vehicle-motor-type`, `vehicle-hand` et de nouvelles actions, essentiellement l'action d'achat `e-buy` et l'action de vente `e-sell`. Afin de répondre à ce besoin, nous avons construit manuellement les représentations IF-CATS d'un corpus de 100 exemples (tirés du corpus CATS) pour ce nouveau domaine.
- Nettoyage de l'IF : suppression des domaines inutiles parmi la liste des domaines programmés.

Nous ajoutons à l'ensemble de ces opérations une transformation qui tient compte de la nature de l'entrée. En effet, dans le projet Nespole !, l'étape d'analyse vers l'IF est précédée d'une étape de reconnaissance de la parole qui gère le dialogue de communication et les phatiques comme Allô, tu m'écoutes, oui j'entends, Oh, Ah, d'accord, ok, ouais, etc.

Tableau 2. Répartition de l'effort pour le portage externe

Adaptation de FR-IF	Dictionnaire	Règles
CATS		
Temps de travail (H)	90	140
% du code modifié/ajouté	20	15
IMRS		
Temps de travail (H) (Fr ; En, Ar)	(20 ; 30, 25)	(10 ; 20,15)
% du code modifié/ajouté (Fr ; En, Ar)	(3 ; 6 ; 7)	(2 ; 4 ;3)

Dans notre cas, la nature de l'entrée de CATS est un texte envoyé par SMS. Nous devons donc tenir compte de la gestion des abréviations : `TBE` pour `très bon état`, `CT` pour `contrôle`

`technique, ttes options` pour `toutes options`...
Nous avons adapté la spécification de l'IF au domaine de l'achat et de la vente de voitures d'occasion pour que l'extracteur (analyseur) puisse traiter l'exemple suivant : je veux vendre une voiture Renault Clio bleue TBE en produisant l'IF suivant (IF-CATS).

```
a:give-information+disposition+vehicle(di
sposition=(desire, who=i), action=e_sell,
vehicle-spec=(car, vehicle-make=Renault,
vehicle-model=Clio, vehicle-color=blue,
vehicle-condition=good))
```

4. Portage par TA

Si l'on veut porter une application d'extraction de contenu à partir d'énoncés spontanés sans aucun accès aux ressources internes de l'application (code, représentation interne), la seule approche envisageable est de passer par la traduction automatique (TA) pour utiliser l'extracteur de contenu de l'application originale comme service. L'idée consiste à traduire les énoncés du sous-langage approprié de la nouvelle langue L2 vers le sous-langage original de L1. L2 est alors « cible » du portage, mais « source », pour la traduction. La Figure 13 décrit l'architecture de cette méthode de portage par TA.

Quelle que soit l'approche linguistique choisie pour cette TA (Boitet, 1990), il faut créer un système spécialisé, et donc disposer d'un corpus parallèle L2/L1. La question qui se pose est la taille du corpus nécessaire.

Si l'on utilise une approche computationnelle de la TA fondée sur des corpus – TA statistique (Koehn, 2004), TA par analogie (Lepage, 2005) –, il faut d'énormes corpus s'il s'agit de langue générale (entre 50 et 200 millions de mots) (Koehn, 2004), bien plus grands que ceux disponibles après deux ou trois ans de fonctionnement d'un e-service. Il est toujours possible que dans le cas de sous-langages restreints des corpus beaucoup plus petits suffisent.

Si l'on utilise une approche computationnelle par règles, le corpus de développement n'a pas besoin d'être très grand ; il suffit qu'il soit représentatif. Il faut cependant disposer de linguistes

computationnels pour chaque couple (L2, L1), ce qui est rare et/ou à coût élevé.

Le portage par réalisation d'un système de TA L2→L1 est possible en théorie. En pratique, on peut le faire, sans linguiste qualifié, par des méthodes d'apprentissage automatique. En effet, nous avons appliqué un système de TA statistique au portage linguistique de l'arabe au français de CATS. Il s'agit d'un sous-langage très restreint. Nous ne disposions que d'un très petit corpus parallèle, lié à l'application choisie (petites annonces portant sur la vente et l'achat d'automobiles d'occasion).

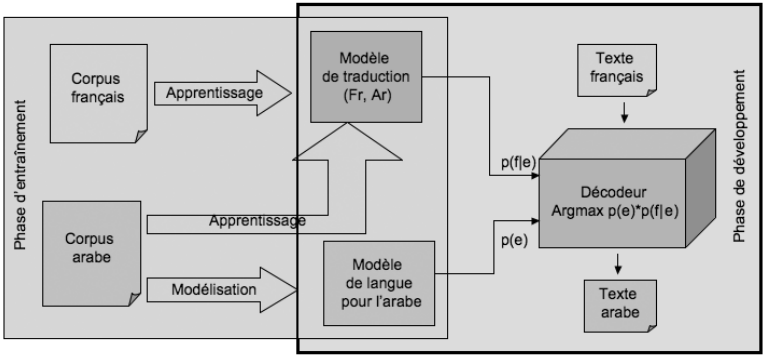

Figure 13. Méthode de portage par TA

Si les sous-langages sont très proches, comme dans les cas que nous visons, nous avons l'intuition que cela peut aider la TA statistique et nécessiter une taille de corpus beaucoup moins importante pour obtenir de bons résultats. Nous pensons que le corpus nécessaire pour le cas d'un sous-langage très petit comme celui de l'auto d'occasion dans CATS (*Cars*) peut ne pas être très volumineux. L'espoir est que les corpus dont nous disposons suffisent (entre 1 000 et 2 000 SMS). L'expérience que nous avons conduite vise à confirmer ou infirmer cette hypothèse.

4.1 Préparation des données

Nous avons proposé une méthode pour construire un corpus bilingue parallèle par déconversion[8] d'un corpus représenté sous forme CRL-CATS (CRL de CATS) vers la langue L2. En effet, nous avons un corpus en L1 (arabe) même s'il est petit (1 100 SMS de taille moyenne 8,75 mots) et la représentation interne (CRL-CATS) correspondante de ce corpus.

Nous avons déconverti les représentations CRL-CATS à l'aide de l'outil DeCo (Uchida *et al.*, 2006) développé dans le cadre du projet UNL (Uchida, 2003). DeCo est un environnement de programmation linguistique qui permet de générer des phrases à partir d'une représentation interne (CRL-CATS dans notre cas). Il utilise un dictionnaire (français CRL-CATS) et des règles à construire pour permettre la génération des énoncés.

Nous avons généré par déconversion une première version de textes français. Nous avons mis 704 min (11,7 h) pour réviser le résultat obtenu, qui donne un corpus français de 10 800 mots (l'équivalent de 43 pages standard). Nous tenons à rappeler que notre but n'est pas d'obtenir une traduction parfaite, mais plutôt une traduction fonctionnelle. Dans le cas où le corpus parallèle obtenu ne suffit pas, la même méthode de déconversion permet de générer un corpus parallèle vers les deux langues.

Nous avons utilisé GIZA++ (Och *et al.*, 2003) comme outil d'alignement au niveau des mots pour aligner le corpus parallèle obtenu qui est composé de 1 100 SMS : un corpus arabe et un corpus français obtenu par déconversion suivi d'une révision humaine afin de rendre plus fonctionnelle et plus naturelle la sortie du déconvertisseur.

La longueur moyenne d'un SMS français est plus élevée (9,93 mots) que pour l'arabe (8,75 mots). Cela peut varier en fonction de la nature des données sources utilisées et de l'utilisation des variantes lexicales. En effet, les données sources sont des données réelles rédigées par de vrais utilisateurs en Jordanie. Aucune règle ne les empêche d'écrire l'équivalent arabe de « VOLKSWAGEN » sous forme d'un seul mot ou de deux mots

8. Déconversion : génération d'énoncés à partir d'une représentation du contenu.

«VOLKS WAGEN» ou encore «LANDROVER» et «LAND ROVER», etc. Le même type de variation peut exister en français.

4.2 Expérimentation

La Figure 14 montre l'architecture générale d'un système de traduction statistique. Une première phase consiste à construire un modèle de langue et apprendre un modèle de traduction. Une deuxième phase consiste à mettre en œuvre un décodeur pour traduire. Plusieurs décodeurs pour la traduction statistique sont offerts sur le Web. Les plus connus sont Pharaoh (Koehn, 2004) et Moses. Ils sont gratuits et en utilisation libre. Pour nos premières expérimentations[9], nous avons choisi Pharaoh.

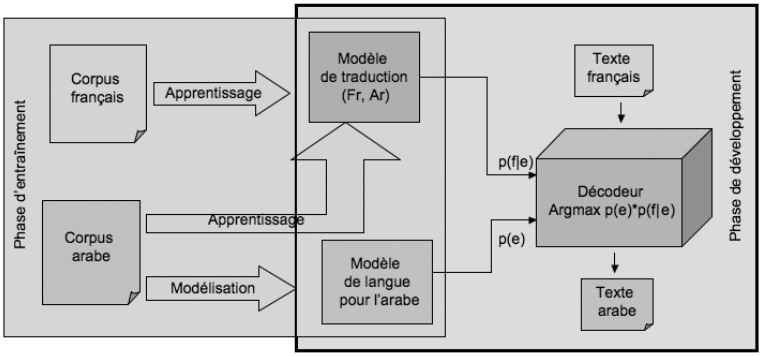

Figure 14. Architecture générale d'un système de traduction statistique

Un modèle de langue pour la langue cible est nécessaire. Dans notre cas, l'arabe est la langue source de l'application et la langue cible pour la traduction. Nous devons donc construire un modèle de langue pour l'arabe. C'est une langue pour laquelle très peu de ressources sont disponibles et gratuites. Nous avons seulement trouvé un modèle de langue pour le Coran, ce qui n'est pas du tout adapté au cas du sous-langage traité. Nous disposons aussi au laboratoire d'un modèle de langue pour l'arabe mésopotamien (iraki), mais il s'agit d'un dialecte et d'une typologie très différents de celui des SMS concernant les voitures d'occasion à Amman.

9. Nous avons refait les mêmes expérimentations avec Moses, mais n'avons pas obtenu de meilleurs résultats qu'avec Pharaoh.

Nous avons donc dû construire un modèle de langue pour le sous-langage du domaine de l'automobile en nous basant sur les données disponibles (1 100 SMS en arabe).

Nous avons utilisé la boîte à outils de modélisation de la langue SRILM offert gratuitement sur le Web (Stolcke, 2002). Nous avons entraîné le modèle de langue sur le même corpus que celui utilisé pour l'entraînement du décodeur de traduction. Le décodeur cherche à maximiser une somme pondérée. Il y a huit poids à fixer dans le fichier de configuration du décodeur : le poids de chacun des cinq coûts de la table de traduction, le poids du modèle de langue, le poids de la pénalité sur le réordonnancement des syntagmes/fragments (phrase, chutes ?), et le poids de la pénalité sur le nombre de mots dans la phrase cible. Le choix de ces huit poids est critique pour que le décodeur produise des traductions de qualité. Nous les avons donc ajustés à l'aide d'un critère de minimisation du taux d'erreur.

En mode développement, nous avons essayé de nouvelles données (200 SMS) qui n'ont pas été utilisées dans le mode apprentissage. Le résultat obtenu dépend de la taille du corpus d'entraînement. En tenant compte de la complexité et de la richesse de la langue arabe, le résultat obtenu pour cette taille est encourageant. En effet, très peu de mots sont inconnus, et les SMS en arabe semblent assez bons. Cela sera précisé par les mesures usuelles de l'évaluation de la TA.

4.3 Évaluation des résultats

Nous avons utilisé dans notre évaluation les scores NIST (Doddington, 2002) et BLEU (Papineni *et al.*, 2002). Notre objectif n'est pas d'avoir une traduction parfaite (le résultat est destiné à un programme et non à un humain), mais d'extraire l'information pertinente dans un énoncé. Il s'agit dans le cas de CATS d'extraire les propriétés importantes (type, modèle, prix, année, etc.) d'une voiture à partir d'un SMS arabe qui est le résultat d'une traduction statistique d'un SMS français. De la même façon que pour les deux autres méthodes de portage, nous mesurerons le rappel et la précision pour la tâche d'extraction de contenu.

L'évaluation par NIST et BLEU suppose d'avoir au moins une version de référence, une version candidate et la version

source. Dans notre cas, la version de référence est le SMS arabe original, la version candidate est le résultat produit par le système Pharaoh, et la version source est celle du corpus d'évaluation en français. Les Figure 15 et Figure 16 présentent les mesures de NIST et BLEU obtenues pour le corpus d'évaluation (200 SMS) en fonction de la taille du corpus d'entraînement. On observe que ces scores n'augmentent presque plus à partir de 500 SMS : ils sont aux alentours de 25 % pour BLEU, loin derrière les meilleurs systèmes statistiques arabe-anglais (qui atteignent un peu plus de 30 %), mais assez élevés tout de même pour espérer que les résultats soient utilisables pour en extraire l'essentiel du contenu.

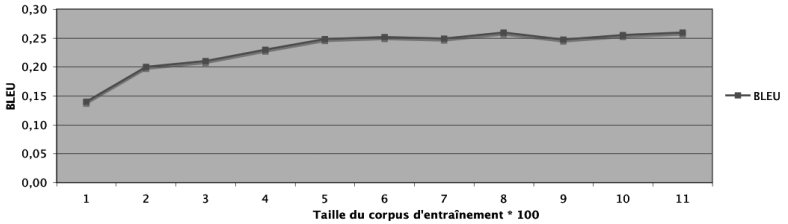

Figure 15. Score BLEU en fonction de la taille du corpus d'entraînement

La courbe de la Figure 15 montre une croissance très faible de la mesure BLEU à partir de la valeur 0,26, qui correspond à une taille du corpus d'entraînement égale à 800 SMS.

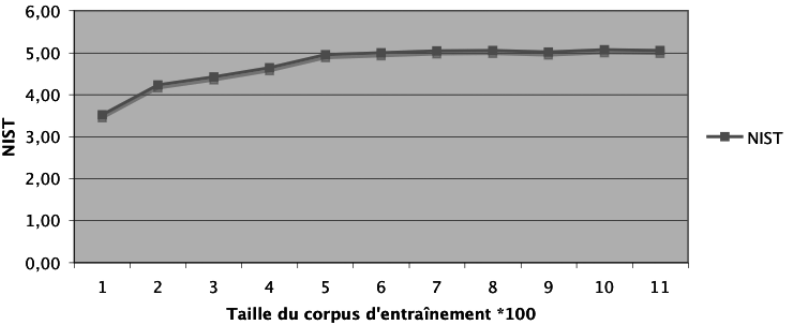

Figure 16. Score NIST en fonction de la taille du corpus d'entraînement

À partir de cette même taille de corpus, la courbe de la Figure 16 représentant la mesure NIST croît (et décroît parfois) aussi très faiblement à partir de la valeur 5,05. Cela indique qu'une augmentation de la taille du corpus d'entraînement ne modifie presque pas la valeur des mesures BLEU et NIST, entre 500 et 1 100 SMS. Rien ne peut être conclu sur l'évolution de ces mesures si la taille croissait beaucoup (p. ex. 50-110 K SMS), mais cela n'est pas important, puisque la qualité obtenue avec 600 à 800 SMS est suffisante pour le portage visé ; pour obtenir un corpus 100 fois plus grand, il faudrait générer ce corpus (génération aléatoire de CRL-CATS et déconversion), et surtout le post-éditer dans les deux langues.

Nous ne garantissons donc pas que les mesures ne puissent pas s'améliorer après une grande augmentation de la taille du corpus d'entraînement ou l'ajout d'autres outils comme un analyseur morphologique pour le français. Mais, rappelons-le, notre objectif est de proposer des solutions de multilinguïsation simples et applicables sur le terrain à un coût le plus faible possible. Or, un examen rapide des résultats nous indique qu'il semble n'y avoir que peu ou pas de perte d'information. Nous allons vérifier ce point en appliquant l'extracteur de contenu à ces résultats, et en comparant les CRL-CATS obtenus à ceux obtenus à partir des SMS originaux. Cela nous laisse dire que NIST et BLEU semblent saturer assez tôt, beaucoup plus tôt que lors des expériences menées sur des corpus plus généraux. D'autres expériences sur le sous-langage de la musique mènent à la même constatation.

5. Évaluation de l'extraction d'information

Après avoir porté CATS de l'arabe vers le français en utilisant les trois précédentes approches, nous avons effectué une évaluation au niveau de la représentation interne. Nous avons utilisé deux mesures : une mesure informationnelle basée sur un corpus de 200 SMS et une mesure plus fine basée sur l'ensemble des énoncés constituant un corpus de 1 100 SMS.

5.1 Évaluation informationnelle

Nous avons traduit manuellement le corpus d'évaluation utilisé pour l'évaluation de la version arabe (originale) du système. Ce corpus est constitué de 200 SMS réels (100 SMS d'achat +

100 SMS de vente) envoyés par des utilisateurs en Jordanie. Nous avons mis 289 min pour traduire les 200 SMS arabes (2 082 mots, soit environ 10 mots/SMS) de l'arabe vers une traduction brute (littérale), soit 35 min par page. Nous avons obtenu 200 SMS français jugés fonctionnels (1 361 mots, soit 6,8 mots/SMS).

Afin de comparer les résultats obtenus par rapport à une version de référence « CRL-CATS-REF », nous avons corrigé manuellement les 200 CRL-CATS correspondant au corpus d'évaluation. Les erreurs concernent essentiellement les propriétés dont les valeurs sont des nombres comme « prix » et « année ». 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*) définies comme suit :

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

(P = 0 si nombre total d'entités identifiées par le système = 0)

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

(R = 0 si nombre d'entités identifiées par l'humain = 0)

F = 2*P*R/(P+R)

Par convention F = 0 si P = R = 0

Le Tableau 3 présente les pourcentages de portage de ces trois méthodes, calculés par évaluation informationnelle sur le petit corpus CorpusEvalFr200SMS. Les pourcentages de portage (rapport des F-mesures) par portage interne (par rapport à la version originale) varient entre 95 % et 100 % (moyenne de 98 %).

Tableau 3. Récapitulatif des résultats d'évaluation informationnelle

	Par rapport à la version originale			**Par rapport à la version de référence**		
Portage	**Minimum**	**Moyenne**	**Maximum**	**Minimum**	**Moyenne**	**Maximum**
Interne	95%	98%	100%	83%	91%	99%
Externe	46%	77%	99%	41%	72%	99%
Par TA	85%	93%	98%	75%	86%	96%

Les pourcentages de portage externe (par rapport à la version originale) varient entre 46 % et 99 % (moyenne de 77 %). Ce sont les propriétés traitant les chiffres comme prix et années qui rendent faible la valeur du pourcentage du portage externe, mais son avantage est qu'elle ne nécessite qu'un simple accès à la représentation interne de l'application.

5.2 Évaluation fine

Afin d'effectuer une évaluation sur la totalité du corpus de 1 100 SMS (CorpusEvalFr1100SMS), nous avons utilisé une deuxième mesure automatique, plus fine que la précédente. Celle-ci permet d'aller au-delà de la détection des noms des attributs et de comparer leurs valeurs par rapport à celles d'une version de référence. Nous supposons que la version originale produite par le système est une version de référence. Ainsi, nous calculons la distance entre la version de référence et la version candidate résultant d'une méthode de portage. Nous avons utilisé l'algorithme de calcul de distance d'édition entre arbres de Selkow (1977). Nous passons à un calcul des mesures connues (rappel R, précision P, et F-mesure F) par simple appel à la formule suivante :

Ref : une CRL de référence (triée), avec |Ref|= m
Cand : une CRL candidate (triée), avec |Cand|= n
K = |{attributs corrects dans Cand}| : la longueur d'une sous chaîne maximale commune entre une CRL candidate et référence

Alors K = |sscmax (Ref, Cand)| et D(Ref, Cand) = m + n - 2K
Donc R = K/m, P = K/n,
d'où F = 1 - D(Ref, Cand)/(m+n)

Dans l'exemple précédent, m = 4 et n = 3 d'où K = 3 et F = 6/7. Nous calculons ensuite la moyenne des valeurs des rappels, des précisions et des F-mesures trouvées. Le Tableau 4 (page suivante) résume les résultats de l'évaluation générale sur tout le corpus obtenus par calcul de distance entre les résultats d'extraction de contenu obtenus par chaque méthode de portage par rapport à ceux obtenus dans la version originale (Hajlaoui, 2008).

Tableau 4. Résultats d'évaluation générale sur CorpusEvalFr1100SMS

	Portage interne	Portage externe	Portage par SMT (TA stat)
F-mesures	0,718	0,540	0,901

La méthode par TA prend de l'avance par rapport à celle par portage interne, mais ne la dépasse pas beaucoup en termes de coûts (Tableau 5). Elle peut être la plus avantageuse, puisqu'elle ne nécessite aucun accès (au code source ou à la représentation interne).Nous sommes satisfaits des trois solutions du point de vue des coûts estimés (voir Tableau 5). Ces solutions sont applicables sur le terrain ; le choix dépend de la situation traductionnelle.

Tableau 5. Récapitulatif des coûts des trois méthodes de portage

	Portage interne	Portage externe	Portage par SMT (TA stat)
Homme/mois d'informaticien		3	1,1
Homme/mois d'informaticien linguiste	1,9		1

Il est important de noter que les chiffres incluent le travail de préparation des données. Par exemple, le travail de TA statistique a demandé un total de 1,1 homme/mois d'informaticien, dont une bonne partie a été consacrée au travail préparatoire des données et à l'expérimentation avec les outils libres (GIZA++, SRLIM, Pharaoh/Moses). À ce coût s'ajoute le coût du travail de déconversion qui a coûté presque 1 homme/mois d'informaticien linguiste. En effet, il a fallu d'abord comprendre la spécification de l'outil DeCo et s'entraîner avec pour construire les règles de déconversion. Cela constitue un avantage pour le portage vers une deuxième langue, puisque la mise en place de l'infrastructure est déjà acquise.

Conclusion

Nous avons présenté le détail de trois approches de portage linguistique d'applications de traitement des énoncés spontanés d'un sous-langage. L'approche basée sur la traduction automatique ne demande pas d'accès au code de l'application ni à la représentation interne de l'application à porter et elle donne de meilleurs résultats. Pour expérimenter la traduction statistique, les

systèmes statistiques et les outils liés à ce travail sont disponibles et se sont améliorés ces dernières années. De plus, c'est la méthode qui demande le moins d'expertise linguistique et qui est donc *a priori* la plus largement applicable.

Malgré la petite taille du corpus d'entraînement utilisé (1 100 SMS), nous avons obtenu de bons résultats de traduction statistique et, en conséquence, de bons résultats d'extraction d'informations, ce qui donne un pourcentage de portage égal à 93 % par rapport à l'application originale et de 86 % par rapport à une version de référence. La construction semi-automatique du corpus d'entraînement n'introduit pas de régularités qui rendent les sous-langages proches. En effet, nous sommes arrivé à la même conclusion par la méthode de portage interne (Hajlaoui, 2008) dans laquelle nous n'avons modifié que 5 % des règles pour passer du sous-langage de l'arabe vers le français. C'est une autre confirmation de l'hypothèse de Kittredge sur les sous-langages, selon laquelle deux sous-langages qui correspondent dans deux langues différentes sont très proches, souvent plus proches qu'ils ne le sont chacun de leur langue-mère respective, ce qui permet de les considérer et de les traiter comme des variantes l'un de l'autre.

Références

BESACIER, Laurent (2007). *Transcription enrichie de documents dans un monde multilingue et multimodal*. Habilitation à diriger les recherches. Université Joseph Fourier, Grenoble, France.

BIBER, Douglas (1993). « Using Register-Diversified Corpora for General Language Studies ». *Journal of Computational Linguistics*, 19, 2, p. 219-241.

BLANCHON, Hervé (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.* Université Joseph Fourier, Grenoble, France.

BOITET, Christian (1990). « La TAO à Grenoble en 1990 ». *Rap. école d'été de Lannion sur le TALN*. Lannion, France.

BROWN, Peter *et al.* (1993). « The Mathematics of Statistical Machine Translation: Parameter Estimation ». *Computational Linguistics*, 19, 2, p. 263-311.

BROSS, I. D. J., P. A. SHAPIRO et B. B. ANDERSON (1972). « How Information is Carried in Scientific Sub-Languages ». *Science*, 176, 4041, p. 1303-1307.

CHANDIOUX, John (1988). « 10 ans de METEO (MD). Traduction assistée par ordinateur ». *In* A. Abbou, dir. *Actes du séminaire international sur la TAO et dossiers complémentaires*. Paris. OFIL, p. 169-173.

DAOUD, Maher (2006). *It is Necessary and Possible to Build (multilingual) NL-based Restricted e-commerce Systems with Mixed Sublanguage and Contend-oriented Methods.* Université Joseph Fourier, Grenoble, France.

DEVILLE, Guy (1989). *Modelization of Task-oriented Utterances in a Man-Machine Dialogue System.* University of Antwerpen, Belgique.

DODDINGTON, George (2002). « Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics ». *Proceedings of the Second International Conference on Human Language Technology Research*, March 24-27, 2002, San Diego, California, p. 128-132.

GRISHMAN, Ralph et Richard KITTREDGE (1986). *Analyzing Language in Restricted Domains*. Hillsdale NJ, Lawrence Erlbaum Associates.

HAJLAOUI, Najeh (2008). *Multilinguïsation de systèmes de e-commerce traitant des énoncés spontanés en langue naturelle*. Université Joseph Fourier, Grenoble, France.

HAJLAOUI, Najeh et Christian BOITET (2007). « Portage linguistique d'applications de gestion de contenu ». *in TOTh. Terminologie et Ontologie : Théories et Applications*, Actes de la première conférence TOTh - Annecy, 1er juin 2007. Annecy. Institut Porphyre, Savoir et Connaissance.

HAJLAOUI, Najeh, Maher DAOUD et Christian BOITET (2008). « Methods for Porting NL-Based Restricted E-Commerce Systems into Other Languages ». *Proceedings of LREC 2008*. Marrakech, Maroc.

HARRIS, Zellig (1968). « Mathematical Structures of Language ». *The Mathematical Gazette*, 54, 388, p. 173-174.

KITTREDGE, Richard et John LEHRBERGER (1982). *Sublanguage - Studies of Language in Restricted Semantic Domain*. New York, Walter de Gruyter.

KOEHN, Philipp (2004). « Pharaoh: a Beam Search Decoder for Phrase-Based SMT ». *Proceedings of AMTA*, Washington, DC, p. 115-124.

KUMAMOTO, Tadahico (2007). « A Natural Language Dialogue System for Impression-based Music-Retrieval ». *Proceedings of CICLING-07*. Mexico, 12-24 février 2007, p. 19-24.

LEPAGE, Yves (2005). « Translation of Sentences by Analogy Principle ». *Proceedings of Language & Technology Conference*. Poznań, Poland, 21-23 avril 2005.

MOSES, décodeur pour la traduction statistique. [http://www.statmt.org/moses/].

OCH, Franz Josef et Hermann NEY (2003). « A Systematic Comparison of Various Statistical Alignements Models ». *Computational Linguistics*, 1, 29, p. 19-51.

PAPINENI, Kishore *et al.* (2002). « BLEU: a Method for Automatic Evaluation of Machine Translation ». *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, July 2002, p. 311-318.

PHARAOH, décodeur pour la traduction statistique. [http://www.isi.edu/licensed-sw/pharaoh/].

PROJET CSTAR, (1986-2003). [http://www.c-star.org/].

SEKINE, Satoshi (1994). « A New Direction for Sublanguage NLP ». *Proceedings of the International Conference on New Methods in Natural Language Processing*. Manchester, England, p. 123-129.

SELKOW, Stanley M. (1977). « The Tree-to-Tree Editing Problem ». *Information Processing Letters*, 6, p. 184-186.

SRILM, Stanford Research Institute Language Modeling toolkit. [http://www.speech.sri.com/projects/srilm/].

STOLCKE, Andreas (2002). « SRILM: an Extensible Language Modeling Toolkit ». *Proceedings of the International Conference on Spoken Language Processing*, vol. 2. Denver, USA, p. 901-904.

UCHIDA, Hiroshi et Meiying ZHU (2003). *The Universal Networking Language specification*. Rap. UNU/IAS, Tokyo.

UCHIDA, Hiroshi et Meiying ZHU (2005-2006). *Universal Networking Language*. UNDL Foundation.

Najeh Hajlaoui
Laboratoire LIG
Université Joseph Fourier
BP 53 385, rue de la bibliothèque
F - 38041 Grenoble Cedex 9 France
Najeh.Hajlaoui@gmail.com