Université de Montréal

Déploiement automatique d'une application de routage téléphonique d'une langue source vers une langue cible.

par
Jérôme Tremblay

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Août, 2010

Université de Montréal
Faculté des arts et des sciences

Ce mémoire intitulé:

Déploiement automatique d'une application de routage téléphonique d'une langue source vers une langue cible.

présenté par:

Jérôme Tremblay

a été évalué par un jury composé des personnes suivantes:

Jian-Yun Nie,	président-rapporteur
Philippe Langlais,	directeur de recherche
Réal Tremblay,	Codirecteur
Matthieu Hébert,	Codirecteur
El Mostapha Aboulhamid,	membre du jury

Mémoire accepté le: . . . . . .

RÉSUMÉ

Les modèles de compréhension statistiques appliqués à des applications vocales nécessitent beaucoup de données pour être entraînés. Souvent, une même application doit pouvoir supporter plusieurs langues, c'est le cas avec les pays ayant plusieurs langues officielles. Il s'agit donc de gérer les mêmes requêtes des utilisateurs, lesquelles présentent une sémantique similaire, mais dans plusieurs langues différentes. Ce projet présente des techniques pour déployer automatiquement un modèle de compréhension statistique d'une langue source vers une langue cible. Ceci afin de réduire le nombre de données nécessaires ainsi que le temps relié au déploiement d'une application dans une nouvelle langue.
Premièrement, une approche basée sur les techniques de traduction automatique est présentée. Ensuite une approche utilisant un espace sémantique commun pour comparer plusieurs langues a été développée. Ces deux méthodes sont comparées pour vérifier leurs limites et leurs faisabilités.

L'apport de ce projet se situe dans l'amélioration d'un modèle de traduction grâce à l'ajout de données très proche de l'application ainsi que d'une nouvelle façon d'inférer un espace sémantique multilingue.

Mots clés: Traduction automatique, Classification, Domaine sémantique, Noyau sémantique.

ABSTRACT

Statistical understanding models applied to dialog applications need a lot of training data. Often, an application needs to support more than one language. This is relevant for countries that have more than one official language. In those applications, users queries convey the same meanings but in different languages. This project presents techniques to automatically deploy statistical comprehension models from a source language to a target language. The goal is to reduce the training data needed and the time requiered to deploy an application in a new language.

First, an approach using machine translation techniques is presented. Then, an approach that uses a common semantic space to compare both languages has been developed. Those methods are compared to verify their limits and feasibility.

This work present an improvement of the translation model using in-domain data and a novel technique for inferring a multilingual semantic space. **Keywords: Machine translation, classification, semantic domain, semantic Kernel.**

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES

ASR	Automatic Speech Recognition
DM	Domain Model
DS	Domain Space
ESA	Explicit Semantic Analysis
HV	Hors Vocabulaire
IDF	Inverse Document Frequency
LSA	Latent Semantic Analysis
SMT	Statistical Machine Translation
VSM	Vector Space Model

NOTATION

$\mathbb{R}$	Ensemble des Réels
D_c	Matrice de domaine cible
D_s	Matrice de domaine source
I^{IDF}	Matrice diagonale où $I_{i,i} = IDF(w_i)$
V_C	Vocabulaire de la langue C
c_1^J	Séquence de mots c_1 à c_J
$\overline{c_i}$	i^{eme} fragment.

CHAPITRE 1

INTRODUCTION

It's as if we're higher apes who had a language faculty inserted

-Noam Chomsky

Depuis l'avènement de l'intelligence artificielle, les chercheurs ont tenté de développer des systèmes en apparence intelligents. Une des branches de l'intelligence artificielle est la compréhension des langues naturelles. Aujourd'hui ces systèmes sont pour la plupart construits sur des bases statistiques, ce qui demande une quantité importante de données pour les entraîner. Le déploiement d'une application de ce type dépend donc de la quantité, mais aussi de la qualité des données disponibles.

Ce projet s'intéresse particulièrement aux applications de routage téléphonique qui aiguillent un utilisateur selon une demande prononcée oralement vers le bon opérateur ou le système automatisé adéquat. Plus précisément, les systèmes de reconnaissance automatique de la parole (Automatic Speech Recognition ASR) comprennent plusieurs composantes : modèles acoustiques, modèles de langage statistiques et modèles de compréhension statistiques. Chacune de ces composantes nécessite une très grande quantité de données pour être entraînée. Plus les données sont proches de l'application (dans le domaine), plus le système sera performant [2]. Ceci constitue un défi quant à la portabilité d'un système à un nouveau domaine [6]. Ce défi a pour conséquence d'augmenter le coût de déploiement d'une nouvelle application. Car à chaque déploiement d'une nouvelle application, il y a une étape de collecte et de traitement des données qui doit être réalisée manuellement, ce qui prend un temps et des efforts importants.

Ici, la tâche sera une application téléphonique du type "*Comment puis-je vous aider ?*" (CPVA) caractérisée par une interface de commande vocale demandant à l'uti-

lisateur de formuler sa requête dans ses propres mots. Ce genre d'invite ouverte est fréquemment rencontré dans les applications commerciales en traitement des langues naturelles. Dans ce type d'application, la compréhension des requêtes est généralement gérée par des classifieurs statistiques. Ce sont ces classifieurs qui donnent une étiquette aux énoncés produits par les utilisateurs.

L'entraînement de ces systèmes demande beaucoup de données transcrites et étiquetées. Souvent, une compagnie possède beaucoup de données étiquetées pour des applications dans une langue hautement représentée (e.g. l'anglais), mais peu ou pas pour des langues peu représentées. L'idée principale de ce projet est la réutilisation des données d'entraînement d'une langue hautement représentée pour déployer un système dans une autre langue, possiblement peu représentée ; donc la réutilisation de données à travers la barrière des langues. L'entraînement est l'étape où le classifieur *apprend* à mettre une étiquette sur les énoncés qu'on lui demande de classifier. C'est cet *apprentissage* qui doit être fait pour chaque langue.

Les prochaines sections présenteront une définition du problème, les objectifs à réaliser ainsi que les systèmes utilisés tout au long du projet.

1.1 Définition du problème

Le problème se résume comme suit : on veut entraîner un classifieur prenant en entrée des énoncés dans une langue source et classifier avec les mêmes étiquettes les énoncés d'une langue cible. Par exemple, entraîner le classifieur avec des énoncés en anglais et tester avec des énoncés en français. Les données d'entraînement sont constituées de transcriptions d'énoncés oraux prononcés par des utilisateurs d'un système de routage téléphonique. Ces données possèdent une étiquette sémantique définissant le sens de la requête de l'utilisateur. Ces étiquettes sont définies préalablement à la cueillette des

données. L'étiquette est un mot qui décrit globalement ce que l'utilisateur demande. Les étiquettes sont définies lors de la conception de l'application et leur nombre est fixe.

Par exemple, dans une application de fournisseur d'électricité dans laquelle on peut effectuer des opérations sur son compte comme un changement d'adresse. Les données de la forme *donnée :étiquette* pourraient être comme à la figure 1.1

Figure 1.1 – Exemple de données d'un système de routage téléphonique

```
je voudrais faire un changement d'adresse:ChangementAdresse
connaître ma consomation:Consomation
faire un paiement:Paiement
```

La section 1.3 définira plus en détail le contexte de l'application et les systèmes utilisés. La prochaine section présente les objectifs liés à ce projet.

1.2 Objectifs

Les meilleurs systèmes de classification utilisent une classification supervisée, donc qui font usage des données étiquetées. Les corpus nécessaires pour entraîner leurs modèles demandent que les échantillons annotés soient représentatifs des utilisateurs de l'application. Il est important d'avoir des données réelles pour que l'application soit plus performante. Plus les données d'entraînement sont représentatives des requêtes des utilisateurs, plus l'application sera performante. Car elle aura bien appris à gérer les requêtes d'usagers réels.

L'objectif du projet est la réduction de la quantité de données nécessaire au déploiement d'une application en minimisant les pertes de performance, et ce, en réutilisant les données d'une application dans une autre langue ; en d'autres termes : utiliser les données d'une langue source pour déployer une application dans une langue cible. Il

s'agit de déployer la même application d'une langue source vers une langue cible. Ce projet chevauche deux domaines : la traduction automatique et l'apprentissage machine. L'apprentissage machine, car il s'agit de classification d'énoncés et l'entraînement du classifieur statistique. La traduction automatique pour traduire les données. Les objectifs détaillés du projet sont :

- Vérifier la faisabilité de réutiliser les données d'entraînement d'un système d'une langue source vers une langue cible.
- Comparer les performances d'un système de classification entraîné en utilisant des données traduites à l'aide de techniques de traduction automatique avec un système utilisant des techniques de classification multilingue. Comparer ces deux techniques avec le système entraîné sur les données réelles dans les deux langues (source et cible).
- Développer une nouvelle plate-forme pour le déploiement d'une application de classification multilingue.

Ce document présente la réalisation de ces objectifs avec deux techniques : une technique provenant de la traduction automatique et une technique utilisant un espace sémantique commun qu'on appellera à l'avenir *classification multilingue*. Ces deux techniques sont comparées afin d'en étudier leurs forces et faiblesses.

1.3 Description des systèmes

Ce projet utilise un système de routage téléphonique complet développé à l'interne chez Nuance[1][2], voici comment le système est construit et utilisé.

La figure 1.2 montre le schéma d'utilisation d'un système de routage téléphonique typique. La liste suivante décrit chaque étape du système.

[1] www.nuance.com

[2] Les données proviennent de clients de Nuance et ne peuvent pas être divulguées.

1. L'utilisateur énonce une requête orale. Il s'agit de l'entrée du système.

2. Cette requête orale est ensuite reconnue par un système de reconnaissance de la parole. La sortie de cette étape est le texte associé à la requête orale de l'utilisateur.

3. Ce texte est fourni à un classifieur statistique qui décide de la sémantique de l'énoncé (étiquette) d'entrée et donne une *destination* en sortie, soit une étiquette sémantique.

4. La sortie du système représente la sémantique reliée à la requête de l'utilisateur.

Les blocs ombragés de la figure 1.2 ne sont pas utilisés pour les expériences afin d'enlever la variable reconnaissance vocale. Comme on veut comparer le système de référence anglais et celui entrainé avec les données en français, c'est la différence entre les deux systèmes qui est importante et non pas les performances absolues des systèmes. Ce sont donc les vraies transcriptions qui sont utilisées en entrée, comme le montre la Figure 1.3 . Ces transcriptions ont été réalisées par des gens entraînés à cette tâche qui ont écouté chaque fichier audio et transcrit le texte associé.

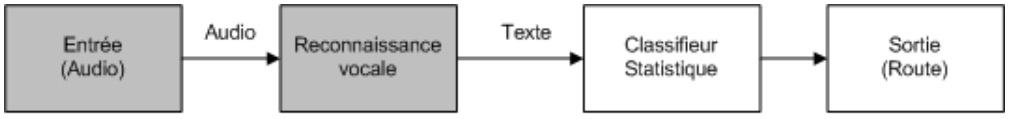


Figure 1.2 – Schéma d'utilisation d'un système de routage

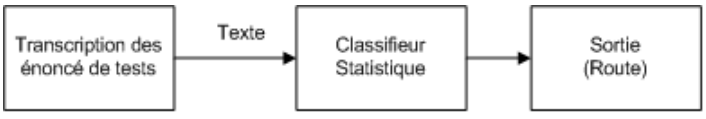


Figure 1.3 – Schéma du système utilisé pour les expériences

Pour résoudre ce problème de réutilisation des données entre deux langues, deux approches différentes ont été tentées.

- Premièrement, une approche de traduction automatique qui utilise les techniques du domaine de la traduction automatique statistique afin de traduire les données de la langue source vers la langue cible.

- Une seconde approche de classification multilingue qui utilise une projection dans un espace sémantique commun afin de pouvoir comparer les deux langues dans le même espace.

C'est deux techniques demandent des corpus externes qui doivent être disponibles ou construits spécialement pour l'application. Ces données sont décrites au chapitre 4. Afin de bien définir ces deux approches, le prochain chapitre introduit l'état-de-l'art pour chacune de ces deux techniques. Par la suite, le chapitre 3 décrira en détail ces deux approches et leurs applications à la résolution de ce problème de réutilisation des données. Cette section se terminera par une comparaison entre ces deux techniques. Ensuite, le chapitre 4 décrira les données utilisées. Enfin, le chapitre 5 présentera les métriques utilisées et les résultats obtenus pour ces deux approches.

CHAPITRE 2

ÉTAT DE L'ART

> I have bought this wonderful machine- a computer [...] it seems to me to be an Old Testament god with a lot of rules and no mercy.
>
> *-Joseph Campbell*

Ce mémoire utilise deux techniques tirées de deux domaines différents afin de résoudre le même problème, celui du déploiement automatique d'une application de routage téléphonique d'une langue source vers une langue cible. Il convient donc de présenter l'état de l'art de chacun de ces domaines afin de débuter sur une base théorique solide. Les domaines de recherche présentés dans ce chapitre seront la classification d'énoncés et la traduction automatique. Le chapitre aborde en premier lieu la traduction automatique en donnant un bref résumé du problème et des techniques proposées pour le résoudre. Par la suite l'état de l'art sur la classification d'énoncés est présenté.

2.1 Traduction automatique

Traduire automatiquement consiste à transformer un énoncé dans une langue source en un énoncé d'une langue cible. Il existe plusieurs techniques pour construire un système de traduction automatique, mais ici seulement la traduction automatique statistique sera abordée, car c'est cette technique qui est utilisée pour ce projet. Ce type de traduction a été choisi, car il est adaptable et ne demande pas d'expertise linguistique. De plus, il est cohérent avec la chaîne de traitement statistique utilisée par le système de routage téléphonique et se classe parmi les types de traduction qui offre les meilleurs résultats.

La description de la traduction automatique statistique suivra la notation suivante : la langue source sera notée S et la langue cible C. Plus formellement : on dénote l'énoncé

de J mots sources $s_1, s_2, ...s_J$ où $s_1^J \in V_S^J$. V_S est le vocabulaire de la langue source. Donc l'ensemble des séquences de J mots appartiennent au vocabulaire de langue source. De la même façon on définit l'énoncé de I mots cible $c_1, c_2, ..., c_I$ où $c_1^I \in V_C^I$. V_C est le vocabulaire de la langue cible.

On veut trouver l'énoncé cible qui maximise $P(C|S)$ [3], on veut donc maximiser la probabilité de la séquence cible C d'être la traduction de la séquence d'entrée S. La meilleure séquence de sortie sera celle qui maximise la probabilité $P(C|S)$.

$$\hat{C} = argmax_C P(C|S) \tag{2.1}$$

La notation *argmax* signifie l'argument qui maximise la probabilité $P(C|S)$ sur l'ensemble des C. En utilisant la règle de Bayes, on peut réécrire :

$$argmax_C P(C|S) = argmax_C \frac{P(C)P(S|C)}{P(S)} \tag{2.2}$$

$P(C)$ est le modèle de la langue C, soit une distribution proabiliste sur l'ensemble des séquences de mot et $P(S|C)$ est le modèle de traduction. Ces modèles sont présentés dans les prochaines sections. On peut ne pas tenir compte du dénominateur car il sera constant pour la séquence S d'entrée.
Les modèles de traduction se basent sur l'alignement de mots. Il s'agit d'un alignement entre les mots des phrases sources et cibles. Avant d'introduire la notion d'alignement, le modèle de langue ainsi que le modèle de traduction sont définis.

2.1.1 Modèle de langue

Le modèle de langue va servir au calcul de la probabilité $P(C)$ ou $P(c_1^I)$, soit la probabilité de la séquence de mots cibles en sortie. Le modèle de langue utilisé pour la traduction automatique est un modèle n-gramme estimé sur la langue cible afin que les traductions en sortie soient en accord avec la langue cible. Ce modèle prédit le prochain mot sachant les $n-1$ mots précédents. Il faut donc estimer la probabilité [17] :

$$P(c_n|c_1,...,c_{n-1}) \tag{2.3}$$

La prédiction est réalisée en tenant compte de l'historique des $n-1$ mots précédents. Le modèle est construit en faisant l'hypothèse que le mot qu'on tente de prédire dépend uniquement des $n-1$ mots qui le précèdent. Il s'agit de la propriété de Markov. Si on veut trouver la probabilité d'une séquence de mots, grâce à la règle de chaînage, on peut écrire pour le modèle bigramme :

$$P(c_i^I) = \prod_{n=1}^{I} P(c_n|c_{n-1}) \tag{2.4}$$

Ici, la probabilité d'un mot ne dépend que du mot qui le précède. Si on veut formaliser au n-gramme, la probabilité d'un mot dépend de la probabilité conditionelle des $n-1$ mots qui le précède, l'équation devient :

$$P(c_i^I) = \prod_{n=1}^{I} P(c_n|c_{n-I+1}^{n-1}) \tag{2.5}$$

2.1.2 Modèle de traduction

Après le modèle de langue, il faut un modèle qui définira le lien entre la langue source et la langue cible, c'est ce que fait le modèle de traduction. Le modèle de traduction présenté est un modèle de fragment, au lieu d'utiliser le mot comme unité de base, on utilise un fragment : soit une courte séquence de mots. La figure 2.1 montre un exemple d'un alignement par fragments.

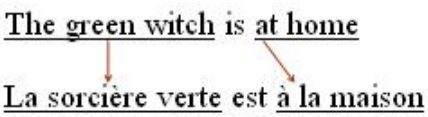


Figure 2.1 – Exemple d'alignement par fragments

Le modèle par fragments présenté par Koehn et al. [15] est décrit par la relation

suivante :

$$P(S|C) = \prod_{i=1}^{I} \phi(\overline{s_i}, \overline{c_i}) d(a_i - b_{i-1}) \tag{2.6}$$

Dans ce modèle $\phi(\overline{s_i}, \overline{c_i})$ est la probabilité que le fragment $\overline{s_i}$ soit la traduction de $\overline{c_i}$. La seconde partie de l'équation réfère à la distortion. La distortion d indique si un mot est traduit puis inséré à une position différente de sa position d'origine. L'ordre des mots n'est pas toujours le même dans deux langues différentes et la distorsion a pour but de pallier ce problème. Dans le cas présent, la distorsion est paramétrisée par $d(a_i - b_{i-1})$, où a_i est la position initiale de la séquence de la langue source générée par la i^s séquence cible $\overline{c_i}$. b_{i-1} réfère à la position finale de la séquence langue source générée par la $(i-i)_{eme}$ séquence cible $\overline{c_{i-1}}$.

La figure 2.2 montre un exemple du modèle par fragment. Étant donné que les fragments et leur traduction sont au même endroit, la distorsion sera de 1 pour tous les fragments. Dans cet exemple le modèle de traduction serait :

$$P(S|C) = \quad P(La\ sorcire\ verte|The\ green\ witch) \times d(1) \times \tag{2.7}$$

$$P(est|is) \times d(1) \times P(\ la\ maison|at\ home) \times d(1) \tag{2.8}$$

Figure 2.2 – Exemple de traduction par fragment

Position	**1**	**2**	**3**
Anglais	The green witch	is	at home
Français	La sorcière verte	est	à la maison

Dans ce modèle : $\phi(\overline{s_i}, \overline{c_i})$ est obtenu à l'aide d'alignements mot à mot et d'heuristiques [15] décrits dans les prochaines sections.

2.1.3 Alignement mot à mot

Dans leur article, Brown et al. [3] ont présenté 5 modèles d'alignement : les modèles IBM 1-5. Ici, on ne décrit que le modèle IBM 1 car il sera utilisé dans les approches pour faire de la traduction mot-à-mot. Dans le modèle IBM 1, on ne permet d'aligner chaque mot cible qu'à un mot source. Par contre, un mot source peut être aligné à plusieurs mots cibles. Soit $A(C,S)$ les alignements possible entre C et S la relation entre l'alignement et la traduction peut s'écrire [13] :

$$P(S|C) = \sum_{A} P(S,A|C) \tag{2.9}$$

Ici, $P(S,A|C)$ est la probabilité de générer une phrase en français sachant la phrase anglaise selon un alignement déterminé. Donc pour calculer $P(S|C)$ on somme sur tous les alignements valides, comme on cherche un alignement on doit trouver :

$$\hat{A} = argmax_A P(S,A|C) \tag{2.10}$$

2.1.4 Obtention des fragments

Afin d'extraire les fragments utilisés par le modèle de traduction par fragments, on utilise un alignement bidirectionnel. Pour ce faire, il faut entrainer un aligneur source-cible et cible-source [20] avec un alignement mot-à-mot comme décrit à la section précédente. Ensuite, on combine les deux alignements en prenant l'intersection ou l'union des deux alignements pour en extraire les fragments à l'aide d'une heuristique qui choisira comment les découper [15]. L'intersection des deux alignements permet d'avoir des points de confiance élevés pour extraire les fragments. La figure 2.3 montre un exemple d'alignement bidirectionnel et de son utilisation pour en extraire les fragments. Les fragments trouvés dans cet exemple pourraient être ceux de la figure 2.4.

Lorsque les fragments sont trouvés, il est possible d'estimer leur probabilité de traduction en faisant le compte de leur coocurence dans le corpus parallèle :

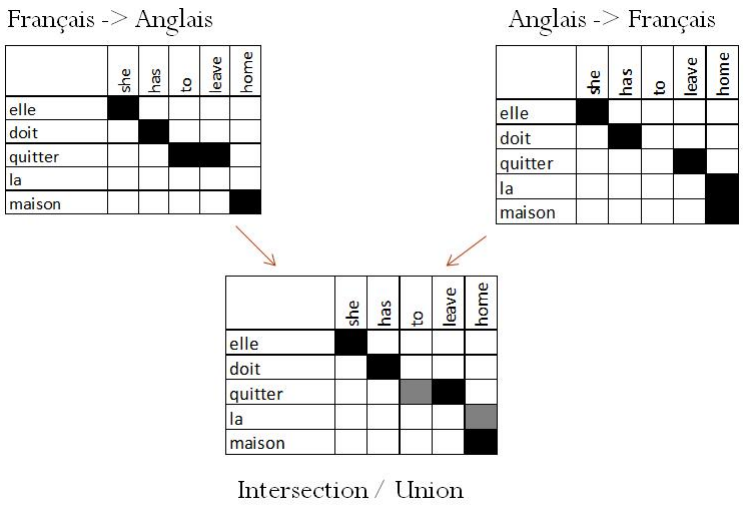


Figure 2.3 – Exemple d'alignement bidirectionnel

Figure 2.4 – Fragments trouvés dans l'exemple d'alignement bidirectionnel

```
elle | she
doit | has
quitter | to leave
doit quitter | has to leave
la maison | home
```

$$\phi(\overline{s},\overline{c}) = \frac{compte(\overline{s},\overline{c})}{\sum_{\overline{s}} compte(\overline{s},\overline{c})} \tag{2.11}$$

Ces valeurs sont ensuite sauvegardées dans un fichier qu'on appelle la *phrase-table*.

2.1.5 Représentation des modèles de traduction

Les données du modèle par fragment sont conservées dans la *phrase-table*, il s'agit d'une table qui conserve les probabilités de traduction des fragments. Afin d'illustrer le contenu de cette table, la figure 2.5 montre des extraits de la *phrase-table* utilisée pour traduire les données d'entrainement. Ces extraits présentent les fragments dans la langue source puis dans la langue cible.

Figure 2.5 – Extrait de la *phrase-table* utilisée pour effectuer les traductions utilisant le modèle par fragment

1. de façon unilatérale , sans aucune ||| , unilaterally and without any
2. acheminement d' information vers ||| flow of information to

Pour ce qui est du modèle de traduction mot-à-mot, la figure 2.6 montre un extrait de la table de traduction du modèle IBM 1. Il s'agit uniquement des probabilités de traduction mot-à-mot. C'est la probabilité maximale qui est utilisée pour la traduction. Ce modèle est utilisé sans décodage et sans modèle de la langue cible, ce qui n'est pas optimal. Un meilleur système de traduction aurait pu être réalisé en utilisant un décodage qui utilise le modèle de la langue source. Par contre, étant donné que le modèle VSM utilisé pour la classification ne tient pas compte l'ordre des mots et que les énoncés à traduire sont très différents d'un document écrit, la traduction mot-à-mot est tentée afin de vérifier ses performances.

Figure 2.6 – Extrait de la table de traduction utilisée pour les traductions mot-à-mot

```
bureau desk 0.0373522
bureau table 0.0366491
bureau him 0.000547862
bureau see 0.000335664
bureau law 0.00104647
bureau it 0.000102859
```

2.2 Classification d'énoncés

La classification d'énoncés est importante pour certaines applications de traitement des langues naturelles. Malgré des différences, les techniques utilisées en recherche d'information et de classification de texte peuvent être appliquées efficacement à ce problème.

La classification d'énoncés diffère de la classification de texte, car les énoncés proviennent de la langue parlée et non de textes écrits. Bien que ce document traite des énoncés comme s'il s'agissait de texte, un énoncé de la langue parlé est différent d'un énoncé écrit. Le nombre de mots dans un énoncé est aussi beaucoup plus faible que dans un texte. Ce type de classification utilise généralement une représentation *sac-de-mot*, c'est-à-dire une représentation vectorielle dans laquelle chaque mot différent représente une dimension. Cette représentation s'appelle *modèle d'espace vectoriel* (Vector Space Model VSM en anglais).

Par contre, les modèles VSM ne prennent pas en compte les relations entre les mots ni l'information sémantique. Comme il sera indiqué dans la section décrivant le VSM, ces modèles font face à un problème de dispersion des données. Une méthode pour pallier ce problème est l'utilisation de noyaux sémantiques. Les noyaux sémantiques vont réduire la disparité de la représentation VSM [28] en insérant de l'information sémantique. Comme les méthodes à base de noyaux, les noyaux sémantiques vont projeter le vecteur VSM dans un espace de dimension plus élevée et plus sémantiquement discriminante. D'un autre côté, les techniques basées sur les LSA (Latent Semantic Analysis) utilisent une décomposition en valeurs singulières ou SVD (Singular Value Decomposition)[1], ce qui fera en sorte de réduire la dimensionnalité de l'espace de classification.

2.2.1 Le VSM

Les modèles vectoriels ont été les modèles état de l'art pour la représentation de document pendant trois décennies et donne de bons résultats [25]. La représentation vectorielle apportée par Salton a d'abord été développée pour être utilisée en recherche d'informations pour être ensuite appliquée à la classification de documents. De la même

[1]Qui est analogue à l'analyse en composante principale (PCA Principal Component Analysis). En traitement de signal on parle parfois de KLT (Karhunen-Loève Transform)

manière qu'on décrit un ensemble de documents, ici, on utilise cette représentation pour représenter un ensemble d'énoncés présentant la même sémantique dans l'espace vectoriel des mots. Chaque mot du vocabulaire du corpus d'entraînement est considéré comme une dimension du VSM. On suppose que les énoncés sont déjà étiquetés avec une étiquette représentant leur sémantique. Plus formellement, soit un énoncé e, on peut le représenter par un vecteur ligne :

$$\phi : e \rightarrow \phi(e) = (tf(t_1, e), ..., tf(t_E, e)) \in \mathbb{R}^E \tag{2.12}$$

Où la fonction $tf(t_i, e)$ est la fréquence du terme t_i dans l'énoncé e et où E est la taille du dictionnaire du corpus d'entraînement. En utilisant cette représentation on crée une matrice T qui représente la fréquence des termes dans chacune des classes.

$$T = \begin{bmatrix} tf(t_1, e_1) & ... & tf(t_E, e_1) \\ tf(t_1, e_2) & ... & tf(t_E, e_2) \\ ... & & ... \\ tf(t_1, e_n) & ... & tf(t_E, e_n) \end{bmatrix} \tag{2.13}$$

Chaque ligne de la matrice T représente un vecteur de poids pour chaque mot les reliant à chaque classe d'énoncés. Chaque colonne est un vecteur pour chaque classe d'énoncés présentant le poids de chaque mot pour cette classe d'énoncés. Le plus grand désavantage de cette technique est que le vecteur représentant un énoncé est très dispersé. En fait la plupart des valeurs reliées aux dimensions sont à 0 car ces mots ne sont pas présents dans l'énoncé. De plus, un énoncé étant généralement très court il n'y a que peu de dimensions qui ont une valeur non nulle. Les données sont donc sous représentées dans l'espace des mots. Un moyen de remédier à cela consiste à utiliser des noyaux sémantiques, qui vont venir augmenter la matrice T avec de l'information relative entre les mots.

2.2.2 Classifieur

Le modèle VSM est utilisé par les classifieurs statistiques du système de routage téléphonique pour l'entrainement et la classification. Il s'agit de classification supervisée, car les données possèdent une étiquette. Les données textuelles sont traduites en VSM puis fournies à l'entrée du classifieur. Il existe plusieurs types de classifieurs et leur description dépasse le cadre de ce document. De façon générale, un classifieur est un modèle qui prend une décision basée sur des données d'entrées. Ici l'entrée sera le vecteur VSM et la décision du classifieur, l'étiquette des données. Les étiquettes sont aussi appelées les *classes* du classifieur. Quand on parle des classes d'un problème, il s'agit des étiquettes liées aux données.

Ce modèle est entrainé sur des données d'entrainement et testé sur des données de test. Il est important que les données de test ne proviennent pas des données d'entrainement ; si c'est le cas, les résultats obtenus ne mesureront pas les performances de généralisation du modèle. On dit qu'un modèle *généralise* bien, s'il est capable de prendre la bonne décision sur des données qu'il n'a pas encore vues. C'est ce qu'on cherche à mesurer avec les données de test. Car en pratique, il n'est pas possible de fournir toutes les données à classifier lors de l'entrainement.

Pour reprendre ce dont il est question dans ce document : il existe un classifieur entrainé sur des données dans une langue source et on veut en entrainer un second avec les mêmes données, mais qui prendra une décision sur une langue cible.

2.2.3 Noyau sémantique

Les noyaux sémantiques [28] sont une technique pour injecter de l'information sémantique (entre autres) dans les applications de classification d'énoncés. La version la plus simple est la relation de premier ordre. La matrice S est la matrice qui détient l'in-

formation qui relie les mots de la matrice T entre eux.

$$S = TT^t \tag{2.14}$$

Cette représentation n'utilise pas de données externes à la tâche et encode seulement les relations de premier ordre entre les mots. Il est possible d'utiliser des données externes pour inférer la matrice S. De manière plus générale on peut définir la matrice S comme [26] :

$$S = RP \tag{2.15}$$

où R est une matrice diagonale de poids pour chacun des mots du vocabulaire et P la matrice de proximité qui encode les relations entre les mots. Les poids de la matrice R sont généralement associé à l'IDF (*Inverse Document Frequency*) ; le poids de chacun des terme t_i devient alors :

$$R(i,i) = w(t_i) = \log\left(\frac{l}{df(t_i)}\right) \tag{2.16}$$

où l représente le nombre de classes ou de documents et $df(t_i)$ le nombre de documents dans lesquels le terme t_i apparaît. La valeur de l'IDF est dans $\mathbb{R}^+$. Intuitivement, cela signifie que plus un terme est fréquent dans plusieurs documents, moins il est discriminant. L'équation 2.16 est souvent utilisée en traitement des langues naturelles lorsqu'on veut associer un poids aux mots d'une application de type recherche de documents.

La matrice P reflète la proximité entre les mots, une façon de la créer est d'utiliser une base de données lexicale par exemple WordNet[2] pour trouver les relations entre les mots présents dans l'application. Cette approche est toutefois tributaire de la disponibilité de telles ressources. De plus, ce type de base de données n'est pas disponible dans toutes

[2]http ://wordnet.princeton.edu/

les langues. Il est aussi possible de la créer en utilisant un corpus externe et en calculant les cooccurrents des mots.

2.2.4 Les domaines sémantiques

L'approche de classification multilingue se base sur un espace sémantique commun. La création de cet espace tire ses fondations du concept des domaines sémantiques lequel est tiré de la théorie des champs lexicaux. Ces domaines correspondent à une signification précise d'une idée ou d'un concept. L'idée d'utiliser cette représentation est qu'un concept représente un sujet ou un sens en particulier tandis qu'un mot peut en avoir plusieurs [9]. L'hypothèse de cette théorie est que le lexique est structuré en champs lexicaux. Les relations sémantiques entre les concepts appartenant au même champ lexical sont très denses, tandis que les concepts appartenant à des champs lexicaux différents sont pratiquement indépendants.

Un autre point fort de cette théorie est qu'il existe de fortes relations entre les champs lexicaux de différentes langues. Les concepts sont les mêmes dans deux langues, tandis que les mots reliés à ces concepts sont différents[3].

Les domaines sémantiques deviennent donc un groupe de mots reliés à un domaine bien précis. Il est alors intuitivement plus discriminant de classifier dans l'espace des domaines que dans l'espace des mots. Un domaine sémantique comprend un ensemble de mots qui lui sont reliés, un mot peut faire partie de plusieurs domaines (polysémie). La figure 2.7 montre un exemple de domaines sémantiques pour trois domaines et les mots qui leurs sont reliés.

[3] Bien que dans certaines langues les mots peuvent se ressembler.

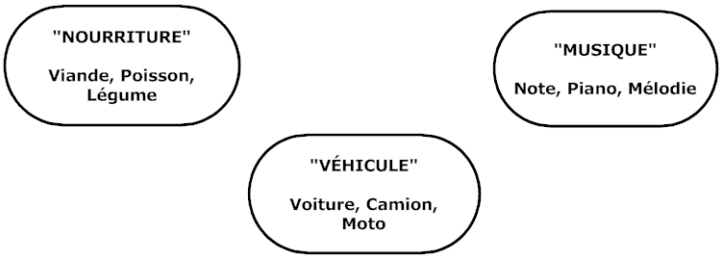


Figure 2.7 – Exemple de domaines sémantique

2.2.4.1 Le modèle de domaine

Le *domain model* (DM) [11] est un modèle pour représenter les domaines sémantiques en définissant des *clusters* de mots pour chaque domaine sémantique. Chaque *cluster* représente un domaine sémantique.

Le DM est défini par un ensemble de domaines :

$$DM = \{D_1, D_2, ..., D_k\} \tag{2.17}$$

et une matrice D qui comporte la relation entre les termes et les domaines. Cette matrice D encode le poids qu'ont les termes envers les domaines, donc la force avec laquelle les termes sont associés aux domaines.

$$D = \begin{array}{c} \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & \dots & D_k \end{array} \\ \begin{pmatrix} H(w_1,D_1) & H(w_1,D_2) & \dots & H(w_1,D_k) \\ H(w_2,D_1) & H(w_2,D_2) & \dots & H(w_2,D_k) \\ \vdots & \vdots & \ddots & \vdots \\ H(w_n,D_1) & H(w_n,D_2) & \dots & H(w_n,D_k) \end{pmatrix} \end{array} \tag{2.18}$$

où $H(w_n, D_k)$ est la fonction d'appartenance du terme au domaine. Le tableau 2.1 montre un exemple de DM tiré de [11].

Tableau 2.1 – Exemple de Domain Model

	MEDECINE	INFORMATIQUE
HIV	1	0
virus	0.5	0.5
laptop	0	1

2.2.4.2 L'espace des domaines

Avec le *Domain Model* et la matrice D, il est facile de projeter les termes dans l'espace des domaines. Par contre, si on veut projeter une phrase ou un texte dans l'espace des domaines, il faut faire la transformation linéaire suivante :

$$s'_j = s_j(I^{IDF}D) \tag{2.19}$$

où s_j est le vecteur correspondant à l'énoncé à projeter et I^{IDF} une matrice diagonale dont les valeurs sont $IDF(w_i)$. On pondère par une matrice IDF afin de donner plus d'importance aux mots qui contiennent plus d'information. Cette transformation fera en sorte que le vecteur s_j dans l'espace des mots sera projeté dans l'espace des domaines grâce à la matrice D qui contient l'information qui relie chaque mots aux domaines.

La projection du vecteur terme dans l'espace des domaines permet d'avoir un système plus flexible et moins sensible au bruit. Il permet aussi d'avoir un système facilement améliorable, dans le sens où de nouveaux domaines peuvent être ajoutés dès qu'un nouveau corpus est disponible. Cette approche permet aussi de contrôler la dimensionnalité du problème, car les dimensions du problème deviennent le nombre de domaines. Le nombre de domaines peut aussi être limité (e.g. par un algorithme de *clustering*). Par contre, le *clustering* des domaines peut devenir nécessaire afin de ne pas faire gonfler le modèle inutilement si beaucoup de corpus sont ajoutés.

L'avantage des domaines sémantiques est que n'importe quel corpus externe peut être ajouté, il n'a pas besoin d'être dans un format particulier. La seule particularité est qu'il faut le découper, soit par paragraphe, par texte, par phrase, etc.. Ce découpage doit

être fait, afin de créer des morceaux de données pour inférer les domaines. Ensuite, il faut utiliser un algorithme de clustering pour trouver les domaines à l'intérieur de ces données.

Une autre technique est d'utiliser immédiatement des domaines créés par des utilisateurs. Par exemple, l'encyclopédie en ligne Wikipédia fournit déjà une collection d'articles qu'il est possible d'utiliser pour créer des domaines sémantiques. C'est l'idée derrière l'ESA (Explicit Semantique Analysis) [8] : comme chaque article Wikipédia représente un sujet particulier, les domaines choisis sont ceux déjà créés par ces pages.

2.2.4.3 Création des domaines sémantiques

Les domaines sémantiques peuvent être créés avec n'importe quel algorithme de *clustering* qui permet de rassembler des ensembles de termes reliés à des concepts. Il s'agit de trouver une relation entre la matrice D et un espace dans lequel chaque dimension représente un concept précis. De cette manière, il est possible de pallier aux problèmes de polysémie et d'homonymie.

Dans le cadre de ce mémoire, deux algorithmes de création de domaines ont été analysés : l'analyse sémantique latente (Latent Semantic Analysis LSA) et l'analyse sémantique explicite (Explicit Semantic Analysis ESA). Le premier algorithme cherche à trouver des concepts latents présents dans les textes tandis que le second se base sur des séparations naturelles des concepts. L'algorithme de LSA a été utilisé par Gliozzo pour créer les domaines sémantiques [10].

2.2.4.3.1 LSA Le Latent Semantic Analysis (LSA) est une technique qui permet principalement d'évaluer la similarité entre des termes et des textes. Elle cherche à découvrir des concepts latents dans un corpus. Il s'agit de concepts latents, car ces concepts et leur nombre sont cachés. Le LSA exploite la décomposition en valeurs singulières de la matrice de termes afin d'extraire ces concepts latents. LSA crée donc un espace de concepts en regroupant les termes sémantiquement reliés ensemble. Il est possible d'uti-

liser son principe de base pour inférer des clusters dans le but de créer un DM. L'idée est d'utiliser une décomposition en valeurs singulières pour ensuite ne sélectionner que les dimensions les plus pertinentes. On décompose la matrice de VSM par les matrices :

$$T = U\Sigma V^t \tag{2.20}$$

Où Σ est une matrice diagonale, U et V sont des vecteurs dont les valeurs sont les valeurs propres de T^tT et TT^t. Les termes qui sont souvent en cooccurrences dans le corpus utilisé vont avoir tendance à s'aligner sur la même valeur propre. Lorsqu'on va reconstruire la matrice T, on va donc projeter la matrice dans un espace de *concepts* dans lequel les termes s'alignent le mieux.

2.2.4.3.2 ESA La seconde technique utilisée pour inférer l'espace sémantique est ESA ou Explicit Semantic Analysis qui est une technique qui utilise un découpage naturel des textes pour créer l'espace sémantique. Gabrilovich [8] a utilisé les concepts présents dans Wikipédia pour créer un espace sémantique. Dans sa représentation, chaque page Wikipédia est considérée comme une dimension, les mots présents dans cette page sont les mots reliés à cette dimension. Suivant cette idée, il est possible de créer un espace sémantique en utilisant n'importe quel partitionnement naturel de concepts.

Cette méthode dépend toutefois de la disponibilité de telles séparations naturelles de texte, Wikipédia est une source intéressante de documents généraux et variés. Par contre, cette technique est pertinente pour une compagnie qui possède des données étiquetées de plusieurs applications. Ces données peuvent être réutilisées en utilisant la technique de ESA.

2.2.5 Classification multilingue

Il existe peu de littérature scientifique dans le domaine de la classification multilingue. Gliozzo et al. [10] réalisent cette tâche en utilisant des corpus comparables et les mots communs à ces deux corpus afin de créer un espace sémantique commun. Vinokourov et al. [30] utilisent une technique appelée le Kernel Canonical Correlation Analysis (KCCA) pour inférer une représentation bilingue à l'aide de corpus alignés. Littman et al. [16] utilisent le Latent Semantic Indexing (LSI) pour créer un espace sémantique multilingue.

Suite à une revue de ces différentes méthodes, c'est l'approche de Gliozzo et al. sur les domaines sémantiques qui a été retenue pour développer le système de déploiement d'une application de routage téléphonique. Cette méthode offre de la flexibilité pour la création des domaines et de l'espace sémantique qui sera utilisé pour la classification. De plus, l'inférence de l'espace avec cette méthode ne demande pas de corpus parallèle. Cette approche est décrite dans la section 3.2.

CHAPITRE 3

APPROCHES UTILISÉES

> The flaw must lie in our methods of description, in languages, in social networks of meaning, in moral structures, and in philosophies and religions - all of which convey implicit limits where no limits exist.
>
> *-Frank Herbert*

Afin de résoudre le problème du déploiement d'une application de routage téléphonique dans une autre langue, deux approches ont été réalisées. Premièrement, la section 3.1 présentera l'approche orientée traduction automatique ainsi que le travail réalisé afin d'appliquer cette technique à la résolution du problème. Ensuite, la section 3.2 décrira l'approche classification multilingue. Enfin, la section 3.3 comparera les deux approches afin de montrer leurs différences ainsi que leurs avantages et inconvénients.

3.1 Traduction des données

L'approche la plus intuitive est de traduire les données d'entraînement de la langue source à la langue cible et de tester avec les données dans la langue cible. Une fois les données d'entraînement traduites, on construit le système de la même manière qu'on le construirait normalement, à l'exception qu'on teste maintenant avec les données de la langue cible. Pour effectuer cette approche, il est nécessaire d'avoir un modèle de langue et un système de traduction.

Cette approche peut être utilisée de deux façons :

- Traduire les données d'entraînement du système source et tester sur les données de la langue cible.

- Utiliser le système de la langue source et tester sur les données de la langue cible traduite vers la langue source.

Les énoncés des utilisateurs sont traduits automatiquement de la langue source vers la langue cible et les classes associées à ces énoncés sont conservées. C'est la technique qui traduit les données d'entrainement qui a été retenue, car elle présente de meilleurs résultats et que la seconde technique demande d'avoir un système de traduction qui traduit les énoncés des utilisateurs en temps réel lors du déploiement de l'application.

3.1.1 Modèles de traduction

Il est possible d'utiliser différents modèles de traduction ; ceux qui ont été utilisés ici sont le modèle de mots IBM 1 et un modèle par fragment comme décrit à la section 2.1. Ces modèles ont été choisis, car ils sont représentatifs des modèles couramment utilisés en traduction automatique. L'utilisation d'un modèle IBM 1 fera uniquement une traduction mot à mot, l'ordre des mots dans la langue cible n'a pas d'importance pour cette tâche de classification, car le modèle VSM ne prend pas en compte l'ordre des mots.

La section suivante présente une méthode pour créer un corpus parallèle (matière première pour l'entraînement des modèles de traduction) proche de l'application dans le but d'améliorer la traduction.

3.1.2 Forage de corpus alignés

La qualité des traductions est dépendante de la qualité des données utilisées pour la création du modèle de traduction. Une étape de cette approche a été d'améliorer le modèle de traduction en forant des corpus alignés sur le web. Comme le corpus devait être le plus proche de l'application, le site web corporatif de l'entreprise a été choisi. Il est très probable qu'une compagnie désirant déployer un système dans une autre langue possède déjà un site web bilingue. Ces données peuvent être utilisées pour améliorer le modèle

de traduction, car le texte présent sur le site web corporatif d'une compagnie contiendra des expressions et des mots directement reliés au vocabulaire de l'application. Ces données sont directement reliées au contexte de l'application, donc très pertinente.

Un système a été développé pour extraire un corpus aligné à partir d'un site web bilingue. Il est décrit dans la section suivante.

3.1.2.1 Système d'alignement de corpus Web

Dans le but d'améliorer les modèles de traduction, des corpus alignés on été construits à partir de sites corporatifs de la compagnie dans le domaine de la tâche à traiter.

L'alignement de corpus à partir de site web a été réalisé à l'aide d'URL appariées par [5] et [24] qui ont respectivement développé les systèmes PTMiner et STRAND. Ces systèmes utilisent des URL appariées comme présentées à la figure 3.1. Par contre, deux URL appariées n'indiquent pas nécessairement que les documents sont alignés [5, 24, 27]. Ces systèmes utilisent donc la longueur des documents afin de vérifier que les documents sont bien alignés. Deux documents de longueurs très différentes ne sont probablement pas des traductions.

Figure 3.1 – Exemple d'URLs appariés

1. http ://www.example.com/page.fr
   http ://www.example.com/page.en
2. http ://www.example.com/fr/information
   http ://www.example.com/en/information

Ici, les sites à aligner sont très ciblés, car il s'agit de sites web corporatifs d'une compagnie bien précise. Le but étant d'extraire un corpus aligné avec des données très

proches de l'application à déployer. Un système plus personnalisé a été développé, spécialement pour l'extraction de certains sites web bien précis.

La Figure 3.3 montre les parties du système d'alignement. L'alignement d'un corpus web se fait en trois étapes : le téléchargement, l'alignement des documents et l'alignement du texte à l'intérieur des documents alignés.

Premièrement, le site web est *aspiré*[1], c'est-à-dire qu'il est complètement téléchargé. Ensuite, les pages web téléchargées sont analysées et les documents sont alignés entre eux. Certains sites utilisent des métadonnées incluses dans le code HTML afin de spécifier la langue de la page. Par exemple, la figure 3.2 montre le code associé à l'anglais et au français.

Figure 3.2 – Métadonnées spécifiant la langue

```
<META HTTP–EQUIV="Content–Language" CONTENT="EN">
<META HTTP–EQUIV="Content–Language" CONTENT="FR">
```

Lorsque les pages de la langue source sont alignées aux pages de la langue cible, deux types d'architecture de site web ont été observés :

1. Lorsque la langue est spécifiée dans le nom de la page
2. Lorsque la langue est spécifiée par un dossier dans l'arborescence des fichiers sur le serveur.

Ces deux types d'architecture sont bien différents et doivent être pris en compte séparément.

Page Lorsque le nom des pages correspond et que la langue est spécifiée par une requête à un script serveur (PHP, CGI, etc..). Deux pages alignées pourraient être par exemple : *CustomerClarity.page@lang=fr* et *CustomerClarity.page@lang=en*. Ici

[1] À l'aide de la commande Unix *wget*

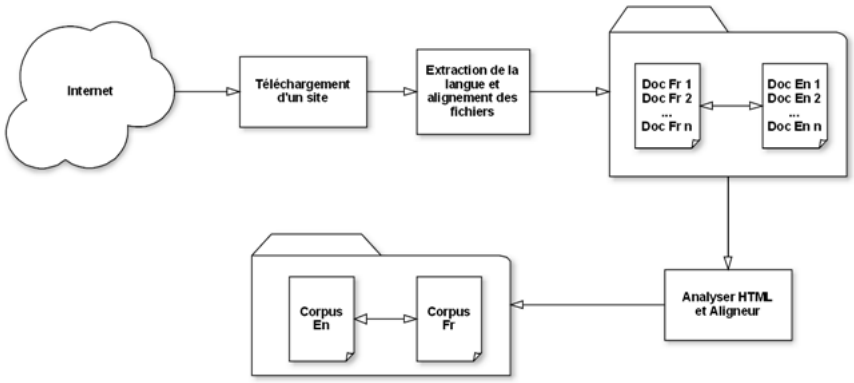


Figure 3.3 – Schéma type du système d'alignement

le nom de la page *CustomerClarity* a sa langue spécifiée par un argument passé par l'URL. Le système d'extraction de la langue reconnaît donc la langue de chacune des pages et enlève la partie du nom relative à la requête (@lang=...). L'argument relié à la requête doit être spécifié au système afin qu'il sache sur quelle partie se baser pour extraire la langue. Enfin, les fichiers sont renommés en : *CustomerClarity.page.en* et *CustomerClarity.page.fr* pour être utilisés par l'aligneur de texte.

Dossier Lorsque l'architecture du site web est telle que les documents alignés ont le même nom, mais sont dans des dossiers différents. Par exemple : *help/support.html* et *aide/support.html*. Ce type d'architecture est plus difficile à aligner, car il faut spécifier à la main les correspondances des dossiers. Il faut aussi que les noms de fichiers correspondent à l'intérieur des dossiers. Dans le cas contraire, il faut aligner les documents à la main ou utiliser un système dédié [22].

Selon l'une ou l'autre de ces architectures, les documents sont alignés un à un. Une fois les documents alignés entre eux, il faut aligner le texte à l'intérieur des documents afin d'en extraire un corpus parallèle. L'aligneur utilise la structure HTML des documents pour aligner le texte ligne par ligne. La structure HTML est alignée pour la plupart des pages[2], mais le texte ne l'est pas toujours (insertion de ligne vide, etc..). Donc

[2]En général, un site bien construit utilise un langage de programmation pour générer l'HTML

l'utilisation de la structure HTML comme guide pour l'alignement aide à l'alignement d'un plus grand nombre de données.

Figure 3.4 – Exemple d'alignement de données sur un site corporatif

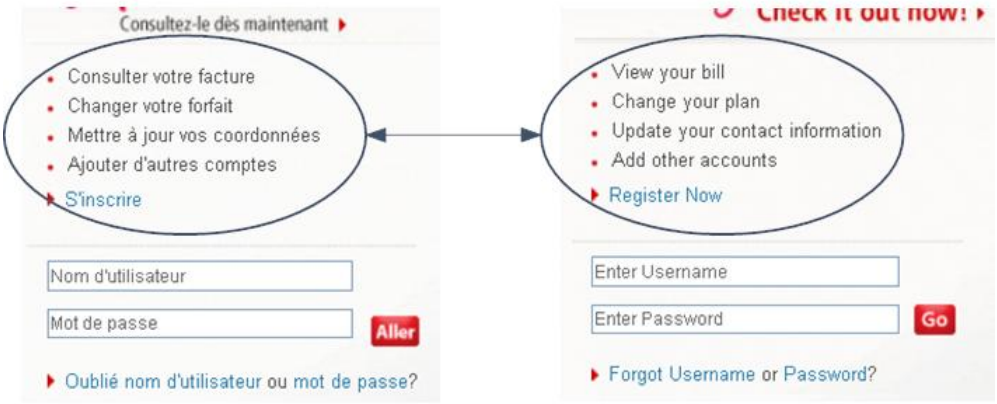


Figure 3.5 – Code HTML anglais

```
<ul>
  <li>View your bill</li>
  <li>Change your plan</li>
  <li>Update your contact information</li>
  <li>Add other accounts</li>
</ul>
```

Figure 3.6 – Code HTML français

```
<ul>
  <li>Consulter votre facture</li>
  <li>Changer votre forfait</li>
  <li>Mettre à jour vos coordonnées</li>
  <li>Ajouter d'autres comptes</li>
</ul>
```

La figure 3.4 présente un exemple d'un alignement présent sur un site web corporatif[3] réalisé à l'aide de la structure HTML et les figures 3.5 et 3.6 montrent le code HTML

[3]L'exemple provient du site web de Rogers http ://www.rogers.com

associé à cette portion de site pour la version anglaise et française. Comme les sites sont identiques dans les deux langues, le texte sera imbriqué dans les mêmes balises HTML, comme le montrent les exemples de code. On voit que les données sont imbriquées dans la même structure et dans le même ordre. Ici, on peut aligner tous les éléments <*li*> imbriqués dans la balise <*ul*>. Il suffit d'aligner les structures puis d'en extraire le texte.

Il n'y a pas que le texte visible sur les pages qui est utilisé, les métadonnées présentes sur les images sont aussi utilisées ainsi que celles définissant la page ; par exemple les mots-clés qui définissent la page. Les figures 3.7 et 3.8 montre un exemple de métadonnées présentes sur un site web bilingue. Ces données sont, entre autres, crées pour être utilisées par les engins de recherche qui indexent les pages web.

Figure 3.7 – Exemple métadonnée en français

```
<META NAME="keywords" CONTENT="Rogers communications,
sans-fil, Internet haute vitesse, téléphonie résidentielle,
télévision par câble, service local, offres groupées,
téléphones cellulaires, télévision numérique, tvhd">
<META NAME="description" CONTENT="Rogers Communications inc.
est le premier fournisseur canadien de services sans-fil,
 de télévision par câble, d'Internet haute vitesse et de
téléphonie résidentielle aux consommateurs et aux entreprises.">
```

Figure 3.8 – Exemple métadonnée en anglais

```
<META NAME="keywords" CONTENT="rogers communications, wireless,
 high speed internet, home phone, cable tv, local service,
 bundles, mobile, cell phones, digital television, hdtv">
<META NAME="description" CONTENT="Rogers Communications Inc.
is a leading provider of Wireless, Digital Cable TV,
High Speed Internet and Home Phone services to consumers
and businesses in Canada.">
```

Les données qui seront alignées de la sorte seront des expressions et des mots employés par la compagnie, donc qui ont de fortes chances de l'être par un utilisateur de

cette compagnie. Par exemple, si une compagnie possède plusieurs modèles d'item différents, ces modèles d'items se retrouveront sur son site web. Les métadonnées n'ont pas besoin d'être alignées, car leur nom est commun aux deux langues. Par exemple dans les figures 3.7 et 3.8 il s'agit de "*keywords*" et "*description*".

Cette technique permet d'extraire un corpus aligné d'un site web bilingue et est très pratique lorsqu'on veut adapter un modèle de traduction à une tâche corporative bien précise.

Ce système à toutefois des limitations : il peut uniquement extraire un corpus d'un site Web dont les pages sont écrites en HTML. Si le site web est construit en utilisant des applets Java ou Flash[4], le système d'alignement ne pourra pas aligner le texte. Cela est causé par le fait que la technique utilisée pour obtenir le texte des pages web ne peut pas extraire le texte dans ce cas.

3.1.3 Exemple

Pour donner un exemple de la traduction automatique, reprenons l'exemple donné en introduction. Soit une application de fournisseur d'électricité présentant les données d'entrainement françaises de la figure 3.9. On veut déployer la même application, mais en anglais. Il faut donc prendre chacun des énoncés : "je voudrais faire un changement d'adresse","connaître ma consommation","faire un paiement" et les traduire en anglais en conservant la même étiquette. La figure 3.10 montre la traduction des données du français vers l'anglais tout en conservant les mêmes étiquettes.

Une fois les données traduites, on les utilise comme données d'entrainement pour un nouveau classifieur statistique. Ce classifieur sera utilisé pour déployer une application en anglais sans avoir à transcrire et à étiqueter des données en anglais.

[4]http ://www.adobe.com/products/flash/

Figure 3.9 – Exemple de données d'un système de routage téléphonique

```
je voudrais faire un changement d'adresse:ChangementAdresse
connaître ma consomation:Consomation
faire un paiement:Paiement
```

Figure 3.10 – Exemple de données d'un système de routage téléphonique en anglais

```
i would like to change my address:ChangementAdresse
know my usage:Consomation
make a paiement:Paiement
```

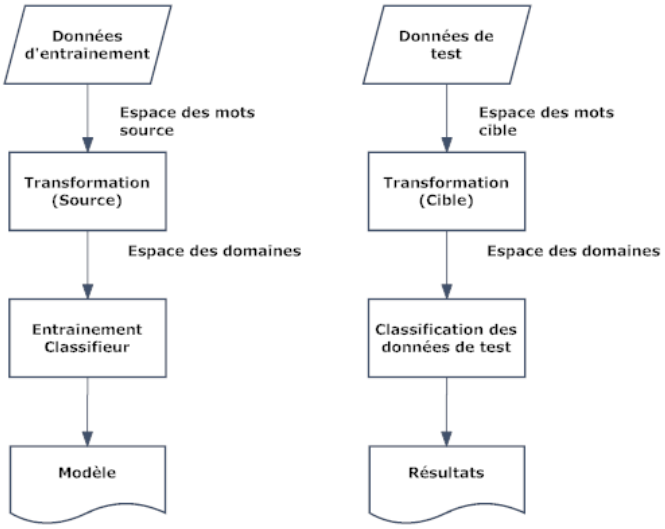
3.2 Classification multilingue

La section suivante présente la deuxième technique pour déployer un système de routage téléphonique d'une langue source à une langue cible. Cette technique utilise un espace sémantique commun pour comparer les deux langues. Elle est appelée : *classification multilingue*, car plusieurs langues sont classifiées en utilisant le même espace.

La classification multilingue utilise les techniques de projection tirées de la théorie des champs lexicaux. Tout comme décrit à la section 2.2, la classification d'énoncés utilisant ces modèles requière la création d'un espace sémantique (le Domain Space) afin d'y projeter les vecteurs à classifier. Par contre, pour ce qui est de la classification multilingue, les deux langues doivent être projetées dans le même espace sémantique afin de pouvoir effectuer la classification sur les mêmes caractéristiques. L'espace doit donc être commun, car autrement il ne serait pas possible de comparer les énoncés de deux langues différentes. Par conséquent, il faut créer un espace sémantique qui sera commun aux deux langues. La méthode utilisée par Gliozzo et al. [10] crée un DM avec les deux langues en se basant sur les mots communs aux deux langues pour inférer l'information sur les autres mots. Ici contrairement à l'approche développée par Gliozzo et al., on crée d'abord un DM dans la langue source puis on utilise une traduction mot à mot qui servira d'ancrage pour inférer le modèle dans la langue cible.

La technique de classification multilingue ajoute une étape de prétraitement avant la phase d'entrainement et de test du classifieur. Cette étape a pour but de projeter les données dans l'espace des domaines. Par la suite, le classifieur est entraîné normalement. Pour ce faire, une matrice dans chaque langue est créée. Il s'agit des matrices de transformation source D_s et cible D_c. La figure 3.11 montre un schéma bloc de l'étape introduite par la classification multilingue. Les deux transformations sont différentes, car elles vont projeter deux langues différentes dans le même espace.

Figure 3.11 – Méthode de classification multilingue



Avant d'entrainer le classifieur, il faut créer les deux matrices qui vont projeter les deux langues dans le même espace. L'inférence des domaines sémantiques multilingues peut être réalisée à l'aide de corpus comparables ; des corpus en deux langues traitant du même sujet. Premièrement, les domaines sont créés avec le corpus de la langue source en utilisant les techniques de LSA ou de ESA tels que décrits à la section 2.2.4.3. Ce sont ces domaines qui définiront l'espace des domaines qui sera utilisé pour effectuer la classification. Ceci définira une matrice D_s représentant le poids de chacun des mots de la langue source relié aux domaines. Par la suite, la partie cible du corpus comparable

est utilisée pour créer une matrice D_c qui représentera le poids de chacun des mots de la langue cible vers les mêmes domaines que la matrice D_s. Donc avec les deux matrices D_s et D_c, l'information nécessaire pour projeter les deux langues dans le même espace sémantique est disponible. Ce sont ces matrices qui seront utilisées pour faire la projection des vecteurs VSM dans l'espace des domaines communs aux deux langues. La section suivante présente les étapes de création de la matrice D_s, celles de la matrice D_c pour ensuite présenter un exemple de création de ces deux matrices.

3.2.1 Inférence de la matrice D_s

La matrice D_s est la matrice inférée à partir du corpus de la langue source, il s'agit des données étiquetées utilisées pour l'entrainement du système de routage téléphonique. Elle peut être inférée de plusieurs manières en utilisant un algorithme de *clustering*, ici on utilisera deux méthodes : l'analyse sémantique latente (Latente Semantic Analysis LSA) et l'analyse sémantique explicite (Explicit Semantic Analysis ESA).

On prend premièrement le corpus de la langue source, on doit ensuite le découper soit en document, paragraphe ou ligne, tout dépendamment du type de corpus qu'on utilise. On définit ensuite le vocabulaire du corpus, soit l'ensemble des mots différents. Ensuite, en utilisant le découpage fait préalablement, on passe à travers tout le corpus pour faire le compte des mots du vocabulaire dans chacune des sections découpées. On obtient ainsi une matrice de type VSM. L'étape suivante est l'utilisation d'un algorithme de représentation d'espace commun pour inférer les domaines. Les domaines correspondent aux étiquettes reliées aux données.

Ici, l'algorithme de LSA peut être utilisé pour trouver les sections découpées les plus pertinentes. Il faut par contre choisir un nombre de dimensions à conserver. Il n'y a pas de règle définissant le nombre de dimensions à conserver pour avoir des performances

optimales. Les expériences empiriques semblent indiquer que le nombre de dimensions doit se situer entre 50 et 400 [1].

La matrice D_s peut aussi être inférée en utilisant ESA, cette technique est facile à implémenter si un corpus étiqueté pour une autre tâche est déjà disponible. Dans ce cas, on prend les étiquettes de cette tâche comme dimensions. Si la tâche est assez proche de l'application qu'on cherche à déployer, les dimensions choisies seront pertinentes. Dans le cas contraire, il faudrait l'augmenter d'un autre corpus général afin de capturer un nombre de domaines plus élevé.

Ici, on utilisera comme espace les classes du corpus source. Les deux langues seront donc projetées dans un espace défini par les classes du corpus source. Par contre, cet espace pourrait être augmenté par les classes d'un autre corpus afin qu'il soit plus riche. C'est cet espace qui est utilisé par le classifieur pour l'entrainement et la classification. Les classes du corpus source sont donc utilisées directement comme un domaine. Étant donné qu'on utilise un corpus étiqueté, le poids d'un mot relié à une classe devient la fréquence de ce mot pour cette classe. La matrice résultante sera :

$$D_s = \begin{array}{c} \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & \dots & D_k \end{array} \\ \begin{pmatrix} tf(w_1,D_1) & tf(w_1,D_2) & \dots & tf(w_1,D_k) \\ tf(w_2,D_1) & tf(w_2,D_2) & \dots & tf(w_2,D_k) \\ \vdots & \vdots & \ddots & \vdots \\ tf(w_n,D_1) & tf(w_n,D_2) & \dots & tf(w_n,D_k) \end{pmatrix} \end{array} \tag{3.1}$$

Les mots w_1 à w_n appartiennent au vocabulaire source. Les poids de ces mots correspondent à l'algorithme ESA, car ils proviennent d'une séparation naturelle du corpus. Il est possible d'effectuer LSA sur cette matrice, si on veut trouver des domaines latents qui seraient plus discriminants. Dans les expériences qui suivront, lorsqu'on parle de l'algorithme LSA, la décomposition en valeur singulière est réalisée sur la matrice D_s

afin de trouver des dimensions plus discriminantes.

3.2.2 Inférence de la matrice D_c

Pour inférer la matrice D_c qui projettera les énoncés de la langue cible vers les domaines inférés avec la partie source du corpus, on commence en prenant chaque colonne de la matrice D_s. Le vecteur colonne associé au domaine j est :

$$d_j = [w_{1,j}, w_{2,j}, \ldots, w_{n,j}] \tag{3.2}$$

Où $w_{1,j}$ est le poids du mot 1 associé au domaine j. Il s'agit de la j^e colonne de la matrice D_s, Ceci donnera le vecteur représentant le poids de chacun des mots reliés au domaine j. Les mots sont ensuite pondérés par la matrice I^{IDF} [5], afin de lisser le poids des mots selon leur pertinence. Cette étape est importante si on ne veut pas donner trop de poids à des mots communs qui ont une fréquence élevée dans le corpus (e.g. déterminants, mots de liaison, etc.).

$$d'_j = d_j I^{IDF} \tag{3.3}$$

Le vecteur d'_i est ensuite trié par ordre décroissant du poids des mots. Les k mots dont le poids est le plus élevé sont ensuite choisis. Ce sont ces mots qui serviront *d'ancrage* pour inférer la matrice D_c. La traduction de ces mots en utilisant un modèle de traduction mot à mot est utilisée pour trouver ses cooccurrences. Cela sera donc utilisé pour trouver les mots de la langue cible qui sont pertinents pour le domaine qu'on est en train d'inférer. Étant donné que le but est d'inférer une matrice de poids pour les mots dans une langue cible, il convient d'utiliser une méthode pour relier les mots de la langue source à ceux de la langue cible. Pour ce faire, les ϕ mots qui maximisent probabilité de traduction calculée selon un modèle IBM1 sont prises pour chacun des k mots choisis

[5] Matrice diagonale où les valeurs de la diagonale représentent les poids IDF de chacun des mots. Il y a une matrice IDF par langue.

précédemment dans le vecteur d'_j. La valeur ϕ représente le nombre de traductions qu'on choisit pour chacun des k mots par domaine. On appelle ϕ la fertilité.

Chacun des k mots qu'on a choisis pour chacun des domaines a donc un ensemble de traductions qui lui est relié. L'union de tous ces ensembles : Q, est utilisé pour inférer l'appartenance des mots de la langue cible à l'espace sémantique créé avec le corpus de la langue source. Par exemple, si on choisit un $k = 3$ et $\phi = 2$, il y aura au plus 6 mots dans cet ensemble. Il est important que les corpus utilisés soient comparables afin que les cooccurrences des mots dans chacune des langues soient consistantes. Si les deux corpus ne sont pas comparables, les cooccurrences des ancrages utilisés pour inférer la matrice D_c risquent de ne pas être dans le même contexte que leur contrepartie dans la langue source. Les mots les plus importants dans chacun des domaines seront utilisés pour aller chercher d'autres cooccurrences dans le corpus cible. Donc les mots qui ne sont pas traduits par le modèle IBM 1 ou qui ne se trouvent pas dans la table de transfert vont se retrouver tout de même dans le modèle cible grâce aux cooccurrences des ancrages.

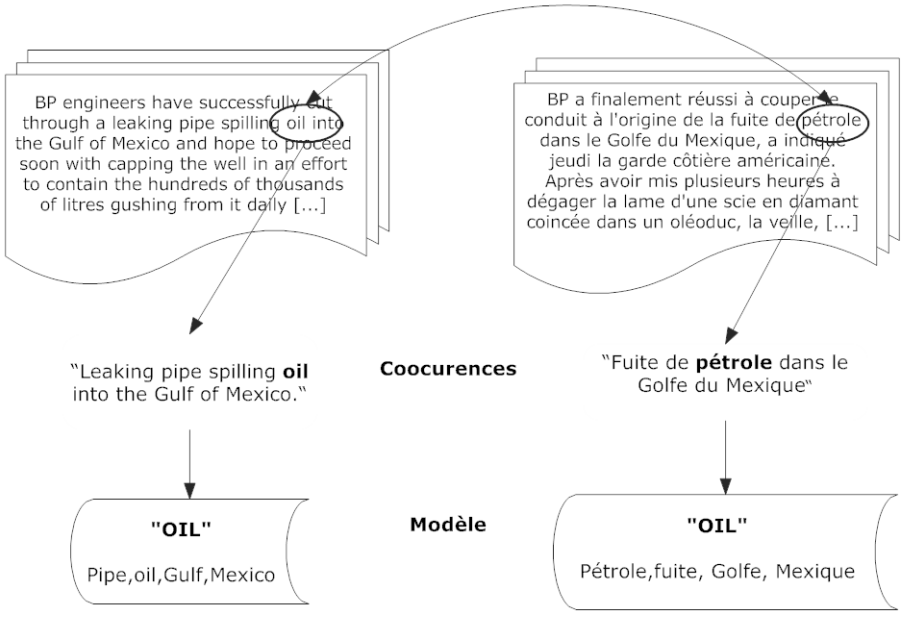
La figure 3.12 présente un exemple d'inférence d'un domaine sémantique multilingue en utilisant un corpus comparable[6]. Ce corpus est ici composé de deux articles de nouvelles décrivant le même événement[7]. Puisque que les données sont confidentielles, cet exemple est tiré d'un autre domaine.

Pour chacun des domaines, on regarde tous les énoncés du corpus cible. Ensuite, si parmi ces énoncés il y a un mot cible qui est présent dans l'ensemble de traduction Q des k mots pour le domaine en cours, on ajoute un poids à tous les mots de l'énoncé cible

[6]Les deux articles proviennent d'articles de journaux pris sur internet.
http ://www.radio-canada.ca/nouvelles/International/2010/06/03/002-maree-noire-jeudi.shtml
http ://www.cbc.ca/world/story/2010/06/03/oil-rig-shears.html

[7]La fuite de la plateforme pétrolière de BP dans le Golfe du Mexique.

Figure 3.12 – Exemple d'inférence de la matrice D_c en utilisant un corpus comparable.



dans la matrice D_c selon l'équation :

$$\eta = \alpha w_{i,j} + \beta p(w^t | w_i^s) \tag{3.4}$$

α et β sont des hyper paramètres du système qui ont pour but de venir pondérer l'apport de $w_{i,j}$ et de $p(w^t|w_i^s)$ qui sont respectivement le poids du i^e mot du domaine j de D_s et la probabilité que le mot cible soit la traduction du mot source. Une fois que cette étape est réalisée pour tous les domaines de la matrice D_s, la matrice D_c est complétée. L'algorithme 1 montre le pseudocode de la fonction d'inférence de la matrice D_c.

La matrice résultante sera :

Algorithm 1 Inférence de la matrice D_c

Require: k,ϕ,α,β
  for $i = 1$ à Nombre de domaines sources **do**
    Choisir les k mots les plus discriminants du domaine i
    $Q = \phi$ traductions des k mots
    for $j = 1$ à Nombre d'énoncés cibles **do**
      M = Mots de l'énoncé j
      if $Q \cap M \neq \emptyset$ **then**
        Ajuster le poids des mots M de D_c pour le domaine i
      end if
    end for
  end for

$$D_c = \begin{array}{c} \\ w_1 \\ w_2 \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & \dots & D_k \end{array} \\ \begin{pmatrix} P(w_1,D_1) & P(w_1,D_2) & \dots & P(w_1,D_k) \\ P(w_2,D_1) & P(w_2,D_2) & \dots & P(w_2,D_k) \\ \vdots & \vdots & \ddots & \vdots \\ P(w_m,D_1) & P(w_m,D_2) & \dots & P(w_m,D_k) \end{pmatrix} \end{array} \tag{3.5}$$

Les mots w_1 à w_m appartiennent au vocabulaire cible et $P(w_m,D_k)$ correspond au poids trouvé lors du processus d'inférence. D_1 à D_k sont les mêmes domaines que ceux utilisés par la matrice D_s.

3.2.3 Entrainement et classification

Les deux matrices D_s et D_c sont créées avant d'effectuer l'entrainement du classifieur. Lorsqu'on possède ces deux matrices, il est possible de faire l'entrainement et la classification. Pour l'entrainement, on prend chaque énoncé du corpus d'entrainement qu'on projète dans l'espace des domaines à l'aide de la matrice D_s :

$$s'_s = s_s \times (I_s^{IDF} \times D_s) \tag{3.6}$$

Ici s_s correspond à un énoncé source sous la forme VSM, donc dans l'espace des

mots de la langue source. Le résultat s'_s correspond à l'énoncé projeté dans l'espace des domaines. Lorsque la transformation a été effectuée pour tous les énoncés, on les envoie au classifieur afin qu'il soit entrainé. Pour l'étape de classfication ou de test, c'est le même processus à l'exception que la matrice qui effectue la transformation est la matrice D_c :

$$s'_c = s_c \times (I_c^{IDF} \times D_c) \tag{3.7}$$

Les deux matrices IDF sont différentes pour les deux langues.

3.2.4 Exemple

Par exemple, la matrice ci-dessous représente une partie de la matrice D_s inférée avec le corpus source de l'application de routage téléphonique.

$$D_S = \begin{array}{c} \\ compte \\ facturation \\ facture \\ adresse \end{array} \begin{array}{c} \begin{array}{ccc} d_1 & d_2 & d_3 \end{array} \\ \begin{pmatrix} 0.348 & 0.042 & 0.08 \\ 0.11 & 1 & 0.101 \\ 0.049 & 0.174 & 0.044 \\ 0.111 & 0.107 & 1 \end{pmatrix} \end{array} \tag{3.8}$$

On prend la première colonne, c'est-à-dire le poids des mots pour d_1, ce vecteur est ensuite pondéré par la matrice IDF :

$$d_1 = [0.348 \quad 0.11 \quad 0.049 \quad 0.111] \tag{3.9}$$

$$d'_1 = d_1 \times I^{IDF} \tag{3.10}$$

On trie le vecteur d'_1 et on prend les k mots les plus discriminants pour le domaine. Par exemple pour $k = 2$, les mots seraient : *compte* et *adresse*. On prend ensuite les

ϕ meilleures traductions de ces mots selon un modèle de traduction mot à mot ou un dictionnaire pour créer un ensemble de mots cibles. Cet ensemble contient les meilleures traductions des mots les plus pertinents pour le domaine qu'on est en train d'inférer. Dans cet exemple pour $\phi = 1$, Q, l'ensemble des mots cibles est :

$$Q = \{compte, address\} \tag{3.11}$$

Le but est d'inférer une matrice D_c qui projette les mots de la langue cible vers l'espace sémantique commun crée à partir du corpus source. On va ensuite analyser tous les énoncés du corpus cible, lorsqu'on trouve un mot présent dans l'ensemble de mots cibles Q, on ajoute tous les mots de l'énoncé à la matrice D_c. Donc, lorsque dans le corpus cible on trouve un mot qui correspond à une traduction d'un des mots les plus pertinents pour un domaine, on ajoute tous les mots de l'énoncé à la matrice D_c. Ceci a pour but de capturer tous les mots du corpus cible dans la matrice D_c.

La matrice 3.12 présente les résultats de la matrice D_c.

$$D_c = \begin{array}{c} \\ account \\ billing \\ bill \\ address \\ information \end{array} \begin{array}{c} \begin{array}{ccc} d_1 & d_2 & d_3 \end{array} \\ \begin{pmatrix} 0.234 & 0.047 & 0.096 \\ 0.0002 & 1 & 0.0115 \\ 0.0006 & 0.043 & 0.012 \\ 0.008 & 0.054 & 1 \\ 0.052 & 0.18 & 0.15 \end{pmatrix} \end{array} \tag{3.12}$$

3.2.5 Paramètres et modèles

Le modèle de classification sémantique multilingue présente plusieurs hyper paramètres à ajuster lors de la création du modèle :

Nombre de mots : Le nombre de mots utilisés dans chaque domaine pour inférer le modèle sémantique dans langue cible : paramètre k

Fertilité : Le nombre de mot générés par le modèle de traduction : paramètre ϕ

Taille du modèle sémantique : Le nombre de domaines utilisés pour créer l'espace sémantique.

Le choix des corpus comparables joue aussi un rôle important, il doit être relié à la tâche à réaliser et aussi être assez riche pour couvrir tout le vocabulaire de l'application.

3.2.6 Domaines multilingues

Jusqu'à présent le DM n'a été construit qu'à partir du corpus source, ces mêmes domaines ont ensuite été inférés dans la langue cible. Il est possible d'augmenter l'espace en utilisant le même processus, mais dans l'autre sens ; c'est-à-dire créer les domaines avec le corpus de la langue cible et inférer la matrice dans la langue source. Ceci aura pour effet d'augmenter le nombre de domaines, donc le nombre de dimensions de l'espace dans lequel projeter les énoncés pour l'entraînement et la classification. Cela va aussi utiliser l'information contenue dans les deux langues du corpus comparable donc maximiser l'utilisation de l'information disponible. Il serait aussi possible d'augmenter cette technique avec l'utilisation de plusieurs paires de langues. Cela n'a pas été fait, mais serait une amélioration à envisager.

3.3 Comparaison des approches

Jusqu'à présent les deux approches décrites pour réaliser le système de déploiement automatique ont été présentées en parallèle, la prochaine section compare les forces et les faiblesses de chacune des deux approches tant théorique que pratique.

Tableau 3.1 – Comparaison des approches

	Traduction automatique	Classification multilingue
Entraînement	Nécessite un corpus parallèle général et du domaine	Nécessite un petit corpus parallèle (ou un dictionnaire) ainsi qu'un corpus comparable dans le domaine.
Classifieur	La traduction automatique requière l'entraînement d'un classifieur par langue	La méthode multilingue ne requière l'entraînement que d'un classifier pour toutes les langues.
Performance	Les performances ne dépendent que de la qualité de la traduction.	Les performances dépendent du corpus comparable, de l'espace sémantique et de la projection des deux langues dans cet espace.

Le point le plus avantageux de l'approche classification multilingue est qu'elle ne requière l'entrainement que d'un classifier pour toutes les langues. Il ne faut donc pas entrainer un nouveau classifieur à chaque nouveau déploiement. C'est un avantage pratique qui peut s'avérer fort utile, car l'ajout d'une nouvelle langue à un système devient plus facile. Par contre, elle nécésite aussi la création des matrices D_c et D_s avec l'aide d'un corpus comparable.

Quant à la technique de traduction automatique, sont point le plus avantageux est sa simplicité. Une fois le modèle de traduction créé, il ne s'agit que de faire les traductions du corpus d'entrainement. Pour simplifier encore cela, la création d'une base de données de corpus parallèles ayant des données forées dans plusieurs domaines pourrait être créée. Avec cette base de données, il s'agirait de créer un modèle de traduction dans la langue et dans le domaine de l'application à déployer sur demande.

CHAPITRE 4

DONNÉES

> It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.
>
> *-Sir Arthur Conan Doyle*

Le chapitre suivant décrit les données utilisées et présente une description des corpus d'entraînement et de test qui sont utilisés dans les expériences décrites à la section 5. Ces corpus sont utilisés pour entrainer le classifieur qui assigne une étiquette à l'énoncé d'un utilisateur. Ils sont aussi utilisés pour entrainer les modèles de traduction ainsi que les modèles sémantiques utilisés dans la classification multilingue.

4.1 Corpus utilisé

Il y a trois types de corpus qui sont utilisés par les deux approches de déploiement automatique d'un système de routage téléphonique : *les corpus unilingues*, *les corpus bilingues alignés* et *les corpus bilingues comparables*. Premièrement, les types de corpus sont définis pour ensuite être décrit en détail selon leur utilisation.

Les corpus unilingues sont utilisés pour entraîner les systèmes de classification statistique. Ce sont des données provenant de transcriptions de fichiers audio de conversations téléphoniques. Des personnes appellent à une centrale téléphonique et sont dirigées selon leurs questions ou leurs problèmes vers la personne qui va pouvoir les aider. Pendant la collecte des données, les personnes sont dirigées manuellement par un humain. Une fois les données récoltées, elles sont transcrites et étiquetées. L'étiquette de chaque donnée est reliée à une *classe* du système.

Les corpus parallèles sont utilisés pour entraîner les systèmes de traduction automatique statistique. Ce sont des corpus comprenant une collection de documents dans deux langues. Chacun des documents dans chacune des langues est en relation de traduction ligne par ligne. C'est-à-dire que pour deux documents alignés, la première ligne dans un document est la traduction de la première dans le second et ainsi de suite.

Les corpus comparables sont utilisés pour l'inférence des modèles sémantiques. Ce sont des collections de corpus dans deux langues, mais pas aligné comme les corpus parallèles. Les collections de textes sont uniquement sur un même sujet et décrivent les mêmes concepts.

4.1.1 Corpus unilingue

Le tableau 4.1 présente les caractéristiques des données des corpus unilingues utilisées pour entrainer les systèmes de routage téléphonique chez Nuance. Ces données proviennent d'une application réelle présentement en utilisation. Il s'agit d'un centre d'appel qui comporte deux systèmes. Un premier système de classification en anglais et un second en français. Ce sont ces données qui ont été utilisées pour toutes les expériences réalisées. Il s'agit de la même application, mais déployée dans deux langues différentes.

Tableau 4.1 – Corpus d'entrainement pour le système de routage téléphonique

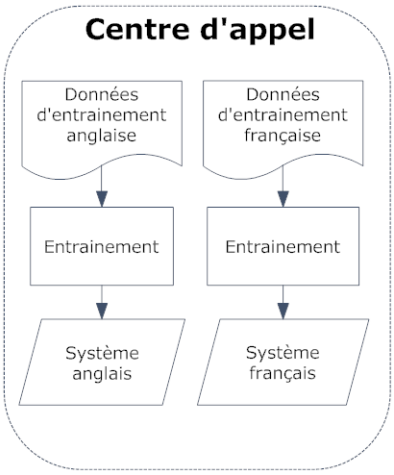
Nom	Énoncés	Vocabulaire (Mots)	Mot total	Classes
Entrainement Anglais	92 657	6 000	503 792	1 414
Entrainement Francais	91 290	9 362	844 993	1 448
Entrainement Anglais	40 099	4 163	224 939	1 143
Entrainement Francais	40 171	6 326	331 757	1 340
Test anglais	1915	1149	14 564	379

Les deux premières lignes du tableau présentent les données totales disponibles, les

deux secondes lignes sont les données qui ont été utilisées pour faire les expériences. La dernière ligne du tableau présente les caractéristiques des données de test utilisées pour réaliser les expériences. Le jeu de test provient de la même distribution, mais n'est pas compris dans les données d'entrainement.

La figure 4.1 montre un schéma du centre d'appel dont les données ont été utilisées. Les deux systèmes à l'intérieur du même centre d'appel ont été déployés de manière

Figure 4.1 – Centre d'appel bilingue



indépendante en ayant en commun uniquement le nom des classes.

4.1.1.1 Particularités

Les données sont des transcriptions d'énoncés prononcés oralement par un utilisateur. Le niveau de langage est donc très différent de celui rencontré dans un document écrit. La figure 4.2 montre un exemple de requêtes qui pourraient être prononcées par des utilisateurs[1].

Comme on peut le voir dans ces énoncés, tout ce qui est prononcé par l'utilisateur est transcrit : les hésitations, les répétitions, les mots coupés, etc. Ces énoncés sont donc très différents d'un texte écrit dans lequel la syntaxe est plus stricte. Parfois les énoncés

[1] Il s'agit de transcriptions fictives.

Figure 4.2 – Exemple d'énoncés d'utilisateurs d'une application de routage téléphonique

1. *c'est quoi le solde de mon compte*
2. *j'aurais euh besoin de euh besoin d'information*
3. *j'voudrais avoir les infor- information sur mon compte*
4. *mon compte*
5. *information*

comportent seulement un mot, le modèle VSM est donc très dispersé. Il est aussi à noter qu'il n'y a aucune ponctuation : seulement les mots reconnus par la reconnaissance vocale sont présents dans les énoncés.

4.1.2 Corpus parallèle

Un corpus parallèle est un corpus bilingue dans lequel un des textes des deux langues est une traduction du second. C'est-à-dire qu'un des textes est écrit dans sa langue d'origine tandis que le second est une traduction de ce texte dans une autre langue. C'est ce type de corpus qui est utilisé pour entrainer les modèles de traduction. Le tableau ci-dessous décrit les différents corpus d'entraînement utilisés pour entraîner les modèles de traductions. Le corpus Europarl [14] provient des débats parlementaires européens, Hansard[2] provient des débats parlementaires canadiens. Le corpus Web est un corpus foré sur le web en utilisant un aligneur spécialement conçu pour extraire des informations bilingues alignées de sites web (section 3.1.2.1). Toute l'information obtenue est disponible publiquement sur le site. Les caractéristiques des différents corpus parallèles utilisés pour effectuer les traductions des données sont présentées dans le tableau 4.2.

[2]Il s'agit de la version 2001-1a disponible à http ://www.isi.edu/natural-language/download/hansard/

Tableau 4.2 – Corpus d'entraînement pour les modèles de traduction

Nom	Paires Phrases	Vf (mots)		Ve (mots)	
		Diff	Total	Diff	Total
Europarl	1 288 074	127 841	36 742 417	108 357	32 916 947
Hansard	947 969	92 752	16 627 518	79 345	14 633 980
Web Corpus	30 513	7 754	140 552	6 825	113 482

4.1.3 Corpus comparable

Un corpus comparable est un ensemble de documents dans une ou plusieurs langues décrivant les mêmes sujets [7]. Le corpus n'est pas aligné comme le sont les corpus utilisés pour la création d'un modèle de traduction. Dans des corpus comparables, ce sont les sujets des textes qui sont alignés. Un exemple d'un corpus comparable serait deux articles de journaux écrits par deux journalistes dans deux langues différentes, mais décrivant le même événement.

L'avantage de ce type de corpus est qu'il est facile à obtenir. Par exemple, Wikipédia[3] renferme plusieurs articles sur un même sujet, mais dans plusieurs langues différentes. Les fils de nouvelles sont aussi un bon exemple, chaque jour des nouvelles sont écrites dans plusieurs langues décrivant les mêmes événements.

Les corpus comparables utilisés ici seront des tâches semblables à celles dont les données sont utilisées pour faire les expériences. Il s'agit de données d'entrainement d'autres systèmes de routage téléphonique. Ces données sont étiquetées, mais n'ont aucun lien avec l'application qu'on tente de déployer d'une langue source vers une langue cible. Les caractéristiques de ces données sont présentées dans le tableau 4.3. Les deux jeux de données sont nommés : *Tâche 1* et *Tâche 2*, car ils ne peuvent pas être identifiés. Le nombre de classes de ces deux applications est beaucoup moins élevé que dans l'application qu'on tente de déployer d'une langue source vers une langue cible, soit environ

[3] www.wikipedia.org

10 fois moins.

Tableau 4.3 – Corpus comparable pour l'inférence des domaines multilingues

Nom	Énoncés	Vocabulaire (Mots)	Mot total	Nombre de classe
Tâche 1	90 061	5 732	773 244	126
Tâche 2	106 000	4 766	467 317	77

Les corpus parallèles peuvent aussi être utilisés comme corpus comparable pour inférer les DMs.

CHAPITRE 5

RÉSULTATS

> If knowledge can create problems, it is not through ignorance that we can solve them.
>
> *-Isaac Asimov*

Dans le but de bien comprendre les performances de chacune des deux méthodes ainsi que leurs particularités, la section suivante présente les résultats obtenus pour chacune d'entre elles. Le chapitre débutera par l'explication des métriques d'évaluation, présentera ensuite les résultats de la méthode par traduction automatique pour poursuivre avec la classification multilingue. Le chapitre sera conclu avec une comparaison des résultats des deux approches.

5.1 Métriques d'évaluation

Voici la définition des métriques utilisées pour comparer les performances des différentes approches. La courbe ROC et la mesure de la précision sont utilisées pour mesurer les performances des techniques réalisées.

5.1.1 Courbe ROC

Dans une application de routage téléphonique, il n'est pas toujours pratique d'utiliser le système dans l'état où il donnera le meilleur taux de classification. Plus d'utilisateurs seront bien entendu dirigés vers la bonne destination, mais il y en aura aussi plus qui seront dirigées vers la mauvaise, car on accepte toujours la réponse du classifieur. Dans ce genre de système, il est important de minimiser les mauvaises classifications, car on veut minimiser les utilisateurs qui sont envoyés au mauvais endroit. Une technique pour réduire le nombre de mauvaises classifications est de faire varier un seuil sur la sortie du classifieur. Cette sortie sera alors vue comme une mesure de *confiance* en la décision : si

la confiance descend en deçà d'un certain seuil, on rejette la donnée puis on redemande à l'utilisateur de prononcer sa requête. Dans ce cas, on dit qu'on rejette la requête de l'utilisateur.

La courbe ROC (Receiver Operating Characteristic) permet de visualiser cet effet. On fait varier un seuil sur la sortie du classifieur, plus le seuil est faible, plus on accepte de candidats avec une confiance plus faible. Au début seulement les candidats pour lesquels le classifieur à une confiance élevée sont acceptés, à mesure que le seuil diminue, on accepte plus de candidats.

Les courbes ainsi que les résultats présentés seront les vrais positifs (Correct Accepted CA) par rapport aux faux positifs (False Accepted FA[1]). On parle de CA lorsque le classifieur classifie correctement un énoncé et de FA lorsque le classifieur donne une mauvaise étiquette. Le tableau 5.1, appelé une matrice de confusion, illustre les types d'erreurs de décision[2]. Ce tableau représente un exemple de classification à deux classes (C1 et C2) ; la référence se trouve à l'horizontale et la sortie du classifieur à la verticale. Lorsque la classe C1 est demandée et que le classifieur répond C1, il s'agit d'une bonne décision donc d'un CA. Par contre, lorsque la référence est C1 et que le classifieur répond C2, il s'agit d'un faux positif ou FA, car le classifieur a pris une décision qui est erronée. *Mutatis mutandis* lorsque la référence est C2.

Tableau 5.1 – Exemple de resultats de classification

		Référence	
		C1	**C2**
Résultat	**C1**	CA	FA
	C2	FA	CA

[1] Parfois : *False Positive* ou erreur de type I.

[2] Comme on a un système de classification multi-classes on ne parle pas de faux négatif comme dans la classification binaire.

5.1.2 Précision

Une autre mesure de classification qui sera utilisée est la précision[3] calculée comme suit :

$$\frac{CA}{CA+FA} \tag{5.1}$$

La précision est en fait le taux de bonne classification ; elle sera utilisée dans les expériences de classification multilingue.

5.2 Tâche et système de référence

Le système de routage téléphonique de référence utilisé pour comparer les expériences est le système anglais, car les expériences vont du français vers l'anglais. Donc, à l'exception des systèmes de référence ou lorsque mentionné, toutes les expériences réalisées utilisent les données du système français pour déployer une application en anglais. Les systèmes de référence sont entrainés avec les données des utilisateurs réels dans chacune de leur langue respective. Le tableau 5.2 montre les performances des systèmes de référence anglais et français. Les résultats du système de référence français sont présentés à titre d'information, car ce sont ces données qui sont utilisées pour effectuer les expériences du français vers l'anglais. Ces deux systèmes ont un jeu de test différent qui est dans leurs langues, ce qui explique les différences de performances. Les données d'entrainement du système français sont probablement plus représentatives des données de test, c'est pourquoi il présente un meilleur taux de bonne classification. Ce sont les performances du système développé par Nuance. Une des particularités des systèmes de Nuance est qu'une étiquette contient tous les énoncés qui sont *rejetés* ; c'est-à-dire qu'il ne contiennent pas une sémantique pertinente pour l'application. Lorsque les résultats CA/FA sont présentés, ils ne somment pas à 100% car le pourcentage qui est correctement relié à l'étiquette *rejeté* n'est pas considérée. Le taux d'énoncés correcte-

[3] En anglais on parle d'*Accuracy* ce qui est différent de *Presicion*.

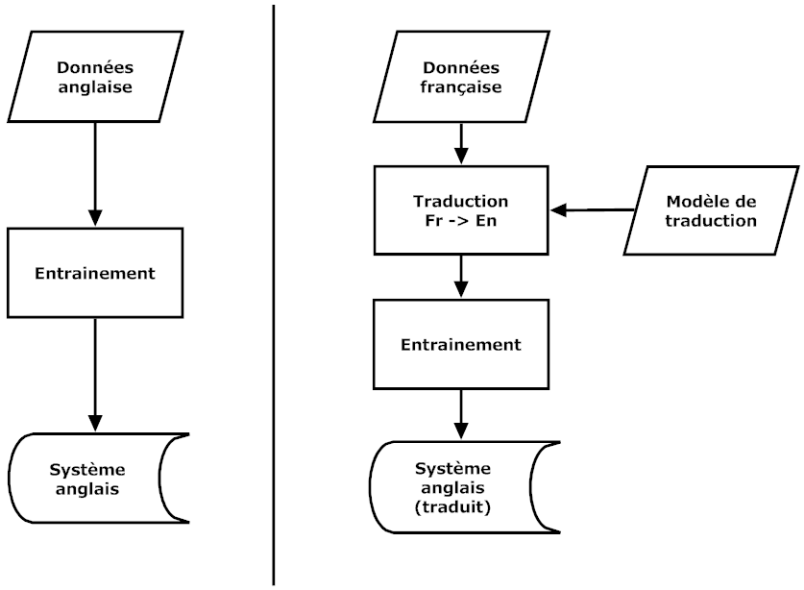
ment rejetés est la différence : $100 - (CA + FA)$. Les résultats du tableau 5.2 signifient que, par exemple, le système anglais présente un taux de bonne classification de 73.79% et un taux de mauvaise classification de 23.39%.

Tableau 5.2 – Performances des systèmes de référence anglais et français

	CA (%)	FA (%)
Système de référence anglais	73.79	23.39
Système de référence français	79.41	5.58

La figure 5.1 montre une forme schématisée des expériences réalisées en utilisant l'approche par traduction. À gauche du schéma on voit le système de référence et à droite le système traduit. Les différences entre les deux systèmes sont les données d'entrées qui ne sont pas dans la même langue ainsi que l'étape de traduction automatique.

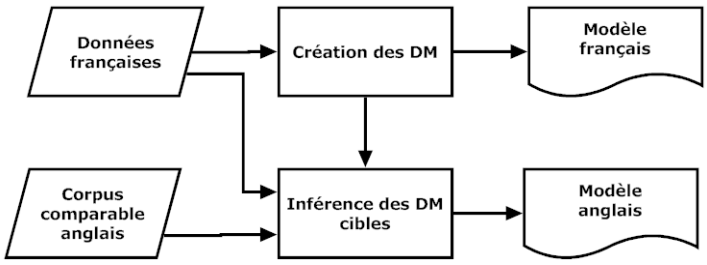
Figure 5.1 – Description schématisée des expériences de traduction



Quant aux résultats de classification multilingue, la figure 5.2 montre la configuration des expériences qui seront présentées dans cette section. Dans les expériences de cette approche, les parties qui vont varier sont la création des DM et l'inférence des DM cibles. Pour l'inférence des DM, il s'agit des deux algorithmes de *clustering* : LSA et

ESA ; pour l'inférence des DM cibles le nombre de mots de chaque domaine et la fertilité. Les deux modèles ainsi créés sont utilisés pour projeter les deux langues dans le même espace de classification.

Figure 5.2 – Description schématisée des expériences de classification multilingue



5.3 Traduction automatique

Pour effectuer les traductions automatiques le logiciel *Moses* [12] a été utilisé afin de créer le modèle de traduction par fragment et de construire la *phrase-table*. Le script d'entrainement fournit par *Moses* utilise le logiciel *GIZA++* [19] pour créer l'alignement de mots et SRILM [29] pour créer le modèle de langue. Ces logiciels sont disponibles sous license libre. Moses est également utilisé pour effectuer le décodage et fournir la traduction d'un énoncé source. Les modèles entrainés avec Moses ont été ajustés en utilisant le *Minimum Error Rate Training* (MERT) [18]. Cette technique est utilisée pour ajuster le poids des caractéristiques utilisées par le modèle de traduction.

Les résultats des expériences présentées dans cette section utilisent le système de routage téléphonique développé par Nuance. Les données d'entrainement sont traduites puis le système est entrainé avec ces données. Les résultats sont présentés sous forme de courbe ROC ainsi que les valeurs de CA/FA lorsque le système accepte toutes les décisions prises par le classifieur. Les expériences présentent d'abord les méthodes utilisées pour améliorer la traduction telle que décrite à la section 3.1. Ensuite, une analyse des résultats est présentée.

5.3.1 Effet de différents modèles de traduction

Parmi tous les types de modèle de traduction, deux modèles de traduction différents ont été expérimentés, le modèle par fragment et un modèle par mot utilisant uniquement les probabilités de traduction provenant d'un alignement IBM1. Le même corpus parallèle a été utilisé pour entrainer les deux modèles. Par contre, le modèle par mot n'utilise pas de décodage comme le modèle par fragment. Ce dernier prend uniquement le mot cible maximisant $p(e|f)$ pour chaque mot source.

Les performances du système traduit avec le modèle par fragment sont nettement supérieures à celles de celui traduit avec le modèle mot à mot. Les modèles par fragments offrent de meilleures traductions [21], ce qui a une répercussion directe sur les performances d'un système déployé avec ce type de traduction. La figure 5.3 et le tableau 5.3 montrent les résultats de cette expérience.

Figure 5.3 – Performance du système de routage téléphonique entrainé avec deux modèles de traduction différents

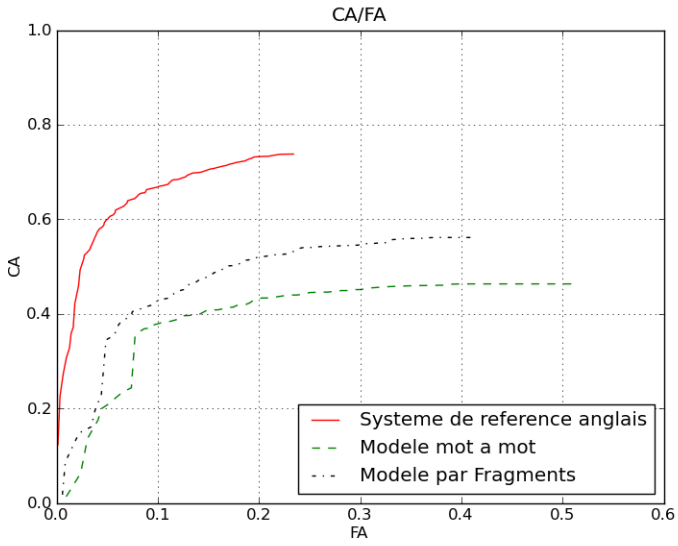


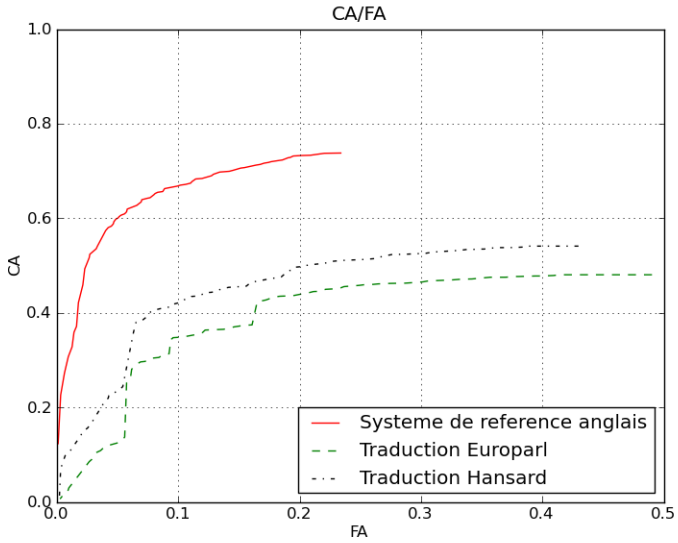
Tableau 5.3 – Tableau des résultats du système de routage téléphonique entrainé avec deux modèles de traduction différents

	CA (%)	FA (%)
Système de référence anglais	*73.79*	*23.39*
Traduction IBM1	50.86	46.32
Traduction Fragment	**56.14**	**40.89**

5.3.2 Effets des corpus parallèles utilisés

L'utilisation de différents corpus pour entraîner le modèle de traduction a des répercussions sur les performances du système. Plus les données utilisées pour entrainer ce modèle sont proches de l'application qu'on tente de déployer, plus les performances de l'application seront élevées. Choisir un corpus parallèle proche de l'application est donc important.

Figure 5.4 – Comparaison du modèle de traduction entrainé avec deux corpus parallèles différents



La figure 5.4 montre les performances du système traduit avec le modèle de traduction entraîné sur Europarl et le Hansard canadien. Les performances du système traduit avec le Hansard canadien sont nettement supérieures à celles du système entraîné avec

Tableau 5.4 – Résultats de la comparaison du modèle de traduction par fragment entrainé avec deux corpus parallèles différents

	CA (%)	FA (%)
Système de référence anglais	*73.79*	*23.39*
Traduction Hansard	**54.10**	**42.98**
Traduction Europarl	48.04	49.03

Europarl. Les deux corpus parallèles proviennent du même domaine, soit des débats parlementaires. La différence est donc causée par le fait que les données utilisées pour entraîner le système sont des données provenant de l'anglais et du français canadien, le corpus Hansard est plus approprié à cette tâche, car il plus proche de la langue de la tâche. Les différences entre les dialectes français/anglais canadiens et leur contrepartie européenne se reflète dans les performances du système. Comme les mots et les expressions utilisés sont différents, la précision des traductions s'en voit affectée. Ce qui a des répercussions sur les performances de la classification. Une expérience a été réalisée en utilisant les deux corpus ensemble comme données d'entrainement du modèle de traduction, mais aucun résultat concluant n'est sorti de cette expérience.

5.3.3 Ajout du corpus foré sur le web

Afin d'adapter le modèle de traduction à l'application, un corpus parallèle directement lié à la tâche a été foré sur le web. L'ajout de ce corpus améliore les performances du système. Comme ce corpus provient du site corporatif de la compagnie, les données sont très reliées à la tâche et comporte des mots et des expressions que les utilisateurs ont de fortes chances d'utiliser. Les données forées du web ont été ajoutées au corpus Hansard afin d'avoir une base générale pour pouvoir traduire un ensemble de mots plus grand. La figure 5.5 et le tableau 5.5 présentent les résultats de cet ajout par rapport aux corpus Hansard et Europarl.

Le corpus foré a été copié plusieurs fois afin de biaiser les traductions vers ce corpus ; ceci, afin d'ajuster les traductions vers le corpus web qui est dans le domaine de

Figure 5.5 – Performance du modèle de traduction par fragment avec l'ajout du corpus web

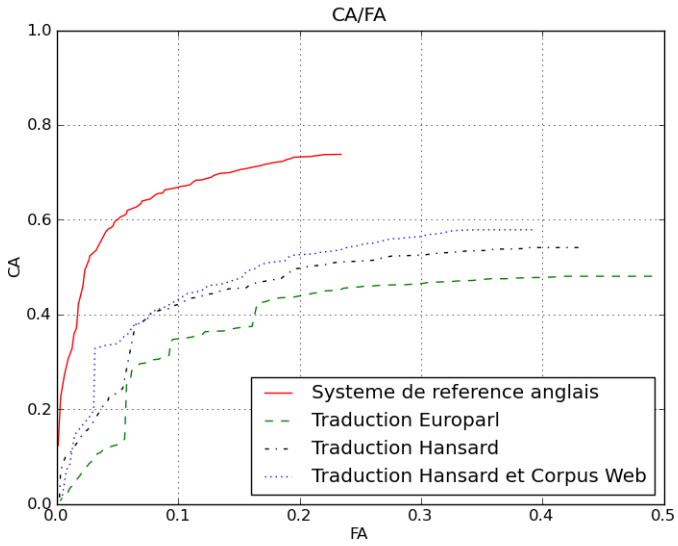


Tableau 5.5 – Résultats des performanges du modèle de traduction par fragment avec l'ajout du corpus web

	CA (%)	FA (%)
Système de référence anglais	*73.79*	*23.39*
Traduction Hansard + corpus web	**57.86**	**39.16**
Traduction Hansard	54.10	42.98
Traduction Europarl	48.04	49.03

l'application. Le tableau 5.6 présente l'effet sur les performances du système par rapport au nombre de (N) fois où le corpus web est dupliqué.

Tableau 5.6 – Performances du modèle de traduction par fragment avec l'ajout du corpus web *N* fois

N	CA (%)	FA (%)
2	56.14	40.89
3	56.45	40.68
4	56.19	40.78
8	56.08	40.94
33	**57.86**	**39.16**
66	56.87	40.21

Les performances optimales sont obtenues lorsque la taille du corpus web est égale à celle du Hansard, c'est-à-dire lorsque le corpus est copié 33 fois. On remarque une baisse des performances lorsqu'on copie le corpus au-delà de la taille du Hansard. Cela montre que le Hansard aide à traduire des énoncés plus généraux et qu'il ne faut pas trop biaiser le modèle en faveur des données forées du web. Le modèle de traduction entrainé uniquement avec le corpus web performe moins bien que celui entrainé avec uniquement le Hansard, car il possède trop peu de mots pour pouvoir traduire efficacement les données d'entrainement.

5.3.4 Analyse de la performance

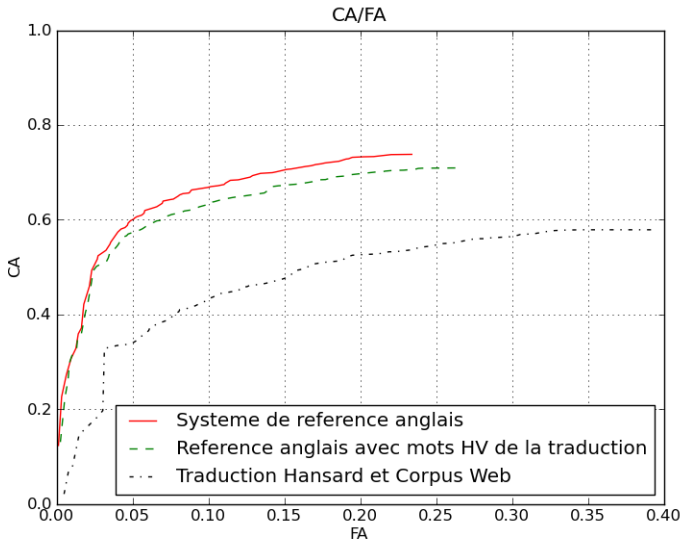
Le meilleur système traduit présente un grand écart avec le système de référence anglais. Les erreurs produites par ce système sont dues en partie à la traduction, comme aucune référence n'est disponible pour mesurer la qualité des traductions, il faut mesurer la perte de performance du système de routage téléphonique. Une méthode pour effectuer cela est de vérifier si tous les mots *hors vocabulaire* (HV) dans le jeu de test qui n'ont pas été générés par la traduction sont discriminants pour le système de classification. C'est-à-dire que tous les mots importants se retrouvent dans le corpus traduit. On entend par mot hors vocabulaire, un mot qui se trouve dans le jeu de test, mais qui ne se

retrouve pas dans le corpus d'entrainement.

Afin d'effectuer cette expérience, un nouveau corpus d'entrainement a été créé. Ce corpus est créé à partir du corpus d'entrainement du système de référence anglais, mais ne contient que les mots présents dans la traduction automatique. C'est-à-dire que tous les mots qui sont dans le corpus d'entrainement anglais et qui ne se retrouvent pas dans le corpus d'entrainement de la traduction du français vers l'anglais ont été retirés. En retirant ces mots, on mesure la perte de performance reliée aux mots hors vocabulaire.

La figure 5.6 montre les résultats de l'effet des mots hors vocabulaire. Dans cette expérience les données utilisées pour entrainer le système de référence ne contiennent que les mots communs entre les données d'entrainement anglaises et la traduction automatique. La dégradation est donc due aux mots hors vocabulaire qui ne se retrouvent pas dans le corpus généré par la traduction automatique.

Figure 5.6 – Effet des mots hors vocabulaire sur le système de référence



Cette dégradation est faible comparée à la différence entre le meilleur système de

traduction et le système de référence anglais. Les erreurs ne sont pas uniquement dues à la traduction. On parle d'une dégradation d'environ 3.5% absolus, en comparaison avec plus de 16% pour la différence entre le système de référence anglais et la meilleure traduction automatique. Une hypothèse est qu'il existe une inconsistance entre les deux applications (française et anglaise). La section suivante présente les résultats de l'analyse de cette inconsistance dans l'étiquetage des données.

Le tableau 5.7 montre les mots hors vocabulaire les plus fréquents, ainsi que leur rang dans le corpus d'entrainement anglais. À l'exception des mots *center*, *inquiries* et *inquiry* les autres mots sont des contractions. Il est donc légitime de penser que les contractions pourraient être une cause de la perte de performance. Considérant que les données proviennent de la langue parlée, les contractions sont beaucoup plus présentes que dans un texte écrit.

Tableau 5.7 – Liste des mots hors vocabulaire présent dans la traduction automatique

Mot	Count	Rang HV	Rang corpus anglais
i'm	3802	1	27
i'd	2572	2	39
it's	2352	3	42
center	1382	4	73
don't	1222	5	86
can't	1170	6	87
there's	1007	7	99
doesn't	987	8	101
inquiries	944	9	104
inquiry	936	10	107
i've	737	11	121
what's	558	12	138

L'expérience suivante montre l'effet des contractions sur les performances du système de référence anglais. Afin de vérifier cet effet, les contractions ont été enlevées du corpus d'entrainement. La figure 5.7 et le tableau 5.8 montre les résultats du système de référence anglais et de ce système entrainé avec les contractions enlevées du corpus. Les contractions ont peu d'impact sur les performances du système de routage télépho-

nique, il y a moins de 1% absolu de dégradation des performances. Donc sur les 3.5% de dégradation due aux mots HV, 1% est dû aux contractions. Par contre, cela est difficile à gérer, car il faut que ces contractions soient présentes dans le corpus d'entrainement du modèle de traduction. Une expérience a tenté de remplacer les contractions par leur forme longue dans les données de test, mais cela n'a pas apporté de gain.

Figure 5.7 – Effet des contractions présentes dans le corpus d'entrainement sur les performances système de référence

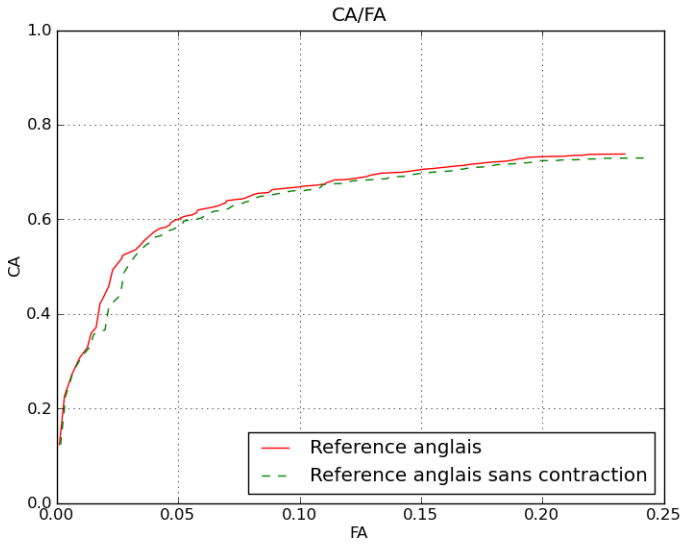


Tableau 5.8 – Résultats de l'effet des contractions dans le corpus d'entrainement sur le système de référence

	FA (%)	CA (%)
Système de référence anglais	73.79	23.39
Système de référence anglais sans contraction	72.95	24.18

Le tableau 5.9 montre le pourcentage de mots hors vocabulaire dans le jeu de test en comparaison aux données d'entrainement du système de référence et à celui traduit. Bien que les données d'entrainement traduites présentent environ 15% de mot hors vocabulaire dans le jeu de test, ces mots sont peu discriminants, car lorsqu'ils sont enlevés des données du système de référence, les pertes de performances sont d'environ 4%.

Tableau 5.9 – Proportion des mots hors vocabulaire du corpus du système de référence et de la traduction automatique comparée sur le jeu de test

Corpus	Mots HV différent (%)	Mots HV totals (%)
Entrainement anglais	3.2	1.82
Meilleur système traduit	14.97	4.19

5.3.4.1 Mots non traduits

En plus des mots hors vocabulaire, il y a les mots qui n'ont pas été traduits du français vers l'anglais. Ces mots n'étaient pas présents dans le modèle de traduction, le système de traduction ne pouvait donc pas fournir de traduction pour ces mots. Le tableau 5.10 montre les mots non traduits les plus fréquents[4], le nombre de fois qu'apparaissent ces mots dans le corpus français ainsi que leur rang comparativement aux autres mots.

Dans ce tableau, le mot non traduit le plus fréquent est *y'*[5] une contraction de "*Il y*". *Canceller* et *cancellation* sont des emprunts lexicaux à l'anglais du mot *cancel* auquel la flexion de la syntaxe française a été ajoutée. Les mots masqués sont des mots particuliers aux données de la compagnie et qui n'ont pas de traduction, les mots sont les mêmes en français et en anglais. Ces mots sont peu discriminants, car l'expérience précédente a montré que les mots hors vocabulaire avec le corpus traduit avaient peu d'impact sur les performances du système.

5.3.5 Inconsistance des données

Les performances sont toujours en deçà du système de référence d'environ 15% absolu. On pourrait croire que cela est dû au système de traduction, mais il existe aussi une inconsistance dans l'étiquetage des données. Ceci est une propriété inhérente des données. Bien que ces deux applications réalisent la même tâche, elles ont été déployées à des moments différents et étiquetées par des personnes différentes. Le nom des classes

[4] Certains mots ont été retirés, car ils donnaient trop d'information sur les données.

[5] *y'a* devrait aussi être compté, mais le mot a mal été découpé.

Tableau 5.10 – Mots non traduits dans le corpus français par le modèle de traduction

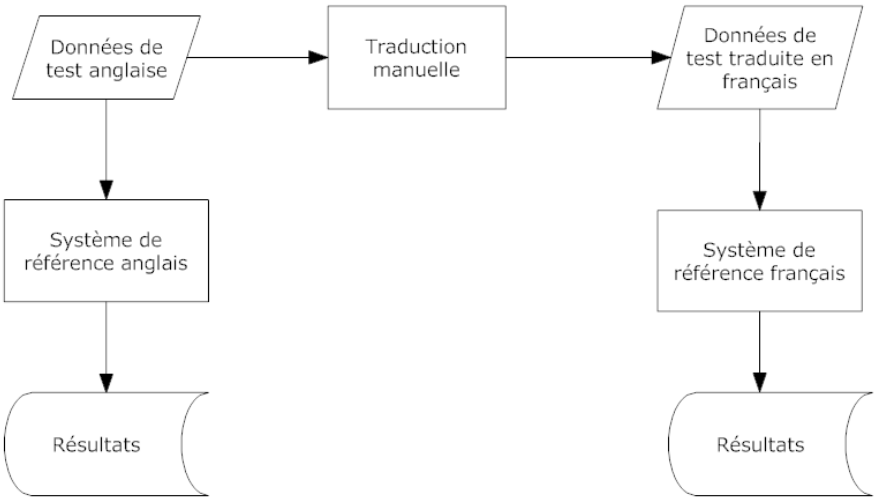
Mot	Count	Rang non-traduit	Rang corpus français
y’	1757	1	96
canceller	1016	2	127
allô	866	3	147
***	574	4	188
***	460	5	211
***	389	6	303
pus	305	7	285
okay	221	8	344
beep	168	9	403
cancellation	143	10	452
facturées	135	11	463
y’a	129	12	473

est le même, mais le sens attaché à ces classes peut avoir été confondu lors de l’étiquetage des deux applications dans les deux langues. Comme il s’agit en fait de deux applications séparées et non une application bilingue, les inconsistances dans l’étiquetage à travers les langues sont plus importantes. Afin d’avoir une idée de cette inconsistance, une expérience a été construite pour le mesurer. La méthode la plus simple aurait été de traduire manuellement les données d’entraînement du français vers l’anglais afin d’avoir de vraies traductions et une référence, par contre ceci reste peu pratique vu le nombre élevé des données d’entraînement.

Une méthode plus réaliste a été conçue avec les données de tests anglaises. Cette méthode consiste à prendre le jeu de test anglais et de le traduire manuellement vers le français. Il existe donc maintenant deux jeux de tests, un en français et un autre en anglais, les deux possédant la même distribution des étiquettes pour les mêmes sens des transcriptions. Comme le jeu de test est traduit manuellement de l’anglais au francais, il s’agit de traduction qu’on peut considérer comme *parfaite*. Il est donc maintenant possible de comparer les systèmes de référence français et anglais sur le même jeu de test. La figure 5.8 présente un schéma de la configuration de l’expérience réalisée pour

mesurer la dégradation due à l'inconsistance de l'étiquetage entre les langues.

Figure 5.8 – Configuration de l'expérience pour mesurer la dégradation des performances



La figure 5.9 et le tableau 5.11 présentent les résultats de l'expérience sur la mesure de la dégradation. Les résultats sur le graphique montrent une grande différence entre le système de référence anglais et le système français testé avec le jeu de test traduit manuellement. Dans cette expérience, les deux systèmes anglais et français sont comparés sur le même jeu de test. Le système de référence français présente une grande différence avec le système anglais. Comme il s'agit de la même application la différence est imputable à l'étiquetage des données d'entrainement. Il y a une différence d'environ 5% absolu entre le meilleur système de traduction automatique et le système français sur le test anglais traduit manuellement en français. Le système traduit est donc assez proche des performances du système de référence français sur le jeu de test de l'application anglaise. Il est donc possible d'avancer que la méthode par traduction automatique résout bien le problème de déploiement d'un système de routage téléphonique d'une langue source à une langue cible.

Cette inconsistance est aussi due au fait que les classes sont sémantiquement très proches. Par exemple les deux énoncés : "Quel est le solde de mon compte" et "solde

du compte" peuvent être reliés à deux classes différentes. Dans ce sens, il se peut que les personnes qui ont étiqueté les corpus dans les deux langues aient assigné les classes différemment.

Figure 5.9 – Résultats de l'expérience démontrant une inconsistance dans les données d'entrainement française et anglaise

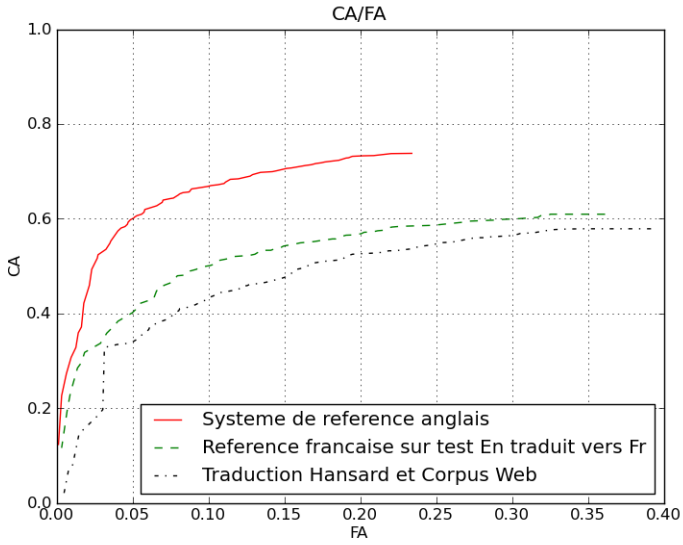


Tableau 5.11 – Résultats de l'expérience sur la mesure de l'inconsistance des données

	CA (%)	FA (%)
Système de référence anglais	*73.79*	*23.39*
Système francais sur test traduit manuellement	60.94	36.14
Traduction Hansard + corpus web	57.86	39.16

5.4 Classification multilingue

La section suivante présente les résultats obtenus avec la deuxième approche réalisée pour le déploiement automatique d'une application de routage téléphonique d'une langue source vers une langue cible : la classification multilingue. Les données utilisées pour effectuer les expériences pour la technique de classification multilingue sont les

mêmes que celles pour la méthode par traduction automatique. La différence est qu'elles ont subi un prétraitement, soit une projection dans un espace sémantique. Elles souffrent donc des mêmes caractéristiques mises en lumière dans la section précédente. C'est-à-dire une inconsistance dans l'étiquetage des deux langues.

Il n'a pas été possible d'utiliser cette technique avec le système de routage téléphonique développé chez Nuance, car l'entrée du système pour l'entrainement doit être une liste d'énoncés étiquetés. Ici, comme on fait une projection dans un espace de concepts, les dimensions ne sont plus les mots. Par contre, cette technique pourrait facilement être implantée dans le système de routage en ajoutant une étape de prétraitement avant l'étape d'entrainement et celle de test. Dans la section 5.5 qui compare les performances des deux systèmes, la même configuration est utilisée pour les deux approches.

Les expériences ont donc dû être réalisées en utilisant un autre système de classification. Pour cela, un classifieur de type SVM entraîné en utilisant la librairie LIBSVM [4] a été utilisé. Les performances seront comparées à un système de référence et ce système a été réentrainé avec le même type de classifieur. Les sections suivantes présentent les résultats des expériences réalisées avec la technique de classification multilingue. Afin de pouvoir effectuer plusieurs expériences et voir l'effet des différents paramètres, un sous-corpus comprenant les 100 classes les plus fréquentes du corpus d'entrainement français a été utilisé. Le système de référence anglais a été ré-entraîné et ses performances ont été recalculées.

5.4.1 Modèle et entrainement

Le modèle est créé en utilisant les données de la langue source, car comme on veut déployer une application d'une langue source vers une langue cible, il existe déjà un corpus étiqueté dans la langue source. Deux algorithmes de *clustering* différents ont été implémentés pour la création du DM : LSA et ESA. Une fois le DM créé, on prend le

corpus d'entrainement de la langue source et on convertit chacun des énoncés en VSM qu'on projette ensuite dans l'espace des domaines à l'aide de la matrice D_s. On utilise ensuite un corpus comparable dans la langue cible afin d'inférer la matrice D_c qui va être utilisée pour projeter les énoncés de la langue cible dans l'espace des domaines qui est l'espace commun de classification.

5.4.2 Algorithme de *clustering*

Le choix de l'algorithme de clustering a un impact sur les performances de classification. Les deux algorithmes utilisés pour inférer le DM sont LSA et ESA. Ces algorithmes sont utilisés pour créer le DM dans la langue source, ensuite la technique de classification multilingue est utilisée pour inférer la matrice D_c qui projette la langue cible dans l'espace des domaines. Les données d'entrainement françaises sont toujours utilisées pour créer les DM dans la langue source. Le tableau 5.12 présente les résultats de l'expérience qui compare les deux algorithmes de *clustering* utilisés pour créer les DM. Les tâches 1 et 2 sont celles décrites à la section 4.1.3

Tableau 5.12 – Résultats de la comparaison des deux algorithmes de *clustering* pour l'inférence des DMs

Algorithme	LSA	ESA
	Précision	Précision
Système de référence anglais	*79.18*	
Tâche 1	47.39	**59.92**
Tâche 2	**51.67**	50.11

L'algorithme ESA performe mieux sur la tâche 1 et LSA sur la tâche 2. La tâche 1 étant plus proche de l'application à déployer, les performances sont supérieures lorsqu'on utilise les séparations sémantiques déjà effectuées. Pour la seconde tâche, qui est plus différente de celle qu'on tente de déployer, l'algorithme LSA va trouver des dimensions un peu plus générales qu'il sera possible d'inférer avec les énoncés de la tâche 2. C'est ce qui explique la faible différence entre LSA et ESA en utilisant cette tâche.

5.4.3 Corpus comparables utilisés

Les différents corpus comparables utilisés pour inférer la matrice D_c de la langue cible ont un impact sur les résultats de la classification. Le corpus doit être représentatif de la tâche et son vocabulaire doit être assez riche pour couvrir le vocabulaire des données de test. De plus, il doit y avoir des domaines sémantiques semblables à ceux présents dans le corpus source, afin de pouvoir inférer l'appartenance des mots cibles aux domaines. Ceci est important, car c'est la matrice D_c qui est utilisée pour projeter la langue cible dans l'espace des domaines qui sera utilisée pour la classification.

Le tableau 5.13 présente les résultats obtenus avec les différents corpus utilisés pour inférer la matrice D_c dans la langue cible. Les meilleures performances sont obtenues en utilisant la tâche 1, par contre ce n'est pas celle qui présente le plus bas taux de mots hors vocabulaire. Comme cette tâche est plus proche de l'application à déployer, l'inférence de la matrice D_c relie les bons mots aux bons domaines de l'espace sémantique de manière plus adéquate. Il y a plus de mots HV, mais ceux-ci sont moins discriminants, car les performances sont supérieures en utilisant un corpus qui en possède un plus grand pourcentage.

Le corpus web quant à lui, a les moins bons résultats. Bien qu'il aide à la traduction, il ne permet pas de faire adéquatement l'inférence des domaines sémantiques dans la langue cible. Son taux de mots hors vocabulaire est aussi assez élevé. Plus du quart des mots ne sont pas présents dans ce corpus. Le fait est qu'il est relativement petit comparé aux autres corpus. On pourrait penser qu'en allant chercher un corpus web plus riche, par exemple en utilisant Wikipédia, on pourrait augmenter les performances.

Tableau 5.13 – Résultats de la comparaison de l'utilisation de différents corpus comparables pour l'inférence des domaines dans la langue cible

Corpus	Précision	HV (% diff)	HV (% total)
Tâche 1	**59.92**	7.85	5.94
Tâche 2	50.11	**1.66**	**1.19**
Corpus Web	39.9	25.86	8.69

5.4.4 Effet du nombre de mots et de la fertilité

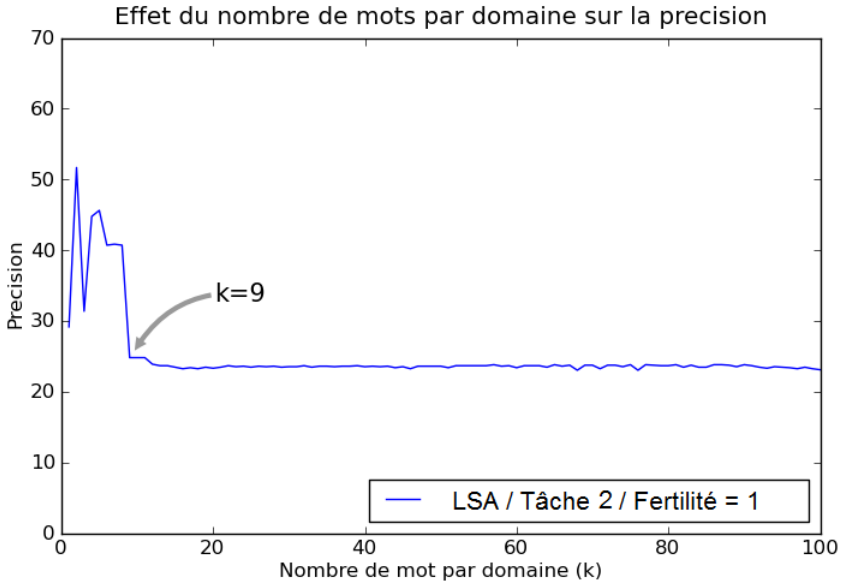
Cette partie analyse l'impact de deux paramètres ajustables lors de la création du DM. Le nombre de mots utilisés dans chaque domaine lors de l'inférence de la matrice D_c ainsi que la fertilité ont un effet sur les performances de la classification. Ces paramètres ont une influence directe sur la création de la matrice D_c qui est utilisée pour projeter la langue cible dans l'espace des domaines.

Le nombre de mots utilisés (k) pour inférer chaque domaine a un impact sur les performances du système, la figure 5.10 montre le résultat de la précision lorsque le nombre de mots par domaine varie de 1 à 100. La meilleure précision est obtenue lorsque le nombre de mots se situe entre 1 et 9. Lorsque la valeur dépasse 9 les performances restent à peu près stables. Les performances optimales sont obtenues lorsque $k = 2$. Donc, parmi tous les mots différents du corpus source, les meilleures performances sont obtenues lorsqu'on utilise de 1 à 9 mots par domaines pour inférer l'appartenance de tous les mots cibles à ces domaines. Lorsque trop de mots sont utilisés, il y a des mots non pertinents qui viennent biaiser la création de matrice D_c ; il faut donc uniquement utiliser les mots les plus importants pour chaque domaine. Ceci fait en sorte que le nombre de traductions requises est assez faible. Par exemple, avec un modèle qui comporte 100 domaines et un $k = 2$, seulement 200 traductions sont nécessaires.

Pour ce qui est de l'effet de la fertilité ou du nombre de mots cibles générés par chaque mot source lors de l'inférence de la matrice D_c, la figure 5.11 présente les ré-

sultats pour une fertilité de 1 à 10 ,pour 3 valeurs de k différents. La fertilité utilisée n'a pas d'effet bénéfique sur les performances de classification. Lorsque la fertilité augmente, les performances sont toujours en baisse ou restent stables. Il est donc préférable de toujours prendre le mot maximisant la probabilité de traduction (une fertilité de 1) lorsqu'on infère la matrice D_c. Les courbe $K = 1$ et $K = 2$ se dégradent plus rapidement car uniquement 1 et 2 mots par domaine sont utilisés pour inférer la matrice D_c. Les mots cibles supplémentaires utilisés lorsque la fertilité augmente créent plus d'ambiguïté.

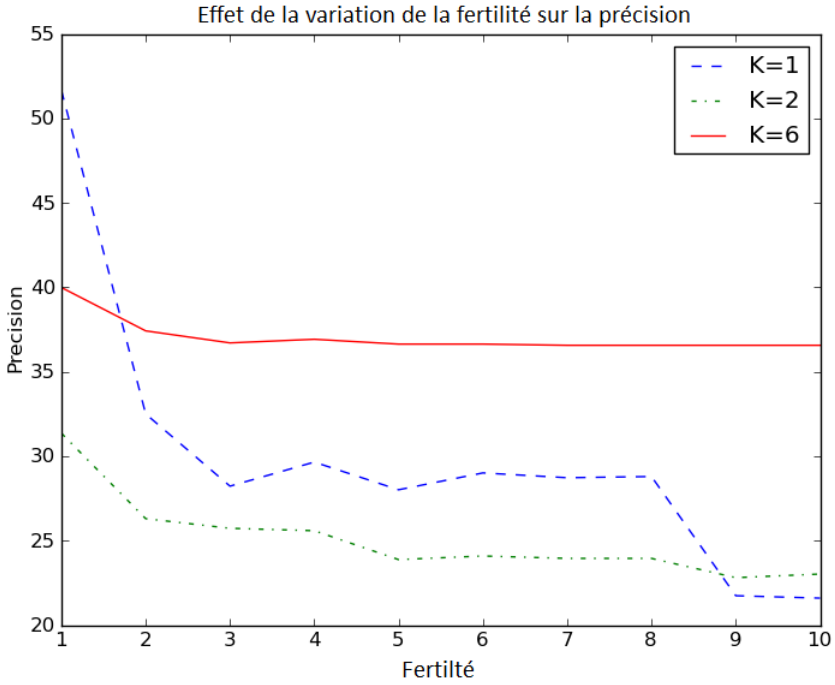
Figure 5.10 – Effet de la variation du nombre de mots par domaine dans l'algorithme d'inférence de la matrice D_c



5.4.5 Effet du nombre de classes du système

L'approche de classification multilingue performe de moins en moins bien lorsque le nombre de classes augmente, c'est-à-dire lorsque le nombre d'étiquettes sémantiques différentes augmente. On observe que les erreurs de classification ont tendance à faire intervenir des classes dont la sémantique est très proche. Étant donné que les données d'entrainement sont des énoncés très courts, la méthode par classification multilingue

Figure 5.11 – Effet de la variation de la fertilité dans l'algorithme d'inférence de la matrice D_c

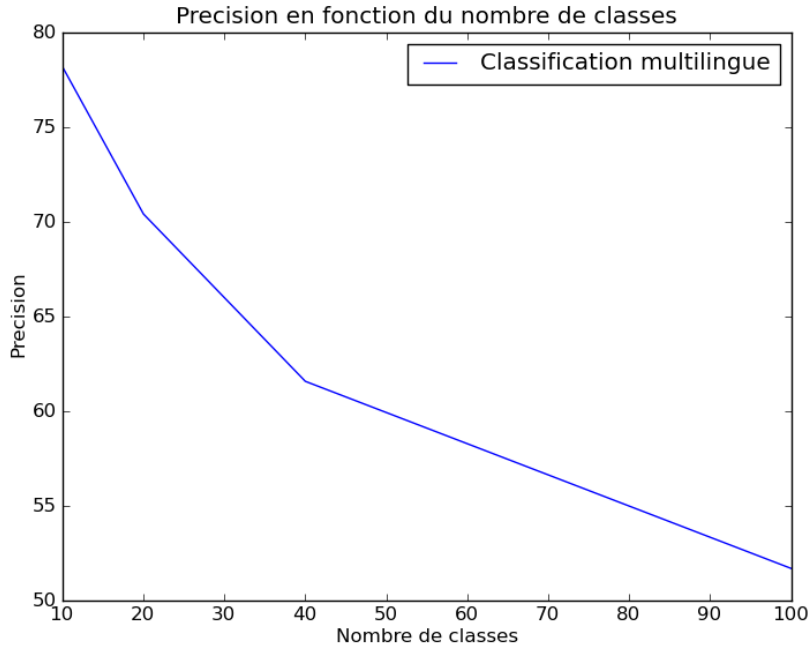


a de la difficulté à capturer les particularités qui discriminent deux classes très proches sémantiquement. De plus, plus le nombre de classes augmente, plus il y a des classes sémantiquement proches ; c'est-à-dire deux classes qui ont presque le même sens. La figure 5.12 montre les performances du système de classification multilingue plus le nombre de classes augmente.

Un autre facteur influençant la classification concerne les données utilisées pour inférer la matrice D_c qui est utilisée pour projeter la langue cible dans l'espace des domaines. Cette matrice doit contenir la plupart de mots discriminants qui seront utilisés par les utilisateurs. Les données doivent contenir le vocabulaire qui sera utilisé dans l'application cible, car les mots qui ne seront pas dans le corpus comparable ne seront pas pris en compte lors de la transformation. Ces mots ne sont pas dans les matrices D_c, il n'y a

donc pas d'information sur eux.

Figure 5.12 – Effet du nombre de classes du corpus d'entrainement du système de routage téléphonique avec la classification multilingue



5.4.6 Analyse des résultats

Il y a une différence d'environ 18% entre le meilleur système de classification multilingue et la système de référence anglais. Outre les particularités liées aux données décrites à la section précédente, il y a d'autres pertes de performance. Cela est dû au fait que les classes sont très proches sémantiquement, par exemple les étiquettes de deux classes différentes pourraient être "*information*" et "*demande d'information*", ces deux classes auront beaucoup de mots en commun. La sémantique est donc très finement découpée, ce que la classification multilingue arrive mal à discriminer. Lorsque ce genre de découpage arrive, le classifieur va généralement assigner l'étiquette liée à la classe qui a une probabilité *a priori*[6] plus élevé, car les énoncés d'entrées sont pratiquement

[6]Nombre d'énoncés de cette classe dans le corpus d'entrainement.

semblables.

Ceci montre l'espace des domaines créé par les algorithmes de *clustering* n'est pas assez fin, sémantiquement parlant, pour pouvoir capter les subtilités qui vont discriminer deux classes très sémantiquement proches. L'algorithme de *clustering* LSA va trouver les domaines les plus discriminants, donc des domaines très généraux et qui sont les plus éloignés possible les uns des autres. Ce n'est pas assez pour capter de petites variations et offrir la possibilité au classifieur de discriminer entre deux classes très proches sémantiquement. L'algorithme ESA, quant à lui, va utiliser les classes du corpus d'entrainement de la langue source comme dimension de l'espace des domaines, mais comme les mots sont pondérés par leur IDF, certains mots qui sont communs, mais discriminants pour une seule classe ont leur poids diminué. Il est important de pondérer les mots par leur IDF si l'on ne veut pas que les mots vides donnent du poids à tous les domaines lors de la projection dans l'espace des domaines.

La classification multilingue arrive à bien faire la distinction entre deux classes qui sont très éloignées, mais performe mal lorsqu'elles sont très proches. On pourrait donc avancer qu'elle serait plus appropriée pour une tâche dans laquelle les classes sont plus éloignées.

Récemment, Prettenhofer et Stein [23] ont utilisé une approche semblable pour développer une technique de classification multilingue. Leur approche utilise un petit nombre de traductions et un corpus comparable. Par contre, leur technique utilise un oracle de traduction[7] et leurs données comportent uniquement deux classes. Les résultats montrent que leur approche est légèrement inférieure à une technique de traduction automatique. Ceci est semblable aux résultats obtenus avec l'approche de classification multilingue présentée ici.

[7] Des traductions réalisées par un expert.

5.5 Comparaison des résultats

La technique utilisant l'approche par traduction automatique utilise un système de routage téléphonique complet, les résultats présentés sont les résultats de ce système. Par contre, la méthode de classification multilingue n'a pas pu être introduite dans le système de routage téléphonique, car elle demande une modification du coeur de l'application. Pour les comparer, les deux approches ont été testées sur un classifieur de type SVM avec la library LIBSVM et son interface Python comme utilisé pour montrer les résultats de la méthode par classification multilingue. Les résultats de cette section pour l'approche de traduction automatique seront donc différents de ceux déjà présentés pour cette méthode. Le tableau 5.14 présente les résultats des deux approches.

L'approche par traduction automatique performe mieux sur le jeu de test anglais que l'approche par classification multilingue. La classification multilingue dépend grandement du corpus comparable utilisé pour inférer la matrice qui va projeter la langue cible dans l'espace des concepts. Cette matrice a un impact sur tous les mots et les domaines tandis que pour la traduction, un mot mal traduit implique seulement un mot.

Tableau 5.14 – Comparaison des techniques de classification multilingue et de traduction automatique

Corpus	Précision
Système de référence anglais	*79.18*
Classification multilingue	59.92
Traduction automatique	**65.14**

Le tableau 5.15 fait la comparaison des deux approches avec un point de vue sur les performances et le processus de déploiement.

Tableau 5.15 – Comparaison des approches de déploiement d'un système de routage téléphonique d'une langue source vers une langue cible

	Traduction automatique	Classification multilingue
Performances	*13 %* de précision absolue de moins que le système de référence.	*19 %* de précision absolue de moins que le système de référence.
Deploiement	• Acquisition des corpus parallèles généraux. • Forage des corpus parallèles spécifiques. • Entrainement des modèles de traduction. • Traduction des données d'entrainement. • Entrainement du système de routage.	• Acquisition des corpus comparables. • Création des DM source. • Inférence des DM cible. • Projection dans l'espace des domaines et entrainements du système de routage.

CHAPITRE 6

CONCLUSION ET PERSPECTIVES

Ce mémoire a présenté deux approches pour résoudre le problème de déploiement automatique d'une application de routage téléphonique d'une langue source à une langue cible. La première technique utilise un système de traduction et la seconde un algorithme de classification tirant profit d'un espace sémantique commun. Ces deux techniques ont été décrites puis comparées.

Il a été démontré que les données utilisées pour les expériences présentaient des inconsistances qui causaient une grande différence entre le système de référence et le système traduit. Ces inconsistances sont dues au fait que les données proviennent du même centre d'appel, mais que les deux systèmes (français et anglais) ont été déployés comme deux systèmes différents et non comme une seule application bilingue.

Le système d'alignement de corpus web a su améliorer les performances du système de traduction en allant chercher des données dans le domaine des énoncées à traduire. Cette technique peut s'avérer utile, par exemple pour améliorer le modèle de langue de la reconnaissance vocale. Les données ainsi forées provenant du site web corporatif sont en effet directement dans le domaine de l'application à déployer. La prochaine étape serait l'utilisation de ce système à plus grande échelle pour forer un plus grand corpus parallèle.

La seconde technique a su utiliser les méthodes de création d'un espace sémantique afin d'effectuer une classification multilingue en utilisant le même espace. Les expériences ont comparé deux algorithmes de création des domaines sémantiques et l'impact des différents hyperparamètres du système. Une prochaine expérience pour améliorer cette méthode serait d'utiliser Wikipédia comme corpus comparable afin de forer un cor-

pus plus riche et ainsi créer un espace sémantique plus discriminant. L'avantage d'une encyclopédie en-ligne comme Wikipédia est qu'elle comprend un grand nombre d'articles dans plusieurs langues qui sont libres d'accès. Cette encyclopédie pourrait être utilisée afin d'améliorer les techniques d'inférence des matrices utilisées pour la projection vers l'espace des domaines. Présentement, les domaines sont uniquement créés en utilisant les données d'entrainement de la langue source, mais l'ajout de données externes pourrait apporter de la richesse au modèle.

Les résultats de l'approche par traduction automatique surpassent l'approche de classification multilingue. Par contre, la technique de classification multilingue présente l'avantage d'avoir uniquement besoin de déployer un seul classifieur et nécessiter moins de données. Dépendemment du type d'application, l'une ou l'autre technique peu s'avérer utile. En déployant uniquement un classifieur, il est facile d'ajouter une langue au système sans avoir à tout réentrainer, on a uniquement besoin d'inférer la matrice de projection de la langue vers l'espace des domaines.

Bien que les résultats utilisant le système traduit présentent une grande différence comparée avec le système de référence, il a été montré qu'une partie de cette différence est due à une inconsistance dans l'étiquetage des deux systèmes. Toutefois, il est pertinent pour une compagnie qui veut déployer un système dans une autre langue d'utiliser cette technique comme méthode de *booststrapping*. Donc une méthode qui sert à démarrer le déploiement d'un nouveau système sans nécessairement être le système final. C'est-à-dire, utiliser cette technique pour déployer un système dans une autre langue, les performances seront plus faible que celles du premier système, mais, pendant ce temps, on récolte des données pour améliorer le système traduit ultérieurement.

Dans une utilisation réelle de ce système, il faudra tenir compte des besoins et du type d'application à déployer pour choisir la bonne technique. La disponibilité des données

jouera un rôle crucial dans la décision de la méthode à utiliser.

BIBLIOGRAPHIE

[1] Gliozzo Alfio et Strapparava Carlo. *Semantic Domains in Computational Linguistics*. Springer-Verlag, 2009.

[2] J. Bellegarda. Statistical language model adaptation : review and perspectives. *Speech Communication*, 42(1):93–108, January 2004. ISSN 01676393. URL `http://dx.doi.org/10.1016/j.specom.2003.08.002`.

[3] Peter E. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra et Robert L. Mercer. The mathematics of statistical machine translation : parameter estimation. *Computational Linguistics*, pages 263–311.

[4] Chih-Chung Chang et Chih-Jen Lin. *LIBSVM : a library for support vector machines*, 2001.

[5] Jiang Chen et Jian-Yun Nie. Parallel web text mining for cross-language ir. Dans *IN IN PROC. OF RIAO*, pages 62–77, 2000.

[6] K. Dayanidhi D. Suendermann, J. Liscombe et R. Pieraccini. Localization of speech recognition in spoken dialog systems : How machine translation can make our lives easier. Dans *Interspeech 2009, 10th Annual Conference of the International Speech Communication Association*, 2009.

[7] Pascale Fung et Lo Yuen Yee. An ir approach for translating new words from nonparallel, comparable texts. Dans *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 414–420, Montreal, Quebec, Canada, August 1998. Association for Computational Linguistics.

[8] Evgeniy Gabrilovich et Shaul Markovitch. Computing semantic relatedness using

wikipedia-based explicit semantic analysis. Dans *In Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611, 2007.

[9] Alfio Gliozzo. Semantic domains and linguistic theory. Dans *In Proceedings of the LREC 2006 workshop*, 2006.

[10] Alfio Gliozzo et Carlo Strapparava. Cross language text categorization by acquiring multilingual domain models from comparable corpora. Dans *ParaText '05 : Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 9–16, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[11] Alfio Gliozzo et Carlo Strapparava. Domain kernels for text categorization. Dans *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL)*, pages 56–63, 2005.

[12] Hieu Hoang, Alexandra Birch, Chris Callison-burch, Richard Zens, Rwth Aachen, Alexandra Constantin, Marcello Federico, Nicola Bertoldi, Chris Dyer, Brooke Cowan, Wade Shen, Christine Moran et Ondrej Bojar. Moses : Open source toolkit for statistical machine translation. pages 177–180, 2007.

[13] Daniel Jurafsky et James H. Martin. *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, second édition, February 2008.

[14] Philipp Koehn. Europarl : A parallel corpus for statistical machine translation. Dans *2nd Workshop on EBMT of MT-Summit X*, pages 79–86, 2005.

[15] Philipp Koehn, Franz Josef Och et Daniel Marcu. Statistical phrase-based translation. Dans *HLT-NAACL*, 2003.

[16] Michael Littman, Susan T. Dumais et Thomas K. Landauer. Automatic cross-language information retrieval using latent semantic indexing. Dans *Cross-*

Language Information Retrieval, chapter 5, pages 51–62. Kluwer Academic Publishers, 1998.

[17] Christopher D. Manning et Hinrich Schuetze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1 édition, June 1999. ISBN 0262133601.

[18] Franz Josef Och. Minimum error rate training in statistical machine translation. Dans *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July 2003. Association for Computational Linguistics.

[19] Franz Josef Och et Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[20] Franz Josef Och et Hermann Ney. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30(4):417–449, 2004. ISSN 0891-2017.

[21] Franz Josef Och, Christoph Tillmann, Hermann Ney et Lehrstuhl Fiir Informatik. Improved alignment models for statistical machine translation. Dans *University of Maryland, College Park, MD*, pages 20–28, 1999.

[22] Alexandre Patry et Philippe Langlais. Paradocs : un système d'identification automatique de documents parallèles. Dans *12e Conference sur le Traitement Automatique des Langues Naturelles (TALN)*, pages 223–232, Dourdan, France, June 2005. URL `http://www-etud.iro.umontreal.ca/~patryale/papers/patry_langlais_2005_taln.pdf`.

[23] Peter Prettenhofer et Benno Stein. Cross-language text classification using structural correspondence learning. Dans *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1127, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P10-1114`.

[24] Philip Resnik et Noah A. Smith. The web as a parallel corpus. *Computational Linguistics*, 29:349–380, 2003.

[25] G. Salton, A. Wong et C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975. ISSN 0001-0782.

[26] John Shawe-Taylor et Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521813972.

[27] Lei Shi, Cheng Niu, Ming Zhou et Jianfeng Gao. A dom tree alignment model for mining parallel data from the web. Dans *In COLING/ACL-2006*, pages 489–496, 2006.

[28] Georges Siolas et Florence d'Alché Buc. Support vector machines based on a semantic kernel for text categorization. Dans *IJCNN '00 : Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)-Volume 5*, page 5205, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0619-4.

[29] A. Stolcke. Srilm – an extensible language modeling toolkit, 2002. URL `http://citeseer.ist.psu.edu/stolcke02srilm.html`.

[30] Alexei Vinokourov, John Shawe-taylor et Nello Cristianini. Inferring a semantic representation of text via cross-language correlation analysis, 2002.