**Université de Montréal**

# Projection multilingue d'annotations pour dialogues avancés

**par Simon JULIEN**

**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 M. Sc. en informatique

Décembre, 2013

# RÉSUMÉ

Depuis quelques années, les applications intégrant un module de dialogues avancés sont en plein essor. En revanche, le processus d'universalisation de ces systèmes est rapidement décourageant : ceux-ci étant naturellement dépendants de la langue pour laquelle ils ont été conçus, chaque nouveau langage à intégrer requiert son propre temps de développement. Un constat qui ne s'améliore pas en considérant que la qualité est souvent tributaire de la taille de l'ensemble d'entraînement.

Ce projet cherche donc à accélérer le processus. Il rend compte de différentes méthodes permettant de générer des versions polyglottes d'un premier système fonctionnel, à l'aide de la traduction statistique. L'information afférente aux données sources est projetée afin de générer des données cibles parentes, qui diminuent d'autant le temps de développement subséquent.

En ce sens, plusieurs approches ont été expérimentées et analysées. Notamment, une méthode qui regroupe les données avant de réordonner les différents candidats de traduction permet d'obtenir de bons résultats.



**Mots-clés : projection syntaxique, dialogues avancés, traduction automatique, traitement par consensus, universalisation.**

## ABSTRACT

For a few years now, there has been an increasing number of applications allowing advanced dialog interactions with the user. However, the universalization of those systems quickly becomes painful : since they are highly dependent on the original development language, each new language to integrate requires an additionnal and significative time investment. A matter that only gets worse considering quality usually rests on the size of training set.

This project tries to speed up the overall process. It presents various methods to generate multilingual versions of a first functionnal system, using statistical machine translation. Information from the source data is projected to another language in order to create similar target data, which then reduces the upcoming development time.

Many approaches were tested and analysed. In particular, a method that regroups data in clusters before reordering the associated translation candidates shows promising results.



**Keywords : syntactic projection, advanced dialogs, machine translation, consensus treatment, universalization.**

TABLE DES MATIÈRES

# LISTE DES TABLEAUX

# LISTE DES FIGURES

# LISTE DES SIGLES ET DES ABRÉVIATIONS

| | |
|---|---|
| **BLEU** | *Bilingual Evaluation Understudy* |
| **BP** | BLEU *Penalty* |
| **CHV** | Canevas hors-vocabulaire |
| **CYK** | Algorithme de Cocke-Younger-Kasami |
| **DIRO** | Département d'informatique et de recherche opérationnelle |
| $\mathbf{F_1}$ | F-mesure combinant précision et rappel, où le paramètre $\beta = 1$ |
| **PROSAC** | *Progressive Sampling Consensus* |
| **RANSAC** | *Random Sampling Consensus* |
| **SMT** | *Statistical Machine Translation* |
| **TALN** | Traitement automatique du langage naturel |

# CHAPITRE 1 — INTRODUCTION

Si le rêve de s'entretenir familièrement avec les machines est presque aussi vieux que leur avènement, la technologie peine à s'élever à la hauteur de l'ambition. Malgré tout, la recherche en ce domaine est jalonnée de réussites historiques — ELIZA, Systran, Siri — ayant ouvert la voie aux solutions actuelles, qui elles offrent une véritable flexibilité de conversation. Leur acuité est telle que l'industrie n'hésite plus à les intégrer à leurs produits, voire à en faire une caractéristique centrale. L'essor couvre un large éventail de domaines, de la téléphonie jusqu'à l'assistance de personnes ayant un handicap visuel ou moteur. Ces solutions restent cependant imparfaites, et les aspects à améliorer ne manquent pas.

En particulier, le temps nécessaire au développement d'une telle application reste non-négligeable. Cela devient particulièrement irritant lors de l'universalisation : chaque langue à intégrer nécessite son propre temps de développement, car les modules de dialogues avancés sont naturellement dépendants du langage pour lequel ils ont été conçus. Or, il existe des systèmes de traduction efficaces. L'intuition suggère donc qu'il doit être possible de diminuer le temps requis pour déployer un module dans une deuxième langue lorsqu'il existe des ressources dans une première langue pour la même application. Une intuition qui se vérifie par l'existence de travaux sur la projection de ressources, notamment l'auto-amorçage d'analyseurs syntaxiques (*bootstrapping syntactic parsers*) [Hwa et al., 2004] et la détection de relations entre différentes entités sémantiques [Kim et al., 2010].

Le gain n'est pas dédaignable : dans une logique marchande, une diminution des coûts rend les marchés de moindre importance plus attrayants. Un constat qui s'affranchit de l'étiquette purement mercantile lorsqu'on considère des applications d'assistance médicale ou destinées à des personnes malvoyantes.

Le présent mémoire s'inscrit dans cette lignée. Il rend compte d'une méthode permettant de générer automatiquement des versions polyglottes d'un premier système fonctionnel. Mais tout d'abord, le premier chapitre effectue un survol de quelques concepts qui aideront le lecteur à mieux comprendre les tenants et les aboutissants du projet de recherche.

## 1.1 Traitement automatique du langage naturel

Le traitement automatique du langage naturel (TALN) concerne tous les logiciels et programmes informatiques s'intéressant aux capacités linguistiques humaines. C'est un domaine qui englobe, entre autres, l'extraction d'information, la stylométrie, la recherche de documents, la correction orthographique, les dialogues avancés et la traduction automatisée. Le TALN n'est pas l'apanage des informaticiens, qui travaillent de concert avec des linguistes, des cogniticiens, des mathématiciens et même des biologistes pour développer des solutions innovantes.

## 1.2 Applications de dialogues avancés

Le travail effectué dans le cadre de ce mémoire s'inscrit principalement dans le volet des dialogues avancés. Cette branche du TALN concentre ses efforts à développer des outils et des applications libérant graduellement les utilisateurs des contraintes imposées par la reconnaissance vocale et textuelle. Historiquement, les modèles se bornaient à distinguer quelques mots-clés prédéfinis, et supposaient une suite d'états finis dans la progression du dialogue. Par exemple, un système qui propose « pour le service en français, dites "français" » ne reconnait généralement que le mot « français » dans un laps de temps précis, sans comprendre des variations logiquement acceptables comme « je suis francophone ». Si cette structure rigide a le mérite d'augmenter l'efficacité de la détection et de limiter les erreurs de reconnaissance, elle demeure lourde et sous-optimale pour l'usager.

Les solutions actuelles de dialogues avancés commencent à s'affranchir de ces contraintes et couvrent maintenant un spectre beaucoup plus large de phrases et de modèles de dialogue. Notamment, l'utilisateur peut fournir plus d'un élément d'information à la fois, et ce, dans

l'ordre qui lui convient, en s'exprimant de façon naturelle au système. Les applications suggèrent un modèle de dialogue de base, mais l'utilisateur peut en déroger selon ses préoccupations courantes. Il en résulte une conversation beaucoup plus fluide et d'autant moins contraignante.

Le fonctionnement d'un système de dialogues avancés suit une architecture relativement simple. D'abord, une application est instanciée pour une tâche donnée : des concepts à capter de l'utilisateur — obligatoirement ou facultativement — sont ajoutés aux commandes globales qui restent toujours à disposition (ex : quitter, répéter, recommencer, aide...). À chaque interaction, un analyseur syntaxique extrait un ensemble d'interprétations possibles, sans tenir compte du contexte, qu'un module de compréhension ramène ensuite en contexte. Le système détermine alors dans quel état il devrait se trouver, selon les concepts qu'il a saisis et ceux qu'il lui reste à apprendre, puis il relance la conversation en conséquence. Notons que dans le cas où l'utilisateur s'exprime verbalement, une étape de reconnaissance vocale s'ajoute au processus. La figure 1.1 résume l'apport des différents composants lors du traitement d'une interaction typique.

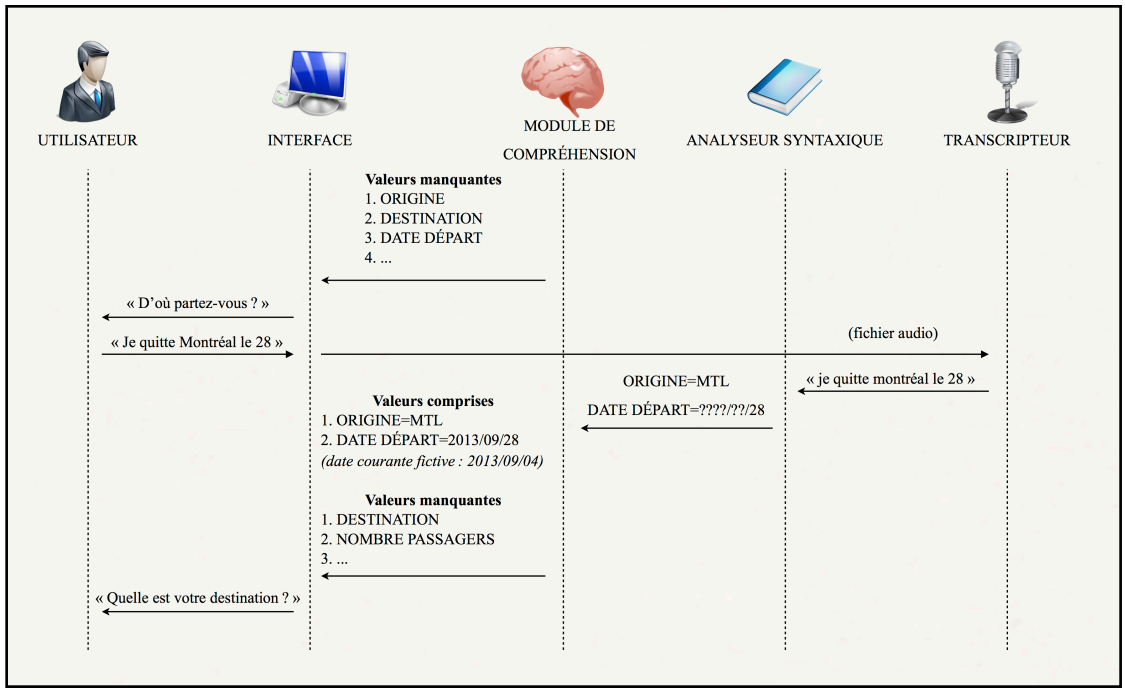



**Figure 1.1** — Traitement typique d'une interaction de dialogues avancés

Dans cet exemple fictif, on remarque que l'interface utilise la première valeur manquante pour relancer le dialogue avec l'utilisateur. Lorsque toutes les valeurs ont été collectées ou qu'il ne reste que des valeurs facultatives, une phase de confirmation est généralement enclenchée. Enfin, la date d'exécution a ici son importance : le module de compréhension connaissant la date du jour (arbitrairement fixée au 4 septembre 2013), il peut en déduire que « le 28 » fait référence au 28$^{e}$ jour du même mois et de la même année.

### 1.3 Analyseur syntaxique et grammaires

L'analyseur syntaxique (*parser* en anglais) est un élément clé d'un système de dialogues avancés. Il lui incombe d'extraire les concepts implicites d'une phrase quelconque[1]. Ainsi, le module de compréhension qui lui succède peut opérer depuis un espace de valeurs réduit. Les concepts comportent généralement deux parties : la première identifie le type de concept et la seconde, sa valeur courante. Différents algorithmes permettent d'effectuer la correspondance (*Shift-Reduce Parsing, CYK, Earley...*), mais tous reposent sur un ensemble de grammaires définissant les règles de simplification acceptables. L'exemple suivant illustre un ensemble de règles possibles et d'éventuelles décompositions :

**Règles**

```
<item>dorval<tag>AIRPORT="YUL"</tag></item>
<item>trudeau<tag>AIRPORT="YUL"</tag></item>
<item>florida<tag>STATE="US-FL"</tag></item>
<item>AIRPORT airport<tag>LOCATION=AIRPORT</tag></item>
<item>STATE<tag>LOCATION=STATE</tag></item>
<item>from LOCATION<tag>ORIGIN=LOCATION</tag></item>
<item>to LOCATION<tag>DESTINATION=LOCATION</tag></item>
```

**Exemples**

| | | |
|---|---|---|
| dorval airport | $\Longrightarrow$ | LOCATION=YUL |
| from trudeau airport | $\Longrightarrow$ | ORIGIN=YUL |
| to florida | $\Longrightarrow$ | DESTINATION=US-FL |

**Figure 1.2** — Grammaire simpliste et exemples associés

[1] Dans sa définition usuelle, l'analyseur syntaxique s'intéresse aux syntagmes plutôt qu'aux concepts.

Des grammaires plus raffinées entraineront une meilleure reconnaissance, et des règles doivent être définies pour chaque langage supporté (raisons qui justifient d'ailleurs la pertinence du présent mémoire). En outre, ces règles doivent permettre l'analyse d'une phrase dans son ensemble, ce qui s'avère problématique de prime abord. En reprenant les règles de l'encadré, par exemple, l'analyse achopperait sur un cas aussi simple que « *to florida please* », car la marque de politesse n'est pas considérée explicitement. Une solution élégante consiste à ajouter une règle universelle, notée par l'astérisque (*), de probabilité extrêmement faible. Cette improbabilité ne sert qu'à favoriser l'émergence des autres règles. Puis, l'énoncé en entrée est transformé en graphe simpliste, sur lequel s'effectue alors l'analyse. L'existence d'au moins un chemin pour l'ensemble des phrases possibles est ainsi assurée. On qualifiera de **robustes** les grammaires pourvues d'un tel mécanisme. La figure 1.3 illustre ce concept.

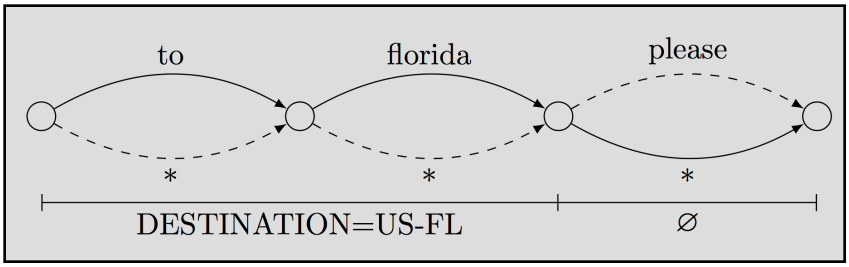



**Figure 1.3** — Analyse optimale à l'aide d'une grammaire robuste

Notons également que les énoncés peuvent admettre plusieurs décompositions. Dans ce cas, l'analyseur s'adjoint les services d'un ré-évaluateur (*rescorer* en anglais) qui ordonne les différentes possibilités selon leur probabilité. En particulier, les concepts dits de **haut niveau** sont toujours favorisés. Pour cette raison, dans l'exemple, le concept LOCATION serait préféré au concept STATE lors de l'analyse de « *florida* ». Il peut arriver que plusieurs décompositions soient équiprobables, notamment lorsque deux valeurs existent pour un même concept (ex. « *london* » pour la ville au Canada ou en Angleterre) ou pour deux concepts non comparables (ex. « *bill* » en tant que prénom ou pour désigner une facture en anglais). Bien qu'il soit possible de régler ces ambiguïtés à l'aide de connaissances ad hoc, les solutions non automatiques ont été sciemment écartées. Précisons qu'en général, les $n$ premières décompositions sont accessibles avec leur score associé.

Finalement, la répartition des responsabilités dans le traitement des dialogues avancés impose généralement que l'analyseur syntaxique opère sans contexte (à ne pas confondre toutefois avec la notion de grammaires « hors-contexte »). Autrement dit, pour une application et une phrase donnée, il devrait toujours effectuer la même analyse. Ce n'est donc pas à lui de déterminer si « *florida* » est une origine ou une destination selon l'état du dialogue. Il doit néanmoins lever le plus d'ambiguïtés possible lorsque la phrase elle-même permet la déduction et c'est pourquoi il peut affirmer que « *from florida* » réfère à une origine.

## 1.4 Ontologies

Dans un cadre restreint aux dialogues avancés, une ontologie est une représentation logique des liens unissant les différents concepts d'une même application. Elle définira par exemple qu'une origine et une destination sont des endroits, et qu'un endroit peut être un aéroport, une ville, un état, un pays, un continent... C'est grâce à l'ontologie que le module de compréhension peut récupérer les concepts transitoires de l'analyseur syntaxique et les superposer aux concepts attendus à la phase courante de dialogue (ex. savoir que « *florida* » est une destination si la dernière question posée était « quelle est votre destination ? » ou une origine si cette question était « d'où partez-vous ? »). La figure 1.4 offre une représentation visuelle possible d'une ontologie simplifiée.

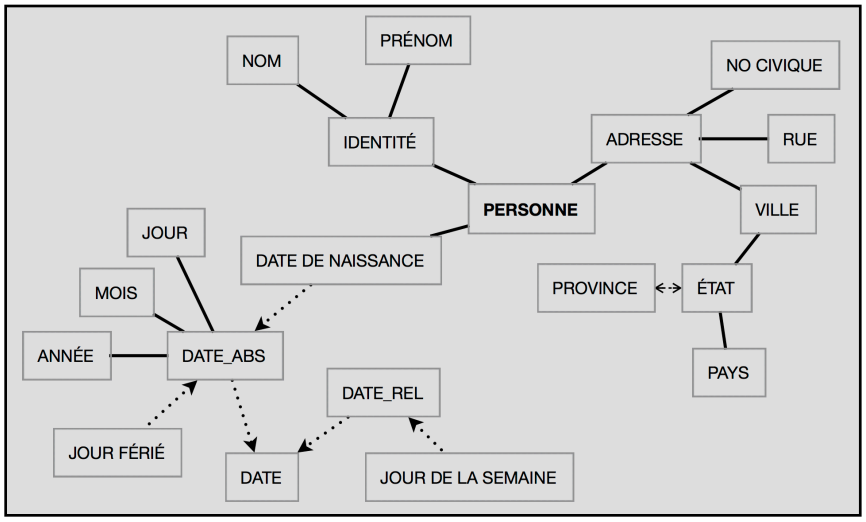



**Figure 1.4** — Représentation visuelle d'une ontologie

L'ontologie définit également des liens depuis des concepts dits **énumérables** vers les grammaires pertinentes. Par exemple, le concept PAYS pointerait vraisemblablement vers la liste des différents pays du monde. Notons que contrairement au reste de l'ontologie, ces liens sont tributaires de la langue courante (une certaine uniformité de notation est toutefois respectée d'une langue à l'autre). En sus, bien qu'ils ne soient pas strictement énumérables, les concepts de DATE et de TEMPS sont suffisamment usités pour avoir eux aussi leurs grammaires sous-jacentes dans chaque langue supportée.

## 1.5 Traduction statistique et modèles de langue

La traduction statistique (SMT — *Statistical Machine Translation*) est une façon courante de trouver la traduction $t$ dans une langue cible $T$ depuis une phrase d'entrée $s$ de la langue source $S$. Cette traduction correspond à l'élément maximisant l'équation 1.1 :

$$\bar{t} = \underset{t \in T}{\operatorname{argmax}} \left( P(t|s) \right) \tag{1.1}$$

Autrement dit, la traduction retenue est la plus probable étant donnée la phrase d'entrée $s$. Pour des raisons pratiques toutefois, cette équation est transformée en vertu du théorème de Bayes avant maximisation.

$$\begin{aligned} \bar{t} &= \underset{t \in T}{\operatorname{argmax}} \left( \frac{P(t) \cdot P(s|t)}{P(s)} \right) \\ &\propto \underset{t \in T}{\operatorname{argmax}} \left( P(t) \cdot P(s|t) \right) \end{aligned} \tag{1.2}$$

La maximisation comprend maintenant la composante $P(t)$, associée au **modèle de langue**, et la composante $P(s|t)$, associée au **modèle de traduction**. Le dénominateur $P(s)$ étant constant sur $T$, il est ignoré.

Le **modèle de langue** cherche à favoriser l'émergence de phrases syntaxiquement valables et repose sur un modèle $n$-grammes pour ce faire. À l'aide d'un corpus unilingue d'entraînement (langue $T$ dans le cas d'intérêt), il peut déterminer la probabilité d'un mot depuis un contexte formé par les $n-1$ mots précédents. Les probabilités individuelles de chaque mot sont ensuite multipliées entre elles pour déterminer la probabilité globale de la phrase pour le $n$-gramme courant. Différents $n$-grammes sont généralement combinés entre eux pour améliorer le modèle de langue, jusqu'à concurrence d'un 5-gramme. La taille du corpus d'entrainement, le nombre de calculs requis et le taux élevé d'hapax[2] restreignent habituellement la pertinence de considérer un modèle d'ordre plus élevé. La figure 1.5 retrace les calculs effectués pour déterminer la probabilité d'une phrase simple (« le chat dort ») dans un modèle quelconque d'ordre 3.

BOS : *Beginning of Sentence*

(1-gramme) $\mathrm{P}_1(t) = \mathrm{P}(\text{le}) \cdot \mathrm{P}(\text{chat}) \cdot \mathrm{P}(\text{dort})$

(2-gramme) $\mathrm{P}_2(t) = \mathrm{P}(\text{le}|\text{BOS}) \cdot \mathrm{P}(\text{chat}|\text{le}) \cdot \mathrm{P}(\text{dort}|\text{chat})$

(3-gramme) $\mathrm{P}_3(t) = \mathrm{P}(\text{le}|\text{BOS BOS}) \cdot \mathrm{P}(\text{chat}|\text{le BOS}) \cdot \mathrm{P}(\text{dort}|\text{chat le})$

$$\mathrm{P}(t) = \sum_{i=1}^{3} w_i \mathrm{P}_i(t)$$

**Figure 1.5** — Calcul de P(t) dans un modèle trigramme

Le **modèle de traduction**, quant à lui, repose sur un ensemble de phrases alignées entre les deux langues, communément appelé bitexte. De ce bitexte, il est possible de déterminer une distribution de probabilité de transfert $f$ d'un mot vers quelques traductions candidates (ex. « *calendar* » vers « calendrier » (0.242), « horaire » (0.151), « annuaire » (0.127), etc.). Cette distribution est ensuite utilisée pour déterminer la correspondance d'alignement optimale entre les mots ou les fragments (groupe de mots). Si la définition large du modèle de traduction ne pose aucune restriction quant à l'alignement retenu, une contrainte courante est d'exiger que chaque mot cible ait au plus un appariement source. Ces modèles, appelés IBM,

[2] Élément rencontré une seule fois lors de l'entraînement.

permettent de réduire l'espace de recherche et le temps de calcul [Brown et al., 1993]. Leur contrecoup est de casser la symétrie entre $S$ et $T$, ainsi que le met en évidence l'exemple de la figure 1.6.

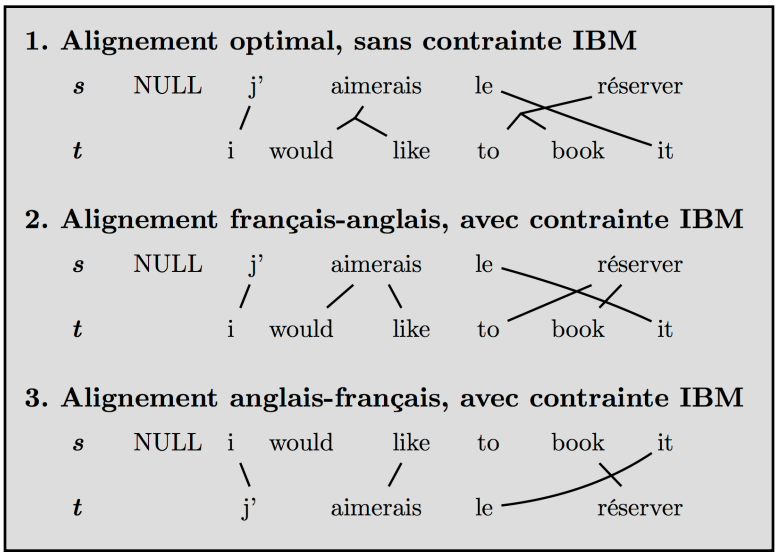



**Figure 1.6** — Différents alignements avec et sans contrainte IBM

La réduction du temps de calcul tient au fait que la probabilité de $s$ étant donné $t$ s'obtient en sommant sur tous les alignements possibles :

$$P(t|s) = \sum_{a} P(t, a|s) \tag{1.3}$$

D'autres hypothèses sur la longueur attendue de la traduction, du rang cible de chaque mot source et du mot cible selon ce rang permettent de simplifier encore davantage l'équation et de réduire le temps de calcul sous un seuil acceptable. Les probabilités à sommer s'obtiennent alors à l'aide du produit suivant[3] :

$$P(t, a|s) = \frac{\epsilon}{(l+1)^m} \times \prod_{i=1}^{m} f(t_i|s_{a_i}) \tag{1.4}$$

[3] Cas du sous-modèle IBM-1

Dans cette équation, $l$ réfère à la longueur de la phrase source et $m$, à celle de la phrase cible. La fonction $f$ intègre directement les probabilités de transfert d'un mot source vers un mot cible, telles que déterminées lors de l'entraînement sur le bitexte. Enfin, $\epsilon$ est une constante.

Lorsque l'ensemble d'entraînement est de taille suffisante, une meilleure performance peut être obtenue en considérant des fragments plutôt que des mots (*phrase-based* en anglais) [Koehn et al., 2003]. Il est également possible, et il s'agit de l'approche retenue dans le cadre de ce mémoire, d'opter pour une traduction hiérarchique [Chiang, 2005]. Celle-ci opère prioritairement sur des fragments, mais garde une certaine latitude au niveau des mots, et peut ainsi raffiner la qualité de la traduction au besoin. Notons que même si elle a donné de meilleurs résultats lors des expérimentations préliminaires, l'approche hiérarchique n'est pas nécessairement une évolution de l'approche par fragments. En fait, ses résultats sont généralement comparables, pour un coût en ressources plus élevé.

Bien qu'elle ne soit pas la seule méthode permettant de traduire d'une langue source vers une langue cible, l'approche statistique figure parmi les plus simples et les plus efficaces. Elle se compare en effet avantageusement aux autres méthodes, ne nécessite aucune connaissance a priori, reste souple et s'automatise complètement. En outre, un système de traduction statistique efficace existe en version publique (Moses[4] [Koehn et al., 2007]). Toutes ces raisons justifient le fait que ce soit cette méthode qui ait été retenue ici.

[4] www.statmt.org/moses/

# CHAPITRE 2 — PROBLÉMATIQUE ET ÉTAT DE L'ART

Le second chapitre s'amorce par une présentation sommaire de la problématique, à laquelle succédera une revue de littérature. En plus d'offrir une base de comparaison, les travaux cités ont permis d'identifier des pistes de recherche prometteuses et de développer un vocabulaire facilitant l'analyse. Puis, certains cas de figure seront décortiqués pour mettre en évidence quelques difficultés inhérentes au TALN. Enfin, l'objectif du mémoire est présenté en termes plus opérationnels en conclusion du chapitre.

## 2.1 Présentation de la problématique

Le développement d'une application de dialogues avancés nécessite des grammaires fiables, capables d'extraire efficacement le sens d'une phrase. Ces grammaires sont habituellement créées à l'aide d'un nombre significatif de données annotées. Or, l'annotation de ces données requiert un temps considérable, et comme elles varient d'une langue à l'autre, le processus d'annotation doit être systématiquement repris pour chaque langue.

Dans cette optique, ce mémoire explore différentes approches visant à projeter des données annotées dans une langue source vers une langue cible, allégeant ainsi la tâche d'annotation lors de l'universalisation.

## 2.2 État de l'art

Il existe peu de travaux axés sur la projection de grammaires dans le cadre proposé. L'absence s'explique par la spécificité des grammaires nécessaires au traitement des dialogues avancés. Alors que la complexité est habituellement proportionnelle au niveau d'étiquetage, les grammaires de dialogues avancés ont un taux élevé d'ambiguïtés et un taux faible de mots étiquetés. Autrement dit, les concepts à extraire ne sont associés qu'à quelques mots-clés, alors que la majorité des mots servent de contexte et n'ont aucune étiquette propre. Ainsi, les méthodes généralement efficaces de projection de ressources ne peuvent être utilisées

directement, car le niveau d'annotations n'est pas adéquat. Une revue de ces méthodes permet de mieux cibler le problème.

Lorsqu'il est question de projeter une grammaire, les approches peuvent être regroupées en deux catégories principales : adaptatives ou centralisatrices. L'approche **adaptative** (*grammar adaptation*) consiste à développer une grammaire complètement indépendante, en utilisant l'information de la grammaire initiale. À cet égard, les travaux de [Kim et al., 2003] et de [Santaholma, 2005] peuvent faire office d'exemples. Les premiers identifient efficacement les règles qui peuvent être conservées telles quelles pour accélérer la transition d'un langage à l'autre, et la seconde superpose des règles de traduction à la structure grammaticale.

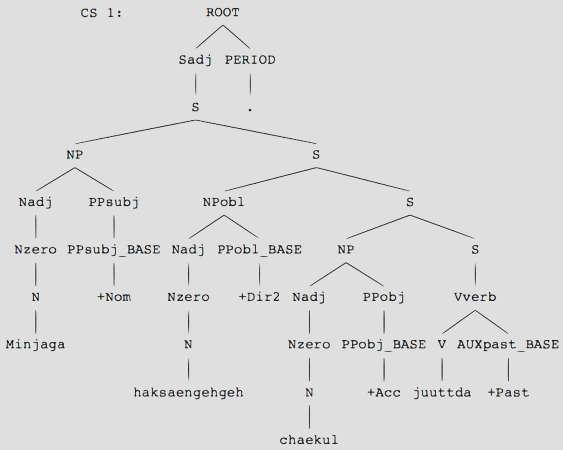
Dans les deux cas, l'étiquetage exhaustif du rôle et du type de chaque mot de la phrase est mis à profit. La logique reste pertinente, mais elle ne saurait être transférée directement au cas présent, où la majorité des mots, non étiquetés, fournissent un contexte précis à quelques mots-clés. C'est en effet l'approche dominante pour le traitement des dialogues avancés : les étiquettes ne s'appliquent qu'aux mots ayant un rôle actif, et c'est le concept associé qui cumule l'information déduite du contexte.

La figure 2.1 permet de mieux cerner la différence avec les travaux cités. Notons qu'il serait également impensable d'ajuster l'étiquetage actuel afin de le rendre plus exhaustif, car le temps requis pour ce faire cannibaliserait tout gain éventuel.

1. Étiquetage dans le cas d'intérêt — « demain, je veux aller à Montréal » (fr)

**DEPARTURE_DATE(DATE(DATE_REL(**demain**)))** je veux aller à **DESTINATION(LOCATION(CITY(**Montréal**)))**

2. Étiquetage [Kim et al., 2003] — « Minjaga haksaengehgeh chaekul juuttda » (ko)




3. Étiquetage [Santaholma, 2005] — « How frequent are your headaches ? » (en)

s : [sem=@fronting_sem(Adj, S), wh=y\/rel, wh=Wh, vform=VForm, inv=Inv, whmoved=y, operator_wrapped=n, takes_adv_type=none, gapsin=null, gapsout=null, elliptical_v=n] -->

adjp : [sem=Adj, wh=Wh, adjpos=pred, gapsin=null, gapsout=null],

s : [sem=S, wh=n, vform=VForm, inv=Inv, whmoved=n, operator_wrapped=n, gapsin=adjp_gap, gapsout=null, elliptical_v=n].

**Figure 2.1** — Différents étiquetages grammaticaux

L'approche **centralisatrice** (*grammar sharing*), pour sa part, cherche à regrouper toutes les langues en une grammaire unifiée. Lors du développement, les caractéristiques structurales partagées entre plusieurs dialectes sont regroupées, ce qui permet l'émergence d'une grammaire universelle hiérarchisée (voir la figure 2.2). De bons résultats ont été obtenus ainsi, notamment par [Bouillon et al., 2006] pour le français et le catalan, et plus largement par [Santaholma, 2008] avec l'anglais, le japonais, le finnois et le grec. Encore une fois, une

granularité élevée quant au rôle de chaque constituant de la phrase est pré-requis. C'est à ce niveau d'abstraction que surgissent les similarités exploitables entre les différents langages.

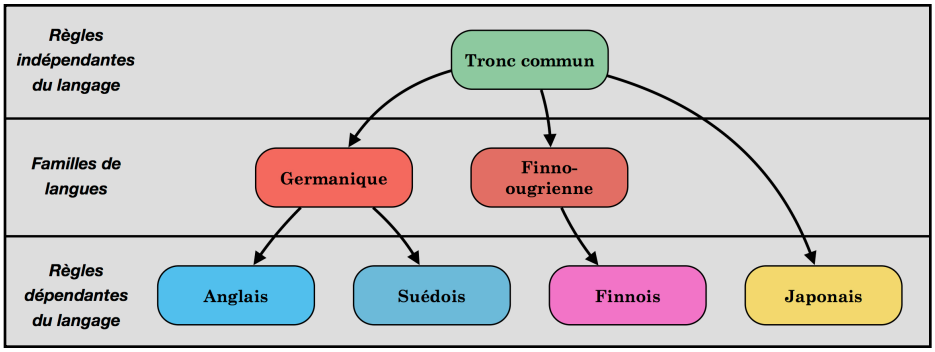



**Figure 2.2** — Structure d'une grammaire centralisatrice [Santaholma, 2008]

## 2.3 Ambiguïtés de langage

L'écriture de grammaires est un exercice beaucoup plus difficile qu'il n'y parait, surtout dans un contexte de dialogues avancés. Les exemples utilisés jusqu'à maintenant peuvent laisser croire qu'un algorithme avoisinant le « rechercher / remplacer » suffirait presque, mais la réalité dément cette fausse impression. En fait, seul le tiers des interactions s'accommode d'une solution aussi simpliste (voir le chapitre 4 à ce propos).

Pour mieux saisir l'ampleur réelle du défi et mieux comprendre le rôle prépondérant du contexte lors de l'analyse, voici un survol des difficultés et ambiguïtés inhérentes à l'étiquetage des données. Les exemples sont issus d'un véritable corpus, décrit au prochain chapitre.

D'abord, il est fréquent qu'un concept se spécialise (ex. de DATE vers DATE DE DÉPART ou DATE DE RETOUR). Son rôle final revêt alors une importance capitale pour éviter les erreurs d'appariement. D'autant que la flexibilité offerte à l'utilisateur implique généralement que les différents cas de figure sont équiprobables (ce n'est pas toujours le cas, mais une solution générique se doit de le supposer).

Le comportement espéré est qu'un humain en viendrait à la même catégorisation. C'est pourquoi étiqueter « Boston » comme un endroit — plutôt qu'une destination — dans la phrase « je vais à Boston » est une erreur significative. Naturellement, y voir une origine plutôt qu'une destination est une erreur encore plus grave.

Un autre réflexe serait de croire que quelques mots-clés dans un voisinage restreint suffisent à un étiquetage adéquat. Ce n'est pas le cas pour trois raisons. Premièrement, certaines prépositions interviennent dans différents contextes et ne peuvent donc pas directement déterminer le rôle de ce qui les suit. Deuxièmement, la modification d'un premier concept peut avoir une incidence directe sur le rôle des autres concepts de la phrase, peu importe leur position. Et finalement, il peut-être ardu de différencier deux concepts distincts d'un seul concept englobant. La figure 2.3 recense quelques cas illustrant ces raisons.

**1. Voisinage identique avec étiquetages différents**

de Montréal *au Brésil* (DESTINATION)

*de Sao Paulo au Brésil* jusqu'à Paris (ORIGINE)

**2. Modification locale avec répercussions globales**

*lundi*, je pars pour Los Angeles (lundi est une date de départ)

**3. Concept englobant ou concepts distincts**

je partirais dans la fin de semaine *du 20 au 22 août* (englobant)

je veux partir *du 13 au 18 mars* (DATE DE DÉPART et DATE DE RETOUR)

**Figure 2.3** — Contexte global et voisinage immédiat

Enfin, plusieurs ambiguïtés locales viennent compliquer la tâche et réaffirment l'importance du contexte. En voici une liste sommaire, élaborée depuis les cas rencontrés dans les corpus retenus pour le projet :

**1. Ambiguïté polysémique (plusieurs sens possibles)**

le jour de *Noël* (DATE) / François *Noël* (NOM)

dans *huit heures* (DURÉE) / à *huit heures* (HEURE)

**2. Ambiguïté anaphorique (le sens dépend du référent)**

j'aimerais revenir à la *même heure que pour l'aller* (HEURE DE RETOUR)

**3. Ambiguïté d'expressivité (sens pertinent ou non)**

*un* billet (expressif) / *un* départ le 20 mai (inexpressif)

*attendez* un instant (expressif) / *attendez*, je voulais dire... (inexpressif)

**4. Ambiguïté de rôle (mot autosuffisant ou modificateur)**

dans le *sud* (autosuffisant) / le *sud* de la France (modificateur)

**5. Ambiguïté quant à la négation (concept affirmé ou rejeté)**

en *soirée* (affirmé) / n'importe quand sauf en *soirée* (rejeté)

**6. Ambiguïté de contraction**

partir et revenir *demain* (DATE DE DÉPART et DATE DE RETOUR)

7. **Ambiguïté d'expansion**

mes *trois* enfants, *ma femme* et *moi* (NB PASSAGERS=5)

**Figure 2.4** — Liste d'ambiguïtés courantes

## 2.4 Génération automatique de grammaires

L'écriture des grammaires étant une tâche non triviale, celles-ci sont souvent élaborées à la main, en tout ou en partie. Néanmoins, il est possible d'automatiser le processus en étiquetant manuellement une certaine quantité d'exemples. Ces exemples définissent le comportement souhaité, duquel il est possible d'extraire une première grammaire. Les avantages de cette façon de faire sont nombreux : en plus du temps économisé, l'arborescence peut être dynamiquement simplifiée, l'identification de la source d'un comportement non désiré est immédiat et les éventuelles modifications de notations se font de manière efficace.

Dans sa forme la plus simple, la génération automatique d'une grammaire concatène bêtement les différentes règles associées à chaque exemple annoté, et permet un certain bruit entre deux composantes (voir la figure 2.5 à cet effet).

**Exemples annotés**

1. montrer le chèque CHECK_NUMBER(vingt et un)
2. PAY(payer) ma facture CREDIT_CARD(mastercard)
3. ce DATE(vendredi)
4. AMOUNT(trois cent deux dollars)
5. NUMBER(six) ORDER(dernières) TRX_LIST(transactions)

**Grammaire résultante**

```
<item repeat="1-">
<one-of>
  <item>*</item> <!-- grammaire robuste -->
  <item>montrer le chèque <ref>CHECK_NUMBER</ref></item>
  <item><ref>PAY</ref> ma facture <ref>CREDIT_CARD</ref></item>
  <item>ce <ref>DATE</ref></item>
  <item><ref>AMOUNT</ref></item>
  <item><ref>NUMBER</ref><ref>ORDER</ref><ref>TRX_LIST</ref></item>
</one-of>
</item>
```

**Quelques exemples reconnus par la grammaire**

ce jeudi 8 août (ce DATE)
dix-sept dollars (AMOUNT)
payer ma facture visa ce samedi (PAY ma facture CREDIT_CARD + ce DATE)
montrer le chèque 104 au montant de 30 dollars (montrer le chèque CHECK_NUMBER + * + * + * + AMOUNT)

**Figure 2.5** — Génération simpliste d'une grammaire

Ainsi construite, la grammaire ne permet la juxtaposition que si ses règles demeurent inaltérées. C'est pourquoi « montrer le chèque **numéro** 21 » serait analysé par la règle universelle (*) uniquement, malgré sa proximité avec l'exemple 1. À ce propos, un filet de sécurité courant consiste à ajouter des liens vers les concepts énumérables définis par l'ontologie (DATE, TIME, etc.). La modification ne sauverait pas l'exemple « montrer le chèque numéro 21 », qui utiliserait la règle ajoutée `<item><ref>NUMBER</ref></item>` en vain. Par contre, il est raisonnable de croire que cette modification suffirait pour une variation de l'exemple 2, disons « payer **la** facture mastercard ». Dans ce dernier cas,

l'identification des concepts énumérables « payer » et « mastercard », respectivement en tant que PAY et CREDIT_CARD, permettrait une bonne analyse.

Il va de soit qu'une méthode inférant de nouvelles règles ou ajoutant des synonymes permettrait d'obtenir de meilleurs scores. Toutefois, comme le mémoire se concentre sur l'étape de traduction, cet algorithme primaire a paru suffisant pour la suite. L'apport réel d'une méthode sur la performance finale est ainsi facilement dissociable d'un gain qui serait attribuable aux étapes de post-traduction.

### 2.5 Définition des objectifs

L'algorithme générant les grammaires de la section précédente permet de circonscrire l'objectif initial dans un cadre plus opérationnel. En effet, la projection d'une grammaire vers une autre langue trouve son équivalence dans la définition de l'objectif qui suit :

> Étant données $n$ phrases annotées dans une langue source, **l'objectif est d'obtenir, avec l'aide d'un système de traduction statistique, les $m$ meilleures phrases pré-annotées dans une langue cible.**

I

Quelques points nébuleux subsistent encore dans cette formulation: la notion de « $m$ meilleures phrases » reste encore à définir. Essentiellement, une grammaire sera dérivée des $m$ phrases en question (grâce à l'algorithme de la section 2.3) et celle-ci sera utilisée pour étiqueter des phrases tests de la langue cible. Un lot $m_1$ de phrases sera considéré meilleur qu'un autre selon sa performance sur les corpus de test. Le chapitre 3 brosse un portrait plus complet des corpus et des métriques utilisés à cette fin.

La qualité des $m$ traductions étant tributaire du système de traduction utilisé, le deuxième objectif, secondaire, cherche à maximiser la fiabilité dudit système :

Étant données $n$ phrases annotées dans une langue source et $m$ traductions pré-annotées dans une langue cible, **l'objectif est d'améliorer, sans intervention humaine, la qualité du système de traduction pour le domaine d'intérêt.** II

Encore une fois, les métriques permettant de départager divers systèmes de traduction font l'objet du prochain chapitre. Précisons aussi que ces objectifs se veulent des orientations principales et non des restrictions.

# CHAPITRE 3 — MÉTHODOLOGIE

Une dernière étape est nécessaire avant de s'atteler directement à la problématique : la définition des outils de recherche. C'est ce à quoi est consacré le présent chapitre. Dans un premier temps, il s'attarde à la description des données utilisées, puis il définit les différentes métriques qui ont orienté la recherche.

## 3.1 Nomenclature

Afin de faciliter la compréhension, voici quelques raccourcis de langage qui seront utilisés. Une donnée est dite **étiquetée** (ou **annotée**) lorsque tous ses concepts pertinents ont été identifiés. Un **concept** est une entité logique à extraire, composée de deux parties : son **étiquette** (ou son **nom**) et son **verbatim**. Le verbatim permet de déterminer la **valeur** du concept. À titre d'exemple, considérons la donnée annotée suivante :

**BOOLEAN**(oui) je veux **GLOBAL**(quitter l'application)

Cette donnée comporte deux concepts. Le premier est formé de l'étiquette BOOLEAN et du verbatim « oui ». Le second, de l'étiquette GLOBAL et du verbatim « quitter l'application ». Leur valeur respective dépend des grammaires, mais pourrait vraisemblablement être [*true*] et [*quit*]. Il peut être plus instructif de préciser que deux verbatims différents, mais identiques aux yeux du système (ex. « oui » et « c'est exact »), engendrent la même valeur. Un concept est **hiérarchisé** lorsque un ou plusieurs concepts s'y trouvent imbriqués. C'est le cas de l'exemple qui suit :

**LOCATION**(**CITY**(Miami) **STATE**(Florida)) please

Les différentes notions s'adaptent intuitivement. Ainsi, le concept au sommet de la hiérarchie (concept **primaire**) aurait ici l'étiquette LOCATION, le verbatim « *Miami Florida* » et sa valeur serait vraisemblablement la concaténation des valeurs de ses deux sous-concepts. Par opposition aux concepts primaires, un concept englobé est qualifié de **secondaire**.

Un **canevas** s'obtient en remplaçant tous les verbatims d'une donnée annotée par leurs étiquettes respectives. Ainsi, le canevas associé au premier exemple serait « BOOLEAN je veux GLOBAL ». En raison des concepts hiérarchisés, une distinction survient entre le **canevas primaire** (ex. LOCATION *please*) et les **canevas secondaires** (ex. CITY STATE *please*). À moins d'une indication contraire, le canevas primaire est utilisé par défaut. Cette notion de canevas sert à offrir une meilleure appréciation des résultats, notamment grâce au pourcentage de Canevas Hors-Vocabulaire (% CHV) entre les corpus d'entraînement et de test. La figure 3.1 offre quelques exemples supplémentaires illustrant cette terminologie.

***donnée :*** le premier août
***annotée :*** le DATE(DATE_ABS(DAY(ORDINAL(premier)) MONTH(août)))
***canevas primaire :*** le DATE
***canevas secondaires :*** le DATE_ABS MONTH / le DAY MONTH / le ORDINAL MONTH
***concepts :***

| Étiquette | Verbatim | Valeur | Hiérarchisé | Primaire |
|---|---|---|---|---|
| DATE | premier août | ????/08/01 | oui | primaire |
| DATE_ABS | premier août | ????/08/01 | oui | secondaire |
| DAY | premier | 01 | oui | secondaire |
| ORDINAL | premier | 01 | non | secondaire |
| MONTH | août | 08 | non | secondaire |

***donnée :*** payer 75 dollars à John Doe
**annotée :** payer AMOUNT(CARDINAL(75) dollars) à PERSON(FIRSTNAME(John) LASTNAME(Doe))
***canevas primaire :*** payer AMOUNT à PERSON
***canevas secondaires :*** payer CARDINAL à PERSON / payer CARDINAL à FIRST_NAME LAST_NAME / payer AMOUNT à FIRST_NAME LAST_NAME
***concepts :***

| Étiquette | Verbatim | Valeur | Hiérarchisé | Primaire |
|---|---|---|---|---|
| AMOUNT | 75 dollars | 75 | oui | primaire |
| CARDINAL | 75 | 75 | non | secondaire |
| PERSON | John Doe | DOE_John | oui | primaire |
| FIRSTNAME | John | John | non | secondaire |
| LASTNAME | Doe | DOE | non | secondaire |

**Figure 3.1** — Résumé de la terminologie employée

## 3.2 Description des données

Bien que les objectifs de recherche se définissent depuis deux langues arbitraires, les contraintes pratiques ont malheureusement forcé des choix plus restrictifs. Et même si le développement s'est effectué en aspirant à une solution extralinguistique, l'universalité des approches n'a pas pu être éprouvée. Ainsi, d'une part, l'anglais s'est imposé comme unique langue source. Un choix qui reste cohérent avec les usages courants. D'autre part, le français a été retenu comme langue cible. En ce qui a trait aux domaines étudiés, deux applications de dialogues avancés ont servi aux tests : la première émule un représentant d'une compagnie aérienne et la seconde, un représentant d'une compagnie financière.

### *3.2.1 Compagnie aérienne*

Les données « *Airline* », en anglais, sont directement issues d'une application de dialogues avancés développée par Nuance Communications[5] — une compagnie qui se spécialise dans le traitement et la synthèse de la parole. Des employés de Nuance étaient invités à échanger avec l'application, soit verbalement par téléphone ou textuellement par messagerie instantanée.

La tâche principale consistait à réserver un vol aller-retour entre deux destinations, puis de sélectionner des vols selon les choix correspondants. L'utilisateur devait obligatoirement préciser l'origine du vol, sa destination, la date et l'heure de départ. En sus, il lui fallait mentionner la durée du voyage, la date de retour ou il pouvait préciser qu'il ne voulait qu'un billet aller. Parmi ses autres options, notons qu'il pouvait spécifier le nombre de passagers, la classe de vol, des contraintes de prix, d'appareil ou de repas et poser plusieurs questions sur les différents choix qui s'offraient à lui. Naturellement, il pouvait s'exprimer sur chacun des points avec une grande latitude. Lorsqu'une heure lui était demandée, par exemple, « à trois heures », « en fin d'après-midi », « entre 8 h 15 et 10 h 35 », « après le souper » ou « aucune préférence » constituent autant de réponses acceptées par le système.

[5] www.nuance.com

Les données « Aviation », en français, ont été traduites manuellement depuis les données *Airline* et l'alignement des traductions a été respecté au moment de créer les corpus. Autrement dit, la traduction d'une donnée d'entraînement *Airline* se retrouve nécessairement dans l'ensemble d'entraînement *Aviation*, tout comme celle d'une donnée test se retrouve nécessairement dans l'ensemble test. La différence de comptes entre les deux corpus s'explique par la suppression des doublons après traduction. Une seule traduction française a été conservée pour chaque entrée anglaise.

| | AIRLINE | | | AVIATION | | |
|---|---|---|---|---|---|---|
| | **TOTAL** | **TRAIN** | **TEST** | **TOTAL** | **TRAIN** | **TEST** |
| **DONNÉES** | 2449 | 1835 | 614 | 2233 | 1693 | 540 |
| **ÉTIQUETTES** | 4163 | 3140 | 1023 | 3849 | 2938 | 911 |
| **CANEVAS** | 910 | 733 | 289 | 969 | 767 | 285 |
| **% CHV** | | | 29,80 % | | | 38,15 % |

**Tableau 3.1** — Description des corpus du domaine aéronautique

### *3.2.2 Compagnie financière*

Le corpus « *Insurance* », en anglais, provient également de données récoltées par Nuance auprès d'employés (92,8 %) et d'utilisateurs (7,2 %). Contrairement à la tâche d'*Airline*, où le dialogue était orienté vers un but clair (la réservation d'un vol), la navigation dans l'application d'*Insurance* est restée très flexible. En tout temps, l'utilisateur pouvait passer d'un service financier à l'autre afin d'effectuer ses opérations courantes (notamment connaître un solde, transférer de l'agent, contracter une assurance ou demander de l'information). Les services couvraient aussi bien les comptes personnels que l'épargne, la retraite, les assurances (véhicule et habitation) et les investissements.

À l'instar du domaine aérien, le corpus *Insurance* a été traduit manuellement en français pour donner le corpus « Assurance ». Les mêmes contraintes d'alignement et de suppression de doublons ont été observées.

| | INSURANCE | | | ASSURANCE | | |
|---|---|---|---|---|---|---|
| | TOTAL | TRAIN | TEST | TOTAL | TRAIN | TEST |
| **DONNÉES** | 2459 | 1844 | 615 | 2313 | 1715 | 598 |
| **ÉTIQUETTES** | 2663 | 2011 | 652 | 2426 | 1793 | 633 |
| **CANEVAS** | 1874 | 1454 | 538 | 1891 | 1459 | 551 |
| **% CHV** | | | 69,27 % | | | 70,40 % |

**Tableau 3.2** — Description des corpus du domaine financier

## 3.3 Description du système de traduction

Bien qu'il soit admis qu'un système de traduction s'améliore en y ajoutant des bitextes du domaine étudié [Koehn et Schroeder, 2007], l'existence de tels bitextes apparaît incertain au moment d'amorcer l'universalisation des ressources. C'est pourquoi le système de traduction retenu ici reste très générique dans un premier temps. Il utilise deux corpus publics sans lien particulier avec les domaines d'intérêt. L'un provient des débats parlementaires canadiens[6] (*Hansard*), et l'autre (*OpenSubtitles2011*) aligne des sous-titrages de films[7] [Tiedemann, 2009]. Éventuellement, les données admises par la problématique serviront à améliorer ce système initial (à cet effet, voir le chapitre 5).

| CORPUS | DOCS | PHRASES | TOKENS (EN) | TOKENS (FR) |
|---|---|---|---|---|
| **OpenSubtitles2011 (fr-ca)** | 24116 | 19,7 x $10^6$ | 119,0 x $10^6$ | 114,5 x $10^6$ |
| **Hansard** | 2 | 1,2 x $10^6$ | 19,8 x $10^6$ | 21,2 x $10^6$ |

**Tableau 3.3** — Description des corpus du système de traduction

## 3.4 Mesures de performance

Cette section brosse un portrait de différentes métriques ayant servi à comparer les approches. À cet égard, le $F_1$ (décrit ci-après) peut être perçu comme le principal critère discriminant, car la hiérarchie des solutions s'est établie suivant son verdict. Quant aux autres mesures, elles

[6] http://www.isi.edu/natural-language/download/hansard/

[7] http://opus.lingfil.uu.se

servaient d'abord à colliger de l'information sur les forces et faiblesses d'une expérience tierce, ou à baliser l'espace de recherche pour éviter que le $F_1$ n'augmente à la faveur de moyens détournés.

### *3.4.1 Précision, rappel et $F_1$*

La précision et le rappel sont des mesures éprouvées permettant de comparer un ensemble de réponses observées avec un ensemble de réponses attendues. La précision évalue la quantité de bruit, alors que le rappel quantifie l'exhaustivité des valeurs reçues. L'ordre est ici sans importance. Adaptée à la problématique, la précision peut être définie par la formule 3.1.

$$précision = \frac{nb\ concepts\ communs}{nb\ concepts\ trouvés} \tag{3.1}$$

De façon analogue, le rappel se calcule ainsi :

$$rappel = \frac{nb\ concepts\ communs}{nb\ concepts\ référence} \tag{3.2}$$

Ces deux mesures complémentaires sont souvent combinées à l'aide d'une moyenne harmonique, notée $F_1$ (parfois $F_{score}$), facilitant l'analyse.

$$F_1 = \frac{2 \cdot (précision \cdot rappel)}{précision + rappel} \tag{3.3}$$

Bien qu'il soit possible de considérer toute la hiérarchie d'étiquettes lors des calculs, les métriques fournies ci-après ne s'intéressent qu'aux concepts primaires. Une hiérarchie souvent déterministe — jumelée à une granularité d'étiquetage parfois élevée — a rendu ce choix nécessaire pour éviter que les résultats ne soient artificiellement dopés. En outre, les concepts primaires revêtent plus d'importance que leurs concepts constituants au moment de l'analyse dans un contexte réel, et l'ajustement reflète bien cette particularité. L'exemple de la

figure 3.2, sur la phrase « départ le 23 mai », permet de mieux saisir la pertinence de cette décision.

1. Calcul du $F_1$ sur toute la hiérarchie de concepts
(ref) **DEPARTURE_DATE**(**DATE**(**DATE_ABS**(**DAY**(23) **MONTH**(mai))))
(tst) **DATE**(**DATE_ABS**(**DAY**(23) **MONTH**(mai)))

$$\left.\begin{array}{l} précision = 4/4 = 100\% \\ rappel = 4/5 = 80\% \end{array}\right\} F_1 = 88,\bar{8}\%$$

2. Calcul du $F_1$ en ne considérant que les concepts primaires
(ref) **DEPARTURE_DATE**(23 mai)
(tst) **DATE**(23 mai)

$$\left.\begin{array}{l} précision = 0/1 = 0\% \\ rappel = 0/1 = 0\% \end{array}\right\} F_1 = 0\%$$

**Figure 3.2** — Différentes méthodes pour calculer le $F_1$

### *3.4.2 Score BLEU*

Le score BLEU (*Bilingual Evaluation Understudy*) sert à évaluer la qualité d'un système de traduction. Il évalue la proximité entre les traductions produites automatiquement et celles qu'un humain aurait choisies (il peut y avoir plus d'une référence par phrase). Son coût modeste de mise en oeuvre et sa forte corrélation avec l'appréciation qualitative en font une mesure très usitée [Papineni et al., 2002]. La valeur BLEU est comprise entre 0 et 1 : la borne inférieure indique une disparité totale, et la borne supérieure, une parfaite identité. Toutefois, une correspondance totale est peu probable dans la pratique, car plusieurs traducteurs humains fourniront immanquablement des références divergentes. BLEU s'obtient en calculant la précision au niveau des fragments de plusieurs $n$-grammes, puis en effectuant une moyenne pondérée des résultats. Un terme pénalisant les phrases trop courtes par rapport à la référence (BP) vient compléter l'équation.

$$\mathrm{BP} = min(1, e^{1-\frac{r}{c}}) \tag{3.4}$$

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{3.5}$$

Dans l'équation 3.4, $r$ correspond à la longueur de la référence et $c$, à celle du candidat. En ce qui a trait à l'équation 3.5, la somme s'effectue sur tous les $n$-grammes (avec $N$ l'ordre maximal considéré), $w_n$ sert à la pondération et $p_n$ reflète la précision de l'ordre courant. Cette précision diffère légèrement de sa définition usuelle, car un compte maximum pour chaque fragment est instauré selon la référence. À titre d'exemple, si une référence compte deux fois le mot « du », alors le nombre d'éléments en commun pour ce mot est borné à deux. Sans cette restriction, les candidats pourraient répéter indéfiniment « du » afin d'obtenir un nombre exagéré de fragments jugés communs. Enfin, la possibilité de considérer plus d'une référence rend le calcul du rappel hasardeux, ce qui explique que BLEU se définisse sans ce contrepoids habituel de la précision.

## 3.5 Évaluation

Les différentes données et mesures présentées dans ce chapitre serviront à hiérarchiser les différentes approches de la section suivante. Ainsi, les divers algorithmes de projection seront appliqués aux données d'entraînement en anglais des corpus *Airline* et *Insurance*, puis une grammaire sera constituée depuis les données cibles pré-annotées résultantes (au besoin, voir la section 2.4 pour l'algorithme de génération de grammaire). Cette grammaire servira alors à annoter automatiquement les données tests en français, et l'annotation retrouvée sera comparée avec l'annotation désirée (précision, rappel et F1). La métrique BLEU servira quant à elle lors de l'évaluation des ressources nécessaires de la section 5.6.

# CHAPITRE 4 — APPROCHES DÉVELOPPÉES

Le chapitre 4 s'attarde aux pistes de solution explorées afin de résoudre l'objectif principal, c'est-à-dire la projection de données annotées d'une langue vers une autre. D'abord, un algorithme n'utilisant aucune donnée source permet d'établir les valeurs de référence. Puis, quelques méthodes d'étiquetage automatique sont présentées. Ensuite, trois approches s'étant imposées par leurs résultats ou par la simplicité de leur implémentation sont décrites et analysées. Enfin, la dernière — et la principale — est explorée plus en détail, et son algorithme fait l'objet d'une optimisation rigoureuse.

## 4.1 Approche de référence

L'approche de référence consiste à ignorer bêtement toutes les données sources et à ne considérer que les données cibles. Ces données sont directement érigées en grammaire, à l'aide de l'algorithme naïf de la section 2.3. Ces grammaires peuvent alors être utilisées pour évaluer la performance sur les jeux de test, selon le nombre d'exemples cibles considérés, avec les résultats suivants en français :

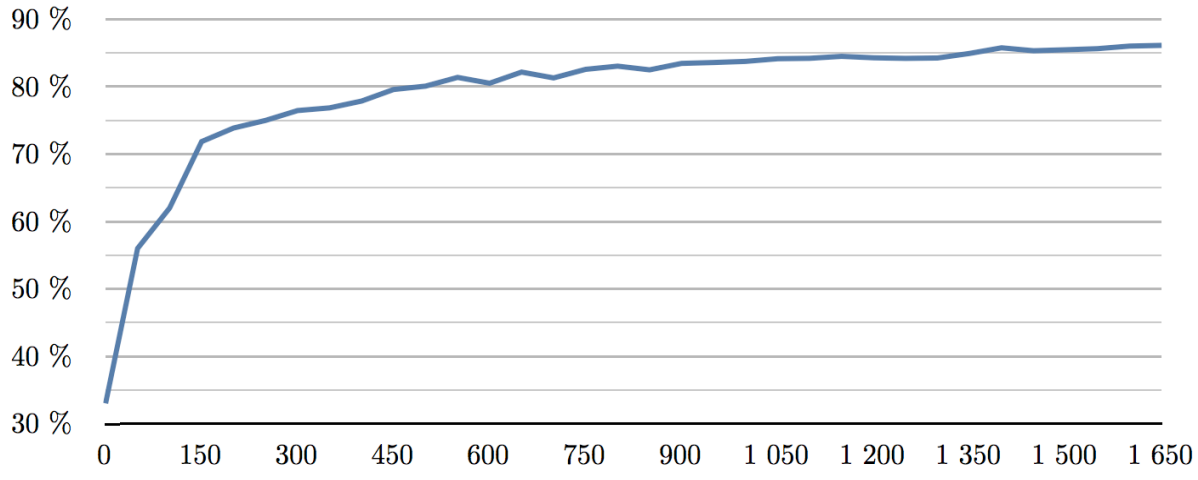



**Figure 4.1** — $F_1$ selon le nombre d'énoncés cibles, modèle de base (Aviation)

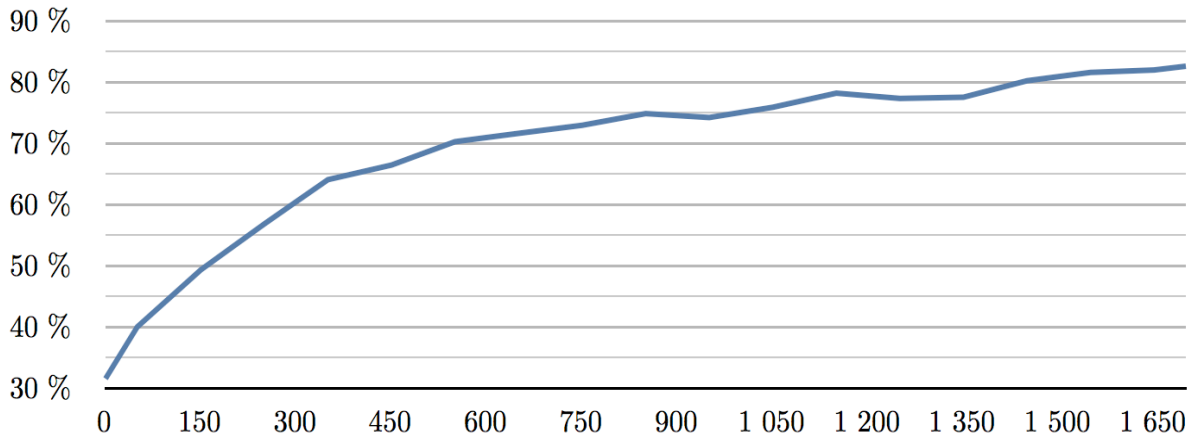



**Figure 4.2** — $F_1$ selon le nombre d'énoncés cibles, modèle de base (Assurance)

Ces résultats ont été obtenus en effectuant 5 itérations avec tirages aléatoires sur toutes les abscisses multiples de 50. Le pourcentage correspond au $F_1$, calculé depuis la précision et le rappel moyens. Les mêmes paramètres ont servi à tous les graphiques présentés dans ce mémoire. Leur description sera donc omise à l'avenir.

Seul le $F_1$ apparait sur la figure, car l'écart restreint entre la précision et le rappel pour ce modèle aurait rendu la confusion entre les trois droites inévitable. La différence moyenne entre les deux métriques n'est que de 1,15 %, avec un écart maximum de 2,22 %. La similitude s'explique par les grammaires de concepts énumérables : comme celles-ci sont intégrées au modèle, le nombre de concepts trouvés diffère très peu du nombre de concepts à trouver. Par exemple, « Lisbonne » qui se retrouve étiquetée comme une ville plutôt qu'en tant que destination dans la donnée « vers Lisbonne » n'entraîne pas de différence de compte ($nb\ concepts\ trouvés = nb\ concepts\ référence = 1$), d'où une précision égale au rappel.

Le modèle de base, par sa performance d'entrée non nulle, permet d'estimer un taux d'étiquettes triviales. Ces étiquettes sont celles qui s'accommodent d'une description hors contexte dans l'ontologie, et qui se présentent sans modificateur dans les données. Les formes

booléennes simples (« oui », « non ») constituent de bons exemples à ce propos. À l'inverse, la trivialité de 33,1 % des annotations d'Aviation et de 31,5 % des annotations Assurance implique que 66,9 % et 68,5 % de ces mêmes annotations nécessitent un algorithme plus élaboré qu'un simple « rechercher et remplacer ».

Autrement, l'allure générale des courbes montre que l'ajout de nouvelles données est généralement bénéfique, même si le gain ralentit au-delà de 500 exemples. Après, la redondance des canevas dans les données augmente le risque de rencontrer une règle déjà apprise, au détriment des cas qui n'ont pas encore été appris (ou qui ne peuvent tout simplement pas être appris par l'algorithme naïf, faute de données similaires dans l'ensemble d'entraînement).

## 4.2 Transposition de l'étiquetage

Avant de présenter les autres méthodes, un détour par les méthodes de transposition de l'étiquetage s'impose. Étant donné un exemple annoté dans la langue source et un candidat de traduction, il est crucial de pouvoir déduire automatiquement l'étiquetage cible. En la matière, les approches intuitives se sont révélées efficaces. Toutefois, puisque l'algorithme générant les grammaires se montre intransigeant envers les faux positifs (un faux positif survient lorsqu'un concept est associé à tort avec un verbatim donné), quelques modifications ont dû être effectuées et cette section en fait état.

### *4.2.1 Transposition en cascade*

La transposition en cascade utilise l'alignement d'étiquetage source et l'alignement de traduction pour déterminer automatiquement l'étiquetage cible. Ces deux alignements n'ajoutent que peu de calculs supplémentaires, ils sont aisément combinables et offrent de bons résultats dans le cas typique. La figure 4.3 fournit l'algorithme et un exemple simple d'étiquetage cible par cette méthode.

**Algorithme**

1. Pour tous les concepts de la donnée source

   $i_{min}$ = -1; $i_{max}$ = -1;

   1.1 Pour tous les mots dans le verbatim du concept $c$

   $a_i$ = alignement($c_i$); // indice dans la traduction du i$^e$ mot de $c$

   si ($a_i$ != -1 && ($i_{min}$ == -1) || $a_i < i_{min}$)) $i_{min} = a_i$;

   si ($a_i$ != -1 && ($i_{max}$ == -1) || $a_i > i_{max}$)) $i_{max} = a_i$;

   1.2 Si ($i_{min}$ != -1 && $i_{max}$ != -1) // concept $c$ trouvé dans la traduction tgt

   tgt.add($c$, $i_{min}$, $i_{max}$); // il commence à $i_{min}$ et finit à $i_{max}$.

**Illustration**

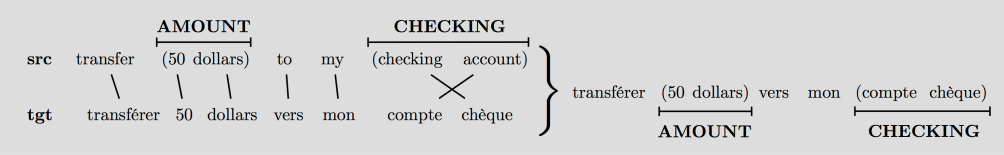



**Figure 4.3** — Transposition de l'étiquetage en cascade

L'exemple illustre le cas simple où AMOUNT est associé aux mots d'indices 1 et 2 et CHECKING est associé aux indices 5 et 6. L'alignement de traduction transpose ensuite ces indices vers ceux de la donnée cible (1 et 2 dans le premier cas, 6 et 5 dans le second), pour retrouver les concepts pertinents.

Malheureusement, cette transposition est sujette à deux lacunes majeures. D'abord, la cohérence entre l'étiquetage source et l'alignement de traduction n'est pas assurée. La déduction d'un étiquetage cible est alors hasardeuse, et souvent peu fiable. Ensuite, cette façon de faire perd de sa robustesse en raison de la contrainte IBM (au besoin, revoir la section 1.5), car lorsqu'un concept est absent d'une traduction, cette contrainte favorise une identification erronée. La figure 4.4 illustre ces défauts.

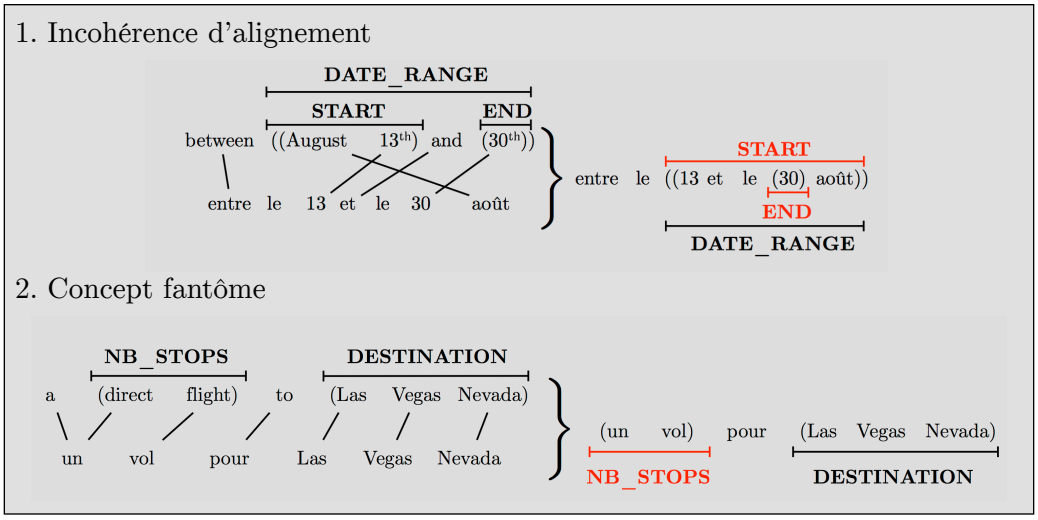



**Figure 4.4** — Ratées de la transposition en cascade

Il est possible de remarquer, dans le premier cas, que l'application directe de l'algorithme entraine le chevauchement de START et de END, alors que « un vol » devient à tort synonyme de « vol direct » dans le second. Pour ces raisons, la transposition en cascade a été délaissée au profit de celle par recherche du verbatim. Même si cette dernière peut paraître plus fastidieuse, elle s'affranchit de ces deux imperfections. L'expérimentation a en outre confirmé l'adéquation de cette préférence.

### *4.2.2 Transposition par recherche du verbatim*

La transposition d'un étiquetage par recherche du verbatim utilise des traductions locales lors de l'étiquetage cible. En plus de la phrase elle-même, le verbatim de chaque concept est traduit indépendamment et ces traductions sont ensuite recherchées dans la traduction principale pour devenir la donnée cible annotée. Comme il existe plusieurs traductions candidates pour chaque verbatim, le nombre de traductions considérées avant de déclarer un concept absent fait l'objet d'un paramètre particulier ($T_{max}$). La figure 4.5 permet de mieux comprendre l'étiquetage par cette méthode.

**Algorithme**

1. Pour tous les concepts de la donnée source

trads = {trad$_0(c)$, trad$_1(c)$, ...} // candidats de traduction pour le concept $c$

1.1 Pour $i = 0$ jusqu'à $T_{max}$

1.1.1 si tgt.contains(trads.at($i$)) // concept trouvé dans la traduction tgt

$i_{min}$ = tgt.indexOf(trads.at($i$));

$i_{max} = i_{min}$ + nbWords(trads.at($i$)) − 1;

tgt.add($c$, $i_{min}$, $i_{max}$);

sortir;

**Illustration**

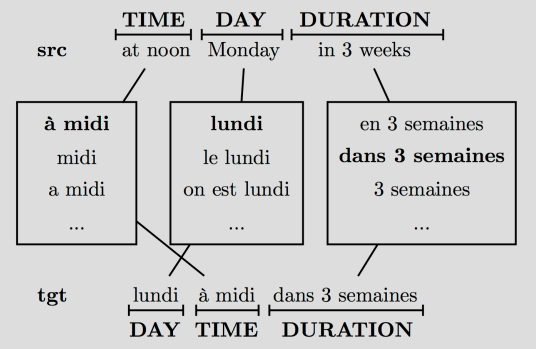



**Figure 4.5** — Transposition de l'étiquetage par recherche du verbatim

L'exemple de la figure 4.5 montre comment les trois concepts initiaux sont retrouvés en utilisant des traductions locales. Notons que le même exemple, avec $T_{max} = 0$, aurait empêché l'identification du concept DURATION, car sa correspondance de traduction se situe à l'indice 1 (typiquement, $T_{max} = 2$).

Si cette méthode pallie les failles de l'approche en cascade, elle se trouve à la merci des ambiguïtés polysémiques. Par exemple, elle ne pourrait étiqueter adéquatement le lieu et la personne dans la donnée fictive « Je vais à **Victoria** en Colombie-Britannique avec **Victoria** ». Dans la mesure où cette situation reste marginale (du moins, selon les données disponibles), elle demeure préférable à l'autre approche. Certaines heuristiques permettent en outre d'amoindrir le problème (par exemple, respecter l'ordre d'apparition des mots entre la

source et la traduction pour certaines paires de langues). Finalement, une taille minimale sur les mots à annoter ($|\text{mot}| \geq 3$) améliore généralement la performance.

### 4.3 Approche par traducteur de canevas

L'approche par traducteur de canevas cherche à adapter le système de traduction aux spécificités des données annotées. La traduction statistique reposant sur des probabilités, une simplification intelligente des données d'entraînement selon la classe de mots permet parfois d'obtenir de meilleurs résultats globaux. Cela se vérifie particulièrement dans le cas d'entités énumérables, par exemple en remplaçant toutes les verbatims de pays (« Canada », « Mexique », « Australie »...) par le terme générique PAYS. L'algorithme de traduction devient alors moins sensible aux faibles fréquences et une table de traduction suffit à retrouver le PAYS sous-entendu d'une étiquette. De ce point de vue, la tâche d'intérêt semble toute désignée pour bénéficier des avantages de cette méthode.

Le *modus operandi* de l'approche débute par la transformation des bitextes : l'ontologie est utilisée parallèlement avec les grammaires disponibles (celles des concepts énumérables — dates, heures, lieux, etc.) pour transformer chaque phrase d'entraînement en canevas associé (ex. « Vancouver 2010 Olympics » et « Jeux olympiques de Vancouver 2010 » deviennent « CITY YEAR Olympics » et « Jeux olympiques de CITY YEAR »). Un système de traduction est ensuite entraîné avec ces données transformées. Celui-ci peut alors servir de deux façons différentes, lesquelles sont illustrées après leurs définitions, aux figures 4.6 et 4.7.

**a) variation déductive**

Au moment de traduire une donnée annotée, celle-ci est également remplacée par son canevas secondaire (celui qui peut être obtenu à l'aide des concepts énumérables uniquement). Le système produit alors un canevas équivalent en langue cible, duquel sont déduits les concepts englobants. Dans ce cas, la transposition de l'étiquetage se fait en cascade pour mieux cibler les termes identiques à annoter différemment (ex. plusieurs ACCOUNT dans une phrase, l'un se spécialisant en CHECKING et l'autre en SAVINGS).

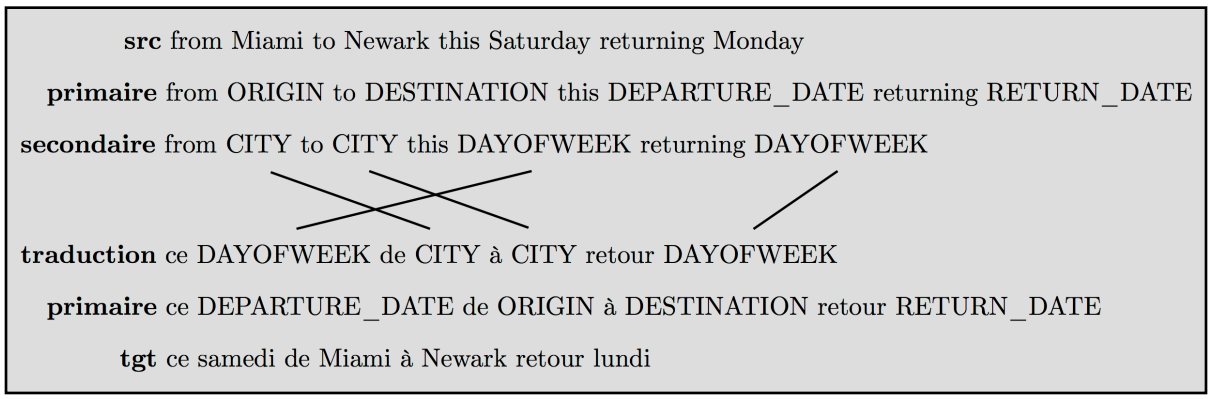



**Figure 4.6** — Approche par traduction de canevas, variation déductive

Dans cet exemple, le canevas secondaire « from CITY to CITY this DAYOFWEEK returning DAYOFWEEK » est traduit par « ce DAYOFWEEK de CITY à CITY retour DAYOFWEEK ».

En remontant l'arbre de correspondance, il est possible de savoir que le premier DAYOFWEEK de cette traduction est associé au premier DAYOFWEEK du canevas secondaire source, lequel a pour antécédents « DEPARTURE_DATE » et « Saturday » (dans le canevas primaire et dans la donnée source respectivement). Ainsi, changer DAYOFWEEK pour DEPARTURE_DATE permet d'obtenir le canevas primaire cible, alors que la traduction de « Saturday » suffit pour former la donnée cible (tgt). Chaque concept du canevas secondaire cible subit le même traitement.

**b) variation restrictive**

Le système de traduction annoté sert en parallèle avec le système de traduction initial. Au moment de traduire une donnée, son canevas secondaire est traduit également. Puis, la donnée cible résultante est annotée automatiquement, mais elle est immédiatement écartée en cas d'incohérence avec le canevas cible estimé. Autrement dit, la donnée retenue pour la suite est la première qui se conforme avec une traduction de canevas attendue, parmi les $T_{max}$ premières ($T_{max}$ étant fixé arbitrairement).

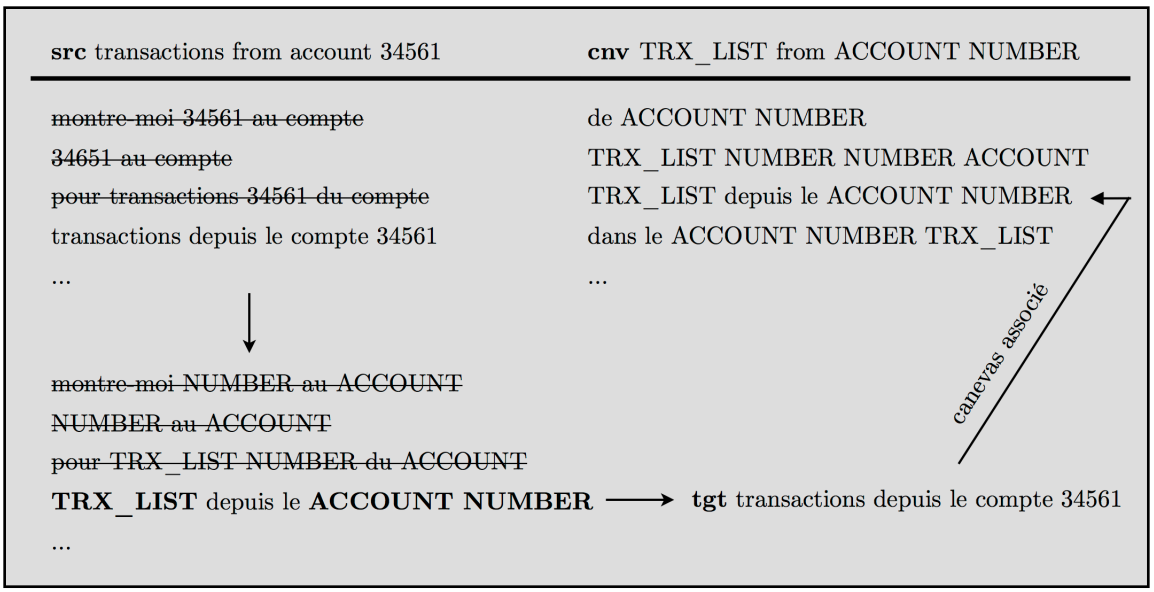



**Figure 4.7** — Approche par traduction de canevas, variation restrictive

Dans l'exemple, la donnée source (src) et son canevas (cnv) sont traduits séparément. Ensuite, chaque traduction candidate de src est annotée en utilisant les grammaires de concepts énumérables disponibles. Puis, la première intersection entre un candidat ainsi annoté et les traductions possibles de cnv est retenue. L'étiquetage primaire est finalement déterminé par recherche du verbatim ou par cascade.

Malheureusement, ni l'une ni l'autre des variations n'offre de résultats satisfaisants et plusieurs raisons peuvent justifier ces échecs. D'abord, l'absence d'une grammaire cible fiable limite les concepts pouvant servir à étiqueter les bitextes. Les étiquettes avancées, qui requièrent un contexte, sont donc inexistantes lors de l'entraînement (ex. MONTH est envisageable, mais pas RETURN_DATE). En outre, comme les données du système de traduction ne sont pas en phase avec les grammaires de la tâche, plusieurs faux positifs viennent plomber la qualité des canevas résultants. La polysémie contribue aussi à la difficulté et, à défaut d'une solution pratique, elle oblige la prise de décisions au jugé ou l'utilisation de l'étiquetage par cascade, avec ses travers (fin de la sous-section 4.2.1). Les cooccurrences sont également problématiques, car plusieurs entités identiques en viennent à se disputer un

même antécédent ou vice versa (ex. lorsque « from CITY to CITY » est traduit par « de CITY »).

### 4.4 Approche par modèle de langue prédictif

Dans la même veine que l'approche par traduction de canevas, le modèle de langue prédictif cherche à exploiter la redondance des étiquettes. Il s'agit en fait d'une approche inspirée d'un modèle de langue $n$-grammes typique (section 1.5).

Supposons l'existence d'un modèle de langue fiable pour les données cibles annotées. Par définition, ce modèle permet de déterminer si un canevas primaire est plus probable qu'un autre. Le traitement d'une nouvelle donnée cible par cette approche se fait alors ainsi. D'abord, une donnée est étiquetée avec les grammaires de concepts énumérables. Comme la donnée fait partie du domaine d'intérêt, peu de faux positifs devraient survenir. Les concepts avancés sont naturellement absents, mais l'ontologie admet un nombre fini de promotions acceptables et tous ces cas de figure sont générés (en identifiant les concepts comme « terminaux[8] » ou « non-terminaux[9] », le risque d'explosion factorielle reste faible). Il ne reste alors qu'à calculer l'adéquation des différents canevas avec le modèle de langue pour découvrir l'étiquetage à conserver. L'exemple de la figure 4.8 devrait aider la compréhension.

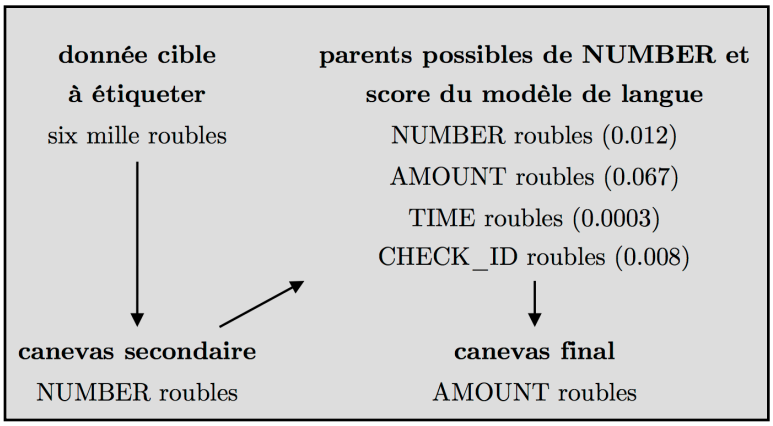



**Figure 4.8** — Approche par modèle de langue prédictif

[8] Concept qui figure au moins une fois dans un canevas primaire, selon les données sources.

[9] Concept qui ne figure jamais dans un canevas primaire, selon les données sources.

Dans cet exemple, NUMBER admet 3 parents terminaux (AMOUNT, TIME et CHECK_ID), en plus d'être lui-même un concept terminal. Le modèle de langue attribue un score d'affinité à ces quatre dérivés, et détermine que « AMOUNT roubles » est l'option la plus probable.

Reste un problème majeur : comment obtenir un tel modèle de langue. Pour ce faire, il faudrait parvenir à entraîner un modèle alors qu'aucune donnée cible n'est encore disponible. Les seules données connues sont celles des bitextes, et l'approche précédente a démontré qu'elles différaient trop du domaine d'intérêt. Une nouvelle utilisation des modèles de langue permet toutefois de contourner ce problème. Il suffit d'entraîner un modèle de langue source avec les données annotées, puis de l'utiliser sur toutes les phrases sources du bitexte. Les phrases ayant un score au-delà d'un certain seuil sont celles qui s'apparentent le plus au domaine d'intérêt, et seront les seules considérées par la suite. Le bitexte tronqué sert finalement à déduire le modèle de langue cible. Pour ce faire, les phrases sources de ce bitexte sont à nouveau annotées, avec les concepts avancés cette fois-ci, et l'étiquetage cible est déduit par recherche du verbatim. La figure 4.9 illustre visuellement ce procédé.

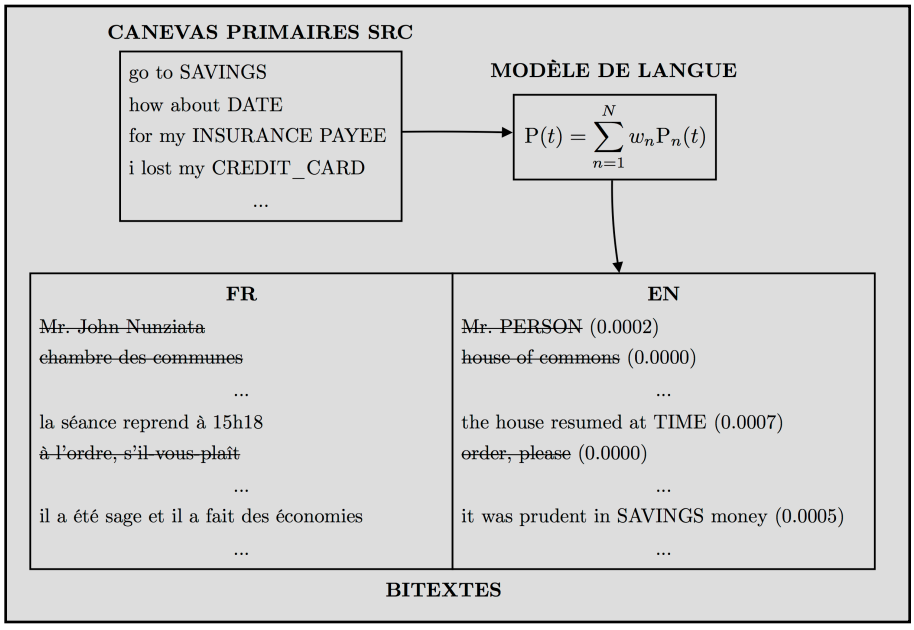



**Figure 4.9** — Identification des données d'un domaine dans les bitextes

L'exemple démontre comment les canevas primaires sources servent à identifier les données pertinentes du bitextes (celles qui s'approchent du domaine d'intérêt). Comme le bitexte comprend un alignement avec les données cibles, celles-ci peuvent aussi être transformées en canevas primaires (au besoin, revoir les techniques de la section 4.2). Une fois ces canevas primaires cibles obtenus, il ne reste qu'à entraîner un nouveau modèle de langue — celui qu'on cherchait à obtenir.

Encore une fois, les résultats ne sont guère impressionnants, et l'évaluation du modèle de langue cible semble être en cause. Les nombreuses étapes qui mènent à son estimation sont jonchées d'imperfections. En particulier, le nombre restreints de canevas primaires pour estimer le modèle source (plus ou moins 2000 exemples, ce qui oblige un modèle unigramme ou bigramme au maximum) et la réutilisation des bitextes, qui servent déjà à entraîner le système de traduction. Ainsi, les phrases du bitexte qui ont été retenues pour entraîner le modèle de langue cible servent à la fois à traduire une donnée et à l'évaluer, ce qui n'est jamais souhaitable. Finalement, même avec un modèle de langue fiable, la possibilité que plusieurs concepts soient englobés par un concept parent unique n'est pas clairement exprimée dans l'ontologie, et fait aussi défaut à la méthode.

Malgré tout, l'idée de cibler les données des bitextes proches d'un domaine apparaissant intuitivement pertinente, elle a été réutilisée pour tenter d'améliorer l'approche par traduction de canevas (avec succès négligeable) et, plus généralement, pour améliorer la qualité du système de traduction (voir la section 5.6).

## 4.5 Approche par traduction et étiquetage

Sous l'hypothèse que les premiers candidats de traduction sont généralement fiables, il existe un algorithme direct permettant de générer des données cibles annotées. En effet, il suffit de traduire tous les exemples sources de l'ensemble d'entraînement, de retenir les $n$ premiers candidats cibles pour chacun, avant de transposer leurs étiquettes en recherchant les verbatims. Ces nouvelles données annotées peuvent alors être soumises à l'algorithme de

génération de grammaire (section 2.3). Fidèle à son processus, cette approche est dite « par traduction et étiquetage ». La figure 4.10 résume l'application la plus directe de cette approche, soit le cas où $n = 1$ (chaque exemple source produit ainsi un unique exemple cible annoté).

| **SRC** | **1er CANDIDAT DE TRADUCTION** | **ÉTIQUETAGE (recherche du verbatim)** |
|---|---|---|
| reimbursement for my windshield | remboursement pour mon pare-brise | CLAIM pour mon WINDSHIELD |
| show interest rate | des taux d'intérêts | des INTEREST_RATE |
| transfer 40$ to checking | transfert de 40$ à vérifier | TRANSFERT de AMOUNT à vérifier |
| | ... | |

**Figure 4.10** — Illustration de la méthode par traduction et étiquetage

La méthode montre des performances encourageantes, et, selon les données disponibles, elle s'améliore en respectant les restrictions suivantes : d'abord, les données cibles doivent contenir le même ensemble d'étiquettes que leur donnée source associée. En particulier, si la traduction d'un verbatim est introuvable dans la donnée cible, celle-ci doit être immédiatement écartée. Dans l'exemple de la figure 4.10, cela se traduirait par un autre candidat de traduction pour « transfer 40 $ to checking », car le concept CHECKING associé à « checking » de la donnée source n'a pas d'équivalent cible, une fois l'étiquetage effectué.

Ensuite, seul le premier candidat de traduction ayant toutes les étiquettes requises doit être retenu. Considérer plus d'une traduction nuit à la performance, car le bruit ajouté supplante la robustesse ainsi obtenue. Enfin, limiter le nombre de candidats potentiels aux $n = 5$ premières traductions semble opportun, mais les différences de score laissent croire que ce paramètre pourrait s'ajuster différemment selon les jeux de test.

Une étude des erreurs subsistantes met toutefois en lumière une grande dépendance de cette méthode à la qualité du système de traduction. La corrélation est inévitable, mais la nature de la tâche magnifie dramatiquement chaque annotation erronée. Rappelons que dans la logique déterministe d'une grammaire, chaque règle est valable et utilisable. Ainsi, en fournissant un seul exemple fautif — disons, un qui prétend que « mais » est une ville —, c'est l'exactitude

de toutes les phrases futures contenant l'erreur qui est compromise. Même s'il est vrai que l'algorithme générant les grammaires retenu pour la thèse (section 2.3) contribue significativement au problème, en retenant bêtement tout ce qui lui est présenté, il apparait important de trouver un moyen d'identifier : soit les données à ne pas traduire, soit les données potentiellement mal traduites ou soit les données potentiellement mal annotées.

Le système de traduction fournissant d'emblée un score pour chaque candidat, quelques recherches ont été entreprises afin d'en déduire un niveau de confiance. Malheureusement, les chiffres bruts du traducteur ne servent qu'à ordonner les divers candidats et n'ont aucun sens autrement. La différence entre les scores ne peut pas non plus servir à établir de seuil quant au nombre de candidats à retenir.

En fait, évaluer la qualité de la traduction d'une SMT (ou, plus largement, d'un système de traduction quelconque) sans référence est un secteur de recherche à part entière, avec sa littérature dédiée. À ce propos, mentionnons le travail de [Raybaud et al., 2009], où différents indices sont efficacement combinés. Cependant, les approches existantes restent à parfaire et elles nécessitent un lot supplémentaire de données alignées. Autant de raisons qui déséquilibrent le gain estimé vis-à-vis des modifications à apporter. C'est pourquoi l'évaluation de la qualité s'est détournée des traductions elles-mêmes pour se concentrer plutôt sur les canevas qui en dérivent (l'approche suivante, par classification et votes).

## 4.6 Approche proposée, par classification et votes

L'approche par classification et votes est une méthode originale qui conserve les avantages de la méthode par traduction et étiquetage. Son ajout principal consiste à utiliser une forme de partitionnement pour favoriser l'émergence de bons candidats de traduction. Elle s'effectue en deux temps : vient d'abord l'identification des canevas pertinents, puis la génération des annotations.

### *4.6.1 Identification des canevas pertinents*

L'objectif de cette étape est de distinguer les canevas cibles pertinents de ceux qui sont incomplets ou erronés. Pour se faire, chaque candidat de traduction est transformé en canevas, lequel est immédiatement écarté s'il y manque un concept source. Ensuite, les canevas restants reçoivent un score selon leur rang de traduction (linéaire dans le modèle de base). Toutes les données sources sont considérées, de sorte que plusieurs canevas sources finissent par se répéter. Le cas échéant, les scores des canevas cibles sont additionnés entre eux.

Une fois toutes les données examinées, les scores finaux permettent d'établir une hiérarchie de pertinence pour tous les canevas cibles rencontrés, étant donné un canevas source. Cette description se formalise mathématiquement par l'équation 4.1, et s'illustre par l'exemple du tableau 4.1 :

$$score(c_{cible}|c_{src}) = \sum_{c_{src}} \left(1.0 - \frac{rang(c_{cible})}{r_{max}}\right) \tag{4.1}$$

Dans cette équation, le rang commence avec un terme d'indice 0, $r_{max}$ réfère au rang maximal considéré (typiquement, le nombre de candidats de traduction renvoyés) et la lettre $c$ est employée comme abréviation de « canevas ».

Il est à noter que l'absence d'un canevas cible ne modifie pas le score (du moins, pas dans cette fonction de pointage), ce qui se traduit dans l'équation par $rang(c_{cible}) = r_{max}$ si $c_{cible}$ n'existe pas. D'autres fonctions de pointage ont été testées, et les résultats font l'objet de la sous-section 4.6.3.

| | I'd like to go from ORIGIN to DESTINATION ($r_{max} = 10$) | | | | | |
|---|---|---|---|---|---|---|
| **ORIGIN** | **Boston** | **Montreal** | **Denver** | **Montreal** | **Chicago Illinois** | **TOTAL** |
| **DESTINATION** | **New York** | **Quebec city** | **Nevada** | **New York** | **London UK** | |
| **je voudrais aller de ORIGIN à DESTINATION** | 1 (+0,9) | 1 (+0,9) | 6 (+0,4) | 0 (+1,0) | 3 (+0,7) | **3,9** |
| **je voudrais aller de ORIGIN pour DESTINATION** | 5 (+0,5) | 5 (+0,5) | 1 (+0,9) | 9 (+0,1) | 2 (+0,8) | **2,8** |
| **je voudrais y aller de ORIGIN à DESTINATION** | 8 (+0,2) | 2 (+0,8) | N/A | 7 (+0,3) | N/A | **1,3** |
| **je voudrais aller à ORIGIN de DESTINATION** | N/A | N/A | 8 (+0,2) | 3 (+0,7) | 1 (+0,9) | **1,8** |
| **je voudrais aller à ORIGIN à DESTINATION** | 0 (+1,0) | 0 (+1,0) | N/A | 8 (+0,2) | N/A | **2,2** |
| **je voudrais aller à ORIGIN pour DESTINATION** | 3 (+0,7) | N/A | N/A | 1 (+0,9) | 9 (+0,1) | **1,7** |

**Tableau 4.1** — Évaluation des canevas cibles ayant un même canevas source

Quant au tableau, le canevas source est le même dans tous les cas : « *I'd like to go from* ORIGIN *to* DESTINATION ». Le nombre principal correspond au rang d'apparition d'un canevas cible pour la traduction d'un exemple où ORIGIN et DESTINATION sont remplacés par les verbatims spécifiés en en-tête. La variation du score associé est indiquée entre parenthèses.

Prenons la première case du tableau : elle indique que « je voudrais aller de Boston à New York » figurait au rang 1 des candidats de traduction pour « *I'd like to go from Boston to New York* » (rappelons que le rang commence avec le terme 0). Son canevas cible associé est « je voudrais aller de ORIGIN à DESTINATION ». Le score de ce canevas s'incrémente donc de 0,9, en vertu de l'équation 4.1. Une fois tous les exemples du tableau considérés, ce canevas finit avec un score de 3,9, ce qui lui vaut le premier rang général (il est donc considéré comme la meilleure traduction du canevas source).

Seulement six canevas figurent dans le tableau : les trois premiers sont arbitrairement considérés comme acceptables et les trois autres, indésirables. Ici, le 2e canevas parvient à se hisser devant le 5e, même si ce dernier figure au premier rang à deux reprises.

### *4.6.2 Génération des annotations*

Une fois les canevas cibles ordonnés, la génération des annotations se fait assez directement. Il suffit de fixer des critères discriminant les canevas valides de ceux à ignorer, puis de garder tous les candidats de traduction se conformant à un canevas valide pour engendrer la grammaire. Toujours en s'appuyant sur le tableau 4.1, par exemple, « je voudrais aller de Boston à New York » serait gardé uniquement si son canevas (je voudrais aller de ORIGIN à DESTINATION) est considéré valide.

En ce qui a trait aux critères de validité, un nombre fixe de candidats par classe (c'est-à-dire considérer valides les $n$ premiers canevas cibles après réordonnancement) suffit presque, mais les cas simplistes s'y accommodent mal. En effet, lorsqu'un canevas cible unique se distingue nettement des autres (ex. le canevas anglais « BOOLEAN » qui se traduit invariablement par le canevas français « BOOLEAN »), en considérer davantage revient à augmenter le bruit. C'est pourquoi la sélection se fait selon deux critères : les $n$ premiers canevas cibles sont valides, sauf si leur ratio de score comparé au meilleur candidat est plus petit qu'un seuil $\lambda$. Ce ratio s'obtient aisément par la formule 4.2 :

$$ratio(c^i_{cible}|c_{src}) = \frac{score(c^i_{cible}|c_{src})}{score(c^0_{cible}|c_{src})} \tag{4.2}$$

Naturellement, ce ratio est tributaire de la fonction de pointage utilisée et le seuil $\lambda$ doit être ajusté en conséquence.

Une première discussion qui s'impose, après avoir pris connaissance de cet algorithme, concerne le taux de répétition effectif des canevas sources. Après tout, cette méthode doit

absolument contenir des doublons de canevas, sans quoi elle est indissociable de l'approche précédente, par traduction et étiquetage (section 4.5). Dans les données d'entraînement, le taux d'hapax des canevas se fixe à 22,6 % pour *Airline* et 48,5 % pour *Insurance*. C'est donc dire que 77,4 % des données *Airline* et 51,5 % des données *Insurance* ont au moins une donnée parente. En considérant intelligemment les concaténations de canevas et les hiérarchies de concepts, ces taux passent à 84,8 % et 63,2 %. L'algorithme altère donc le traitement de la grande majorité des données. Comme une analyse approfondie est prévue au chapitre 5, il n'est pas opportun de discuter immédiatement des balises qui assurent un succès à cette méthode. Toutefois, pour en avoir une bonne intuition, il convient de se rappeler qu'un même concept peut s'exprimer de nombreuses manières (ex. « demain », « 3 mars » et « deuxième lundi de décembre » qui sont tous des dates). C'est ce qui explique que des phrases pointant vers un même canevas n'ont pas nécessairement des candidats de traduction voisins.

### *4.6.3 Fonctions de pointage*

Si le réordonnancement des canevas s'accommode bien d'une fonction linéaire, il convient de se questionner quant à l'optimalité de ce système de pointage. Après tout, l'observation des résultats de traduction montre que la qualité des candidats semble décroître rapidement, et qu'une fonction rendant mieux compte de cette réalité améliorerait les résultats. Dans cette optique, quatre autres fonctions de pointage ont été implémentées, lesquelles sont présentées ci-après avec leur formule respective.

**a) fonction inverse**

Dans sa forme la plus simple, il suffit d'aligner l'hyperbole sur le rang d'indice 0 en divisant 1 par le rang augmenté. Cependant, une forme plus générale introduisant le paramètre $\alpha$ permet une plus grande maniabilité en ce qui a trait à l'importance relative des premiers rangs.

$$score(c_{cible}|c_{src}) = \sum_{c_{src}} \left( \frac{1.0}{rang(c_{cible}) + \alpha} \right) \ (\alpha > 0) \tag{4.3}$$

**b) fonction escalier**

Cette fonction considère les candidats d'un même palier sur un pied d'égalité. Dans l'équation 4.4, les paramètres $\alpha$ et $\beta$ contrôlent respectivement l'écart de valeur des contremarches et le nombre de candidats par palier.

$$score(c_{cible}|c_{src}) = \sum_{c_{src}} \max\left(0, -\alpha\left\lfloor \frac{rang(c_{cible})}{\beta} \right\rfloor + 1\right) \quad (0 < \alpha \leq 1,\, \beta \in \mathbb{N}^*) \quad \textbf{(4.4)}$$

**c) fonction constante**

La fonction constante ne prétend pas être une amélioration par rapport à la fonction linéaire, mais elle aide à mieux comprendre l'algorithme et c'est pourquoi elle figure ici également. Comme elle se borne à compter la fréquence des canevas cibles, sa définition formelle est omise.

**d) fonction charnière**

La fonction charnière est une fonction par parties, d'abord constante puis linéairement décroissante. Par rapport à ses deux constituantes, elle n'introduit qu'un nouveau paramètre servant à marquer le rang frontière ($k \in \mathbb{N}$). À nouveau, sa formule explicite est omise en raison de sa simplicité.

Lorsque toutes les données sources d'entraînement sont utilisées, sans aucune donnée cible, la maximisation des fonctions s'établit comme suit :

| | AVIATION | | | | ASSURANCE | | | |
|---|---|---|---|---|---|---|---|---|
| **FONCTION** | **PARAMS OPTIMAUX** | **P** | **R** | **$F_1$** | **PARAMS OPTIMAUX** | **P** | **R** | **$F_1$** |
| **Linéaire** | | 74,9 % | 81,1 % | 77,9 % | | 64,3 % | 70,2 % | 67,1 % |
| **Inverse** | $\alpha = 1{,}75$ | 73,3 % | 79,4 % | 76,2 % | $\alpha = 1{,}75$ | 63,8 % | 69,0 % | 66,3 % |
| **Escalier** | $\alpha = 0{,}1$ $\beta = 2$ | 74,4 % | 80,6 % | 77,4 % | $\alpha = 0{,}2$ $\beta = 2$ | 65,4 % | 69,9 % | 67,6 % |
| **Constante** | | 71,1 % | 77,4 % | 74,1 % | | 59,8 % | 65,3 % | 62,4 % |
| **Charnière** | $k = 2$ | 73,3 % | 79,1 % | 76,1 % | $k = 3$ | 65,5 % | 70,1 % | 67,7 % |

**Tableau 4.2** — Performance de différentes fonctions de pointage

Pour assurer la pertinence des résultats, les paramètres de fonction qui entraîneraient une simplification n'ont pas été pris en compte. C'est la raison pour laquelle, par exemple, le $F_1$ de la fonction charnière (Aviation) s'établit à 76,1 % pour $k = 2$, même s'il aurait pu être de 77,9 % en simulant une fonction linéaire avec $k = 0$. Il va sans dire que l'optimalité des paramètres concerne uniquement les valeurs explorées (environ 10 valeurs par paramètre).

Même si la fonction retenue semble avoir une incidence restreinte, la sous-optimalité de la fonction constante prouve qu'il est avantageux d'exploiter le rang de traduction retourné par la SMT. Après tout, la fonction constante est la seule qui considère tous les candidats sur un pied d'égalité, ce qui se traduit par un retard de ± 4 % sur les maximums locaux.

### *4.6.4 Surgénération de données*

Même si les canevas orphelins ne bénéficient pas de la méthode actuelle, il est aisé de leur créer des copies proches. Après tout, les grammaires de concepts énumérables, l'étiquetage et le système de traduction sont autant d'outils qui peuvent être mis à profit en ce sens. La question est de savoir si ces données artificielles suffiront à améliorer la performance. Dans l'affirmative, rien n'oblige à limiter cette surgénération aux singletons : tous les exemples y gagneraient peut-être en robustesse. Afin d'évaluer la justesse de ce raisonnement, trois mécanismes de surgénération ont été évalués.

**a) surgénération depuis les grammaires de l'ontologie**

Tel que mentionné au moment d'introduire les ontologies (section 1.4), plusieurs concepts peuvent s'exprimer — du moins en partie — sous forme de liste, et ces listes sont généralement accessibles. Une consultation directe permet donc de remplacer une instance énumérable par une autre de la même famille, créant ainsi une donnée artificielle annotée (ex. dans « *Transfer $25 to John's account* », « *John* » est remplacé par « *Christine* » pour créer « *Transfer $25 to Christine's account* »).

Un problème majeur survient toutefois en raison de l'exhaustivité de l'ontologie. Afin de maximiser la reconnaissance, les listes contiennent un nombre impressionnant de verbatims, souvent sans pondération. Ainsi, s'il serait souhaitable de remplacer « *Calgary* » par « *Melbourne* », « *Shanghai* » ou « *Oslo* », les probabilités sont plus grandes de tomber sur un nom de ville obscur, comme « *Ambanja* » ou « *Pickle Lake* ». Ces noms de ville, inconnus du système de traduction, augmentent le bruit plutôt que la robustesse. Et à moins de connaissances a priori, le tri entre les différents substituts ne peut pas s'automatiser.

**b) surgénération depuis les données d'entraînement**

Il est toutefois possible de déduire un indice de pertinence pour certains substituts, en ajoutant l'hypothèse que les données d'entraînement sont assez représentatives. En effet, il suffit de compter la fréquence de chaque verbatim à travers les données annotées pour effectuer une surgénération intelligente. C'est ainsi que, par exemple, la donnée « *how about Sunday ?* » peut être créée lorsque les données « *how about Monday ?* » et « *returning on Sunday* » lui fournissent respectivement un canevas et un verbatim. Plus généralement, si « *Sunday* » est un concept DAYOFWEEK fréquent à l'entraînement, il sera utilisé dans tous les canevas incluant DAYOFWEEK pour créer autant de nouvelles données (si, bien évidemment, le verbatim remplacé n'est pas lui-même « *Sunday* »).

L'approche est prometteuse, même si les résultats ne sont pas tout à fait concluants. Le $F_1$ augmente imperceptiblement de 77,9 % à 78,2 % sur le corpus Aviation et de 67,1 % à 67,7 % sur le corpus Assurance. La différence n'est pas assez marquée pour être significative, malgré les tentatives d'optimisation des paramètres (notamment, le nombre d'exemples à générer et le seuil sur la fréquence des verbatims). La faiblesse du gain peut s'expliquer en partie par la taille restreinte du corpus d'entraînement : les exemples uniques ont peu de chance d'avoir un parent proche dans les jeux de test, ce qui minimise leur contribution.

Devant cette impasse, c'est plutôt une expérience modifiée qui vient valider l'utilité de la méthode : deux nouveaux ensembles d'entraînement ont été créés ($Airline_{hpx}$ et $Insurance_{hpx}$),

lesquels ne contiennent aucune paire de données pointant vers un même canevas source. Lorsque plusieurs données se disputaient un canevas, l'exemple retenu a été sélectionné au hasard. Il était ensuite possible de calculer les métriques, avec ou sans surgénération locale. Pour mieux apprécier la différence entre les données réelles et les données fictives, des émulations des corpus d'entraînement ont aussi été réalisés. Ces émulations généraient $n-1$ nouvelles données par canevas, où $n$ est le nombre d'antécédents réels dans l'ensemble d'entraînement original. Autrement dit, si 5 données d'Aviation ont pour canevas « from ORIGIN », alors 4 données de ce type étaient créées depuis la seule donnée « from ORIGIN » retenue dans $\text{Aviation}_{\text{hpx}}$. La figure 4.11 illustre les différents ensembles, et le tableau 4.3 fait état des résultats.

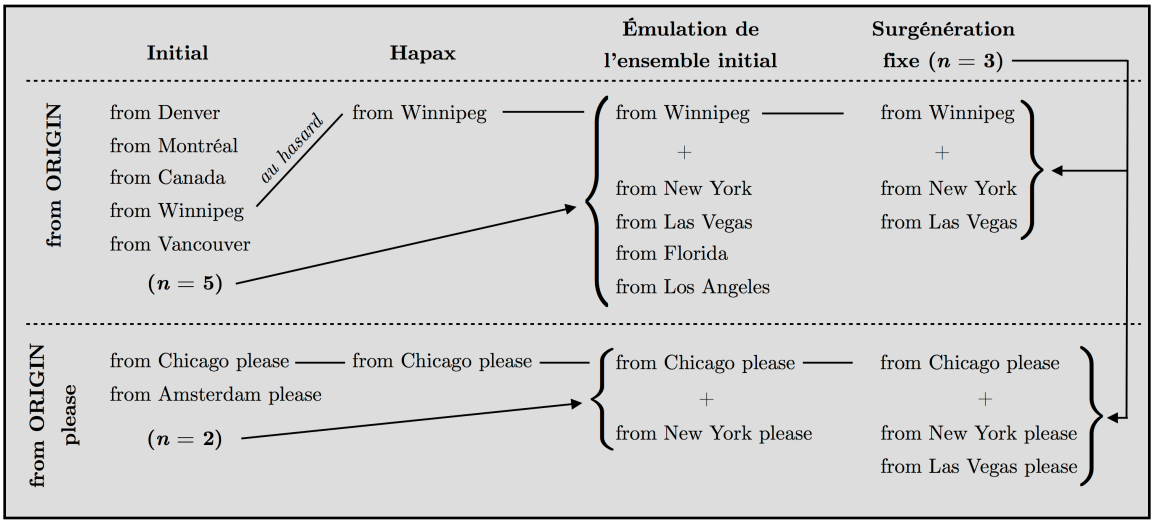



**Figure 4.11** — Ensembles avec surgénération locale de données

| | AVIATION | | | ASSURANCE | | |
|---|---|---|---|---|---|---|
| **ENSEMBLE D'ENTRAÎNEMENT** | **P** | **R** | **$F_1$** | **P** | **R** | **$F_1$** |
| **Initial** | 74,9 % | 81,1 % | 77,9 % | 64,3 % | 70,2 % | 67,1 % |
| **Hapax** | 49,5 % | 53,8 % | 51,6 % | 43,8 % | 44,7 % | 44,2 % |
| **Émulation de l'ensemble initial** | 66,1 % | 68,9 % | 67,5 % | 53,1 % | 58,6 % | 55,7 % |
| **Surgénération fixe ($n = 6$)** | 62,0 % | 64,5 % | 63,2 % | 50,8 % | 54,5 % | 52,6 % |

**Tableau 4.3** — Évaluation de la surgénération de données

Un coup d'oeil au tableau permet de constater que la surgénération locale n'est pas vaine. Même si ils ne permettent pas de retrouver des statistiques s'approchant des données réelles, autant l'émulation de l'ensemble initial que la surgénération fixe permettent d'améliorer les scores obtenus en ne considérant que des canevas singletons. Le gain finit toutefois par se noyer à travers les données réelles de l'ensemble d'entraînement, qui doivent leur meilleure performance à une plus grande représentativité et variabilité.

**c) surgénération depuis le système de traduction**

Finalement, puisque le système de traduction est celui qui renvoie des traductions différentes pour des données au même canevas, l'intuition suggère que ses bitextes pourraient être mis à profit. A priori, il est permis de croire qu'une donnée bien connue du système sera mieux traduite qu'une donnée qui lui est étrangère. Dans le cadre de la tâche courante, cela se traduit par un décompte des verbatims dans les bitextes, puis une génération automatisée selon les fréquences obtenues. Ainsi, si « *dollars* » est une devise usitée, il sera possible d'ajouter « *1320 dollars* » à la donnée existante « *1320 yens* ». À l'inverse, si « *yens* » est rare dans les textes d'entrainement de la SMT, la donnée « *10 dollars and 32 cents* » restera orpheline, sans que sa parente « *10 yens and 32 cents* » soit ajoutée.

Malheureusement, les problèmes rencontrés par l'approche traduisant les canevas (section 4.4) prévalent encore ici. En particulier, les nombreux faux positifs parviennent à noyer les verbatims réellement pertinents. En outre, l'intuition selon laquelle la SMT traduit mieux les phrases contenant des mots courants à l'entraînement est fausse. Une certaine variabilité apparait souhaitable, du moins dans un contexte où les données d'entraînement ne sont pas spécifiques au domaine d'intérêt. À ce propos, voir l'analyse détaillée de la section 5.4.

### *4.6.5 Intégration de vraies données cibles*

L'intégration de vraies données cibles s'effectue simplement : les canevas qu'elles sous-tendent sont considérés comme des vérités indiscutables. Ainsi, tous les canevas qui admettent

un contexte identique sans se conformer à l'un des étiquetages proposés sont automatiquement écartés. C'est ainsi que l'annotation « je veux parler à TIME » sera ignorée si une donnée réelle propose plutôt « je veux parler à PERSON ». Ce mode de fonctionnement peut sembler rigide, mais il fournit de meilleurs résultats comparativement aux approches pondérées ou à seuillage. La seule exception concerne l'absence de contexte (le canevas est un concept, sans rien d'autre), où il est préférable de ne poser aucune restriction. Naturellement, des canevas valables peuvent être exclus dans le processus, mais, même avec très peu de données cibles réelles, la perte de généralité vaut mieux que l'ajout de bruit.

### *4.6.6 Comparaison avec le modèle de base*

L'optimisation de l'approche étant complétée, il est désormais possible de la comparer au modèle de base. Comme elle est dérivée de l'approche par traduction et étiquetage, cette dernière apparait également sur les graphiques des figures 4.12 et 4.13.

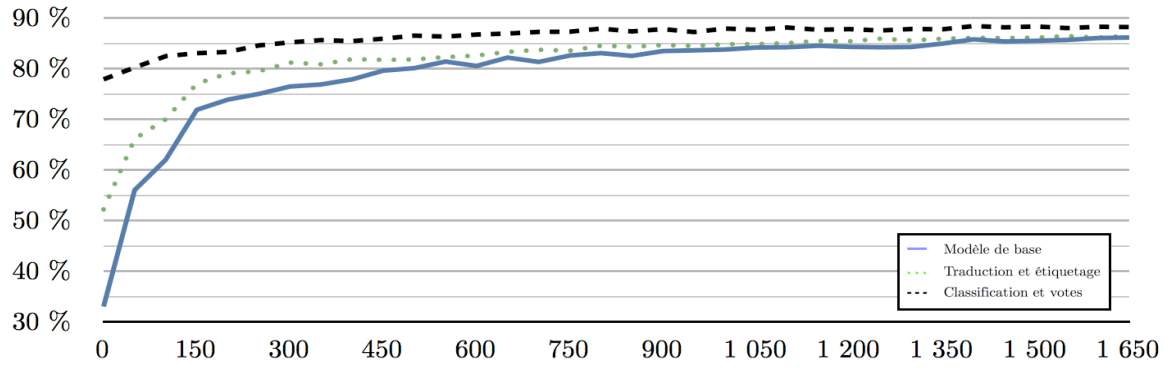



**Figure 4.12** — $F_1$ selon le nombre d'exemples cibles considérés (Aviation)

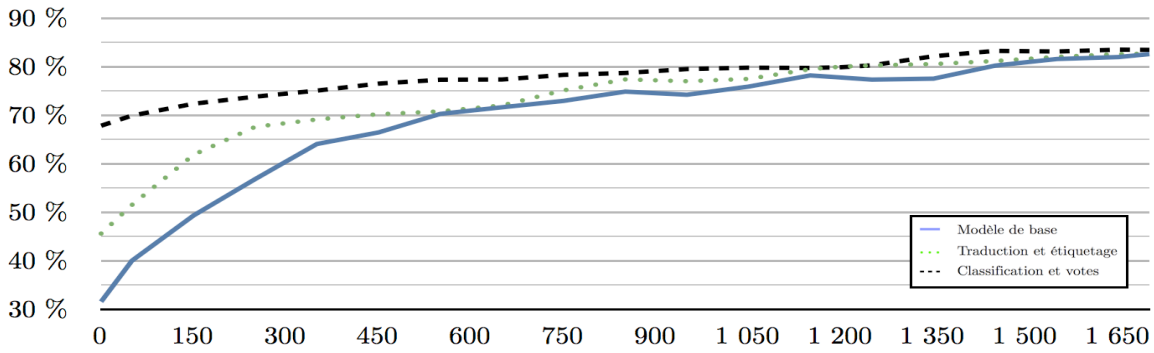



**Figure 4.13** — $F_1$ selon le nombre d'exemples cibles considérés (Assurance)

Les résultats parlent d'eux-mêmes. La méthode par classification et votes offre un net avantage sur les options ignorant les données sources (modèle de base) ou ignorant les regroupements par étiquettes (traduction et étiquetage). La façon proposée pour intégrer les données cibles (section 4.6.5) assure aussi l'optimalité des grammaires résultantes. Cela explique qu'au pire, les courbes « traduction et étiquetage » et « classification et votes » se confondent avec celle du modèle de base.

De façon plus générale, l'avantage d'une méthode sur une autre tient aux hypothèses qu'elle peut considérer. C'est ainsi que le modèle de base proposera toujours une interprétation limitée aux concepts énumérables, ne connaissant que ceux-ci. Or, ces concepts sont souvent « promus » vers des concepts plus complexes, rendant ces hypothèses très incomplètes. L'ajout de données artificielles par traduction et étiquetage permet d'établir une liste plus exhaustive des renvois envisageables. Ce second modèle ajoute cependant beaucoup de bruit, car plusieurs traductions sont acceptées pour chaque canevas. Chaque bonne hypothèse vient alors souvent de paire avec une mauvaise, sans différentiation possible. Ce n'est qu'au terme de l'étape de classification et votes que peut enfin survenir la distinction.

# CHAPITRE 5 — ANALYSE DES RÉSULTATS

L'expérimentation a permis d'établir que la méthode par classification et votes peut s'avérer très efficace. Seulement, la question demeure à savoir quelles conditions doivent être remplies pour qu'elle préserve son efficacité. Dans ce chapitre, une revue de méthodes analogues permet un premier balisage, puis quelques expériences supplémentaires viennent le restreindre. Enfin, les résultats obtenus sont mis à profit afin de réduire la quantité de ressources nécessaires lors de l'implémentation.

## 5.1 Méthodes par consensus

Le principe de la méthode par classification et votes se résumerait ainsi : pour un canevas donné, ses canevas traduits sont réordonnés selon leur fréquence pondérée, et seuls les plus représentatifs sont conservés. Autrement dit, plus un canevas s'approche du résultat moyen, plus il a de chances d'être retenu et vice versa.

Or, il existe des algorithmes reconnus oeuvrant exactement selon le même principe. Le plus célèbre parent en la matière est RANSAC (*Random Sampling Consensus*) [Fischler et Bolls, 1981], une méthode itérative qui se débarrasse progressivement des données aberrantes d'une distribution. Son fonctionnement est simple : un tirage aléatoire estime le modèle, puis les données conformes à ce premier modèle servent à en estimer un second, et ainsi de suite, jusqu'à ce que des critères d'arrêt soient satisfaits.

En comparaison, la méthode par classification et votes ne cherche pas à retenir toutes les données conformes, mais à s'assurer que celles qu'elle retient le sont. Cette visée est compatible avec l'idée du RANSAC, attendu que les mêmes hypothèses sont rencontrées (notamment, qu'il existe une structure modélisable dans les données, donc des canevas traduits acceptables qui se répètent parfois).

Le parallèle avec RANSAC implique un tirage uniforme, dont l'équivalent est une fonction de pointage constante. Il existe toutefois une variation du RANSAC, le PROSAC (*Progressive Sampling Consensus*) [Chum et Matas, 2005], qui utilise une mesure de similarité pour diminuer le temps d'exécution requis. Cette variation aurait une fonction de pointage linéaire comme équivalent, et vient étayer l'intuition selon laquelle le rang de traduction peut être mis à profit.

Les travaux de [Bangalore et al., 2002] s'inscrivent aussi dans cette veine, bien qu'ils misent sur plusieurs systèmes de traduction plutôt que sur une classification a priori des exemples. Ainsi, alors qu'eux dégagent un consensus en comparant les candidats de traductions suggérés par divers SMT, l'approche par classification et votes compare les candidats de traduction d'exemples sources similaires, ce qui permet de n'utiliser qu'une seule SMT.

Finalement, en considérant chaque canevas cible comme une catégorie et chaque traduction d'un canevas source comme une classification, il est possible de rattacher la méthode par classification et votes à celle de boosting adaptatif AdaBoost [Freund et Schapire, 1995]. Cette dernière combine linéairement plusieurs classificateurs faibles pour en créer un plus fort. Les coefficients finaux s'établissent itérativement.

Il est vrai que ces comparaisons sont partielles, et n'ont de sens qu'au moment où un canevas source est fixé. Cependant, elles permettent de supposer quelques caractéristiques cruciales sur la méthode par classification et votes. Ainsi, cette dernière doit contenir un ratio appréciable de données valides pour contrebalancer le bruit, elle a intérêt à exploiter le rang de traduction et elle ne doit admettre qu'un seul modèle cible (c'est-à-dire un seul ensemble de canevas cibles par canevas source, peu importe le contexte). À la lumière de ces observations, les sections qui suivent cherchent à quantifier ces différentes restrictions.

## 5.2 Mesure de disparité

Les méthodes par consensus introduisent une distinction importante entre données aberrantes et données conformes à la distribution. C'est en ignorant les premières (ou en leur attribuant un poids très faible) qu'elles arrivent à avoir du succès. Il serait donc souhaitable, pour faciliter l'analyse de la méthode d'intérêt, d'être en mesure de départager ces deux catégories. N'ayant pas les données requises pour établir un oracle optimal, l'approche retenue s'est plutôt basée sur une mesure de disparité. Ainsi, une donnée est considérée « conforme » si ses résultats locaux s'approchent des résultats moyens, et « singulière » autrement.

Par exemple, si l'algorithme par classification et votes établit que les 5 meilleurs canevas pour traduire « to DESTINATION » sont $(c_0,\ c_1,\ c_2,\ c_3,\ c_4)$ et que la donnée « to Denver » admet — dans l'ordre — les canevas $(c_2,\ c_0,\ c_1,\ c_3,\ c_4)$, alors cette donnée serait plutôt conforme. À l'opposé, si une donnée propose d'abord les canevas $(c_5,\ c_8,\ c_4,\ c_{11},\ c_{10})$, cette donnée sera considérée comme singulière. Un exemple plus détaillé est proposé après la définition mathématique, qui se résume à une distance euclidienne au niveau des rangs pour tous les canevas jugés valides :

$$disparité(donnée_{src}) = \sqrt{\sum_{i=0}^{n_v} \left(i - rang_{cible}\left(c^i_{ref}\right)\right)^2} \tag{5.1}$$

Ici, $n_v$ correspond au nombre de traductions considérées valides, et $c$ sert d'abréviation pour « canevas ». La liste de tous les canevas cibles valides pour un canevas source est établie au moment d'identifier les canevas pertinents (sous-section 4.6.1). Ainsi, pour chaque donnée source, le rang d'un canevas dans la référence est comparé avec le rang d'apparition local de ce même canevas (en cas d'absence, ce rang est fixé au maximum $r_{max}$). Les scores des exemples de la figure 5.1 sont établis suivant ce procédé ($r_{max} = 15$).

| | |
|---|---|
| **cnv src** | how much did i TRANSACTION DATE |
| **cnv tgt valides** | 0. combien ai-je TRANSACTION DATE |
| | 1. combien est-ce que j'ai TRANSACTION DATE |
| | 2. est-ce que j'ai TRANSACTION DATE |
| **donnée src A** | how much did i spend last month |
| **cnv tgt** | 3. combien ai-je TRANSACTION DATE |
| | 8. combien est-ce que j'ai TRANSACTION DATE |
| | 0. est-ce que j'ai TRANSACTION DATE |

$$disparité(A) = \sqrt{(0-3)^2 + (1-8)^2 + (2-0)^2} = \sqrt{62} = 7,87$$

| | |
|---|---|
| **donnée src B** | how much did i earn this week |
| **cnv tgt** | N/A combien ai-je TRANSACTION DATE |
| | 6. combien est-ce que j'ai TRANSACTION DATE |
| | 12. est-ce que j'ai TRANSACTION DATE |

$$disparité(B) = \sqrt{(0-15)^2 + (1-6)^2 + (2-12)^2} = \sqrt{350} = 18,70$$

**Figure 5.1** — Calcul de la disparité d'une donnée

Naturellement, lorsqu'un canevas n'est associé qu'à une seule donnée source, sa disparité est nulle d'emblée (elle s'impose elle-même comme la référence). Plus généralement, pour éliminer l'influence du nombre de canevas servant à établir la référence (**cnv tgt valides**), la surgénération locale a été utilisée lorsque nécessaire pour obtenir exactement 15 données par canevas source (si nécessaire, voir la section 4.6.4 b ayant trait à la surgénération locale). Aussi, comme la métrique varie selon le nombre de candidats considérés valides, ce nombre a été fixé ($n_v = 3$) avant d'effectuer les calculs.

L'intuition derrière cette mesure de disparité se résumerait ainsi : s'il existe une donnée avec un score de disparité très faible, alors cette donnée condense efficacement l'information des autres données ayant le même canevas source, et vice versa pour une donnée singulière.

## 5.3 Recherche de « super-données »

Une fois cette mesure de disparité établie, une première question survient : existe-t-il des données suffisamment conformes pour fournir directement la liste des canevas cibles valides ? Après tout, s'il existe des données avec une faible disparité, il serait souhaitable de pouvoir les identifier avant même de lancer l'algorithme, soit pour leur accorder plus d'importance, voire pour ne considérer qu'elles.

Une façon directe de le déterminer est tout bonnement de calculer la disparité sur l'ensemble (T) des données disponibles (lesquelles sont « réelles » (R) si elles font partie de l'ensemble d'entraînement, et « artificielles » (A) si elles sont issues de la surgénération locale) :

| | AIRLINE | | | INSURANCE | | |
|---|---|---|---|---|---|---|
| **DISPARITÉ** | **T** | **R** | **A** | **T** | **R** | **A** |
| **Moyenne** | 17,8 | 17,1 | 18,6 | 20,4 | 20,8 | 20,3 |
| **Minimale** | 1,4 | 1,4 | 3,2 | 1,4 | 1,4 | 4,5 |
| **Minimale moyenne** | 8,7 | 7,9 | 8,6 | 13,6 | 12,8 | 13,8 |

**Tableau 5.1** — Disparité moyenne, minimale et minimale moyenne

La disparité « minimale moyenne » est calculée en prenant la donnée ayant la plus faible disparité pour chaque canevas source. Comme chaque canevas est associé à exactement 15 données, cela revient à prendre une de ces 15 données (celle ayant la plus faible disparité), avant de calculer la moyenne. Enfin, rappelons que le calcul de la disparité s'effectue en considérant toutes les données. La distinction entre données réelles et artificielles n'arrive qu'au moment de colliger les résultats (et ce, toujours dans l'optique d'éviter qu'une donnée réelle unique ne considère sa disparité nulle, faute de comparaison).

Ce qui retient d'abord l'attention dans ce tableau, c'est la disparité minimale commune aux deux ensembles d'entraînement. Cela est attribuable à l'occurrence « *yes* », présente autant dans *Airline* que dans *Insurance*, et qui doit sa faible disparité (1,4) à sa grande simplicité. Sinon, les statistiques du tableau établissent une limite claire quant à la proximité qu'une

donnée peut avoir avec le classement établit par classification et votes. En moyenne, la donnée la plus proche se situe à une distance de 8,7 pour *Airline* et de 13,6 pour *Insurance*. Comme $n_v$ est fixé à 3, cela équivaut à une différence minimale d'environ 5 rangs pour *Airline* ($sqrt(8{,}65^2 \div 3)$) et de 7 à 8 rangs pour *Insurance*. L'écart est considérable.

Reste à savoir si cet écart est bel et bien à la faveur de la méthode par classification et votes. Deux nouveaux ensembles d'entraînement, $Airline_{hpx^+}$ et $Insurance_{hpx^+}$ permettent d'adresser la question. Ceux-ci sont formés en prenant uniquement les données de disparité minimale par canevas, c'est-à-dire celles responsables des scores obtenus à la ligne « disparité minimale moyenne ». Puis, la précision, le rappel et le $F_1$ obtenus depuis ces deux ensembles peuvent être comparés avec ceux de la méthode par classification et votes (tableau 5.2) :

| | AVIATION | | | ASSURANCE | | |
|---|---|---|---|---|---|---|
| **ENSEMBLE** | **P** | **R** | **$F_1$** | **P** | **R** | **$F_1$** |
| **Initial** | 74,9 % | 81,1 % | 77,9 % | 64,3 % | 70,2 % | 67,1 % |
| **Hapax** | 49,5 % | 53,8 % | 51,6 % | 43,8 % | 44,7 % | 44,2 % |
| **Hapax +** | 63,3 % | 64,0 % | 63,7 % | 46,3 % | 53,0 % | 49,4 % |

**Tableau 5.2** — Évaluation de la disparité

La méthode d'intérêt conserve l'avantage, ce qui suggère qu'une certaine combinaison des données est souhaitable. Cela dit, l'optimalité de l'ensemble hapax+ n'est garantie que par une métrique tributaire de la méthode analysée, et les résultats doivent donc être tempérés en conséquence. Malgré tout, le net avantage d'hapax+ sur hapax (où les données étaient retenues au hasard — section 4.6.4 b), montre que la disparité n'est pas complètement étrangère aux performances. Bref, autant il parait important de mélanger l'information de traduction de plusieurs données ($F_1$(inital) > $F_1$(hapax+)), autant il parait prometteur d'altérer judicieusement la pondération de ces différentes données ($F_1$(hapax+) > $F_1$(hapax)). Ainsi, la pondération des exemples s'établit comme amélioration éventuelle de la méthode proposée.

## 5.4 Corrélation avec le système de traduction

La section précédente permet de confirmer ce que l'intuition suggérait : pour un même canevas, certaines données se trouvent mieux traduites que d'autres. La disparité, bien qu'imparfaite, permet une première identification en ce sens.

Une autre approche qui mérite d'être explorée repose plutôt sur la SMT. Comme la traduction est statistique, il apparait logique de supposer qu'un comportement fréquemment observé lors de l'entraînement affichera de bons résultats par la suite. En d'autres termes, une donnée source formée de mots fréquents dans les bitextes d'entraînement devrait être généralement bien traduite, et vice versa. Dans le cas d'intérêt, cela revient à croire que « *going to Paris* » a plus de chances d'être bien traduit que « *going to Reykjavik* », si la fréquence de « Paris » dans les bitextes d'entraînement de la SMT dépasse celle de « Reykjavik ».

Encore une fois, la création de sous-ensembles d'entraînement permet l'évaluation. Ces sous-ensembles — $Airline_{SMT}$ et $Insurance_{SMT}$ — sont aussi constitués d'une seule donnée source par canevas. Cette fois, c'est un modèle de langue entraîné sur la portion source des bitextes de la SMT qui guide le choix de cette donnée. Seule celle ayant le score $n$-gramme le plus élevé (donc la plus grande proximité avec la SMT) est retenue. Arbitrairement, l'ordre de ce modèle de langue a été fixé à 2. Le tableau 5.3 présente les résultats obtenus par ces ensembles :

| | AVIATION | | | ASSURANCE | | |
|---|---|---|---|---|---|---|
| **ENSEMBLE** | **P** | **R** | **$F_1$** | **P** | **R** | **$F_1$** |
| **Initial** | 74,9 % | 81,1 % | 77,9 % | 64,3 % | 70,2 % | 67,1 % |
| **Hapax** | 49,5 % | 53,8 % | 51,6 % | 43,8 % | 44,7 % | 44,2 % |
| **Hapax +** | 63,3 % | 64,0 % | 63,7 % | 46,3 % | 53,0 % | 49,4 % |
| **SMT** | 52,6 % | 57,1 % | 54,7 % | 47,0 % | 49,0 % | 48,0 % |

**Tableau 5.3** — Évaluation de la proximité avec le système de traduction

Les résultats d'Aviation peuvent sembler décevants, surtout lorsque comparés à ceux des ensembles hapax+. Pourtant, l'évaluation plus détaillée permet, encore une fois, de blâmer les données hors domaine des bitextes, comme ce fut le cas avec l'approche par traducteur de canevas (section 4.3) et la surgénération d'exemples (section 4.6.4 c). Le fait est que, lorsqu'une donnée s'approche de la SMT, alors sa traduction tient davantage des débats parlementaires que de l'aviation. D'ailleurs, un certain chevauchement de concepts entre l'assurance et les débats parlementaires explique que pour ce domaine, la différence soit plus faible (49,4 % pour l'ensemble hapax+, 48,0 % pour l'ensemble SMT).

Ainsi, lorsque les données de traduction ne sont pas ajustées à la tâche courante, il ne semble pas avantageux de faire l'inverse, c'est-à-dire d'ajuster les données courantes au système de traduction (ce qui était, en somme, la logique derrière les ensembles SMT). Pour s'en convaincre une dernière fois, il suffit de faire appel à la corrélation linéaire de Bravais-Pearson.

Bien connue, cette corrélation permet de cerner la présence d'un lien linéaire entre deux variables. Un coefficient de valeur absolue inférieure à 0,5 indique un lien faible (jusqu'à absence de lien lorsque $r_p = 0$) et un lien fort autrement (jusqu'à corrélation parfaite pour $r_p = \pm 1$). Une valeur négative ou positive indique une progression décroissante ou croissante respectivement. Ce coefficient s'établit selon la formule 5.2 :

$$r_p = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{5.2}$$

Le numérateur, $\sigma_{xy}$ désigne la covariance entre les variables $x$ et $y$, alors que le dénominateur se compose du produit entre les deux écarts-types $\sigma_x$ et $\sigma_y$. Dans le cas d'intérêt, la corrélation sera évaluée entre la disparité d'une donnée source (section 5.2, équation 5.1) et sa proximité avec la SMT, telle que calculée par le modèle $n$-gramme d'ordre 2 (section courante). Les résultats apparaissent dans le tableau 5.4 :

| | AIRLINE | INSURANCE |
|---|---|---|
| $r_p$ | -0,17472 | -0,46435 |

**Tableau 5.4** — Corrélation entre proximité avec la SMT et disparité

Les résultats vont dans le sens des observations précédentes. La corrélation faible obtenue pour l'ensemble *Airline* implique que les exemples de disparité faible (donc les exemples représentatifs d'un canevas) ne sont pas systématiquement plus fréquents dans les bitextes de la SMT. Conformément à l'analyse des résultats du tableau 5.3, la corrélation est plus forte en ce qui a trait à *Insurance*.

Bien que le nombre restreint de domaines ne permette aucune conclusion définitive, il est permis de croire que l'adéquation entre la SMT et le domaine étudié ait un impact significatif sur l'approche par classification et votes. C'est, du moins, la tendance esquissée par les deux domaines retenus. À l'extrême, c'est-à-dire lorsque le système de traduction est parfaitement en phase avec le domaine considéré, il pourrait être préférable de ne retenir que quelques données par canevas, celles que la SMT « connaît » le plus. Des expériences avec traductions oracles (c'est-à-dire, lorsque des traductions « parfaites » sont disponibles) permettraient sans doute un meilleur discernement.

## 5.5 Caractéristiques des groupes

La méthode par classification et votes s'appuie sur la redondance de canevas. Une question qui s'impose est donc : combien d'exemples faut-il, par canevas source, pour espérer de bons résultats ?

Pour évaluer ce nombre, commençons par une légère simplification de langage. L'expression **groupe source** sera utilisé pour faire référence à l'ensemble des données sources ayant un même canevas. Cela permettra d'utiliser la cardinalité[10]. Accessoirement, le canevas associé au groupe source en question peut être précisé. Par exemple, le groupe source « a

[10] Nombre d'éléments dans un ensemble.

FLIGHT_CLASS flight to DESTINATION » est composé de deux données — « *a first class flight to Toronto* » et « *a first class flight to Chicago* ». Sa cardinalité s'établit donc à 2.

Différents échantillonnages de ces groupes permettent de jouer sur la variable d'intérêt. Pour se faire, il suffit de déterminer un nombre fixe de canevas sources, puis d'incrémenter progressivement leur cardinalité. La figure 5.2 offre une représentation visuelle de cette approche :

| | |
|---|---|
| **cnv src** | pay BILL |
| **groupe src (4)** | pay electric bill<br>pay cable bill<br>pay mastercard bill<br>pay renter's bill |
| $\boldsymbol{n = 1}$ | pay mastercard bill |
| $\boldsymbol{n = 2}$ | pay electric bill<br>pay mastercard bill |
| $\boldsymbol{n = 3}$ | pay cable bill<br>pay mastercard bill<br>pay renter's bill |

**Figure 5.2** — Échantillonnage de groupes sources

Le nombre de canevas sources retenus est en fait contraint par les données d'entraînement. Comme il est impossible d'augmenter la cardinalité d'un groupe sans créer de nouvelles données, et qu'une certaine uniformité est requise, le comportement à la cardinalité $n$ ne peut s'observer qu'avec les groupes de taille initiale $m \geq n$. Autrement illustré, si, pour un canevas donnée, il n'existe qu'un seul exemple, alors l'effet d'avoir deux exemples pour ce même canevas ne peut être mesuré. Le canevas ne peut donc pas être utilisé lorsque $n = 2$. Et pour éviter un avantage indu, il ne peut pas non plus être retenu lorsque $n = 1$.

Dans le cas d'intérêt, la limite supérieure de l'analyse a été fixée à $n = 4$. Au-delà, le nombre de canevas encore éligibles diminuait drastiquement (surtout dans le cas d'*Insurance*). La méthodologie se résume donc ainsi : toutes les données d'entraînement appartenant à un

groupe source ayant une cardinalité inférieure à 4 ont été ignorées. Puis, des ensembles à cardinalité restreinte ont été dérivés pour tous les groupes restants, par tirage aléatoire uniforme, sans remise. Ainsi, si un groupe comportait 9 données :

- 1 donnée sur 9 a été retenue pour l'ensemble $n = 1$
- 2 données sur 9 ont été retenues pour l'ensemble $n = 2$
- 3 données sur 9 ont été retenues pour l'ensemble $n = 3$
- 4 données sur 9 ont été retenues pour l'ensemble $n = 4$

Cette façon de faire assure que les 4 ensembles possèdent exactement le même nombre de canevas sources. Seul le nombre d'antécédents par canevas varie d'un ensemble à l'autre. Les résultats ainsi obtenus figurent dans le tableau 5.5 :

| | AVIATION | | | ASSURANCE | | |
|---|---|---|---|---|---|---|
| **CARDINALITÉ** | **P** | **R** | **$F_1$** | **P** | **R** | **$F_1$** |
| $n = 1$ | 44,1 % | 51,8 % | 47,6 % | 37,4 % | 43,6 % | 40,2 % |
| $n = 2$ | 51,2 % | 56,5 % | 53,7 % | 42,1 % | 48,2 % | 44,9 % |
| $n = 3$ | 56,0 % | 60,8 % | 58,3 % | 50,4 % | 55,0 % | 52,6 % |
| $n = 4$ | 59,5 % | 63,2 % | 61,3 % | 54,3 % | 56,0 % | 55,2 % |

**Tableau 5.5** — Évaluation de la cardinalité des groupes sources

Dans les limites des données disponibles, il est difficile de dégager une réponse définitive. Une plus grande cardinalité se traduit immanquablement par un meilleur $F_1$, mais, malgré un ralentissement perceptible, aucun plateau n'est encore atteint. Ce qui est flagrant, en revanche, c'est que la robustesse bonifiée affecte davantage la précision. Elle croît plus vite que le rappel, croissance attribuable au nombre moins important de faux-positifs. Comme il y a moins de concepts trouvés en général, le dénominateur diminue, d'où un score plus élevé.

## 5.6 Réduction des ressources nécessaires

Pour conclure cette section, il convient de donner suite aux perspectives de l'approche par modèle de langue prédictif de la section 4.4. Cette approche suggérait un moyen d'évaluer la

pertinence des paires de phrases utilisées lors de l'entraînement de la SMT, selon le domaine d'intérêt. En bref, trois étapes permettent d'obtenir cette pertinence :

i. créer le modèle $n$-gramme $\mathrm{P}(s)$ depuis les données sources annotées
ii. annoter le côté source des bitextes de la SMT
iii. utiliser $\mathrm{P}(s)$ sur les phrases sources des bitextes

Dans le cadre de l'approche par modèle de langue prédictif, ces étapes servaient à déduire un nouveau modèle de langue — du côté cible, cette fois — et cette surenchère d'estimations finissait par avoir raison des performances. Par contre, dans l'optique de répondre à l'objectif II (section 2.4), c'est-à-dire en vue d'améliorer la qualité de la SMT, cet algorithme apparaît encore prometteur. Les travaux de [Moore et Lewis, 2010] et de [Axelrod et al., 2011] explorent d'ailleurs des pistes similaires avec de bons résultats.

Commençons par définir une référence pour le système de traduction actuel, soit celui décrit à la section 3.3. Le score BLEU sera utilisé à cette fin (au besoin, revoir la section 3.4.2). Afin de le calculer, les traductions candidates retournées par la SMT doivent être comparées à une référence. Une telle référence pour chaque domaine est donc nécessaire. Or, il se trouve qu'un nombre raisonnable de phrases ont déjà été manuellement traduites dans le cadre de ce mémoire, lors de la création des ensembles Aviation et Assurance. Ce sont ces paires de phrases qui seront utilisées. Le tableau 5.6 présente ces résultats initiaux :

| | AIRLINE | INSURANCE |
|---|---|---|
| **Nb Paires** | 2449 | 2459 |
| **BLEU** | 24,30 | 25,16 |

**Tableau 5.6** — Score BLEU initial pour chaque domaine

À titre informatif, il est courant de multiplier le score BLEU par 100 au moment de la comparaison, même si ce score est originalement compris entre 0 et 1. Les références unitaires

qui en découlent ne sont pas rares non plus. Ainsi, il est commun de parler « d'augmentation d'un point de BLEU », pour indiquer une augmentation effective de 0,01. Quant aux résultats, un score inférieur à 15 est typiquement assimilé à du bruit, alors qu'un score de l'ordre de 40 est vu comme excellent. Cet intervalle restreint permet de mieux saisir la pertinence (et la difficulté) d'aller chercher quelques points BLEU supplémentaires.

Dans le cas présent, l'évaluation de pertinence décrite en début de section a été mise à profit. Deux modèles *n*-grammes, entraînés respectivement sur les corpus annotés d'*Airline* et d'*Insurance*, ont servi à identifier les paires les plus pertinentes des bitextes pour chaque domaine. Puis, selon l'affinité d'une phrase source de la SMT avec le modèle de langue, cette phrase était soit ignorée, soit considérée, soit répétée un certain nombre de fois (afin d'augmenter son poids à l'entraînement). Naturellement, la phrase cible associée du bitexte subissait le même traitement que sa parente.

Quelques configurations ont été testées, et la meilleure solution locale fonctionne avec les multiples d'un seul paramètre $\Delta$. De façon générale, le premier $k \in \mathbb{N}$ tel que $\mathrm{P}(s) < k \cdot \Delta$ est calculé, puis $s$ est répété $k-1$ fois. Ainsi, si $\mathrm{P}(s) < \Delta$, alors $s$ est ignoré ; s'il est plus grand que $\Delta$, mais plus petit que $2 \cdot \Delta$, il est considéré une seule fois ; s'il se trouve plutôt dans l'intervalle $[2 \cdot \Delta,\ 3 \cdot \Delta]$, il est répété deux fois ; et ainsi de suite. Une telle façon de faire a permis d'obtenir les résultats suivants :

| BLEU | AIRLINE | INSURANCE |
|---|---:|---:|
| **Initial** | 24,30 | 25,16 |
| **Adapté** | 24,78 | 25,34 |

**Tableau 5.7** — Score BLEU des SMT adaptés aux domaines

Le gain est modeste, d'autant que le nombre limité de données servant aux calculs de BLEU augmente sa variabilité. Ce qui est plus impressionnant, en revanche, c'est la diminution de ressources nécessaires à l'obtention de tels scores (tableau 5.8).

| | INITIAL | AIRLINE | INSURANCE |
|---|---|---|---|
| **Nb de paires** | $20{,}9 \times 10^6$ | $13{,}0 \times 10^6$ | $11{,}3 \times 10^6$ |

**Tableau 5.8** — Nombre de paires de phrases dans les différentes SMT

Même avec le dédoublement de certaines entrées, la nouvelle SMT d'*Airline* comporte 37,8 % moins de phrases, alors que le corpus d'entraînement d'*Insurance* diminue de près de 45,9 %. Le tout, sans que les performances de traduction ne soient affectées (négativement). Malheureusement, ces gains ne se reflètent pas dans la projection de grammaires par classification et votes. La différence en BLEU n'est pas assez significative. Quant à la réduction de la taille des bitextes, cette avenue peut sembler intéressante, mais le temps supplémentaire requis n'est pas négligeable — le temps gagné à l'entraînement du modèle est perdu lors de l'identification des phrases à conserver.

Une expérience avec plus de données sources serait de mise. Le modèle de langue résultant serait d'autant plus fiable, ce qui se traduirait peut-être par un gain de BLEU plus conséquent, et, ultimement, par une meilleure projection de grammaire.

# CHAPITRE 6 — CONCLUSION

En conclusion, la pertinence de mettre à profit des données sources annotées au moment de projeter une grammaire ne fait aucun doute, et ce, malgré l'absence d'un système de traduction dédié. Pour ce faire, plusieurs méthodes ont été présentées, parmi lesquelles s'est distinguée l'approche par classification et votes. Celle-ci a fait l'objet d'une optimisation et d'une analyse rigoureuse.

Au-delà de ses performances, c'est sa logique algorithmique qui est révélatrice : à n'en pas douter, la clé d'une projection réussie, dans le cadre proposé, repose sur l'exploitation efficace de l'information véhiculée par les annotations. C'est ce surplus d'information — d'abord ignorée par la SMT — qui permet d'augmenter la robustesse et d'obtenir de meilleurs candidats de traduction à terme. Exploiter la redondance de canevas est une façon de faire en ce sens, montrant de très bons résultats sur les données disponibles. En outre, la possibilité de créer des données fictives par surgénération locale est un moyen intéressant de simuler cette redondance.

D'autres stratégies intuitives se sont révélées plus décevantes. Ainsi, l'expérimentation tend à décourager l'idée de rapprocher les données sources du système de traduction. Dans un tel cas, les données ont tendance à dévier du domaine d'intérêt pour rejoindre les domaines d'entraînement de la SMT, ce qui n'est pas souhaitable. À l'inverse, un pré-traitement des bitextes pour pondérer différemment certaines paires de phrases, selon leur proximité avec le domaine de projection, est envisageable.

Pour faire suite aux travaux du mémoire, plusieurs avenues se dessinent. En tête de liste figure une meilleure intégration des données cibles. L'ajout d'un certain nombre d'exemples annotés dans la langue cible coïncide avec le développement usuel d'une application de dialogues avancés. Et lorsque certaines données sont disponibles, il est important de combiner

efficacement l'information provenant des deux langues. En particulier, comment les données cibles annotées peuvent-elles améliorer la traduction des données sources, et leur étiquetage automatique ?

Tester de nouvelles paires de langues serait aussi de rigueur, entendu que la proximité entre le français et l'anglais introduit un biais non négligeable. Les méthodes gagneraient en crédibilité si elles affichaient de bons résultats en ayant le russe, le coréen ou le japonais comme langue cible. Et même en cas de piètres résultats, l'analyse subséquente permettrait sans doute de mieux baliser les conditions de succès de chacune.

Sinon, un moyen simple de pondérer intelligemment les exemples sources serait souhaitable. La version actuelle est encore sensible aux données aberrantes, et cela est dû en partie au poids constant affecté à chaque exemple d'entrée. En estimant l'importance relative de chaque donnée, la performance devrait augmenter. Plus largement, la possibilité de surgénérer de nouveaux exemples ou d'en dériver syntaxiquement — à l'aide de grammaires et de dictionnaires, par exemple — se concilierait bien avec une pondération flexible.

Enfin, plus de temps pourrait être investi à l'amélioration du système de traduction. Des résultats décents ont déjà été obtenus dans le cadre de ce mémoire, bien que les expériences en ce sens n'étaient pas prioritaires. Les bitextes des SMT étant incroyablement volumineux, il est même permis de croire que de nouvelles paires annotées pourraient être déduites de ces corpus.

Car au final, le nerf de la guerre reste inchangé : plus grande est la quantité de données fiables disponibles, meilleure sera la performance. La projection de données sources par classification et votes n'est qu'un outil à cette fin.

# BIBLIOGRAPHIE

AXELROD, Amittai ; HE, Xiadong et GAO, Jianfeng (2011). *Domain Adaptation via Pseudo In-Domain Data Selection,* Dans : Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, p. 355-362.

BANGALORE, Srinivas ; MURDOCK, Vanessa et RICCARDI, Giuseppe (2002). *Bootstrapping Bilingual Data Using Consensus Translation for a Multilingual Instant Messaging System,* Dans : Proceedings of the 19th International Conference on Computational Linguistics, p. 1-7.

BOUILLON, Pierrette ; RAYNER, Manny ; NOVELLAS, Bruna ; NAKAO, Yukie ; SANTAHOLMA, Marianne ; STARLANDER, Marianne et CHATZICHRISAFIS, Nikos (2006). *Une grammaire multilingue partagée pour la traduction automatique de la parole*, Dans : Proceedings of Traitement Automatique des Langues Naturelles, p. 155-173.

BROWN, Peter F. ; DELLA PIETRA, Stephen A. ; DELLA PIETRA, Vincent J. et MERCER, Robert L. (1993). *The Mathematics of Statistical Machine Translation : Parameter Estimation*, Computational Linguistics, p. 263-311.

CHIANG, David (2005). *A Hierarchical Phrase-Based Model for Statistical Machine Translation*, Dans : ACL '05 Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, p. 263-270.

CHUM, Ondrej et MATAS, Jiri (2005). *Matching with PROSAC — Progressive Sample Consensus*, Dans : Proceedings IEEE Conference Computer Vision and Pattern Recognition, p. 220-226.

FISCHLER, Martin A. et BOLLS, Robert C. (1981). *Random Sample Consensus : a Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography*, Dans : Communications of the ACM, vol. 24 nº 6, p. 381-395.

FREUND, Yoav et SCHAPIRE, Robert E. (1995). *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*, Dans : EuroCOLT '95 Proceedings of the Second European Conference on Computational Learning Theory, p. 23-37.

HWA, Rebecca ; RESNIK, Philip ; WEINBERG, Amy ; CABEZAS, Clara et KOLAK, Okan (2004). *Bootstrapping Parsers via Syntactic Projection across Parallel Texts,* Dans : Natural Language Engineering, vol. 11 nº 3, p. 311-325.

KIM, Roger ; DALRYMPLE, Mary ; KAPLAN, Ronald ; KING, Tracy H. ; MASUICHI, Hiroshi et OHKUMA, Tomoko (2003). *Multilingual Grammar Development via Grammar Porting*, Dans : Proceedings of the ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development, p. 49-56.

KIM, Seokhwan ; JEONG, Minwoo ; LEE, Jonghoon et LEE, Gary Geunbae (2010). *A Cross-lingual Annotation Projection Approach for Relation Detection,* Dans : Proceedings of the 23rd International Conference on Computational Linguistics, p. 564-571.

KOEHN, Philipp ; OCH, Franz Joseph et MARCU, Daniel (2003). *Statistical Phrase-Based Translation,* Dans : Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, p. 48-54.

KOEHN, Philipp et SCHROEDER, Josh (2007). *Experiments in Domain Adaptation for Statistical Machine Translation,* Dans : StatMT ’07 Proceedings of the Second Workshop on Statistical Machine Translation, p. 224-227.

KOEHN, Philipp ; HOANG, Hieu ; BIRCH, Alexandra ; CALLISON-BURCH, Chris ; FEDERICO, Marcello ; BERTOLDI, Nicola ; COWAN, Brooke ; SHEN, Wade ; MORAN, Christine ; ZENS, Richard ; DYER, Chris ; BOJAR, Ondrej ; CONSTANTIN, Alexandra et HERBST, Evan (2007). *Moses : Open Source Toolkit for Statistical Machine Translation*, Dans : Annual Meeting of the Association for Computational Linguistics, p. 177-180.

MOORE, Robert et LEWIS, William (2010). *Intelligent Selection of Language Model Training Data,* Dans : Proceedings of the ACL 2010 Conference Short Papers, p. 220-224.

PAPINENI, Kishore ; ROUKOS, Salim ; WARD, Todd et ZHU, Wei-Jing (2002). *BLEU : a Method for Automatic Evaluation of Machine Translation,* Dans : ACL ’02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, p. 311-318.

RAYBAUD, Sylvain ; LAVECCHIA, Caroline ; LANGLOIS, David et SMAÏLI, Kamel (2009). *Word- and Sentence-level Confidence Measures for Machine Translation,* Dans : Proceedings of the 13th Annual Conference of the EAMT, p. 104-111.

SANTAHOLMA, Marianne (2005). *Linguistic Representation of Finnish in a Limited Domain Speech-to-Speech Translation System,* Dans : Proceedings of the 10th Conference on European Association of Machine Translation, p. 226-234.

SANTAHOLMA, Marianne (2008). *Multilingual Grammar Resources in Multilingual Application Development*, Dans : Proceedings of Grammar Engineering Across Frameworks Workshop, p. 25-32.

TIEDEMANN, Jörg (2009). *News from OPUS — A Collection of Multilingual Parallel Corpora with Tools and Interface*, Dans : Recent Advances in Natural Language Processing volume V, p. 237-248.