Un cadre d'apprentissage intégralement discriminant pour la traduction statistique

Thomas Lavergne[1, 2] Alexandre Allauzen[1, 2] François Yvon[1, 2]
(1) Université Paris Sud 91 405 Orsay
(2) LIMSI/CNRS rue John von Neuman 91 405 Orsay
{lavergne,allauzen,yvon}@limsi.fr

RÉSUMÉ

Une faiblesse des systèmes de traduction statistiques est le caractère *ad hoc* du processus d'apprentissage, qui repose sur un empilement d'heuristiques et conduit à apprendre des paramètres dont la valeur est sous-optimale. Dans ce travail, nous reformulons la traduction automatique sous la forme familière de l'apprentissage d'un modèle probabiliste structuré utilisant une paramétrisation log-linéaire. Cette entreprise est rendue possible par le développement d'une implantation efficace qui permet en particulier de prendre en compte la présence de variables latentes dans le modèle. Notre approche est comparée, avec succès, avec une approche de l'état de l'art sur la tâche de traduction de données du BTEC pour le couple Français-Anglais.

ABSTRACT

A fully discriminative training framework for Statistical Machine Translation

A major pitfall of existing statistical machine translation systems is their lack of a proper training procedure. In fact, the phrase extraction and scoring processes that underlie the construction of the translation model typically rely on a chain of crude heuristics, a situation deemed problematic by many. In this paper, we recast machine translation in the familiar terms of a probabilistic structure learning problem, using a standard log-linear parameterization. The tractability of this enterprise is achieved through an efficient implementation that can take into account all the aspects of the underlying translation process through latent variables. We also address the reference reachability issue by using oracle decoding techniques. This approach is experimentally contrasted with a state-of-the-art system on the French-English BTEC translation task.

MOTS-CLÉS : Traduction Automatique, Apprentissage Discriminant.

KEYWORDS: Machine Translation, Discriminative Learning.

1 Introduction

L'objectif d'un système de traduction statistique (STS) consiste à calculer, pour toute phrase en langue source $\mathbf{s}$, la traduction $\mathbf{t}^*$ qui lui est la plus probablement associée. Ce résultat est typiquement obtenu en maximisant une fonction de score $\Phi_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{t})$, paramétrisée par le vecteur $\boldsymbol{\theta}$, sur l'ensemble de toutes les traductions possibles de $\mathbf{s}$. Un choix raisonnable pour Φ est la probabilité conditionnelle de $\mathbf{t}$ *sachant* $\mathbf{s}$ $p_{\boldsymbol{\theta}}(\mathbf{t} \mid \mathbf{s})$.

Étant donnée la taille des espaces d'entrée et de sortie sur lesquels de tels modèles probabilistes

doivent être définis, un modèle pour **t** *sachant* **s** doit être décomposé en modélisant la traduction par une séquence d'étapes de dérivation. Dans les systèmes à base de segments (*phrase-based systems*) (Zens *et al.*, 2002; Koehn *et al.*, 2003), qui seront considérés dans cette étude, ces étapes de dérivation correspondent à des décisions qui portent (a) sur la délimitation des unités de traduction en langue source, (b) sur le choix d'un équivalent de traduction pour chaque unité définie en (a) ; enfin sur l'ordre relatif dans lequel sont réarrangées (on dira *réordonnées*) les unités cibles sélectionnées en (b). Dans la mesure où l'apprentissage se fonde uniquement sur l'observation des paires $(\mathbf{s}, \mathbf{t})$, ces dérivations ne sont pas observées pendant l'apprentissage et doivent être incorporées sous la forme de *variables latentes*.

Chacune de ces étapes de dérivation doit être modélisée et associée à un paramètre numérique, qui est réglé de façon à ce que le système résultant engendre les meilleures traductions possibles. Ainsi, dans les systèmes à base de segments, chaque unité de traduction source est nantie d'un ensemble de paramètres qui valuent les différentes alternatives de traduction et de réordonnancement pour ce segment.

Dans la plupart des systèmes de traduction (voir (Koehn, 2010) pour un état de l'art récent, ou, en français (Allauzen et Yvon, 2011)), l'apprentissage de ces paramètres s'effectue en deux temps : (i) en premier lieu, plusieurs modèles probabilistes sont estimés de manière indépendante, en utilisant de très gros corpus monolingues ou bilingues *parallèles*. Une étape supplémentaire (ii) d'apprentissage (souvent désignée sous le nom de *tuning*) est ensuite nécessaire pour équilibrer la contribution de chacun de ces modèles à la fonction de score. Cette seconde étape, réalisée sur des corpus de développement de taille réduite, conduit au calcul de paramètres globaux (un pour chaque modèle estimé en (i)), qui sont réglés de manière discriminante, c'est-à-dire en cherchant à maximiser explicitement une mesure de qualité de la traduction, sous l'hypothèse que les scores se combinent linéairement. Ceci implique, par exemple, que le paramètre $\boldsymbol{\theta}_{(\bar{s},\bar{t})}$ qui évalue la plausibilité que le segment [1] source $\bar{s}$ se traduise $\bar{t}$ est calculé comme le produit d'un poids global, réglé de manière discriminante sur un ensemble de développement, avec un score local, calculé de manière heuristique sur de larges corpus. Comme souligné dans de nombreuses études, ce processus à deux étages conduit à des paramètres sous-optimaux ; pour obtenir des résultats stables, il est également nécessaire de limiter le nombre de modèles combinés en (ii) à quelques dizaines d'unités (voir cependant (Liang *et al.*, 2006; Chiang *et al.*, 2009; Blunsom *et al.*, 2008; Simianer *et al.*, 2012) pour des tentatives de contourner cette limitation).

Dans ce travail, à la suite de (Liang *et al.*, 2006; Blunsom *et al.*, 2008; Dyer et Resnik, 2010), nous explorons une approche alternative, dans laquelle **tous les paramètres du modèle** sont appris *simultanément* (plutôt qu'indépendamment) et de *manière discriminante* (plutôt qu'heuristique) ; cet apprentissage est réalisé en optimisant une fonction objectif bien connue sur **l'intégralité des données d'entraînement** (plutôt qu'un petit ensemble de développement). Notre architecture permet de se dispenser presqu'entièrement du besoin d'estimer des modèles séparés puis de régler les paramètres pour les recombiner : ces deux étapes sont ici réalisées simultanément.

Dans cette approche, l'apprentissage ne demande que (a) un corpus parallèle, (b) un inventaire des unités de traductions et (c) un mécanisme pour produire des hypothèses de réordonnancement. Il est important de noter que (b) peut être obtenu de plusieurs manières, par exemple en fouillant des corpus *comparables*, et/ou en exploitant des dictionnaires et des terminologies bilingues. De même, plusieurs options existent pour (c), comme d'utiliser des modèles de réordonnancement simples tels que IBM-n (Tillmann et Ney, 2003) et WJ-n (Kumar et Byrne, 2005)

1. La situation est un peu plus complexe car les systèmes standard comprennent plusieurs modèles de traduction.

ou encore d'apprendre les règles de réordonnancement, comme nous le ferons ici.

L'implantation d'un cadre discriminant intégré pour la traduction statistique implique toutefois de résoudre plusieurs problèmes pratiques et théoriques liés à la présence de variables latentes dans le modèle et à l'impossibilité de disposer de données de supervision pour certaines paires de phrases lorsque la traduction de référence ne peut être produite par le modèle (on dit alors que la référence est *non atteignable*). Ces problèmes sont résolus respectivement en sommant (marginalisant) sur toutes les dérivations possibles et en recourant à des *traductions oracles*.

Les contributions de ce travail, qui développe et étend la proposition présentée dans (Lavergne *et al.*, 2011) en s'affranchissant du besoin de disposer d'alignements de référence, sont multiples : la conception d'un modèle intégré pour la traduction automatique, qui rend possible l'utilisation d'un grand nombre de traits linguistiques ; une implémentation modulaire qui, en s'appuyant sur le formalisme des transducteurs finis pondérés (WFST), bénéficie d'algorithmes efficaces aussi bien pour l'apprentissage que pour l'inférence ; et l'étude de plusieurs manières de traiter le problème des références non atteignables. Notre contribution est aussi expérimentale, puisque nous montrons que le système ainsi construit s'avère capable de surpasser un système très performant sur une tâche de complexité moyenne.

Le reste de cet article est organisé comme suit. Nous commençons par clarifier, à la section 2, les concepts nécessaires à la formulation de notre cadre discriminant et comparons notre approche avec d'autres implantations de l'apprentissage discriminant en traduction automatique. Nous introduisons ensuite plus précisément (section 3), notre modèle de traduction et discutons plusieurs détails d'implantation. Les sections ultérieures sont consacrées respectivement à deux aspects pratiques : le problème des références non atteignables (Section 4), puis la conception d'un ensemble performant de descripteurs (section 5). Nous décrivons à la section 6 les principaux résultats expérimentaux obtenus sur la tâche de traduction français-anglais utilisant les données du corpus BTEC. Les sections conclusives permettent finalement de positionner notre travail par rapport à l'état de l'art (section 7), puis de présenter brièvement diverses extensions de cette approche.

2 Apprentissage discriminant en traduction statistique

2.1 Inférence

Comme expliqué supra, les STS modélisent le processus de génération d'une traduction sous la forme d'une successions d'étapes de dérivation. Ainsi, dans l'approche à base de n-gramme (Mariño *et al.*, 2006; Crego et Mariño, 2007), sur laquelle nous nous appuyons principalement dans cet article, les traductions sont engendrées de la manière suivante [2] :

1. la phrase source est réordonnée de manière non-déterministe et transformée en un graphe de réordonnancement ;
2. ce graphe est ensuite étendu en considérant toutes les décompositions possibles de la phrase source en *segments* ;

2. Les dérivations des systèmes à base de segment telles que formulées dans (Koehn *et al.*, 2007) ou dans (Kumar *et al.*, 2006) utilisent essentiellement le même ensemble de variables latentes, alors que le modèle hiérarchique de Chiang (2005) utilise les dérivations d'une grammaire hors-contexte synchrone.

3. un modèle de traduction est alors appliqué sur cette entrée étendue, de manière à générer le graphe de recherche de toutes les traductions possibles ;
4. ce graphe est finalement parcouru pour rechercher la traduction de meilleur score.

Chaque hypothèse de traduction $\mathbf{t}$ d'une phrase source $\mathbf{s}$ est ainsi associée à une ou plusieurs dérivations latentes $\mathbf{a}$, où $\mathbf{a}$ représente toutes les variables qui sont impliquées dans les étapes de dérivation (1–3). Chaque triplet $(\mathbf{s}, \mathbf{a}, \mathbf{t})$ est représenté comme un vecteur de caractéristiques $\mathbf{G}$ et son score est calculé par le produit scalaire ($\boldsymbol{\theta}$ est le vecteur de paramètres) :

$$\Phi(\mathbf{s}, \mathbf{a}, \mathbf{t}) = \boldsymbol{\theta}^T \mathbf{G}(\mathbf{s}, \mathbf{a}, \mathbf{t}) \tag{1}$$

Il est aisé de transformer ces scores en probabilités en définissant $p_{\boldsymbol{\theta}}(\mathbf{t}, \mathbf{a} \mid \mathbf{s})$ comme suit :

$$p_{\boldsymbol{\theta}}(\mathbf{t}, \mathbf{a} \mid \mathbf{s}) = \frac{\exp\left(\boldsymbol{\theta}^\top \mathbf{G}(\mathbf{t}, \mathbf{a}, \mathbf{s})\right)}{\sum\limits_{\substack{\mathbf{a}' \in .\mathscr{A}(\mathbf{s}) \\ \mathbf{t}' \in \mathscr{T}(\mathbf{a}', \mathbf{s})}} \exp\left(\boldsymbol{\theta}^\top \mathbf{G}(\mathbf{t}', \mathbf{a}', \mathbf{s})\right)}, \tag{2}$$

où $\mathscr{A}(\mathbf{s})$ est l'ensemble de toutes les assignations possibles des variables latentes et où $\mathscr{T}(\mathbf{a}, \mathbf{s})$ représente l'ensemble de toutes les traductions possibles de $\mathbf{s}$ sachant une assignation particulière de $\mathbf{a}$. La probabilité conditionnelle de $\mathbf{t}$ sachant $\mathbf{s}$ s'en déduit par sommation selon :

$$p_{\boldsymbol{\theta}}(\mathbf{t} \mid \mathbf{s}) = \sum_{\mathbf{a} \in .\mathscr{A}(\mathbf{s})} p_{\boldsymbol{\theta}}(\mathbf{t}, \mathbf{a} \mid \mathbf{s}) \tag{3}$$

La règle d'inférence optimale consiste à choisir la meilleure traduction $\mathbf{t}^*$ pour $\mathbf{s}$ selon :

$$\mathbf{t}^* = \arg\max_{\mathbf{t}} p_{\boldsymbol{\theta}}(\mathbf{t} \mid \mathbf{s}) = \arg\max_{\mathbf{t}} \sum_{\mathbf{a} \in .\mathscr{A}(\mathbf{s})} p_{\boldsymbol{\theta}}(\mathbf{t}, \mathbf{a} \mid \mathbf{s}), \tag{4}$$

La somme (4) devant être réalisée pour chaque traduction possible $\mathbf{t}$, il s'avère toutefois que l'inférence ainsi définie donne lieu à un problème combinatoire NP-difficile. C'est pourquoi la plupart des systèmes de traduction se contentent d'utiliser une approximation, dite *de Viterbi*, qui correspond à l'utilisation de la règle d'inférence plus simple suivante :

$$\mathbf{t}^* = h_{\boldsymbol{\theta}}(\mathbf{s}) = \arg\max_{\mathbf{t}, \mathbf{a}} p_{\boldsymbol{\theta}}(\mathbf{t}, \mathbf{a} \mid \mathbf{s}), \tag{5}$$

On notera que cette règle permet également de recouvrer la dérivation latente optimale $\mathbf{a}^*$.

2.2 Apprentissage discriminant (version standard)

Le modèle introduit ci-dessus est suffisamment général pour rendre compte de la plupart des systèmes à base de segments et peut être instancié de multiples manières. Comme mentionné plus haut, l'architecture la plus utilisée (Koehn, 2010) s'appuie sur plusieurs couches de modélisation statistique. La première couche correspond à l'estimation, sur des corpus monolingues et/ou parallèles, d'un ensemble de modèles probabilistes, les plus importants étant le modèle de langue, le modèle de traduction et le modèle de réordonnancement, qui sont usuellement estimés au

maximum de vraisemblance [3]. Chaque modèle ainsi calculé correspond à une composante du vecteur **G** introduit en (1) : $G_k(\mathbf{t},\mathbf{a},\mathbf{s})$ est le score, pour le $k^{\text{ème}}$ modèle, de la dérivation **a** qui produit **t** à partir de **s**.

Le seconde couche d'apprentissage est effectuée de *manière discriminante* : son implantation la plus utilisée, *Minimum Error Rate Training (MERT)* (Och, 2003), consiste à résoudre le problème d'optimisation suivant : étant donné un ensemble de couples entrée/sortie $\{(\mathbf{s}^n,\mathbf{t}^n), n=1\ldots N\}$, trouver les paramètres optimaux satisfaisant :

$$\boldsymbol{\theta}^* = \arg\max_{\text{BLEU}} \Big(\{(\mathbf{s}^n, h_{\boldsymbol{\theta}}(\mathbf{s}^n), \mathbf{t}^n), n=1\ldots N\} \Big), \tag{6}$$

où BLEU (Papineni *et al.*, 2002) est une mesure automatique de la qualité de traduction. La résolution de ce problème n'est en pratique faisable que lorsque $\boldsymbol{\theta}$ est de dimension réduite. On retiendra également que sa résolution requiert d'identifier une dérivation optimale, par exemple celle qui conduit au meilleur score BLEU parmi une liste de n meilleurs candidats.

3 Apprentissage discriminant (version intégrée)

Dans cette section, nous proposons une autre instanciation du cadre d'apprentissage décrit ci-dessus, dans lequel l'estimation de **tous les paramètres du modèle** est réalisée de manière intégrée et discriminante, ce qui constitue une différence fondamentale avec la plupart des autres approches discriminantes en traduction statistique (voir également la discussion de la section 7). Comme on le verra, notre modèle s'inspire largement du modèle des champs aléatoires conditionnels (CRF, voir (Lafferty *et al.*, 2001)) qu'il étend de plusieurs manières.

3.1 Apprentissage du modèle

L'apprentissage est réalisé en maximisant la (log) vraisemblance conditionnelle définie par :

$$\mathscr{L}(\boldsymbol{\theta}) = \sum_n \left[\log \sum_{\mathbf{a}\in\mathscr{A}(\mathbf{s}^n)} \exp\left(\boldsymbol{\theta}^\top \mathbf{G}(\mathbf{t}^n,\mathbf{a},\mathbf{s}^n)\right) - \log \sum_{\substack{\mathbf{a}\in\mathscr{A}(\mathbf{s}^n)\\ \mathbf{t}\in\mathscr{T}(\mathbf{a},\mathbf{s}^n)}} \exp\left(\boldsymbol{\theta}^\top \mathbf{G}(\mathbf{t},\mathbf{a},\mathbf{s}^n)\right) \right] \tag{7}$$

Comme expliqué ci-dessus, nous ne considérons que des dérivations **a** qui sont rationnelles et correspondent à la série d'étapes (1-3) introduites à la section 2.

L'introduction de variables latentes fait que la fonction objectif (7) n'est pas convexe, contrairement au cas des CRF standard (Sutton et McCallum, 2006). En pratique, son optimisation reste possible, et, si elle ne conduit qu'à des optimums locaux, les résultats obtenus ne semblent pas trop dépendants des conditions initiales. Comme détaillé à la section 3.3, l'optimisation repose

3. L'estimation du modèle de traduction est en fait plus complexe et implique un empilement d'étapes heuristiques : calcul d'alignements de mots asymétriques, symétrisation des alignements, extraction et évaluation des couples bilingues de segments, etc.

sur un algorithme de descente de gradient qui demande de calculer le gradient suivant :

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} = \sum_n \left[\sum_{\mathbf{a} \in \mathcal{A}(\mathbf{s}^n)} G_k(\mathbf{t}^n, \mathbf{a}, \mathbf{s}^n) - \sum_{\substack{\mathbf{a} \in \mathcal{A}(\mathbf{s}^n) \\ \mathbf{t} \in \mathcal{T}(\mathbf{a}, \mathbf{s}^n)}} \boldsymbol{\theta}_k G_k(\mathbf{t}, \mathbf{a}, \mathbf{s}^n) p_{\boldsymbol{\theta}}(\mathbf{t}, \mathbf{a} \mid \mathbf{s}^n), \right] \quad (8)$$

Dans cette équation, les deux termes représentent respectivement l'espérance empirique et l'espérance pour le modèle calculées sur l'ensemble des données d'apprentissage.

En théorie, dans cette approche, les composants de G peuvent tester des propriétés arbitraires du triplet $(\mathbf{t}^n, \mathbf{a}, \mathbf{s}^n)$; en pratique, toutefois, le choix des caractéristiques a un impact sur la complexité computationnelle des algorithmes d'inférence et d'apprentissage. Dans cette étude, nous nous limitons à des caractéristiques de *portée locale*, reproduisant les dépendances locales qui sont modélisées dans un CRF linéaire standard (Lafferty *et al.*, 2001) : la portée d'une caractéristique ne peut excéder un bigramme de segments cibles. Cette restriction permet de calculer efficacement les deux termes de l'équation (8) en utilisant une variante de l'algorithme *forward-backward* (voir, par exemple, (Dreyer *et al.*, 2008) pour une présentation détaillée de l'apprentissage de modèles globalement normalisés avec des variables latentes).

La fonction objectif est usuellement augmentée d'un terme de régularisation pour limiter les problèmes de sur-apprentissage. Dans cette étude, nous utilisons une régularisation ℓ_1 (Tibshirani, 1996), qui permet d'aboutir à des ensembles de paramètres « creux » et donc implicitement de sélectionner les caractéristiques les plus importantes.

3.2 Inférence

L'inférence est définie par l'équation (4), qui exige en principe de sommer sur toutes les variables latentes pour calculer l'hypothèse de traduction optimale. Cette tâche correspond à un problème NP-difficile ; en pratique, il est possible de l'approximer de manière efficace en élaguant et déterminisant l'espace de recherche, comme expliqué section 3.3.

Il est important de noter que les dépendances qui sont modélisées se limitent à des bigrammes de segments cibles qui ne fournissent qu'une très mauvaise approximation des contraintes syntaxiques à respecter en langue cible. Pour compenser cette faiblesse, nous utilisons durant l'inférence un modèle de langue n-gramme d'un ordre supérieur à deux, ce qui permet d'améliorer sensiblement les performances du seul modèle CRF.

3.3 Détails d'implantation

Transducteurs finis Toutes les opérations nécessaires pour réaliser l'apprentissage et l'inférence sont implantées comme des opérations standard sur des transducteurs pondérés. Pour l'essentiel, nous nous reposons sur les fonctionnalités génériques de la bibliothèque OpenFst (Allauzen *et al.*, 2007) ; pour des raisons d'efficacité, nous avons toutefois réimplanté une version optimisée de l'algorithme *forward-backward* et des interactions avec le modèle de traduction.

Pour l'essentiel, notre décodeur est donc implanté comme une cascade de transducteurs finis, impliquant les étapes suivantes : (i) réordonnancement et segmentation de la phrase source ;

(ii) application du modèle de traduction et (optionnellement) (iii) composition avec un modèle de langue cible, une architecture très similaire à celle proposée par (Kumar *et al.*, 2006). Plus précisément, étant donnés un modèle de réordonnancement et un inventaire d'unités, nous dérivons les transducteurs suivants :
– I, un accepteur pour la phrase source $\mathbf{s}$;
– R, qui implémente les règles de réordonnancement ;
– C, qui regroupe des séquences de mots sources en segments de taille variable ;
– T, qui réalise l'association entre segments sources et toutes leurs traductions possibles.
Si l'on note $\circ$ l'opération de composition entre transducteurs, alors $S = I \circ R \circ C \circ T$ définit l'espace de recherche qui est utilisé pour l'apprentissage et pour l'inférence.

Apprentissage du modèle L'optimisation de la log-vraisemblance (équation (7)) est effectuée en utilisant l'algorithme R-Prop (Riedmiller et Braun, 1993) qui implémente une stratégie de descente de gradient adaptée à l'optimisation des modèles log-linéaires à grande échelle. Cet algorithme demande de calculer les espérances définies par l'équation (8). Le premier terme est obtenu en collectant les statistiques pour les caractéristiques actives dans le transducteur défini par $S \circ O$, où O est l'accepteur représentant la traduction de référence. La seconde espérance demande de collecter ces mêmes statistiques sur l'intégralité de l'espace de recherche S, de nouveau par application de l'algorithme *forward-backward*.

Inférence des traductions Dans notre implantation, l'inférence est réalisée en quatre temps : S est tout d'abord reparcouru pour calculer la probabilité *a posteriori* de chaque arc ; nous déterminisons ensuite le transducteur ainsi repondéré, ce qui a pour effet de réaliser la somme impliquée par l'équation (4) ; le score du modèle de langue est ensuite ajouté simplement par une opération de composition pondérée (le poids du modèle de langue est obtenu par une recherche sur un corpus de développement) ; finalement, le meilleur chemin dans le transducteur est extrait. Dans la mesure où l'opération de déterminisation est la plus exigeante en temps, nous la réalisons de manière approchée en ne considérant à ce stade que les n-meilleures hypothèses de l'espace de recherche. La somme (4) est donc seulement calculée sur ces n meilleures hypothèses, ce qui ne semble pas trop limitant en pratique.

4 Les références non atteignables

Un problème spécifique qui se pose dans le cadre de l'apprentissage discriminant pour la traduction est celui de la *non atteignabilité des références*, correspondant aux situations où la traduction de référence ne peut pas être dérivée dans le modèle (Liang *et al.*, 2006) . Cela arrive, par exemple, quand on utilise un inventaire d'unités bilingues trop restreint, ou que l'on considère des réordonnancements trop limités. Il est alors possible qu'une traduction de référence contienne une traduction inconnue d'un mot source connu, ou bien des déplacements de groupes qui vont au-delà de ceux qu'explore le décodeur. Un remède radical consiste alors à supprimer ces cas problématiques du corpus d'apprentissage (Blunsom *et al.*, 2008; Dyer et Resnik, 2010) – conduisant ainsi à abandonner de nombreux exemples potentiellement utiles.

Une autre solution simple, utilisée dans plusieurs études, consiste à utiliser des *pseudo-références oracles*, qui sont les meilleures hypothèses (au sens de la métrique d'évaluation) réellement

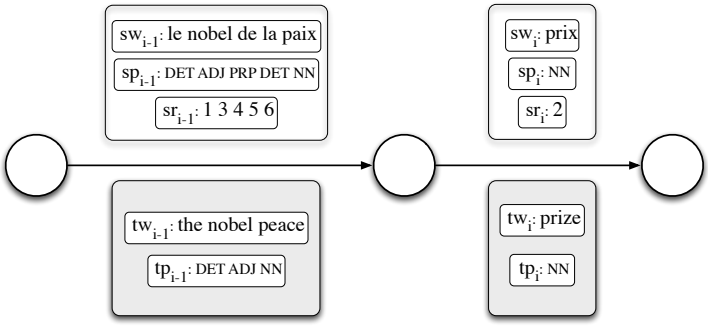

FIGURE 1 – Deux arcs consécutifs dans l'espace de recherche : informations dont sont dérivées les caractéristiques.

présentes dans l'espace de recherche. Comme le soulignent les auteurs de (Liang *et al.*, 2006), une bonne traduction (au sens de la métrique) peut toutefois s'appuyer localement sur des étapes de dérivation qui ont une très faible probabilité, et ne devraient pas être utilisées comme exemple. Cette observation suggère des stratégies plus prudentes, selon lesquelles l'oracle est choisi parmi les n hypothèses les plus probables (au sens du modèle). Des stratégies hybrides sont également envisageables, selon lesquelles les oracles sont choisis parmi les hypothèses qui sont à la fois proches de la référence et bien évaluées par le modèle.

Diverses stratégies ont été implantées et évaluées dans nos expériences. La première consiste à supprimer tous les exemples non atteignables. Une seconde alternative consiste à augmenter le modèle localement de façon à compenser les lacunes du modèle : dans notre architecture, cela revient par exemple à simuler l'existence d'unités de traduction qui seraient manquantes. La troisième alternative, qui s'est avérée la meilleure, consiste à utiliser des pseudo-référence oracles calculées non pas sur des listes de n-meilleures hypothèses, mais sur l'intégralité de l'espace de recherche (voir (Sokolov *et al.*, 2012) pour une description des algorithmes permettant de calculer ces oracles lorsque la métrique mesurant la qualité des traductions est le score BLEU).

5 Caractéristiques

Pour présenter les principales caractéristiques utilisées dans notre modèle, reportons nous à la Figure 1 qui donne à voir deux arcs consécutifs dans l'espace de recherche S. Chaque arc porte toutes les informations nécessaires au calcul des caractéristiques : les segments source et cible (sw et tw), les séquences de parties du discours (POS) associées (sp et tp), ainsi que les positions originales (avant réordonnancement) des mots sources (sr). Les indices i et $i-1$ servent seulement à noter le fait que l'arc $i-1$ précède l'arc i et constitue le contexte gauche de l'arc courant. Étant donnée cette représentation, il est possible de définir des caractéristiques binaires qui chacune teste une propriété particulière du couple d'arcs $(i-1, i)$. La liste de descripteurs de base est dans le tableau 1.

La forme des caractéristiques de base simule les dépendances d'un modèle de langue bigramme en cible : ainsi, les caractéristiques notées **LM :*** correspondent à des modèles unigrammes

LM :uni-tphr	$\mathbb{I}(tw_i = tw)$					
LM :uni-tpos	$\mathbb{I}(tp_i = tp)$					
LM :big-tphr	$\mathbb{I}(tw_i = tw$	$\wedge$	$tw_{i-1} = tw')$			
LM :big-tpos	$\mathbb{I}(tp_i = tp$	$\wedge$	$tp_{i-1} = tp')$			
TM :ci-phrp	$\mathbb{I}(tw_i = tw$	$\wedge$	$sw_i = sw)$			
TM :ci-posp	$\mathbb{I}(tp_i = tp$	$\wedge$	$sp_i = sp)$			
TM :ci-mixp	$\mathbb{I}(tw_i = tw$	$\wedge$	$sp_i = sp)$			
TM :cd-phrs	$\mathbb{I}(tw_i = tw$	$\wedge$	$sw_i = sw$	$\wedge$	$sw_{i-1} = sw')$	
TM :cd-poss	$\mathbb{I}(tp_i = tp$	$\wedge$	$sp_i = sp$	$\wedge$	$sp_{i-1} = sp')$	
TM :cd-phrt	$\mathbb{I}(tw_i = tw$	$\wedge$	$tw_{i-1} = tw'$	$\wedge$	$sw_i = sw)$	
TM :cd-post	$\mathbb{I}(tp_i = tp$	$\wedge$	$tp_{i-1} = tp'$	$\wedge$	$sp_i = sp)$	

TABLE 1 – Caractéristiques de base avec les notations de la Figure 1.

et bigrammes respectivement de segments de mots et de POS. L'autre groupe principal de caractéristiques, noté **TM :*** modélise les relations de traduction. Il comprend des caractéristiques indépendantes du contexte (qui ne regardent que le segment courant) **TM :ci-phrp** et **TM :ci-posp** qui testent respectivement l'association d'un segment source avec un segment cible au niveau lexical et au niveau des étiquettes grammaticales ; les caractéristiques dépendantes du contexte gauche (**TM :cd***) sont plus spécifiques et prennent en compte le segment précédent.

Les réordonnancements sont évalués par un autre ensemble de caractéristiques intégrant des tests qui simulent les modèles de réordonnancement lexicalisés standard (Tillman, 2004; Crego *et al.*, 2011). Dans notre approche, cinq classes de déplacements sont considérées : '*monotone*', '*swap*', *'left discontinuous*', '*right discontinuous*' and *'other*'. Pour chaque catégorie, deux caractéristiques testent respectivement l'association avec le segment cible et la séquence de POS correspondante.

Nous utilisons finalement deux caractéristiques supplémentaires : la première est toujours active et permet d' « encourager » l'insertion de nouveaux segments dans la phrase en construction. La seconde est relative aux recopies, et est active quand les mots source et cibles sont identiques, ce qui permet de « récompenser » la recopie d'un mot source inconnu dans la cible, une stratégie qui s'avère souvent gagnante. (pour les noms propres, les dates, etc)

6 Expériences

6.1 Corpus et système de base

La tâche de traduction considérée utilise les données parallèles français/anglais du *Basic Traveling Expression Corpus* (BTEC), tel qu'il a été utilisé dans les évaluations internationales de l'atelier IWSLT. Ce corpus contient des phrases semblables à ce que l'on peut trouver dans des guides touristiques, en plusieurs langues (Takezawa *et al.*, 2002). Le corpus de développement est *devel03*, qui contient 506 lignes et 16 références par lignes ; nous utilisons comme jeu de test les corpus *test09* et *test10* qui contiennent respectivement 469 lignes et 464 lignes, avec 7 traductions de référence. Notre mesure principale de la qualité des traductions est le score BLEU calculé en utilisant le maximum de références disponibles. Cette tâche est souvent considérée comme

relativement simple, au vu de la longueur moyenne des phrases, et du relativement faible nombre de données d'apprentissage : l'utilisation de notre cadre intégré d'apprentissage discriminant implique toutefois d'entrainer le système sur environ 20K phrases, soit 10 fois plus que ce qui est usuellement utilisé pour entrainer discrimininativement (avec MERT) des systèmes standard sur des « grosses » tâches.

Notre système de base est *n*-code [4] (Crego *et al.*, 2011), une implantation domaine public de l'approche à base de *n*-gram introduite dans (Mariño *et al.*, 2006). Selon cette approche, le modèle de traduction est représenté par un transducteur stochastique correspondant à un modèle *n*-gramme de *couples de segments* ($n = 3$ dans nos expériences). L'entraînement d'un tel modèle demande au préalable de réordonner les phrases sources pour reproduire l'ordre des mots en langue cible. Ce réordonnancement est également effectué par un transducteur fini non-déterministe, qui utilise des informations morpho-syntaxiques (calculées par le TreeTagger [5]) pour généraliser les règles de reordonnancement au niveau des POS.

Le modèle complet utilise quatorze caractéristiques : le modèle de traduction, un modèle (trigramme) de langue cible, quatre modèles d'alignement lexicalisés [6], six modèles de réordonnancement lexicalisés (Tillman, 2004; Crego *et al.*, 2011) ; un modèle de distortion ainsi que deux modèles supplémentaires qui encouragent respectivement la génération de mots et de segments cibles. Les poids des différents modèles sont estimés en utilisant la procédure MERT (Och, 2003).

Pour toutes nos expériences, le modèle de langue cible est estimé en utilisant un lissage de Kneser-Ney modifié (Chen et Goodman, 1996). Notons également que tous les systèmes évalués ci-desous utilisent le même inventaire d'unités de traduction et le même mécanisme de réordonnancement, qui sont ceux construits pour le système de base, ce qui permet une comparaison équitable entre systèmes. Tous nos résultats respectent les contraintes de la tâche spécifiée pour la campagne IWSLT 2010, et peuvent être directement comparés avec les résultats de (Paul *et al.*, 2010).

6.2 Résultats

Le tableau 2 récapitule nos principaux résultats en termes de scores BLEU. Première observation : le système de base est légèrement meilleur que le meilleur système ayant participé à la campagne IWSLT 2010 ((Paul *et al.*, 2010, p.20) mentionne un score de 52,69 pour le meilleur système). Trois configurations différentes du système discriminant sont comparées : la première réalise l'inférence en utilisant l'approximation dite de Viterbi (équation (5)) et obtient des performances très inférieures au système *n*-code ; la seconde configuration implante la procédure de marginalisation approximative décrite à la section 3.3, ce qui permet une légère amélioration des performances. La troisième configuration (+LM cible) intègre également, comme c'est le cas pour les systèmes *n*-code, un modèle trigramme en langue cible et permet de surpasser légèrement le système de base sur les deux jeux de test.

À l'initialisation de l'apprentissage, le modèle de traduction contient environ 13 millions de caractéristiques. Au terme de l'apprentissage, seulement 4% sont sélectionnées, les autres étant éliminées du modèle sous l'action de la pénalité ℓ_1. Au total, apprendre un tel modèle prend une dizaine de minutes sur un gros serveur de calcul et la traduction du jeu de test ne demande que

4. Accessible depuis `ncode.limsi.fr/`.
5. Accessible depuis `www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`.
6. Ces modèles sont similaires à ceux qui sont utilisés dans les systèmes standard.

Configuration	*devel03*	*test09*	*test10*
Système *n*-code			
Modèle de traduction 2g	68,7	61,1	–
Modèle de traduction 3g	68,0	61,6	53,4
Système entraîné discriminativement			
Inférence Viterbi	64,0	58,8	51,5
+ marginalisation	64,7	59,3	52,0
+ LM cible	67,7	**61,7**	**53,9**

TABLE 2 – Performance des systèmes de traduction (scores BLEU).

deux ou trois minutes.

Le tableau 3 compare les différentes manière de gérer les références non atteignables (voir section 4) [7]. Il apparait clairement que supprimer les exemples pour laquelle la référence est non atteignable est la pire, puisque dans notre cas elle conduit à abandonner environ 8% des exemples. Augmenter localement le modèle de traduction permet d'améliorer très nettement les résultats ; la stratégie la plus efficace consiste toutefois à utiliser des pseudo-références oracles.

Configuration	*devel03*	*test09*
Suppression	59,2	52,6
Augmentation locale	62,4	56,4
Pseudo-références	64,0	58,8

TABLE 3 – Différentes manières de gérer les références non atteignables

7 Discussion

L'approche standard en traduction statistique, rappelée à la section 2, réalise l'apprentissage des modèles en deux étapes successives et repose grandement sur une procédure d'optimisation *ad hoc*, connue sous le nom de MERT (Och, 2003). De nombreux travaux récents ont tenté de reformuler MERT comme un problème d'apprentissage standard, afin de le rendre plus robuste à des situations où le nombre de caractéristiques est grand. MERT a ainsi été reformulé par exemple comme un problème d'apprentissage structuré (Tillmann et Zhang, 2006; Watanabe *et al.*, 2007; Cherry et Foster, 2012) ou encore comme un problème d'apprentissage de fonction d'ordonnancement (Hopkins et May, 2011). Ces approches visent à améliorer la seconde étape de l'apprentissage, sans remettre toutefois en cause l'architecture globale du système. Par comparaison, les travaux cherchant à définir des cadres d'apprentissage intégrés sont plus rares.

Un pas important dans cette direction est le modèle de Liang *et al.* (2006), qui utilise un perceptron structuré pour apprendre les paramètres du modèle. Cette approche requiert toutefois de fixer la valeur des variables latentes impliquées dans une dérivation aussi bien à l'apprentissage que lors de l'inférence, là où nous utilisons une procédure de marginalisation. Une autre différence avec notre travail est l'utilisation d'un modèle de réordonnancement plus simple. Une autre source

7. Ces résultats sont obtenus pour la stratégie d'inférence dite de Viterbi.

d'inspiration est le travail décrit dans (Blunsom *et al.*, 2008), qui décrit une version discriminante du modèle hiérarchique de Chiang (2005). Comme dans notre approche, l'apprentissage repose sur l'optimisation de la log-vraisemblance conditionnelle, impliquant de sommer sur toutes les dérivations (hors-contexte) d'une traduction. La complexité de l'algorithme de parsage sous-jacent au calcul du gradient $O(|\mathbf{t}|^3|\mathbf{s}|^3)$ semble toutefois limiter l'approche à des phrases courtes [8]. Une différence significative avec notre travail est la gestion des références non-atteignables, qui sont purement et simplement supprimées du corpus d'apprentissage. Le travail plus récent de Dyer et Resnik (2010) mérite enfin mention, puisqu'il utilise la même architecture que la nôtre, à la différence près que le modèle de réordonnancement est un modèle hors-contexte plutôt que rationnel. Ce travail est toutefois focalisé sur l'apprentissage du modèle de réordonnancement et conserve le besoin d'entraîner séparément le modèle de traduction.

En résumé, notre approche se distingue de la plupart des approches discriminantes en traduction statistique en ceci que nous réalisons l'apprentissage simultané de **tous les paramètres du modèle de manière intégrée**, par optimisation d'une fonction objectif bien fondée théoriquement (la log-vraisemblance conditionnelle régularisée).

Conclusion

Nous avons présenté une architecture intégrée pour réaliser en une seule étape l'apprentissage discriminant de tous les paramètres des systèmes de traduction. Cette architecture, qui emprunte beaucoup à des techniques d'apprentissage bien connues, permet d'introduire dans le modèle un très grand nombre de caractéristiques. En utilisant cette architecture, nous avons développé un système qui surpasse un système de base très performant sur la tâche de traduction du BTEC. Notons en particulier que notre approche conduit à des meilleurs scores BLEU que *n*-code, qui est pourtant spécifiquement entraîné pour optimiser cette métrique. Une propriété importante de notre approche est son caractère modulaire, puisqu'elle s'accomode d'inventaires d'unités et de modèles de réordonnancement variés.

Dans le futur, la priorité principale sera de réaliser des expériences sur des tâches plus complexes, impliquant à la fois de plus gros corpus d'apprentissage et des langues plus éloignées. Diverses améliorations du modèle présenté ici sont également à l'étude : ainsi l'utilisation de modèles de réordonnancement plus puissants, à la manière de Dyer et Resnik (2010) ; l'utilisation d'unités de traduction avec trous, poursuivant les propositions de (Simard *et al.*, 2005; Crego et Yvon, 2009) ; ou l'utilisation d'une fonction objectif intégrant une mesure plus directe de la qualité de traduction, à l'instar par exemple de (Gimpel et Smith, 2010).

Remerciements

Ce travail a été partiellement financé par OSEO dans le cadre du programme Quaero.

8. Les résultats de (Blunsom *et al.*, 2008) utilisent des phrases de moins de 15 mots.

Références

ALLAUZEN, A. et YVON, F. (2011). Méthodes statistiques pour la traduction automatique. *In* GAUSSIER, E. et YVON, F., éditeurs : *Modèles statistiques pour l'accès à l'information textuelle*, chapitre 7, pages 271–356. Hermès, Paris.

ALLAUZEN, C., RILEY, M., SCHALKWYK, J., SKUT, W. et MOHRI, M. (2007). OpenFst : A general and efficient weighted finite-state transducer library. *In Proc. of CIAA*, pages 11–23.

BLUNSOM, P., COHN, T. et OSBORNE, M. (2008). A discriminative latent variable model for statistical machine translation. *In Proc. ACL/HLT*, pages 200–208.

CHEN, S. F. et GOODMAN, J. T. (1996). An empirical study of smoothing techniques for language modeling. *In Proc. ACL*, pages 310–318.

CHERRY, C. et FOSTER, G. (2012). Batch tuning strategies for statistical machine translation. *In Proc. of the 2012 Conf. HLT-NAACL*, pages 427–436.

CHIANG, D. (2005). A hierarchical phrase-based model for statistical machine translation. *In Proc. ACL*, pages 263–270.

CHIANG, D., KNIGHT, K. et WANG, W. (2009). 11,001 new features for statistical machine translation. *In Proc. NAACL/HLT*, pages 218–226.

CREGO, J. M. et MARIÑO, J. B. (2007). Improving SMT by coupling reordering and decoding. *Machine Translation*, 20(3):199–215.

CREGO, J. M. et YVON, F. (2009). Gappy translation units under left-to-right SMT decoding. *In Proc. of the conf. EAMT*, pages 66–73.

CREGO, J. M., YVON, F. et MARIÑO, J. B. (2011). N-code : an open-source Bilingual N-gram SMT Toolkit. *Prague Bulletin of Mathematical Linguistics*, 96:49–58.

DREYER, M., SMITH, J. et EISNER, J. (2008). Latent-variable modeling of string transductions with finite-state methods. *In Proc. EMNLP*, pages 1080–1089.

DYER, C. et RESNIK, P. (2010). Context-free reordering, finite-state translation. *In Proc NAACL/HLT*, pages 858–866, Los Angeles.

GIMPEL, K. et SMITH, N. A. (2010). Softmax-margin CRFs : training log-linear models with cost functions. *In Proc. HLT-NAACL*, HLT '10, pages 733–736.

HOPKINS, M. et MAY, J. (2011). Tuning as ranking. *In Proc. EMNLP*, pages 1352–1362.

KOEHN, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

KOEHN, P., HOANG, H., BIRCH, A., CALLISON-BURCH, C., FEDERICO, M., BERTOLDI, N., COWAN, B., SHEN, W., MORAN, C., ZENS, R., DYER, C., BOJAR, O., CONSTANTIN, A. et HERBST, E. (2007). Moses : Open source toolkit for statistical machine translation. *In Proc. ACL*, pages 177–180.

KOEHN, P., OCH, F. J. et MARCU, D. (2003). Statistical phrase-based translation. *In Proc of the conf. HLT-NAACL*, pages 127–133.

KUMAR, S. et BYRNE, W. (2005). Local phrase reordering models for statistical machine translation. *In Proc. HLT-EMNLP*, pages 161–168.

KUMAR, S., DENG, Y. et BYRNE, W. (2006). A weighted finite state transducer translation template model for statistical machine translation. *Natural Language Engineering*, 12(1):35–75.

LAFFERTY, J., MCCALLUM, A. et PEREIRA, F. (2001). Conditional random fields : probabilistic models for segmenting and labeling sequence data. *In Proc. ICML*, pages 282–289.

LAVERGNE, T., ALLAUZEN, A., CREGO, J. M. et YVON, F. (2011). From n-gram-based to CRF-based translation models. *In Proc. WMT*, pages 542–553.

LIANG, P., BOUCHARD-CÔTÉ, A., KLEIN, D. et TASKAR, B. (2006). An end-to-end discriminative approach to machine translation. *In Proc. ACL*, pages 761–768.

MARIÑO, J. B., BANCHS, R. E., CREGO, J. M., de GISPERT, A., LAMBERT, P., FONOLLOSA, J. A. et COSTA-JUSSÀ, M. R. (2006). N-gram-based machine translation. *Comp. Ling.*, 32(4):527–549.

OCH, F. J. (2003). Minimum error rate training in statistical machine translation. *In Proc. ACL*, pages 160–167.

PAPINENI, K., ROUKOS, S., WARD, T. et ZHU, W.-J. (2002). BLEU : a method for automatic evaluation of machine translation. *In Proc. ACL*, pages 311–318.

PAUL, M., FEDERICO, M. et STÜCKER, S. (2010). Overview of the IWSLT 2010 Evaluation Campaign. *In* FEDERICO, M., LANE, I., PAUL, M. et YVON, F., éditeurs : *Proc. IWSLT*, pages 3–27.

RIEDMILLER, M. et BRAUN, H. (1993). A direct adaptive method for faster backpropagation learning : The RPROP algorithm. *In Proc. ICNN*, pages 586–591.

SIMARD, M., CANCEDDA, N., CAVESTRO, B., DYMETMAN, M., GAUSSIER, E., GOUTTE, C., YAMADA, K., LANGLAIS, P. et MAUSER, A. (2005). Translating with non-contiguous phrases. *In Proc. HLT-EMNLP*, pages 755–762.

SIMIANER, P., RIEZLER, S. et DYER, C. (2012). Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT. *In Proc. ACL*, pages 11–21.

SOKOLOV, A., WISNIEWSKI, G. et YVON, F. (2012). Computing lattice BLEU oracle scores for machine translation. *In Proc. EACL*, pages 120–129.

SUTTON, C. et MCCALLUM, A. (2006). An introduction to conditional random fields for relational learning. *In* GETOOR, L. et TASKAR, B., éditeurs : *Introduction to Statistical Relational Learning*. The MIT Press.

TAKEZAWA, T., SUMITA, E., SUGAYA, F., YAMAMOTO, H. et YAMAMOTO, S. (2002). Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. *In Proc. of LREC*, volume 1, pages 147–152.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the Lasso. *J.R.Statist.Soc.B*, 58(1):267–288.

TILLMAN, C. (2004). A unigram orientation model for statistical machine translation. *In* DUMAIS, S., MARCU, D. et ROUKOS, S., éditeurs : *HLT-NAACL 2004 : Short Papers*, pages 101–104.

TILLMANN, C. et NEY, H. (2003). Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Comp. Ling.*, 29(1):97–133.

TILLMANN, C. et ZHANG, T. (2006). A discriminative global training algorithm for statistical mt. *In Proc. of the conf. of the ACL*, pages 721–728.

WATANABE, T., SUZUKI, J., TSUKADA, H. et ISOZAKI, H. (2007). Online large-margin training for statistical machine translation. *In Proc. of EMNLP-CoNLL*, pages 764–773.

ZENS, R., OCH, F. J. et NEY, H. (2002). Phrase-based statistical machine translation. *In* JARKE, M., KOEHLER, J. et LAKEMEYER, G., éditeurs : *KI-2002 : Advances in AI*, volume 2479 de *LNAI*, pages 18–32. Springer.