Chapitre 7

Méthodes Statistiques pour la Traduction Automatique

7.1. Introduction

.

7.1.1. *La traduction automatique à l'heure de l'Internet*

La traduction automatique constitue un domaine de recherche déjà ancien, qui a conduit au développement d'un certain nombre de technologies qui ont largement diffusé aussi bien auprès des professionnels que des particuliers. On date, en général, le début des recherches sur la traduction automatique du début des années 50, sous l'influence en particulier du mathématicien américain Warren Weaver, qui, comparant une langue étrangère avec un code (secret), orienta dès l'origine les travaux en traduction sous l'angle du décodage. L'histoire du développement des technologies de traduction automatique n'a pas été linéaire, loin s'en faut, et a suscité bien des espoirs et des désillusions : on se reportera, en particulier, aux nombreux écrits de John Hutchins sur la question (par exemple [HUT 92, HUT 01, HUT 03]). Si l'on s'est souvent gaussé de la lenteur des progrès des systèmes de traduction automatique, et si l'on a souvent moqué les traductions produites, la situation réelle du domaine, telle qu'elle prévalait au milieu des années 90, était pourtant loin d'être aussi sombre.

Chapitre rédigé par Alexandre ALLAUZEN et François YVON.

Il existait, d'une part, une offre de traduction entièrement automatisée, à destination majoritairement d'industriels ou d'institutions telles que la communauté européenne, le gouvernement canadien, etc : cette offre, dont le système Systran est un des exemples les plus aboutis, s'appuyait principalement sur des systèmes à base de règles, intégrant un savoir linguistique très fin accumulé au fil des années sur quelques domaines restreints. L'exemple du système MÉTÉO [CHE 78], développé pour la traduction automatique de bulletins météorologiques entre l'anglais et le français est souvent cité comme représentatif de la gamme d'applications, principalement techniques, à laquelle ces systèmes s'adressaient. La recherche en *traduction à base de règles* progressait régulièrement, au rythme du développement des outils d'analyse automatique et de génération de textes. Il existait, d'autre part, une seconde offre, à destination cette fois-ci des traducteurs professionnels, d'outils intégrés d'aide à la traduction, et comportant en particulier des dictionnaires de termes, des dictionnaires bilingues et des *mémoires de traduction*[1]. Un des intérêts de ces outils est, d'une part, de fournir aux traducteurs un moyen d'éviter d'avoir à retraduire des énoncés ou des passages figurant de manière répétitive dans les documents qu'ils ont à traiter et d'autre part de garantir la cohérence de la traduction des termes. L'exploitation automatique des mémoires de traduction, dans le but d'augmenter leur rendement, faisait également l'objet de recherches et de développements actifs, à la suite des propositions de [NAG 04], dans ce cadre du paradigme de la *traduction à base d'exemples* (voir, par exemple, [CAR 03] pour des développements récents de ce domaine de recherche). Pour le grand public enfin, la principale offre portait sur les dictionnaires bilingues ainsi que sur les dispositifs transportables d'aide à la traduction, principalement destinés à être utilisés dans un contexte touristique.

L'accélération de la mondialisation des échanges économiques, des migrations de populations, et surtout l'avènement d'un média universel, Internet, ont, en l'espace d'une décennie, singulièrement bouleversé ce paysage. Via Internet, chacun est confronté au besoin d'accéder à des informations qui ne sont pas disponibles dans sa langue d'origine, voire à communiquer électroniquement avec d'autres internautes parlant ou écrivant des langues différentes. Le rendement inégal de l'utilisation de telle ou telle langue pour leurs publications conduit, de surcroît, un nombre croissant d'utilisateurs de ces nouvelles technologies à rédiger et à diffuser leurs informations dans une langue qui n'est pas leur langue maternelle. Ces différents facteurs convergent pour susciter de nouvelles demande pour des services de traduction, qui ne peuvent être réellement satisfaites que par des outils de traduction automatique.

Seule la traduction automatique permet, en effet, de répondre à ces nouvelles demandes[2] qui se distinguent, par de multiples aspects, des demandes auxquelles font

1. Voir par exemple l'offre de la société Trados http://www.trados.com/fr/.

2. Une indication de l'importance de cette demande est le nombre de requêtes que les services de traduction en ligne offerts par des éditeurs de logiciels (Systran (http://

usuellement face les traducteurs humains. Celles-ci se caractérisent en premier lieu par leur non-solvabilité ; par leur caractère d'urgence ; par l'imprévisibilité des types de textes à traduire, dont certains ont d'ailleurs une durée de vie très courte (SMS, *tweets*, *e-mails*, échanges dans une *chat room*, commentaires sur un forum, bribes de blogs ou de pages web statiques, dépêches d'agence de presses, articles scientifiques) ; et par la diversité des thèmes abordés dans les textes à traduire (des textes d'actualité aux correspondances privées en passant par des documentations techniques de logiciels libres). En contre-partie, ces nouveaux usages de la traduction automatique semblent s'accomoder de traductions qui seraient grossières ou imparfaites. Cette demande correspond à plusieurs grands types de pratiques : d'une part le besoin d'accéder à des informations qui ne seraient disponibles sur le Web que dans une langue étrangère[3], d'autre part l'envie de communiquer électroniquement avec autrui par delà les barrières linguistiques, enfin la nécessité de publier des documents dans une langue qui n'est pas celle dans laquelle ils ont été écrits, afin de leur assurer une plus grande diffusion. Si le souci de disposer d'une traduction de qualité va croissant dans ces trois gammes d'application, on notera qu'aucune de ces applications n'est véritablement en concurrence avec les services de traduction de qualité professionnelle. Des services professionnels qu'il serait d'ailleurs parfois difficile de trouver : rien qu'en Europe, ce n'est pas moins de 23 langues officielles qui sont utilisées pour échanger et diffuser des informations, soit un potentiel de 506 paires de langues possibles.

Il est heureux, mais pas totalement fortuit, que l'accroissement exponentiel du nombre des documents numériques accessibles en ligne qui a résulté du développement d'Internet ait permis de fournir une partie de la réponse à ce problème. Ce développement des corpus électroniques, conjoint à la maturation de techniques statistiques de traitement des langues [MAN 99], y compris de traduction automatique, a permis l'avènement d'une troisième famille de technologies de traduction automatique : les méthodes probabilistes, qui se basent principalement sur l'exploitation statistique de grands corpus de documents électroniques. Le travail séminal des équipe d'IBM [BRO 90], au début des années 90, a posé les fondations d'une nouvelle conception de la traduction automatique, qui, au terme d'une rapide maturation, s'est rapidement imposée comme la réponse la plus adaptée et la plus efficace à ces nouvelles formes de demandes. En premier lieu, l'utilisation de méthodes statistiques garantit une certaine forme de robustesse face à des énoncés bruités ou mal formés qui caractérisent de nombreux sous-genres de documents électroniques. En second lieu, l'utilisation de modèles probabilistes fournit un cadre mathématique bien adapté pour

---

`www.systran.com`), Reverso/Softissimo (`http://reverso.net`), Microsoft (`http://www.microsofttranslator.com/`)) ou par les principaux moteurs de recherche (Google (`http://translate.google.com/`), Yahoo (`http://fr.babelfish.yahoo.com/`)) qui servent quotidiennement des millions de requêtes de traduction.

3. On se reportera sur ces problèmes de *recherche documentaire cross-lingue*, par exemple, à [NIE 09].

aborder les nombreuses ambiguïtés qui font souvent toute la difficulté des traductions. Enfin, le recours à des outils statistiques[4] s'avère être le moyen le plus efficace pour tirer parti, dans un contexte de traduction, de ces immenses corpus disponibles sur Internet.

Il est donc finalement tout naturel d'accorder, dans ce livre consacré aux méthodes probabilistes pour l'accès à l'information, une large part aux méthodes de traduction automatique statistiques. Celles-ci sont appelés à révolutionner notre mode d'accès à l'information, en démultipliant le nombre de sources documentaires, qui, bien que rédigées dans des langues qui nous sont étrangères, nous deviennent accessibles par le truchement d'outils de traduction automatique.

7.1.2. *Organisation du chapitre*

Ce chapitre est organisé comme suit. La section 7.2 est consacrée à une présentation simplifiée de la première génération des modèles de traduction automatique statistique : *modèles à base de mots* (*word-based models*). Cette présentation a pour principaux objectifs de présenter schématiquement les principes de fonctionnement des systèmes de traduction automatique statistique dans un cadre simplifié et de faire comprendre les motivations qui ont conduit à l'élaboration des modèles de l'état de l'art, les *modèles à base de segments* (*phrase-based models*). Nous en profiterons pour introduire un certain nombre de notations et de concepts qui seront utiles dans la suite de la présentation. Nous détaillons dans les sections qui suivent le fonctionnement de ces systèmes, en abordant en premier lieu la question de la conception et de l'estimation des *modèles de traduction* à la section 7.3 : conceptuellement, ces modèles sont semblables à de gros dictionnaires bilingues qui enregistreraient des correspondances entre des groupes des mots en source et en cible. Nous y présentons tout d'abord les techniques de construction d'alignements mot-à-mot (Sections 7.3.1 et 7.3.2), à partir desquels les modèles de traduction sont construits et estimés (Section 7.3.3). La section 7.4 est consacrée à la *modélisation des déplacements*, ou plutôt, à la modélisation des divergences dans l'ordre des mots entre les langages source et cible. Ici encore, nous exposerons dans un premier temps les principes de ces modèles, avant de traiter de la question de leur apprentissage. Au terme de cette section, nous aurons fini avec l'apprentissage des modèles statistiques embarqués dans un système de traduction automatique et nous pourrons nous intéresser à leur utilisation, c'est-à-dire au calcul de nouvelles traductions en réponse à des entrées de l'utilisateur. Ce problème difficile,

4. Qu'on ne s'y méprenne pas : d'autres approches de la traduction automatique continuent à être explorées et développées, souvent avec succès : la traduction automatique experte à base de règles comme la traduction automatique à base d'exemples continuent de fournir des approches alternatives et complémentaires aux approches entièrement statistiques que nous avons choisi de présenter dans ce chapitre.

ainsi que différentes manières pour l'aborder, sont présentés à la section 7.5. Nous abordons enfin, à la section 7.6, la question de l'évaluation des systèmes de traduction statistique : que peut-on dire de la qualité des traductions qu'ils produisent ? Nous verrons que cette question n'a pas de réponse unique, tant la question de l'évaluation est difficile. Nous discuterons également de l'automatisation de ces évaluations, qui permet de quantifier de manière plus objective, mais probablement plus grossière, la qualité des traductions.

Deux brèves sections d'approfondissement terminent ce chapitre. Dans la première (Section 7.7), nous revenons sur le modèle standard pour mettre en évidence quelques-unes de ses limitations et discuter de propositions récentes pour l'améliorer ; la seconde (Section 7.8) donne une série d'indications pratiques et de pointeurs vers des ressources (logiciels et données) qui sont librement accessibles sur Internet et à partir desquelles les lecteurs intéressés pourront aisément construire leurs propres systèmes de traduction et expérimenter avec ces systèmes.

7.1.2.1. *Remarques terminologiques*

La littérature anglaise récente en traduction automatique, et en particulier en traduction automatique statistique, est parfois obscurcie par un usage approximatif de termes dont le sens semblait pourtant établi en informatique ou en linguistique. Il n'est donc probablement pas inutile de donner quelques indication sur les choix (de traduction !) que nous avons opérés dans ce chapitre. Selon l'usage, nous appelons *langue source* (respectivement *langue cible*) pour désigner la langue depuis laquelle (respectivement vers laquelle) on traduit. Une raison (historique) de l'ambiguïté possible de ces termes est liée à l'utilisation d'un modèle de canal bruité (voir la section 7.2.1), qui fait que, pour construire un système traduisant de la langue A vers la langue B, on est amené à construire un modèle de traduction pour traduire de B vers A.

Un problème plus délicat est posé par le terme anglais '*phrase*', qui est utilisé pour qualifier la famille de modèles ('*phrase-based models*') qui sont étudiés dans ce chapitre. '*phrase*' se traduit le plus souvent par 'syntagme' dans un contexte linguistique ; pour marquer le caractère non-linguistique de ces unités, nous avons privilégié l'emploi de 'segment' pour désigner une unité de ces modèles et de 'bisegment' pour désigner un couple apparié de 'segments' en langue source et cible.

Le terme anglais '*alignment*' est également problématique : en informatique, en particulier en reconnaissance des formes, ce terme désigne une relation entre deux séquences de symboles qui préserve l'ordre relatif des séquences. Nous verrons (section 7.3.1) qu'en traduction automatique, ce terme a un sens plus général de relation entre séquences source et cible ; nous avons ici conservé la traduction la plus simple d' 'alignement' qui nous semble finalement prêter assez peu à confusion.

7.2. La traduction probabiliste, une vue d'ensemble

Dans cette section, nous présentons une vue d'ensemble de la traduction probabiliste, en nous focalisant sur les modèles dits « à base de mots » (*word based models*, qui ont constitué, jusqu'au début des années 2000, l'état de l'art en la matière. L'objectif principal n'est pas tant de donner une compréhension technique de ces modèles très complexes que de mettre en évidence les principales hypothèses de modélisation qu'ils intègrent. Après avoir présenté les principes du modèle standard à la section 7.2.1, nous discuterons à la section 7.2.2 d'un certain nombre de réalités linguistiques auxquelles ces modèles ont dû faire face et qui justifient, d'une certaine manière, leur abandon et leur remplacement par les modèles à base de segments, qui en sont les héritiers directs, et qui seront introduits à la section 7.2.3.

Avant de débuter, il n'est pas inutile de bien rappeler que les modèles de traduction que nous présentons dans ce chapitre sont extrêmement limités dans leur comportement. En particulier, ils ne savent traduire que des phrases indépendamment les unes des autres, le plus souvent en produisant une phrase en sortie pour chaque phrase en entrée. Ils sont donc tout à fait incapables de prendre en compte des dimensions liées à l'analyse du discours, et encore moins capables de rendre compte en langue cible de nuances liées au style, au registre d'élocution etc, qui font souvent toute la difficulté du travail de traduction pour les humains. Ces différents aspects font l'objet de recherches actives (dont certaines seront évoquées à la section 7.7) ; pour l'exposé qui suit, ces aspects seront entièrement ignorés. Toutefois, par facilité, nous continuerons par désigner sous le vocable de *systèmes de traduction automatique* ce qu'il serait certainement plus juste d'appeler *systèmes statistiques de traduction automatique littérale phrase-à-phrase*.

7.2.1. *Traduction probabiliste : le modèle standard*

Introduisons tout d'abord quelques notations qui seront utilisées tout au long du chapitre. Une phrase en langue source est notée $\mathbf{f}$, et contient J mots $f_1 \dots f_J$, une phrase en langue cible est notée $\mathbf{e} = e_1 \dots e_I$[5]. Envisagée comme un problème de décision probabiliste, la traduction consiste à résoudre le problème suivant :

$$\mathbf{e}^* = \underset{\mathbf{e}}{\operatorname{argmax}} \, \mathrm{P}(\mathbf{e}|\mathbf{f}) \tag{7.1}$$

On notera que cette formulation simple dissimule deux problèmes considérables : un problème de modélisation, puisqu'il faut donner un sens à la probabilité de n'importe

5. Ces notations sont historiques : dans les travaux de [BRO 90], la traduction s'effectuait du "français" (**f***rench*) vers "l'anglais" (**e***nglish*). Depuis, cette notation s'est étendue et peut faire référence à **f***oreign*, n'importe quelle langue autre que l'anglais.

quelle phrase en langue cible, conditionnellement à toute phrase en langue source ; un problème de recherche, puisqu'il faudra pour trouver la solution de ce problème effectuer une recherche parmi toutes les phrases en langue cible. Historiquement, le problème a plutôt été abordé en le transformant par l'application de la règle de Bayes de la manière suivante :

$$\mathbf{e}^* = \underset{\mathbf{e}}{\operatorname{argmax}} \, \mathrm{P}(\mathbf{f}, \mathbf{e}) = \underset{\mathbf{e}}{\operatorname{argmax}} \, \mathrm{P}(\mathbf{f}|\mathbf{e}) \, \mathrm{P}(\mathbf{e}) \tag{7.2}$$

Ce modèle est connu dans la littérature sous le nom de modèle du *canal bruité* (*noisy chanel model*), parce qu'il décompose la génération d'un couple $(\mathbf{f}, \mathbf{e})$ en deux étapes : le choix d'une phrase $\mathbf{e}$ selon $\mathrm{P}(\mathbf{e})$, phrase qui est « déformée » en $\mathbf{f}$ par son passage dans un canal de transmission dont les déformations sont modélisées par $\mathrm{P}(\mathbf{f}|\mathbf{e})$. L'argument donné par [BRO 93b] pour « inverser » l'équation (7.1) est le suivant. La phrase à traduire $\mathbf{f}$ est supposée grammaticalement bien formée, et l'on souhaite naturellement construire une traduction $\mathbf{e}$ qui soit également bien formée. Le modèle de probabilité impliqué dans (7.1) doit être tel que pour toute phrase source $\mathbf{f}$, il concentre la masse de probabilité sur des phrases en langue cible qui sont à la fois bien formées et qui sont des traductions de $\mathbf{f}$. En essayant de maximiser $\mathrm{P}(\mathbf{f}|\mathbf{e}) \, \mathrm{P}(\mathbf{e})$ on décompose un problème difficile en deux problèmes que l'on espère plus simples. D'un côté, le développement d'un modèle de traduction garantissant que $\mathrm{P}(\mathbf{f}|\mathbf{e})$ est élevé, pour toute phrase cible, grammaticale ou non, appariée avec $\mathbf{f}$. Ce modèle est estimé sur des *corpus bilingues parallèles*, c'est-à-dire qui ont été préalablement alignés phrase à phrase[6]. De l'autre, le développement d'une modélisation probabiliste des phrases en langue cible fournit le terme $\mathrm{P}(\mathbf{e})$. Cette modélisation doit concentrer sa probabilité sur les phrases grammaticales *indépendamment de la phrase source*. Elle peut donc être réalisée sur des corpus monolingues. Sous ces hypothèses, une vue d'ensemble du développement d'un système de traduction statistique est représentée à la figure 7.1.

Au-delà de la justification théorique esquissée ci-dessus, le découplage réalisé par l'équation (7.2) présente un intérêt pratique indéniable puisqu'il sépare le problème de modélisation en deux sous-problèmes qui peuvent être abordés de manière relativement indépendante. En particulier, il existe une littérature considérable sur la question de la construction et l'estimation de modèles probabilistes de langue, qui sont des outils très utiles dans de nombreuses applications de traitement automatique des langues (voir, par exemple [MAN 99] ou [JEL 97]) et de recherche d'information (voir le chapitre **??**). Même si ce problème est loin d'être résolu de manière complètement satisfaisante, il existe aujourd'hui des techniques et des outils éprouvés qui peuvent être directement utilisés dans les systèmes de traduction statistique. D'autre part, ces modèles étant construits indépendamment de la tâche de traduction, en particulier indépendamment de la connaissance de la langue source, ils peuvent être estimés sur

6. L section 7.8 donne des informations sur la disponibilité de telles ressources.

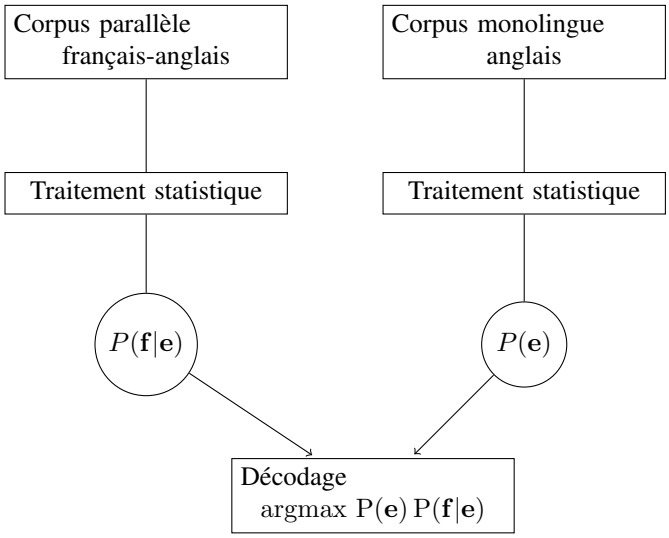


Figure 7.1 – Vue d'ensemble d'un système de traduction probabiliste

des *corpus monolingues*, qui sont aujourd'hui disponibles dans des proportions qui excèdent de plusieurs ordres de grandeur celles des corpus bilingues. Dans ce chapitre, nous ferons l'hypothèse que ces modèles décomposent la probabilité d'une séquence cible $\mathbf{e} = e_1 \dots e_I$ en faisant l'hypothèse de dépendance markovienne d'ordre 1[7] entre les mots de la phrase, soit :

$$\mathrm{P}(\mathbf{e}) = \mathrm{P}(e_1) \prod_{i=2}^{I} \mathrm{P}(e_i|e_{i-1}) \tag{7.3}$$

On se reportera au chapitre A, notamment à la section A.4 pour des compléments touchant à l'estimation de ces modèles, ainsi qu'à des généralisation à des dépendances d'ordre supérieur. Dans la suite de cette section, nous nous intéressons donc principalement aux difficultés posées par la modélisation du terme $\mathrm{P}(\mathbf{e}|\mathbf{f})$, en particulier lorsqu'on envisage la traduction comme un processus de transformation mot-à-mot.

7. En pratique la dépendance est d'ordre supérieur, au moins d'ordre 2 ou 3.

7.2.2. *La traduction mot-à-mot et ses difficultés*

Une première approche simpliste pour modéliser le processus de traduction consiste à envisager qu'il s'effectue mot-à-mot. On écrirait alors :

$$\mathrm{P}(\mathbf{f}|\mathbf{e}) = \prod_{i=1}^{I} \mathrm{P}(f_i|e_i) \tag{7.4}$$

7.2.2.1. *Ambiguités lexicales*

Un premier problème bien connu en traduction est celui du non-déterminisme des appariemments mot-à-mot, qui peut résulter soit d'une homographie quasi accidentelle, soit de cas de polysémie. La première situation est illustrée par des mots comme '*avocat*', qui désigne un juriste ou bien un fruit, et qui se traduira selon les cas en anglais par '*lawyer*' ou '*avocado*' ; ou comme '*mousse*' (le jeune marin ou l'amas de bulle). La seconde situation est illustrée de nouveau par mousse, dont le sens littéral de plante se traduit différemment du sens métaphorique d'amas de bulle. Il n'est pas besoin d'insister sur ce problème abondamment documenté (voir par exemple [FUC 96]), sinon pour examiner comment le choix de l'une ou l'autre traduction est opéré dans le cadre de l'équation (7.4). À supposer que '*lawyer*' ou '*avocado*' soient les deux traductions possibles de '*avocat*', on peut penser que les deux termes $\mathrm{P}(\text{'}avocat\text{'}|\text{'}lawyer\text{'})$ et $\mathrm{P}(\text{'}avocat\text{'}|\text{'}avocado\text{'})$ seront comparables, conférant des valeurs proches à des traductions qui emploieraient l'un ou l'autre de ces termes. Le choix entre l'une et l'autre traduction sera alors déterminé en fonction de la qualité de leur insertion dans le modèle de langue cible. Ce fonctionnement est un peu paradoxal puisque le choix de traduction ne prend en compte que très indirectement le contexte en langue source, point sur lequel nous reviendrons ultérieurement à la section 7.7.1.

$\mathbf{f}$	'*cet avocat est mon défenseur*'
$\mathbf{e}_1$ $\mathrm{P}(\mathbf{f}\|\mathbf{e}_1)$	'*this lawyer is my defender*' $\mathrm{P}(\text{'}cet\text{'}\|\text{'}this\text{'}) \times \mathrm{P}(\text{'}avocat\text{'}\|\text{'}lawyer\text{'}) \times \mathrm{P}(\text{'}est\text{'}\|\text{'}is\text{'}) \times \mathrm{P}(\text{'}mon\text{'}\|\text{'}my\text{'})$ $\times \mathrm{P}(\text{'}défenseur\text{'}\|\text{'}defender\text{'})$
$\mathbf{e}_2$ $\mathrm{P}(\mathbf{f}\|\mathbf{e}_2)$	'*this avocado is my defender*' $\mathrm{P}(\text{'}cet\text{'}\|\text{'}this\text{'}) \times \mathrm{P}(\text{'}avocat\text{'}\|\text{'}avocado\text{'}) \times \mathrm{P}(\text{'}est\text{'}\|\text{'}is\text{'}) \times \mathrm{P}(\text{'}mon\text{'}\|\text{'}my\text{'})$ $\times \mathrm{P}(\text{'}défenseur\text{'}\|\text{'}defender\text{'})$

Si les probabilités de $\mathbf{f}$ sachant $\mathbf{e}_1$ ou $\mathbf{e}_2$ sont proches, le choix entre l'une ou l'autre des deux hypothèses repose uniquement sur les valeurs relatives de $\mathrm{P}(\mathbf{e}_1)$ et $\mathrm{P}(\mathbf{e}_2)$.

Tableau 7.1 – Un choix lexical difficile

Notons que cette situation de non-déterminisme existe même en l'absence de polysémie en langue source : '*river*' (dans le sens de 'cours d'eau') se traduit en français '*fleuve*' ou '*rivière*' selon que le cours se jette ou non dans la mer ; cette distinction n'est pas présente en anglais, rendant la traduction du mot ambiguë. Lorsque l'on

considère comme unités de traduction des formes fléchies (par exemple des verbes conjugués), ce non-déterministe est souvent amplifié. En premier lieu à cause, à nouveau, des homographies accidentelles, telles que par exemple en français '*souris*', qui correspond à plusieurs formes du verbe '*sourire*' (au présent, au passé simple) et désigne également l'animal (au singulier et au pluriel) et le périphérique informatique[8]. En second lieu, parce que les langues source et cible n'opèrent pas, ou plutôt ne marquent pas formellement les mêmes distinctions. Certaines langues, par exemple, distinguent trois genres pour les noms (masculin, neutre, féminin), là où d'autres n'en distinguent que deux ; la situation est similaire (voire plus complexe) pour ce qui concerne la conjugaison : là où l'anglais ne manipule que cinq formes différentes pour chaque verbe, d'autres langues en manipulent plusieurs dizaines (le français, plus généralement les langues latines), voire des milliers (le géorgien). Moins la langue source marque de distinctions, plus le modèle de langage en langue cible sera conduit à jouer un rôle important dans la sélection de la bonne traduction.

Le modèle défini par l'équation (7.4) décompose la probabilité d'un appariemment global entre $\mathbf{f}$ et $\mathbf{e}$ comme un produit d'appariemments mot-à-mot et repose donc sur une hypothèse de compositionnalité des traductions[9]. Cette hypothèse est ici encore très simplificatrice, puisque l'on sait que le sens (et donc souvent la traduction) de nombreux groupes de mots ne se déduit pas directement du sens de chacun des composants. Il existe toute une gradation de situations qui vont du composé idiomatique complètement figé, tel que '*casser sa pipe*' (en anglais '*kick the bucket*'), à des termes techniques dont le sens n'est plus complètement compositionnel. Le modèle précédent attribue pourtant au premier des appariements une probabilité $\mathrm{P}(\text{'}kick\text{'}|\text{'}casser\text{'}) \times \mathrm{P}(\text{'}the\text{'}|\text{'}sa\text{'}) \times \mathrm{P}(\text{'}bucket\text{'}|\text{'}pipe\text{'})$ bien plus faible que celui d'une traduction littérale comme '*break his pipe*'. La traduction littérale est plus probable car la décomposition de sa probabilité contient des facteurs dont la valeur est supérieure ($\mathrm{P}(\text{'}break\text{'}|\text{'}casser\text{'})$ et $\mathrm{P}(\text{'}pipe\text{'}|\text{'}pipe\text{'})$).

7.2.2.2. *Un mot pour un mot*

Une seconde critique majeure que l'on peut formuler à l'encontre du modèle (7.4) est qu'il ne donne une probabilité non nulle que dans la situation où les phrases cible et source ont même longueur. Cette hypothèse est exagérément simpliste. Tout d'abord parce que les langues diffèrent, parfois profondément, sur la notion même de mot. Pour prendre un seul exemple de divergence majeure, là où certaines langues expriment des significations complexes par combinaisons syntaxiques de mots isolés (c'est le cas en première approximation pour le français et l'anglais), d'autres (en particulier les langues *agglutinantes*, comme le finnois, le hongrois, ou encore les langues

8. Cette dernière ambiguïté est préservée en anglais.

9. Dans le modèle, les variables aléatoires correspondants à la traduction de chacun des mots sont indépendantes.

turques) ont également recours à des combinaisons d'unités de signification plus petites que le mot. Ces processus de construction de formes complexes sont illustrés par les exemples de la figure 7.2.

forme	décomposition	glose
görüntülenebilir	görüntüle+n+ebil+ir	visualiser+Passif+pouvoir+être (cela peut être visualisé)
sakarlıklarından	sakar+lık+ları+ndan	maladroite+esse+leur+du fait de (du fait de leur maladresse)

Figure 7.2 – Formes complexes en turc

Des problèmes de même nature se posent pour des langues plus familières, telles que l'allemand, qui forme également des mots composés par concaténation [10] : à partir de '*die Flashe*' ('*la bouteille*'), '*das Wasser*' ('*l'eau*'), '*das Mineral*' (le minéral), on forme ' *die Wasserflasche*' (la bouteille à eau), '*die Mineralwasserflasche*' ('*la bouteille à eau minérale*').

Même dans le cas où langues cible et source sont plus proches, comme le français et l'anglais, les configurations dans lesquelles un mot français se traduit en plusieurs mots anglais, ou le contraire, sont légion. Les mécanismes de formation de mots-composés dans ces deux langues conduisent en effet à des situations où un mot anglais correspond à plusieurs mots en français : '*laptop*'-'*ordinateur portable*', '*widescreen*'-'*grand écran*', etc. Un examen de quelques correspondances entre groupes verbaux illustre la situation inverse. Ainsi, là où l'anglais a recours à l'ajout de particules pour marquer des nuances de sens d'un verbe, le français aura recours à des mots différents : ainsi '*switch on/off*' pour '*allumer/éteindre*' ; '*break out/up/down*' pour respectivement '*(s')évader*', '*(s')achever*', '*briser*' etc. Là où le français utilise une conjugaison particulière pour marquer le conditionnel, l'anglais utilise des verbes modaux, donnant lieu à des appariemments comme '*(I) would come*'-'*(je) viendrais*' etc. On notera que, facteur aggravant, il arrive que les différents mots anglais traduisant un même mot français ne soient pas adjacents : c'est, par exemple, le cas avec les particules comme dans '*She said she would never take her husband back again*', où les deux fragments qui constituent le groupe verbal '*take ... back*' sont séparés ici par l'objet direct du verbe.

Une conséquence est que le passage d'un type de langue à l'autre ne peut se modéliser correctement mot-à-mot. Les modèles de traduction mot-à-mot doivent donc être enrichis pour pouvoir prendre en compte ces phénomènes. Une manière de procéder consiste à concevoir un modèle dans lequel les appariemments de tailles variables

10. La réalité est un peu plus complexe puisque des lettres peuvent s'insérer à la frontière des mots entrant dans la composition.

sont le résultat d'un mécanisme en deux temps : pour chaque mot f_j, on choisit tout d'abord le nombre n_j de mots avec lesquels il doit s'apparier, puis on applique un modèle mot-à-mot pour chacune des n_j copies de f_j. On aurait alors :

$$\mathrm{P}(\mathbf{f}|\mathbf{e}) = \prod_{i=1}^{I} \mathrm{P}(n_i|e_j) \prod_{j=1}^{n_i} \mathrm{P}(e_i|f_j) \tag{7.5}$$

Cette modélisation est notamment utilisée dans les modèles de traduction mot-à-mot développés à IBM dans les années 90 [BRO 90, BRO 93b][11]. En autorisant qu'un mot en langue cible n'ait pas d'équivalent de traduction en langue source (ce qui revient à autoriser que $n_i = 0$), et en ajoutant un mot « vide » fictif dans $\mathbf{e}$ on prendra également en compte d'autres situations communes, comme par exemple la suppression du déterminant indéfini '*des*' qui n'a pas toujours de correspondant en anglais, ou inversement la possibilité d'insérer l'auxiliaire '*do*' quand il n'a pas de correspondant en français.

7.2.2.3. *Restructurations syntaxiques*

La troisième critique, sans doute la plus importante, que l'on peut adresser aux modèles précédents est l'hypothèse que les mots en relation de traduction mutuelle sont disposés dans le même ordre. Cette hypothèse est naturellement inappropriée, sauf peut-être pour des langues très proches, et les différences entre l'ordre des mots en langue cible et langue source sont des faits très communs, qui s'observent aussi bien au niveau de l'agencement général des phrases qu'au niveau plus local de l'organisation des syntagmes nominaux ou verbaux.

Au niveau macroscopique, les classifications typologiques des langues distinguent plusieurs grands types en fonction de l'ordre relatif du sujet (S), de l'objet (0) et du verbe (S) : ainsi le français, comme l'anglais et les langues romanes, est SVO, car l'ordre canonique dans la phrase est sujet-verbe-complément ; le japonais et le turc sont principalement SOV, l'arabe classique est VSO, etc. Il existe, de surcroît, des langues dans lesquelles l'ordre des constituants n'est pas aussi strictement régi, qui sont dites à ordre libre (par exemple le latin) : c'est, dans ces langues, les marques de cas qui vont permettre d'identifier les fonctions des différents constituants qui dépendent du verbe. La modélisation de la relation de traduction entre deux langues appartenant à des familles différentes nécessite donc de prendre en compte les nécessaires restructurations (ou réordonnancements) syntaxiques qui sont opérés lorsque l'on passe de l'une à l'autre. Dans la mesure où ces réordonnancements portent sur les constituants, ils peuvent se traduire, au niveau des mots, par des déplacements très importants.

---

11. Ces modèles, toujours utilisés pour entraîner des modèles de traduction, sont présentés en détail à la section 7.3.

Ces divergences syntaxiques entre langues peuvent également exister à un niveau plus microscopique, celui de la structure interne des syntagmes : ainsi le français et l'anglais diffèrent-ils, entre autres, par le positionnement relatif des adjectifs (et dans une certaine mesure également des adverbes) par rapport aux noms (respectivement aux verbes) qu'ils modifient. On parlera dans ce cas de réordonnancements locaux.

Compte-tenu de la multiplicité et de la finesse de ces divergences entre langues, l'intégration de ces différences d'ordre dans les modèles probabilistes de traduction mot-à-mot est une réelle gageure. Les propositions « historiques » [BRO 90], qui n'ont jamais été réellement améliorées, s'appuient sur un modèle simpliste, qui ajoute une troisième étape de génération au modèle décrit par l'équation (7.5). Une fois choisi le nombre de mots appariés avec chaque mot e_i, on choisit pour chaque mot la position dans laquelle il figure dans $\mathbf{f}$, en utilisant des distributions qui garantissent que les « grands » déplacements seront moins probables que les plus petits.

7.2.3. *Vers les modèles à base de segments*

En réponse à certaines de ces difficultés des modèles de traductions de mots, les modèles à base de segments, initialement décrits dans [ZEN 02, KOE 03b, OCH 04] se sont progressivement imposés comme une alternative performante. Contrairement aux modèles de mots, les unités de traduction considérés dans ces modèles sont des segments de taille variable, et les phrases du langage source sont traduits segments par segments. Cette manière de procéder possède de nombreux avantages :

– en augmentant l'empan des appariemments modélisés, il devient possible de lever bon nombre d'ambiguïtés lexicales, en particulier celles qui se résolvent par examen du contexte local en langue source. Reprenons l'exemple précédent de la traduction depuis l'anglais vers le français. Nous avons évoqué le fait que, dans un modèle basé sur les mots, une forme telle que '*think*' est très ambiguë et peut se projeter dans pratiquement toutes les formes du présent du verbe '*penser*'. Par comparaison, le nombre d'appariements possibles pour '*we think*' est bien plus réduit. Il en ira de même pour les adjectifs, qui sont invariables en genre et en nombre en anglais, mais doivent s'accorder avec le nom en français : si '*small*' est ambigu et peut se traduire par '*petit*', '*petite*' etc., '*small car*' ne l'est plus et a pour seule traduction '*petite voiture*' ;

– en autorisant des appariements entre groupes, elle permet très simplement de prendre en compte les cas où le nombre de mots n'est pas le même en cible et en source, incluant la plupart des traductions d'idiomes ;

– l'utilisation d'unités de taille variable permet une modélisation implicite des réordonnancements locaux. En mémorisant qu'une traduction possible de '*natural language processing*' est '*traitement des langues naturelles*', on se dispense effectivement de modéliser le changement relatif d'ordre qui intervient au sein du groupe nominal lorsque l'on passe d'une langue à l'autre.

Du point de vue de la mise en œuvre de systèmes statistiques, ne subsistent alors que deux grands problèmes. Le premier problème concerne la modélisation des appariemments entre bisegments (ces appariemments définissent ce que nous appellerons dans la suite le *modèle de traduction*) et l'estimation de ces modèles à partir de corpus. Le second problème concerne la modélisation des déplacements des bisegments pendant le processus de traduction ; la définition et l'estimation de ces modèles de réordonnancement fait l'objet de la section 7.4. Une fois ces modèles décrits, nous pourrons finalement revenir, à la section 7.5, sur un problème que nous avons délibérément laissé en suspens, à savoir celui de l'interaction de ces modèles avec le modèle de langue en cible et celui de la recherche de la solution à des problèmes combinatoires tels que celui défini par l'équation (7.2).

7.3. Modéliser les traductions de segments

Dans un système de traduction automatique statistique, le modèle de traduction est la source de connaissance principale qui établit le lien entre les deux langues (source et cible). Son rôle est de guider la construction, pour une phrase source, d'un ensemble d'hypothèses de phrases en langue cible. Comme expliqué ci-dessus, l'hypothèse de traduction finale est sélectionner en utilisant un modèle de langue monolingue, dont le rôle est de favoriser les phrases grammaticalement « bien formées ».

La définition d'un modèle de traduction repose sur le choix des unités de traduction : pour chaque unité en langue source, ce modèle fournit des propositions de traduction possibles en langue cible et des scores associés. Alors que les premiers systèmes de traduction automatique statistiques travaillaient sur des mots[12], l'unité de segmentation utilisée dans les systèmes de l'état de l'art est le segment : un groupe de mots contigus. L'association entre un segment source et une traduction possible en cible forme un bisegment. Notons qu'il est possible qu'un segment admette plusieurs traductions alternatives, donnant lieu à plusieurs bisegments partageant le même segment source. Afin de faire un bon usage de ces bisegments, il est nécessaire de leur associer des mesures, par exemple statistiques, qui quantifient la confiance en l'association ainsi réalisée. Le modèle de traduction rassemble donc l'ensemble de ces informations et nous allons maintenant décrire comment ce modèle est estimé automatiquement à partir de corpus bilingues parallèles.

La figure 7.3 représente schématiquement le processus d'estimation d'un modèle de traduction à partir de corpus bilingue parallèle, tel qu'il est décrit dans [ZEN 02, KOE 03b]. L'exemple choisi pour la figure 7.3 est la paire de langues français/anglais.

12. Rappellons que la notion de mot doit ici être entendue comme une forme graphique.

Sur cette figure, nous avons représenté seulement un couple de phrases ; dans la pratique, les corpus parallèles utilisés[13] contiennent entre quelques dizaines de milliers et quelques millions de couples de phrases.

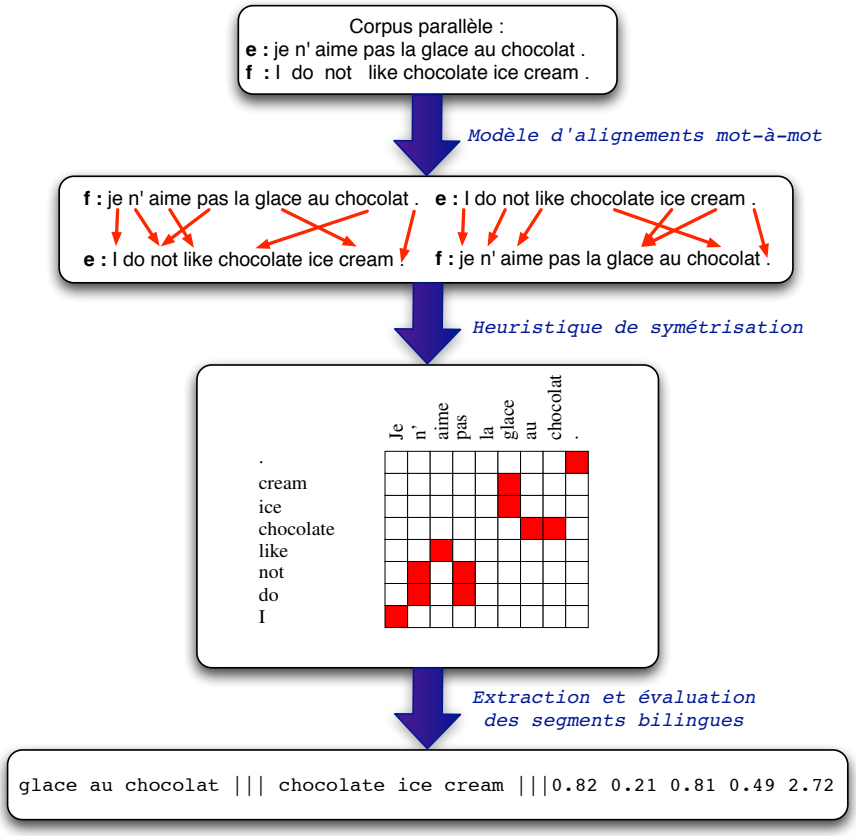


Figure 7.3 – Vue d'ensemble du processus d'extraction des bisegments bilingues

Partant du corpus bilingue, l'objectif est donc l'extraction et l'évaluation de l'ensemble des bisegments. Cette tâche est non-triviale et la construction d'appariements sous-phrastiques à partir de phrases en relation de traduction nécessite des connaissances sur la traduction des unités qui composent ces phrases. Malheureusement, ces

13. Certains de ces corpus sont présentés à la section 7.8.

connaissances ne sont pas disponibles puisque l'objectif est précisément de les aquérir. Une part de la complexité vient du double objectif sous-jacent : repérer les unités pertinentes que sont les segments (*la segmentation*) et mettre en relation les segments qui sont des traductions mutuelles (*l'alignement*).

Afin de simplifier la tâche, l'approche la plus courante considère dans un premier temps le mot comme unité, ce qui a pour effet de fixer une certaine segmentation. La première étape consiste alors à construire des appariements de mots pour chaque couple de phrases, à l'aide de modèles d'alignements mot-à-mot. Cette étape de modélisation, décrite à la section 7.3.1, nécessite à nouveau d'être simplifiée en imposant des restrictions sur les alignements autorisés. Les modèles d'alignement mot-à-mot considérés sont asymétriques et étiquettent les mots d'une phrase avec la position du mot associé dans l'autre phrase, comme représenté sur la figure 7.3. Deux modèles sont donc nécessaires, un dans chaque direction. Pour un couple de phrases donné, deux alignements mot-à-mot sont alors obtenus qui apportent des informations complémentaires. Ces deux alignements sont fusionnés pour créer une représentation unique et symétrique des appariements entre les mots des phrases source et cible.

Disposant des alignements mot-à-mot, il est alors possible d'aborder les questions liées à la segmentation en unités de traduction (les bisegments) et à leur évaluation, questions qui sont abordées à la section 7.3.3. Ainsi, dans l'approche standard que nous allons maintenant détailler, les connaissances utilisées pour la construction d'un modèle de traduction sont extraites automatiquement à partir de modèles statistiques d'alignement mot-à-mot, puis généralisées grâce à des heuristiques au niveau des segments. Cette approche est considérée comme empiriquement la plus efficace, ce qui explique sa popularité ; il en existe de multiples variantes, visant à simplifier un processus qui reste très heuristique et implique des calculs lourds (voir, par exemple, [MAR 02, VEN 03, ZHA 03, VOG 05]).

7.3.1. *Construire des alignements mots-à-mots*

Dans notre modèle, un corpus parallèle est un ensemble de couple de phrases $(\mathbf{f}, \mathbf{e})$: la phrase source est une séquence de J mots $\mathbf{f} = f_1, ..., f_j, ..., f_J$ et la phrase cible est une séquence de I mots $\mathbf{e} = e_1, ..., e_i, ...e_I$. L'alignement mot-à-mot entre $\mathbf{f}$ et $\mathbf{e}$a pour objectif de mettre en relation les mots qui se correspondent, ou dit autrement qui sont traduction l'un de l'autre. Une première représentation de l'alignement entre deux phrases est une matrice binaire $A = (a_{i,j})$, où chaque terme $a_{i,j}$ indique si le mot f_j est aligné avec le mot e_i, comme sur l'exemple représenté Figure 7.4). L'alignement automatique de mots consiste à prédire cette matrice à partir du couple $\mathbf{f}, \mathbf{e}$.

La représentation matricielle d'un alignement mot-à-mot implique cependant $2^{I \times J}$ alignements possibles, autant que de valeurs possibles pour la matrice A. Une manière

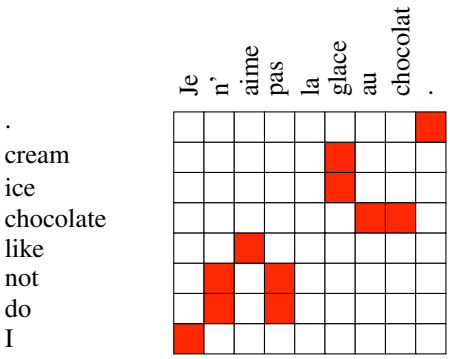


Figure 7.4 – Exemple de matrice d'alignement entre une phrase anglaise et une phrase française. Les termes non nuls de la matrice sont représentés par des carrés pleins. L'ensemble des liens associés à cette matrice est donc $\{(1,1),(2,2),(2,3),(3,4),(4,2),(4,3),(6,6),(6,7),(7,5),(8,5),(9,8)\}$

de restreindre le nombre d'alignements et de simplifier la tâche est de ne considérer que les alignements qui correspondent à des applications de $[0:I]$ dans $[0:J]$: il n'y en a que I^J. Sous cette hypothèse simplificatrice , chaque mot d'une des phrases (appelée dans ce contexte la phrase *source*) est étiqueté avec l'indice de position du mot correspondant dans l'autre phrase (la phrase *cible*). Le modèle n'est plus symétrique, c'est-à-dire que les deux phrases jouent désormais un rôle différent.

Construire un alignement revient alors à déterminer la séquence d'étiquettes[14] $\mathbf{a} = a_1, ..., a_j, ...a_J$ associée à $\mathbf{f}$, où, pour un couple de phrases donné, chaque étiquette a_j est une variable aléatoire ayant comme espace de réalisation les indices des mots dans la phrase cible : $a_j \in \mathcal{A} = 0, 1, ..., I$. Cette étiquette représente l'indice dans la phrase $\mathbf{e}$ avec lequel f_j est aligné : le mot f_j est alors aligné avec le mot e_{a_j}. Si un mot ne peut être aligné, l'étiquette est par convention $a_j = 0$. La notion d'alignement nul implique l'introduction artificielle dans la séquence cible du mot $e_0 =$ `null` ; la séquence cible est désormais constituée de $I + 1$ mots.

14. Formellement, l'alignement correspond à un problème d'étiquetage de séquences analogue à ceux qui sont traités au chapitre 6, et peut être traité avec les mêmes outils.

Les modèles statistiques d'alignement mot-à-mot sont introduits dans [BRO 90]. Dans cet article fondateur, les alignements mot-à-mot représentent les variables cachées du processus de traduction mot-à-mot. Dans la présentation qui suit, nous laissons de côté les aspects liés à la traduction mot-à-mot pour définir directement les modèles d'alignement[15]. Nous nous appuyons également sur la présentation très complète des principaux modèles d'alignement qui est donnée dans [OCH 03b]. Cette présentation comprend les travaux [BRO 90] ainsi que les modèles d'alignement à base de modèles de Markov cachés (*HMM*) proposés par [VOG 96] et différents modèles heuristiques.

La modélisation générative d'un problème d'étiquetage de séquences peut se présenter en décrivant le processus d'estimation de la probabilité conjointe d'une séquence d'observations (la séquence source $\mathbf{f}$) et de la séquence d'étiquettes (les alignements) $\mathbf{a}$ associée. Cette probabilité est ici conditionnée par la phrase cible $\mathbf{e}$ qui est suposée connue et qui contraint l'espace de réalisation des alignements. L'équation (7.6) constitue le point de départ pour définir un modèle génératif :

$$\mathrm{P}(\mathbf{a}, \mathbf{f}|\mathbf{e}) = \mathrm{P}(\mathbf{a}|\mathbf{e})\,\mathrm{P}(\mathbf{f}|\mathbf{a}, \mathbf{e}) \tag{7.6}$$

Cette équation fait intervenir dans son membre de droite deux termes distincts qui correspondent respectivement à la probabilité conditionnelle de la séquence d'étiquettes ($\mathrm{P}(\mathbf{a}|\mathbf{e})$) et à celle de la séquence d'observations ($\mathrm{P}(\mathbf{f}|\mathbf{a}, \mathbf{e})$). Ces termes sont usuellement désignés sous le nom *probabilité de traduction* pour le terme $\mathrm{P}(\mathbf{f}|\mathbf{a}, \mathbf{e})$ et de *probabilité de distorsion* pour $\mathrm{P}(\mathbf{a}|\mathbf{e})$.

Dans [BRO 90], les auteurs proposent 4 modèles génératifs désignés par les acronymes IBM1–IBM4. Ces 4 modèles, de complexité croissante, partent de l'équation (7.6) et proposent chacun un ensemble d'hypothèses simplificatrices permettant d'estimer les paramètres de ces modèles.

7.3.1.1. *Le modèle IBM1*

Afin de simplifier l'équation (7.6), le modèle IBM1 fait les hypothèses suivantes sur le processus de génération :

– chaque alignement a_j est choisi indépendamment parmi tous les alignements possibles répartis de manière uniforme, ainsi :

$$\mathrm{P}(\mathbf{a}|\mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(a_j|\mathbf{e}) = \prod_{j=1}^{J} \frac{1}{I+1} = \frac{1}{(I+1)^J};$$

15. La présentation de ces modèles dans un cadre de traduction mot-à-mot et non pas simplement d'alignement impliquerait de nombreux détails que nous omettons ici : les systèmes à base de mots ne sont plus utilisés que marginalement, car considérés comme moins performants que les modèles à base de segments.

– puis chaque mot de $\mathbf{f}$ est engendré indépendamment[16] à partir du mot e_{a_j} de la phrase $\mathbf{e}$ avec lequel il est aligné :

$$\mathrm{P}(\mathbf{f}|\mathbf{a},\mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(f_j|\mathbf{a},\mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(f_j|e_{a_j}).$$

Le modèle IBM1 est souvent décrit comme un modèle de traduction lexicale puisqu'il a pour seuls paramètres les probabilités de traduction mot-à-mot $P(f|e)$.

7.3.1.1.1. Définition du modèle et inférence

À partir des ces hypothèses, l'équation (7.6) se simplifie puisque chaque élément de la séquence est étiqueté indépendamment des autres :

$$\mathrm{P}(\mathbf{a},\mathbf{f}|\mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(a_j, f_j|e_{a_j}) = \frac{1}{(I+1)^J} \prod_{j=1}^{J} \mathrm{P}(f_j|e_{a_j}) \tag{7.7}$$

Au total, ce modèle est paramétré par l'ensemble des distributions conditionnelles modélisant les équivalents de traduction de chacun des mots cibles e ; le vecteur de paramètres $\boldsymbol{\theta}$ du modèle IBM1 s'écrit donc $\boldsymbol{\theta} = \{\mathrm{P}(f|e), \forall (f,e) \in \mathcal{V}_f \times \mathcal{V}_e\}$, où le vocabulaire $\mathcal{V}_f$ (et respectivement $\mathcal{V}_e$) regroupe l'ensemble des mots rencontrés dans les textes parallèles d'apprentissage pour la langue source (respectivement cible).

Connaissant les paramètres $\boldsymbol{\theta}$, il est aisé d'inférer l'alignement le plus probable pour un couple de phrases donné en appliquant la règle du maximum *a posteriori*.

$$\mathbf{a}^* = \operatorname*{argmax}_{\mathbf{a}} \mathrm{P}(\mathbf{a}|\mathbf{f},\mathbf{e}) = \operatorname*{argmax}_{\mathbf{a}} \mathrm{P}(\mathbf{a},\mathbf{f}|\mathbf{e}) \tag{7.8}$$

La recherche du meilleur alignement $\mathbf{a}^*$est particulièrement efficace puisque l'évaluation de chaque alignement a_j est indépendante des autres alignements présents dans la phrase. La séquence $\hat{\mathbf{a}}$ la plus probable *a posteriori* est alors celle qui maximise la probabilité jointe $\mathrm{P}(\mathbf{a},\mathbf{f}|\mathbf{e})$. D'après l'équation (7.7), cette probabilité jointe est le produit de termes indépendants compris entre 0 et 1. Ce produit est alors maximal quand chaque terme est maximal :

$$\forall j, a_j^* = \operatorname*{argmax}_{a_j \in \mathcal{A}} \mathrm{P}(f_j|e_{a_j}) \tag{7.9}$$

16. C'est-à-dire sans tenir compte du contexte dans lequel il est employé.

7.3.1.1.2. Apprentissage

L'estimation des paramètres du modèle peut se faire par la méthode du maximum de vraisemblance si des données alignées mot-à-mot sont disponibles. Malheureusement, ce type de données est rarement disponible en quantité suffisante. En l'absence de telles données, il reste possible d'utiliser pour l'apprentissage un corpus bilingue parallèle seulement aligné au niveau des phrases. Du point de vue de l'alignement mot-à-mot, ces corpus sont considérés comme non-étiquetés. Les alignements sont alors les variables cachées et les distributions associées peuvent être estimées via l'algorithme EM[17] (*Expectation-Maximization*) de manière à maximiser la *log-vraisemblance* des paramètres pour le corpus d'apprentissage [DEM 77]. Les hypothèses du modèle IBM1, dans lequel chaque mot est aligné indépendamment de son contexte, rendent cette maximisation possible. De plus, les auteurs de [BRO 93b] montrent que la *log-vraisemblance* est concave, ce qui garantit la convergence de l'algorithme vers un optimum global. Les étapes de l'algorithme EM sont alors les suivantes :

– **Initialisation :** choix aléatoire des valeurs pour $\boldsymbol{\theta}$

– **Jusqu'à convergence ou pour un nombre d'itérations prédéfini :**

- **étape E :** Connaissant $\boldsymbol{\theta}$, le modèle est appliqué aux données parallèles afin d'estimer les probabilités *a posteriori* de chaque lien d'alignement :

$$\mathrm{P}(a_j|f_j,\mathbf{e}) = \frac{\mathrm{P}(a_j,f_j|e_{a_j})}{\mathrm{P}(f_j|\mathbf{e})} = \frac{\mathrm{P}(f_j|e_{a_j})}{\sum_{j'=0}^{J}\mathrm{P}(f_{j'}|e_{a_{j'}})} \tag{7.10}$$

Connaissant les probabilités *a posteriori*, l'espérance du nombre d'occurences d'un alignement entre f et e dans le corpus parallèle se calcule en sommant sur tous les couples de phrases $(\mathbf{f},\mathbf{e})$:

$$\mathcal{F}(f|e) = \sum_{(\mathbf{f},\mathbf{e})}\left[\mathrm{P}(a|\mathbf{f},\mathbf{e})\Big(\sum_{j=1}^{J}\delta(f,f_j)\Big)\Big(\sum_{i=0}^{I}\delta(e,e_i)\Big)\right]. \tag{7.11}$$

Dans l'équation (7.11), a désigne le lien d'alignement entre f et e.

- **étape M :** Connaissant ces espérances, les paramètres $\boldsymbol{\theta}$ du modèle s'estiment selon :

$$\mathrm{P}(f|e) = \frac{\mathcal{F}(f|e)}{\sum_{f'\in\mathcal{V}_f}\mathcal{F}(f'|e)}, \tag{7.12}$$

Il est à noter que la somme sur tous les alignements possibles du couple $\mathbf{f},\mathbf{e}$ qui figure au dénominateur de l'expression $\mathrm{P}(a|\mathbf{f},\mathbf{e})$ dans l'équation (7.11) peut être effectuée de manière efficace.

17. Voir le chapitre A, en particulier la section A.3, pour une présentation détaillée de l'algorithme EM appliqué toutefois à un modèle plus simple.

Le modèle IBM 1 est un modèle d'alignement très simple qui utilise comme seule information les cooccurrences entre un mot et ses traductions pour estimer le modèle de traduction selon (7.11). Bien que rudimentaire, ce modèle permet d'obtenir de bons résultats lorsque l'on dispose de très grandes masses de données [BRA 07].

7.3.1.2. *Aligner avec des modèles de Markov cachés*

Une première critique à l'encontre du modèle IBM1 est l'hypothèse selon laquelle toutes les valeurs d'un alignement a_j sont équiprobables. Le modèle IBM2 [BRO 90] introduit une dépendance entre la valeur de a_j et la position j du mot dans la phrase via la distribution $\mathrm{P}(a_j|j, J, I)$. Cette hypothèse permet de favoriser certains alignements au détriment d'autres. En effet pour la paire de langue français/anglais par exemple, un mot en début de phrase est rarement aligné avec un mot en fin de phrase.

Cependant, le modèle IBM2 n'est plus très utilisé car il ne prend pas en compte une propriété de nombreuses phrases parallèles, à savoir la *monotonicité des alignements* : si le mot source $j-1$ est aligné avec le mot cible a_{j-1}, alors il est fort probable que le mot source j soit aligné avec le mot cible $a_j = a_{j-1}+1$ ou, du moins, avec un mot proche. Dit autrement, il semble pertinent de modéliser le saut entre deux alignements consécutifs en se basant la différence $\mid a_j - a_{j-1} \mid$. En guise d'illustration, prenons l'exemple de l'alignement d'une séquence *déterminant nom*, l'alignement a_j du *nom* est influencé par l'alignement a_{j-1} du *déterminant* qui le précède. Le caractère monotone des alignements peut être modélisé grâce aux modèles de Markov cachés [18] (HMM) d'ordre 1, qui introduisent une dépendance entre deux liens d'alignement de deux mots consécutifs a_{j-1} et a_j.

7.3.1.2.1. Définition et inférence

Les modèles de Markov cachés pour l'alignement [VOG 96] utilisent un scénario génératif qui introduit une dépendance entre liens d'alignement comme suit :

– chaque étiquette a_j est choisie parmi toutes les étiquettes possibles conditionnellement à l'étiquette précédente. La probabilité d'une séquence d'étiquettes s'écrit alors :

$$\mathrm{P}(\mathbf{a}|\mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(a_j|a_{j-1}, \mathbf{e})$$

– chaque mot f_j de $\mathbf{f}$ est ensuite produit conditionnellement à l'étiquette a_j, ou plutôt conditionnellement au mot cible en position a_j.

Sous l'hypothèse de dépendance markovienne d'ordre 1, la probabilité jointe de la séquence d'observations et de la séquence d'étiquettes associée se simplifie de la

18. Voir le chapitre A, et notamment la section A.5.

manière suivante :

$$\mathrm{P}(\mathbf{a},\mathbf{f}|\mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(a_j|a_{j-1},\mathbf{e})\,\mathrm{P}(f_j|e_{a_j}), \tag{7.13}$$

où, comme dans le modèle IBM1, $\mathrm{P}(f_j|e_{a_j})$ dénote la probabilité de traduire le mot e_{a_j} par $\mathrm{P}(f_j|)$. Afin de paramétrer les probabilités de transition $\mathrm{P}(a_j|a_{j-1},\mathbf{e})$, les auteurs de [VOG 96] font une hypothèse supplémentaire, qui stipule que cette probabilité de transition ne dépend que de la largeur du saut entre a_{j-1} et a_j :

$$\mathrm{P}(a_j|a_{j-1},\mathbf{e}) = \frac{s(a_j - a_{j-1})}{\sum_{a\in\mathcal{A}} s(a - a_{j-1})}, \tag{7.14}$$

où $\{s(j-j')\}$ est un ensemble de paramètres positifs. L'ensemble des paramètres $\boldsymbol{\theta}$ du modèle est alors constitué des mêmes paramètres que pour le modèle IBM1, auxquels s'ajoutent les paramètres liés aux probabilités de transition. Dans cette formulation [VOG 96], la possibilité qu'un mot source ne soit aligné avec aucun mot cible n'est pas envisagée. Dans [OCH 03b], les auteurs proposent une extension du modèle HMM qui consiste à modifier la séquence cible $\mathbf{e}$ afin d'y ajouter I « mots vides » (désignés par `null`) : $\mathbf{e} = e_1, ...e_I, e_{I+1}, e_{2\times I}$. Dans ce cas, pour $1 \leq j \leq I$, e_j représente le $j$ème mot de la phrase cible, et pour $I \leq j \leq 2 \times I$, $e_j =$ `null` . Connaissant les paramètres $\boldsymbol{\theta}$ du modèle, la séquence d'alignement la plus probable pour un couple de phrases donné se détermine par la résolution de l'équation (7.8). L'introduction de dépendances entre deux étiquettes successives rend le problème plus complexe que le problème équivalent pour le modèle IBM1. Il peut être toutefois être résolu efficacement de manière exacte par l'algorithme de Viterbi[19].

7.3.1.2.2. Apprentissage

Pour l'estimation des paramètres de ce modèle, la démarche est, dans son principe, identique à celle employée pour le modèle IBM1. Le corpus d'apprentissage est constitué d'un ensemble de phrases parallèles où les alignements sont des variables cachées. La mise en œuvre de l"algorithme EM implique alors les étapes suivantes :

– **Initialisation :** choix des valeurs initiales pour $\boldsymbol{\theta}$.

– **Jusqu'à convergence ou pour un nombre d'itérations prédéfini :**

- **étape E :** puisque les étiquettes sont dépendantes entre elles, il est nécessaire de calculer la probabilité *a posteriori* d'une séquence d'alignement. Pour un couple de phrases donné, cette probabilité s'écrit :

$$\mathrm{P}(\mathbf{a}|\mathbf{f},\mathbf{e}) = \frac{\mathrm{P}(\mathbf{a},\mathbf{f}|\mathbf{e})}{\mathrm{P}(\mathbf{f}|\mathbf{e})} = \frac{\mathrm{P}(\mathbf{a},\mathbf{f}|\mathbf{e})}{\sum_{\mathbf{a}'} \mathrm{P}(\mathbf{a}',\mathbf{f}|\mathbf{e})}. \tag{7.15}$$

19. Voir de nouveau le chapitre A, et notamment la section A.5.

Ce calcul nécessite de sommer sur toutes les séquences d'alignement possibles, ce qui, pour les modèles de Markov cachés, peut être réalisé efficacement par l'algorithme de Baum-Welsh. Une fois les probabilités *a posteriori* connues, il est possible de calculer des espérances des comptes $F(f|e)$ et $F(j|j')$ où j et j' sont les deux valeurs prises respectivement par a_i et a_{i-1}. Il suffira, pour cela, d'adapter l'équation (7.11).

- **étape M :** Les paramètres $\boldsymbol{\theta}$ du modèle peuvent alors être ré-estimés en utilisant les espérances des comptes comme dans l'équation (7.12).

Différence majeure avec le modèle IBM1, la *log-vraisemblance* n'est plus une fonction concave des paramètres. Elle présente même de nombreux extréma locaux, supprimant toute garantie de convergence vers le maximum. Dans de nombreux cadres applicatifs, ce défaut n'est pas perçu comme rédhibitoire si le choix des valeurs initiales pour les paramètres est judicieux. Le choix le plus courant consiste à entraîner préalablement un modèle IBM1. Les paramètres obtenus, qui sont de la forme $\mathrm{P}(e|f)$, sont alors utilisés pour initialiser les paramètres correspondant du modèle HMM [20]. .

7.3.1.3. *Modéliser la fertilité, les modèles IBM3, 4 et 5*

Lors de l'alignement d'une séquence $\mathbf{f}$, chaque mot f_j est apparié à un mot unique de la phrase cible e_{a_j}. Cette hypothèse est évidement restrictive, puisqu'il est courant qu'un même concept s'exprime par des groupes de mot de taille différente en langues source et cible. Ce problème est illustré sur la figure 7.5, qui présente deux extraits de phrases parallèles à aligner. Le premier exemple est celui de *pomme de terre*. Ce concept s'exprime dans la phrase française par un mot composé, quand en anglais, il apparait sous la forme d'un mot simple *potatoes*. Lors d'un alignement du français vers l'anglais[21], le modèle d'alignement peut trouver une solution consistant à aligner les trois mots source *pomme*, *de* et *terre* avec le même mot cible *potatoes*. Mais suivant les données d'apprentissage, une autre solution peut émerger conduisant à aligner *pomme* avec *potatoes* et les deux autres mots source avec `null` .

Une manière de guider l'alignement vers la « bonne » solution consiste à introduire la notion de *fertilité* : à chaque mot cible est associé une distribution $n(\phi|e)$ indiquant le nombre de mots dans la phrase source avec lequel il peut être aligné. Ce nouveau modèle probabiliste décrit donc la propension de certains mots cible à "recevoir" un plus ou moins grand nombres de liens d'alignements. Par exemple, la distribution de probabilité du tableau 7.2 exprime la propension de *patatoes* à s'aligner de manière préférentielle soit avec un seul, soit avec trois mots.

---

20. Dans [OCH 03b], les auteurs préconisent d'initialiser l'algorithme à l'étape M en utilisant les statistiques collectées lors de la dernière itération du modèle IBM1.

21. Les mots de la phrase française sont étiquetés avec les indices des mots anglais correspondants.

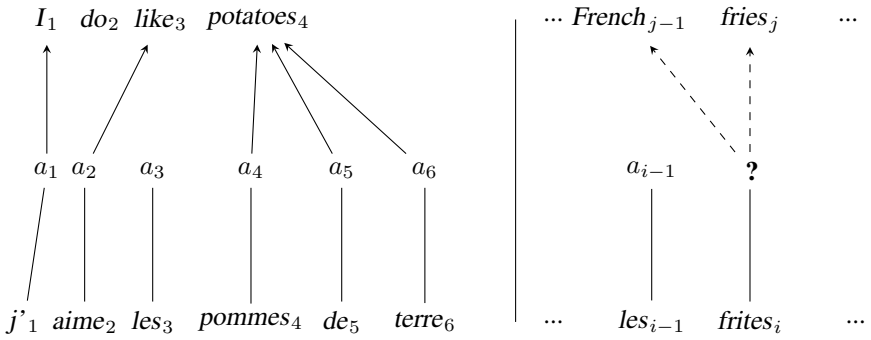


Figure 7.5 – Fertilité, un problème pour l'étiquetage de séquence

ϕ	$n(\phi$|potatoes$)$
0	0
1	0.4
2	0.1
3	0.4
4	0

Tableau 7.2 – Distribution de la fertilité du mot *potatoes*

La fertilité permet également de modéliser la propension de certains mots cible à demeurer non-alignés. C'est, par exemple, le cas d'un auxiliaire tel que *do* dans l'exemple de la figure 7.5. De tels phénomènes seront simplement modélisés en assignant une probabilité non-nulle à l'évènement correspondant à laisser *do* non aligné : $n(0|\text{do}) \neq 0$. Un autre cas particulier intéressant dans cet exemple est celui du mot source *les*, qui doit *a priori* être aligné avec le mot `null` . Ce phénomène est déjà pris en compte dans les modèles IBM1 et HMM, mais la notion de fertilité permet également d'affiner la prédiction des alignements nuls. En partant de l'intuition que, au sein d'une phrase donnée, le nombre de mots alignés avec `null` dépend principalement de la longueur de la phrase, les auteurs de [BRO 90] proposent une distribution particulière pour la fertilité du mot `null` $\mathrm{P}(\phi_0)$ qui est fonction de la longueur de la phrase et d'un paramètre p_0 quantifiant la probabilité *a priori* d'un alignement nul.

Ainsi, dans le modèle IBM3, la probabilité jointe $\mathrm{P}(\mathbf{a}, \mathbf{f}|\mathbf{e})$ enrichit le modèle IBM2 en intégrant deux nouveaux facteurs : la fertilité du mot `null` $(\mathrm{P}(\phi_0))$ et, pour chaque mot cible, sa fertilité $n(\ |.)$ Notons enfin qu'à partir du modèle IBM3, les auteurs de [BRO 90] modifient le scénario génératif et introduisent la notion de *distorsion*, qui se substitue aux probabilités $\mathrm{P}(\mathbf{a}|\mathbf{e}, \mathbf{f})$.

7.3.1.3.1. Inférence et apprentissage

Le modèle IBM3 introduit une modélisation plus fine des interactions entre deux langues, mais au prix d'une complexité algorithmique accrue. D'une part, le modèle inclut un bien plus grand nombre de paramètres que les modèles IBM1 et HMM, puisqu'il incorpore, pour chaque mot cible, les paramètres modélisant la la fertilité de ce mot[22]. D'autre part, il n'est plus possible d'utiliser des algorithmes exacts efficaces pour réaliser l'apprentissage et l'inférence. En effet, dans ce nouveau modèle, la recherche de l'alignement optimal devient un problème NP-difficile [UDU 06], et doit donc être résolue de manière approchée. De même, à défaut de pouvoir envisager tous les alignements possibles, les algorithmes d'apprentissage les plus couramment utilisées recourent à un échantillonnage pour approcher au mieux les quantités nécessaires à l'application de l'algorithme EM.

7.3.1.3.2. Les modèles IBM4, IBM5 et IBM6

Nous résumons ici les raffinements ultérieurs des modèles IBM, le lecteur désirant plus de détails pourra se référer aux articles originaux [BRO 90, OCH 03b]. Le modèle IBM4 améliore le modèle IBM3 en introduisant un modèle de distorsion qui utilise les positions relatives des mots et qui inclue une dépendance markovienne d'ordre 1. De plus, les mots sont regroupés aussi bien en langue source qu'en langue cible dans des classes lexicales, afin de réduire le nombre de paramètres à estimer. Ces classes sont déterminées automatiquement, de manière non-supervisée, à partir des données d'apprentissage [BRO 92].

Le modèle IBM5 corrige un problème théorique posé par les modèles IBM3 et IBM4, qui sont dits *déficients* car une partie de la masse de probabilité est attribuée à des alignements qui sont impossibles. Enfin, le modèle IBM6 est une combinaison des modèles IBM4 et HMM introduite dans [OCH 03b]. Dans la pratique, le surcroit de complexité qu'apportent ces modèle n'est pas payé d'une amélioration très significative des alignements obtenus et ces modèles sont rarement utilisés.

7.3.1.4. *Symétrisation*

Les exemples de la figure 7.5 mettent en évidence les limitations des modèles qui d'envisagent l'alignement comme un simple étiquetage de séquences. Dans le premier exemple, trois mots doivent être alignés avec un seul, ce qui est possible. Cependant, lorsque la direction de l'alignement est inversée, le modèle est par construction dans l'incapacité de proposer une solution satisfaisante. Cet aspect est également illustré par le second exemple, qui illustre le cas inverse d'un mot unique en français *frites* devant s'aligner avec un couple de mots en anglais *French fries* ? Ici, c'est l'alignement du français vers l'anglais qui est problématique, alors que quand on considère la direction opposée, il est possible d'obtenir un alignement satisfaisant.

22. Ce qui représente $|\mathcal{V}_e| \times (F_{max})$ paramètres supplémentaires, avec F_{max} la fertilité maximum envisagée.

Ainsi, les deux directions d'alignement sont complémentaires. Pour les prendre simultanément en compte, il est courant d'utiliser des heuristiques de symétrisation, afin de produire un alignement unique. Ces heuristiques s'expriment simplement lorsque l'on représente un alignement entre deux phrases sous la forme d'une matrice booléenne $J \times I$: un 1 en position (i, j) indique l'alignement entre f_j et e_i.

Une première heuristique simple consiste à prendre *l'union* des deux matrices d'alignements correspondant aux alignements source/cible et cible/source. La matrice de liens qui en résulte exploite au maximum les deux directions d'alignement, au risque de proposer des liens peu sûrs, qui n'existent que pour une seule direction d'alignement. Une deuxième heuristique consiste à sélectionner *l'intersection* des deux alignements d'entrée. Dans ce cas, le nombre d'alignement obtenu est nettement plus faible, mais ces liens peuvent être considérés comme plus fiables puisqu'ils sont identifiés dans les deux directions. Dans [OCH 03b, KOE 03b], les auteurs proposent différentes heuristiques qui ont pour point commun de compléter l'alignement construit intersection avec certains des liens d'alignement qui figurent dans l'union.

7.3.2. *Point de vue synthétique et pratique sur les modèles d'alignement*

Dans leur utilisation actuelle, les modèles d'alignement en traduction statistique servent à produire des alignements mot-à-mot pour les données parallèles d'entraînement. Ainsi, pour chaque couple de phrases, l'alignement est effectué dans les deux directions puis symétrisé. Aujourd'hui, le système d'alignement le plus utilisé est `Giza++` [OCH 03b]. Il propose une implantation librement disponible des modèles principaux modèles d'alignement génératifs, en particulier les modèles IBM [BRO 93b] (décrits à la section 7.3.1).

Parmi les modèles d'alignement présentés ci-dessus, le modèle IBM4 est celui qui semble fournir le meilleur compromis entre temps d'estimation et la qualité des alignements produits[23]. Son estimation nécessite toutefois au préalable l'apprentissage successif des modèles IBM1, HMM et IBM3, puisque chacun de ces modèles sert d'initialisation pour l'estimation du modèle de complexité supérieure (IBM1 sert d'initialisation pour le modèle HMM, et ainsi de suite).

Les performances de `Giza++`, aussi bien du point de vue de la qualité des alignements prédits que des temps de calcul, restent insatisfaisantes et l'étape d'alignement mot-à-mot constitue, aujourd'hui, un facteur limitant dans de nombreux systèmes de traduction automatique [FRA 07]. Des travaux récents explorent l'usage de

23. Pour donner une idée du coût computationnel d'un apprentissage, mentionnons que le temps de calcul pour aligner dans les deux directions le corpus Hansard est de l'ordre d'une journée. Ce corpus français/anglais contient plus d'un million de paire de phrases extraites des comptes rendus du parlement canadien.

modèles discriminants tels que les champs aléatoires conditionnels (CRF, voir le chapitre 6) [BLU 06, NIE 08]. Les modèles discriminants sont effet plus expressifs et permettent de combiner des sources d'informations multiples pour décider de la validité d'un lien d'alignement. Ainsi, par exemple, ces modèles permettent d'intégrer des informations linguistiques issues d'une analyse syntaxique ou morpho-syntaxique préalable. Cependant, l'estimation de tels modèles requiert des données *annotées*, c'est-à-dire des données alignées manuellement au niveau du mot. Compte-tenu du coût d'une telle annotation, rares sont les corpus d'entraînement disponibles, et les modèles discriminants ne semblent pas devoir, dans le court terme, pouvoir remplacer les modèles IBM pour constuire automatiquement des alignements.

7.3.3. *L'extraction de bisegments*

Nous abordons ici la dernière étape du processus de construction des modèles de traduction tel qu'il est résumé à la figure 7.3. Afin de nous situer dans ce processus, rappellons que lors de la première étape, la matrice d'alignement est créée. Cette matrice d'alignement s'obtient à partir des modèles d'alignement décrits précédemment aux sections 7.3.1.1–7.3.1.3. Chaque couple de phrases est donc aligné deux fois, une fois dans chaque direction d'alignement. Ces deux alignements sont ensuite symétrisés à l'aide d'heuristiques (voir la section 7.3.1.4). Il reste donc à extraire et à valuer un ensemble de bisegments. Lors de l'extraction, la méthode la plus communément utilisée considère chaque couple de phases indépendamment des autres. L'évaluation d'une bisegment repose à la fois sur des statistiques accumulées sur l'ensemble du corpus parallèle et sur l'exploitation des modèles d'alignement décrits à la section 7.3.1.

Pour débuter la discussion, considérons l'exemple de la figure 7.6, qui représente la matrice d'alignement d'un couple de phrases. L'objectif de l'algorithme d'extraction est de repérer des bisegments tel que celui qui associe le groupe nominal *la glace au chocolat* et son équivalent anglais *chocolate ice cream*. En effet, les longs bisegments, qui intègrent des élément de contexte, fournissent des traductions plus fiables ; par ailleurs, les bisegments qui correspondent à des constituants syntaxiques ont tendance à fonctionner (y compris pour leur traduction) de manière plus autonome que des séquences de mots arbitraires. Les bisegments longs ont toutefois des nombres d'occurrences plus faibles que les bisegments plus courts, ce qui implique qu'ils sont plus difficiles à valuer et surtout qu'ils sont moins susceptibles de réapparaître dans le futur. Mémoriser ces traductions présente donc un intérêt plus réduit que la mémorisation des bisegments plus courts, tels que le couple *glace* et *ice cream*. Pour réaliser un bon compromis entre la richesse du contexte capturé et le potentiel de réutilisation d'un bisegment, l'heuristique d'extraction doit donc extraire des segments de différentes longueurs. On se reportera à [DEN 06] pour une discussion approfondie de l'intérêt de ces heuristiques d'extraction.

7.3.3.1. *Cohérence d'un bisegment*

L'heuristique d'extraction vise à extraire des bisegments, c'est-à-dire des groupes de mots adjacents en langue source et en langue cible qui correspondent potentiellement à des traductions mutuelles. La notion principale qui fonde cette extraction est celle de *cohérence*, illustrée sur la figure 7.6. considérons le couple ('*I do not*','*je n' *'). Si cette association intègre de nombreux liens d'alignements, elle exclut toutefois le mot *pas*, qui est pourtant également aligné avec *do not* : on peut donc penser que la traduction de *I do not* n'est pas complètement restituée par *je n'* . La solution consiste à étendre le bisegment du côté français afin d'inclure le mot *pas*. Le côté français du bisegment inclut maintenant *aime*, dont le correspondant anglais est extérieur au bisegment. Il faut donc encore étendre l'alignement, jusqu'à extraire ('*I do not like*','*Je n'aime pas*').

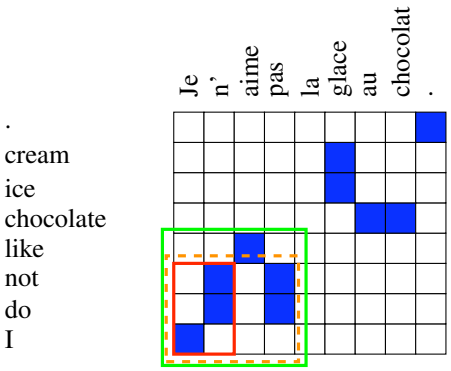


Figure 7.6 – Illustration de la cohérence d'un bisegment étant donnée la matrice d'alignement. Seul le bisegment délimité en vert est cohérent. Les deux autres ne sont pas valides.

À partir de ces considérations, il est possible de définir formellement la cohérence pour un bisegment $(\overline{e}, \overline{f})$ apparaissant dans une matrice d'alignement $\boldsymbol{A}$ selon :

$$\forall e_i \in \overline{e}, \qquad (e_i, f_j) \in \boldsymbol{A} \Rightarrow f_j \in \overline{f}, \text{ et} \tag{7.16}$$

$$\forall f_j \in \overline{f}, \qquad (e_i, f_j) \in \boldsymbol{A} \Rightarrow e_i \in \overline{e}, \text{ et} \tag{7.17}$$

$$\exists e_i \in \overline{e} \text{ et } \exists f_j \in \overline{f}, \qquad (e_i, f_j) \in \boldsymbol{A}. \tag{7.18}$$

Les deux premières règles permettent d'exclure le bisegment('*I do not*','*je n' aime pas*'), car le mot *aime* possède un lien d'alignement avec le mot *like* qui hors du bisegment. Cependant, ces deux règles laissent une certaine latitude pour étendre des segments lorsqu'il existe des mots non-alignés dans $\boldsymbol{A}$. Un mot non-aligné correspond à une colonne ou une ligne vide dans la matrice : c'est par le cas du mot *la* dans la figure 7.6. La dernière règle permet de borner le traitement des alignements nuls en imposant que tout bisegment contienne au moins un lien d'alignement. Les mots non-alignés sont absorbés dans les bisegment disponibles, aussi bien à leur gauche qu'à leur droite. On extraira donc, entre autres bisegments : ('*do not like*','*n' aime pas la*') et ('*chocolate ice cream*','*la glace au chocolat*').

7.3.3.2. *Algorithme d'extraction de bisegment*

Pour extraire tous les bisegments d'un couple de phrases $\mathbf{f}, \mathbf{e}$, la définition de la cohérence propose un critère de sélection et la matrice d'alignement $\boldsymbol{A}$ apporte des contraintes guidant l'algorithme d'extraction. Pour un couple de phrases donné, cet algorithme considère toutes les segmentations possibles d'une des phrases (par exemple la phrase $\mathbf{e}$), puis explore la matrice d'alignement pour chaque segment de $\mathbf{e}$ afin de

sélectionner le segment associé et cohérent. L'algorithme d'extraction est formalisé par l'algorithme 3.

Algorithme 3: Algorithme d'extraction de bisegments pour un couple de phrases alignées au niveau du mot.

```
/* Pour un couple de phrases et la matrice d'alignement     */
Données : f, e, A
/* Ensemble des bisegments                                  */
Résultat : BS

i_d, i_f // les indice de début et de fin du segment cible;
/* Double boucle considérant tous les segments de e         */
pour i_f = 1 ... I faire
    pour i_d = 1 ... i_f faire
        S_f = {j | ∃i, i_d ≤ i ≤ i_f et A(i,j) = 1};
        si cohérence(S_f) alors
            j_d = min(S_f) // indice de début du segment source;
            j_f = max(S_f) // indice de fin du segment source;
            BS = BS ∪ (e_{i_d}^{i_f}, f_{j_d}^{j_f});
            tant que non-aligné(f_{j_f} + 1) faire
                j_f = j_f + 1;
                BS = BS ∪ (e_{i_d}^{i_f}, f_{j_d}^{j_f});
                j' = j_d;
                tant que non-aligné(f_{j'} − 1) faire
                    j' = j' − 1;
                    BS = BS ∪ (e_{i_d}^{i_f}, f_{j'}^{j_f});
                fin
            fin
        fin
    fin
fin
```

La fonction non-aligné retourne **vrai** s'il n'existe aucun lien d'alignement dans la matrice $\boldsymbol{A}$ correspondant à cet indice source. La fonction cohérence retourne **vrai** si les indices contenus dans S_f se suivent ou bien, si un ou plusieurs indices manquent, ils doivent alors correspondre à un mot non-aligné. En pratique, il est nécessaire d'imposer une longueur maximale pour les segments extraits afin d'en limiter le nombre.

7.3.3.3. *Évaluation des bisegments*

Une dernière étape de construction du modèle de traduction consiste à associer un score de confiance aux bisegments ainsi extraits ; ce score sera utilisé lors du décodage (voir la section 7.5) pour orienter les choix de l'algorithme de traduction afin

qu'il utilise les bisegments les plus probables, au détriment des appariemments plus rares, qui sont le souvent des artéfacts induits par des erreurs dans la matrice d'alignement. En pratique, chaque bisegment est valué par un ensemble de scores, qui sont conjointement utilisés lors de la recherche de la meilleure traduction. Nous décrivons ici succinctement les scores de confiance les plus utilisés.

La question à laquelle il faut répondre est au fond la suivante : soit $(\overline{e}, \overline{f})$ un bisegment extrait du corpus parallèle, dans quelle mesure le segment $\overline{e}$ est-il une traduction « adéquate » de $\overline{f}$? Une première manière simple d'évaluer la qualité d'un bisegment $(\overline{e}, \overline{f})$ consiste à estimer sur le corpus parallèle la fréquence relative de cette association rapportée à toutes les occurrences de $\overline{e}$ ou de $\overline{f}$. Les valeurs résultantes sont des estimations de la probabilité conditionnelle de traduction. On définit ainsi :

$$\phi(\overline{e}|\overline{f}) = \frac{F(\overline{e}, \overline{f})}{\sum_{\overline{e}_k} F(\overline{e}_k, \overline{f})}, \tag{7.19}$$

et, de manière similaire pour $\phi(\overline{f}|\overline{e})$. Les estimations fondées sur les fréquences relatives sont toutefois très optimistes, surtout pour les bisegment rares. Considérons le cas extrême où un bisegment est constitué de deux segments qui n'apparaissent qu'une seule fois chacun. Dans ce cas, on aura $\phi(\overline{f}|\overline{e}) = \phi(\overline{e}|\overline{f}) = 1$, que l'association soit valide, ou bien au contraire, qu'elle corresponde à une erreur dans la matrice d'alignement. Pour améliorer ces estimateurs, la solution la plus communément utilisée est de prendre en compte la qualité des alignements de mots qui subsument l'alignement du bisegment : on parle alors de *pondération lexicale* (*lexical weighting*) [KOE 03b]. Il s'agit alors de calculer les scores suivants :

$$lex(\overline{e}|\overline{f}, \boldsymbol{A}) = \prod_{i=i_d}^{i_f} \frac{1}{F(j|\boldsymbol{A}(i,j)=1)} \sum_{j|\boldsymbol{A}(i,j)=1} \frac{F(e_i, f_j)}{F(f_j)}, \tag{7.20}$$

où $F(e_i, f_j)$ et $F(f_j)$ dénotent respectivement la fréquence de l'alignement du mot e_i avec le mot f_j estimée à partir de l'ensemble des alignements de mots dans le corpus parallèle (respectivement la fréquence du mot f_j) et i_d et i_f sont respectivement les indices dans la phrase $\mathbf{e}$ de début et de fin de segment. On déduit de (7.20) un score indépendant de $\boldsymbol{A}$ en prenant la valeur maximale sur tous les alignement de $\overline{e}$ avec $\overline{f}$. Enfin, comme précédemment, en pratique, on utilise conjointement les deux termes de pondération lexicale $\mathrm{lex}(\overline{f}|\overline{e})$ et $\mathrm{lex}(\overline{e}|\overline{f})$.

D'autres fonctions de score ont été proposées dans la littérature et peuvent être utilisées en complément des scores précédents. En fait, toute fonction associant un score numérique à chaque bisegment est candidat à fournir une mesure de confiance. On pourra ainsi utiliser des fonctions booléennes mesurant des propriétés syntaxiques arbitraires d'un bisegment $(\overline{f}, \overline{e})$ telles que '$\overline{e}$ est-il un constituant cible ?' , '$\overline{f}$ est-il un constituant source ?', '$\overline{e}$ et $\overline{f}$ contiennent tout deux un verbe ?', un nom ?, etc. La plus

simple de ces fonctions vaut 1 pour tous les bisegments : son inclusion dans le modèle permettra de forcer le décodeur, toutes choses égales, à préférer de traduire avec peu de bisegments, et donc à choisir des bisegments longs. Ce score supplémentaire est désigné dans la littérature anglaise sous le nom de *phrase penalty*.

En résumé, à chaque bisegment extrait du corpus parallèles sont associées des valuations numériques. Les plus utilisées de ces valuations correspondent aux probabilités de traduction $\phi(\overline{f}|\overline{e})$ et $\phi(\overline{e}|\overline{f})$, aux deux termes de pondération lexicale et à la pénalité de longueur.

7.4. Modéliser les déplacements

Comme discuté à la section 7.2.2.3, une des difficultés principales que rencontrent les systèmes de traduction automatique est que les différentes langues emploient des schémas de construction de phrases qui diffèrent selon les positions relatives des différents constituants. En conséquence, l'ordre des mots ou des segments figurant dans les phrases source et cible peut être très différent. Comme pour les autres différences entre langues source et cible, il va donc falloir (i) modéliser ces différences d'ordre et (ii) construire des fonctions qui évaluent la plausibilité des possibles arrangements de la cible en fonction de l'ordre des mots en source. On attend en particulier de ces modèles qu'ils aident à prédire des restructurations importantes (c'est-à-dire non-locales) de la phrase source, dans la mesure où les restructurations locales sont déjà largement capturées dans les modèles de traduction à base de segments. Le rôle joué par ces modèles est naturellement d'autant plus crucial que les langues diffèrent entre elles quant au positionnement relatif des mots et des syntagmes : pour la traduction entre français et anglais, ces modèles jouent un rôle beaucoup moins essentiel que pour la traduction entre français et allemand, *a fortiori* pour la traduction entre français et japonais (voir [ALO 06, BIR 09] qui proposent des mesures quantitatives des divergences d'ordre entre langues).

Nous présentons dans cette section les modèles de réordonnancement les plus communément utilisés, en distinguant clairement deux facettes de ces modèles : d'une part les réordonnancements qu'ils permettent d'explorer à la section 7.4.1 ; d'autre part les évaluations de la plausibilité des différents réordonnancements à la section 7.4.2.

7.4.1. *L'espace des réordonnancements possibles*

Pour donner l'intuition du rôle que jouent ces modèles, replaçons-nous dans le cadre simple de la traduction mot-à-mot et supposons que l'on souhaite traduire la phrase $\mathbf{f} = a\ b\ c$. Supposons également que chaque mot n'a qu'une seule traduction, α pour a, β pour b et γ pour c. Modéliser les réordonnancements consiste à spécifier

quelles sont les permutations de $\alpha\ \beta\ \gamma$ qui doivent être considérées[24] comme des traductions possibles et à assigner un score de plausibilité à chacune d'entre elles.

Dans un premier temps, nous allons présenter les modèles qui manipulent ou évaluent des modèles de réordonnancement à base de mots, avant de voir comment ils se généralisent aux modèles à base de segments. On supose donc qu'un appariemment mot-à-mot entre une phrase source $\mathbf{f} = f_1 \dots f_J$ et sa traduction $\mathbf{e} = e_1 \dots e_I$ prend la forme $(f_{j_1} : e_1) \dots (f_{j_I} : e_I)$: pour tout indice $t \in [1 : I]$, le mot cible e_t est la traduction du mot source f_{j_t}. On qualifiera alors de *monotone* une traduction vérifiant $\forall t \in [1 : J], j_t = t$. Ces appariements entre source et cible peuvent être représenté de manière plus abstraite sous la forme d'une permutation π des indices de $[1 : J]$: $j_1 j_2 \dots j_l$.

Dans la plupart des modèles présentés ci-dessous, la génération des permutations peut être simulée de la manière itérative suivante : à chaque étape $t \in [1 \dots l]$, on ajoute un nouveau mot e_t dans la phrase cible en traduisant un mot non encore traduit, d'indice j_t. Dans ce cadre, les principaux modèles expriment des contraintes et définissent des valuations fondées sur les positions relatives, dans la phrase source, de mots dont les traductions sont adjacentes dans la phrase cible. On notera j_g le plus petit indice d'un mot source non encore traduit après avoir traduit t mots, et j_d l'indice maximal d'un mot traduit. Un état du calcul des permutations est représenté sur la figure 7.7, réprésentant la situation où l'on a déjà traduit les mots $1, 3, 4$ et 8 d'une phrase de 9 mots. On a alors $j_g = 2$ et $j_d = 8$.

1	2	3	4	5	6	7	8	9
●	○	●	●	○	○	○	●	○

Figure 7.7 – Un état de la génération des permutations, où les mots déjà traduits sont figurés par ●, les alternatives restantes par ○.

7.4.1.1. *Les permutations locales*

L'écart le plus faible à la traduction "monotone" consiste à n'autoriser que des permutations *locales*, c'est-à-dire qui n'impliquent qu'un nombre restreint de mots qui sont voisins en source. La contrainte est paramétrée par une valeur d qui spécifie la taille du voisinage impliqué dans la permutation. Une manière alternative d'exprimer cette contrainte est d'imposer qu'à chaque instant, j_t soit dans l'intervalle $[t-d : t+d]$. Illustrons ce mécanisme pour une valeur du paramètre égale à $d = 1$: à la première étape, on peut choisir de traduire $\mathbf{f}_1$ ou $\mathbf{f}_2$; dans la première hypothèse, on continuera

24. Pour une phrase de trois mots, il est possible d'énumérer et d'évaluer les 3 ! = 6 permutations ; pour des phrases plus longues cela devient impossible.

par $\mathbf{f}_2$ ou $\mathbf{f}_3$; dans le second cas, toutefois, il faut impérativement traduire $\mathbf{f}_1$ pour achever la permutation entre les positions 1 et 2. Cette contrainte présente l'avantage de pouvoir être implantée dans le formalisme des transducteurs finis [KUM 05]. Si $\mathbf{f}$ dénote l'automate canonique représentant la phrase source, l'ensemble des permutations locales de $\mathbf{f}$ peut être calculé comme $\pi \circ \mathbf{f}$, avec π un transducteur fini et $\circ$ dénotant la composition de transducteurs. Il est à noter que, même sous cette contrainte très forte, le nombre de permutations d'une séquence continue de croître rapidement : pour $d = 1$, on montre que le nombre de permutations d'une séquence de longueur J est donné par le $n^{\text{ème}}$ terme de la suite de Fibonacci, qui croît comme $(1 + \sqrt{5})^J$. Les contraintes moins restrictives qui sont décrites ci-dessous autorisent un bien plus grand nombre de permutations.

7.4.1.2. *Les contraintes "IBM"*

Les contraintes dites IBM[25] disent dans quelle mesure il est possible de "retarder" la traduction d'un mot source. À chaque étape, le choix est restreint à la traduction d'un des d premiers mots non encore traduits, avec d un paramètre libre du modèle. Soit alors $\mathbf{f} = f_1 ... f_J$ la phrase source, et supposons que l'on fixe $d = 2$: le premier mot cible est une traduction soit de f_1, soit de f_2 ; si l'on a choisi de traduire d'abord f_1, à l'étape suivante on aura le choix entre f_2 et f_3 ; si, en revanche, on choisit de traduire d'abord f_2, le second choix sera entre f_1 et f_3, etc. Le développement de ces alternatives est représenté à la figure 7.8.

1	2	3	4	5	6	7	8	9	10
$t = 4$									
●	○	●	●	○	●	○	○	⋆	⋆
$t = 5$									
●	●	●	●	○	●	○	○	○	⋆
$t = 6$									
●	○	●	●	●	●	○	○	○	⋆
$t = 7$									
●	○	●	●	○	●	●	○	○	⋆

Figure 7.8 – Développement des alternatives de réordonnancement dans le modèle IBM, pour une valeur de d égale à 4. Les mots inatteignables sont représentés par $\star$. À l'étape $t = 4$, les choix possibles sont 2, 5, 7 et 8.

L'ensemble des permutations autorisées par ces contraintes peut être représenté sous la forme d'un automate fini, comme par exemple à la figure 7.9. Dans cet automate, chaque état correspond à un sous-ensemble de mots cibles déja traduits. Ainsi,

25. [LOP 09] conteste cette appellation, qui est pourtant attribuée à [BER 96] dans [TIL 03].

l'état 4 correspond à une configuration dans laquelle les mots 1 et 2 sont déjà traduits. Pour limiter l'empan des déplacements, il est possible de compléter ce modèle

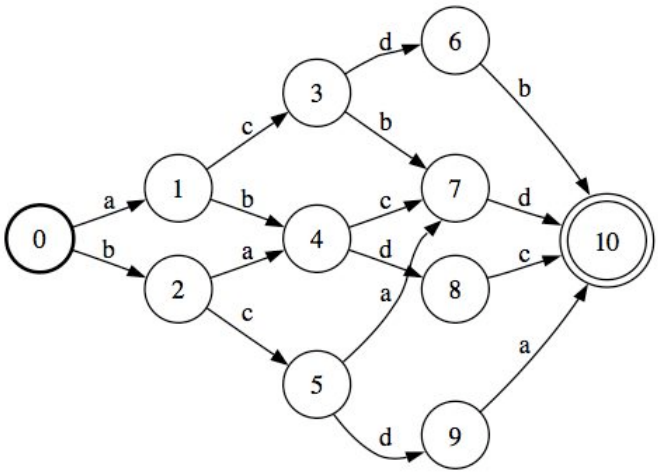


Figure 7.9 – Les permutations de '*a b c d*' autorisées par les contraintes IBM pour $d = 1$.

en ajoutant d'autres contraintes. Par exemple, on pourra spécifier un seuil de distance maximal entre l'indice d'un mot source et l'indice en cible de sa traduction ou bien limiter le nombre total de discontinuités[26].

26. À comprendre ici comme le nombre mots qui restent à traduire, mais dont les successeurs immédiats l'ont déjà été. Ainsi, à l'étape 7 de la figure 7.8, on a deux discontinuités correspondant aux mots d'indice 1 et 4.

Formellement, l'algorithme d'énumération des réarrangements possibles est le suivant :

Algorithme 4: Les permutations « IBM »

Données : d
$J_\circ = [1:d]$;
pour $t \in [1:I]$ **faire**
  $j_t \leftarrow \texttt{Choisir}(J_\circ)$
  si $(t + d \leq J)$ **alors**
    $J_\circ \leftarrow J_\circ \setminus \{j_t\} \cup \{t + d\}$
  fin
  sinon
    $J_\circ \leftarrow J_\circ \setminus \{j_t\}$
  fin
fin

Ces contraintes sont asymétriques et limitent plus les mouvements des mots les plus à droite (en source) que ceux des mots les plus à gauche. En particulier, si une phrase contient plus que k mots, il est impossible que le premier mot cible traduise le dernier mot source, alors que le contraire est toujours permis. Il est donc possible, comme le suggère [KAN 05] d'utiliser les contraintes « inverses » (c'est-à-dire s'appliquant sur la phrase lue de droite à gauche) pour modéliser d'autres situations de réordonnancement importants entre langues source et cible.

7.4.1.3. *Contraintes fondées sur la distance en source*

La contrainte la plus simple dans cette famille consiste à limiter les déplacements trop importants, par exemple en imposant une valeur maximale à la différence $\mid t - j_t \mid$. Cette manière de procéder vaut surtout quand les phrases source et cible ont des longueurs comparables, afin que le calcul de la différence des positions absolues dans les phrases source et cible ait un sens. On s'affranchit facilement de cette limitation en travaillant sur des *distances relatives* et donc en imposant que les différences $\mid j_t + 1 - j_{t+1} \mid$ (qui mesurent l'écart dans la phrase source entre deux mots adjacents dans la phrase cible) restent inférieures à un seuil fixé.

Comme le note [LOP 09], cette manière de procéder garantit simplement que les discontinuités (en source), pour des segments non-encore traduits, ont une taille bornée. Elle ne dit rien, en revanche, de la différence de position en source et en cible. Une contrainte un peu plus forte consiste à imposer qu'à chaque instant la distance entre j_g et j_t soit inférieure à un seuil. C'est la contrainte qui est imposée dans le système Moses [KOE 07b].

7.4.1.4. *Apprendre les réordonnancements possibles*

Les méthodes décrites dans les paragraphes précédents sont toutes fondées sur une définition *a priori* des réarrangements qui sont pertinents pour un couple de langue donné. Pour mieux modéliser les réarrangements effectivement présents dans les données, plusieurs équipes de recherche ont proposé d'utiliser des méthodes d'apprentissage automatique pour inférer les réordonnancements utiles [XIA 04, CRE 07].

Ainsi, la proposition de [CRE 07] consiste à apprendre, à partir de corpus alignés, des règles qui expriment les permutations possibles de la source. La première étape de cet apprentissage consiste à construire un corpus d'apprentissage appariant les phrases telles qu'elles apparaissent originellement avec la version réordonnée correspondant à l'ordre des mots en cible[27]. Un exemple typique se présente alors comme sur la figure 7.10, sur laquelle on a représenté la phrase source originale ($\mathbf{f}$), son alignement avec la phrase cible $\mathbf{e}$, et la phrase source réordonnée ($\pi(\mathbf{f})$).

$\mathbf{f}$ '*c'[PRO] est[VB] une[DET maison[NC] bleue[ADJ]*'
$\mathbf{e}$ '*this is a blue house*'
$\pi(\mathbf{f})$ '*c' est une bleue maison*'

Figure 7.10 – Phrase source réordonnée

Une fois cette base d'apprentissage constituée, il suffit d'extraire des règles de réécriture qui permettent de passer d'une représentation à l'autre. Pour plus de généralité, les auteurs préconisent d'exprimer les règles de réécriture en s'appuyant sur les catégories morphosyntaxiques des mots, plutôt que sur les mots eux-mêmes. Partant de l'exemple précédent, on extraira une règle telle que NC ADJ $\rightarrow$ ADJ NC, décrivant le déplacement de l'adjectif devant le nom au sein du groupe nominal. L'ensemble des réécritures applicables (de manière non-déterministe) à une phrase source définit alors l'ensemble des réordonnancements possibles de cette phrase et constitue l'espace de recherche qui est exploré pendant la construction d'une traduction (voir ci-après la section 7.5).

7.4.1.5. *Les réordonnancements hiérarchiques*

Un dernier modèle de réordonnancement très utilisé est celui des grammaires d'inversion, qui sont une sous-classe de grammaires hors-contextes synchrones, introduites en traduction automatique par les travaux de D. Wu [WU 97]. Une grammaire (hors-contexte) synchrone[28] est un système de récriture définissant des dérivations

27. Cette étape n'est pas sans difficulté, car les alignements ne sont pas toujours mot-mot...

28. La terminologie n'est pas ici complètement fixée : en théorie de la compilation, on parle de *Schéma de traduction dirigé par la syntaxe* en référence au travail original de [LEW 68, AHO 69].

parallèles de deux langages (formels), qui joueront ici le rôle de langues source et cible. Toute règle d'une grammaire G est de la forme $X \rightarrow \alpha_s ; \alpha_t$, avec α_s et α_t respectivement les parties droites en source et en cible ; on impose de surcroît que les non-terminaux de α_s et de α_t soient en *correspondance bi-univoque* (cette correspondance entre non-terminaux est figurée ci-dessous par des indices communs). Une grammaire synchrone dérive donc des couples de mots, et définit ainsi une relation (ou une transduction) entre deux langages. Une grammaire d'inversion est une grammaire hors-contexte synchrone simplifiée mettant en relation une phrase et certaines de ses permutations. Elle ne comprend qu'un seul non terminal X et deux productions génériques : $X \rightarrow X_1X_2 ; X_1X_2$ et $X \rightarrow X_1X_2 ; X_2X_1$. La production $X \rightarrow X_1X_2 ; X_1X_2$ décompose donc X en deux fragments, qui apparaissent dans le même ordre en cible et en source ; à l'inverse, la production $X \rightarrow X_1X_2 ; X_2X_1$ exprime une inversion entre la "source" et la "cible" (voir la figure 7.11). Les terminaux de G sont les mots de la phrase à permuter et G contient donc une production de la forme $X \rightarrow w_i ; w_i$ pour chaque mot de la phrase à permuter.

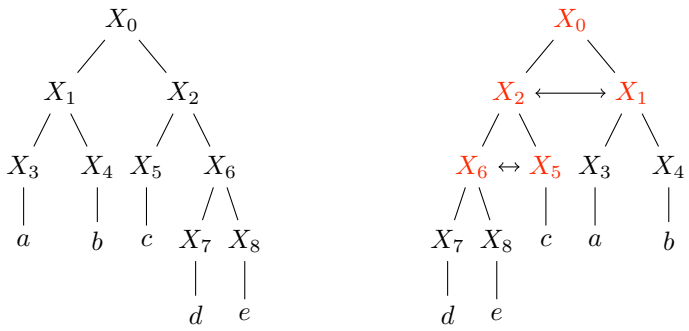


Figure 7.11 – Une permutation de '*a b c d e*' dans une grammaire d'inversion. La figure de gauche représente la dérivation en "source" qui reproduit l'ordre initial ; la figure de droite la dérivation en "cible" dans laquelle certains indices sont permutés. Les correspondances entre non-terminaux sont exprimés par des indices identiques dans les deux arbres ; les inversions entre non-terminaux sont figurées par des flèches.

L'article originel [WU 97] souligne qu'un tel mécanisme ne permet d'engendrer qu'un sous-ensemble des permutations possibles de $[1 : J]$: en particulier, seules 22 des 24 permutations de $[1 : 4]$ sont produites[29]. Plusieurs travaux ultérieurs ont permis de mieux cibler la nature exacte des permutations autorisées et interdites par ces grammaires [ZEN 03] et d'apprécier la portée linguistique de ces

29. On peut montrer (voir les références citées dans [ZEN 03]) que le nombre de permutations d'une phrase de longueur J autorisées par ce formalisme croît comme $(3 + \sqrt{8})^J$

restrictions [KAN 05, WEL 06, HUA 09]. De par leur simplicité, ces modèles de réordonnancement sont donc très utilisés à la fois pour construire des alignements [SAE 09, HAG 09] et pour définir l'espace de recherche de l'algorithme de traduction [EIS 06, ZHA 06].

7.4.1.6. *Généralisation à des segments*

Les contraintes décrites dans les paragraphes précédents se généralisent sans difficulté à des modèles de traduction de segments, à condition toutefois que soit connue la segmentation jointe des phrases source et cible qui sous-tend la traduction. Pour chaque segmentation, chacun des jeux de contraintes défini ci-dessus spécifie un ensemble de réordonnancements possibles et l'ensemble des réordonnancements licites d'une phrase s'obtient en faisant l'union sur toutes les segmentations possibles. À titre d'exemple, considérons la séquence '*a b c d*' : la segmentation '*[a b] [c d]*', associée à l'inversion des deux blocs, donne lieu à la permutation '*c d a b*' ; en revanche, la segmentation '*[a] [b c d]*', associée à la même transposition des segments, donne lieu à la permutation '*b c d a*'.

Pour conclure sur ces modèles, notons qu'ils ont pour la plupart été conçus dans le cadre de modèles de traduction à base de mots, dans lesquels tous les déplacements doivent être explicitement modélisés. Comme expliqué dans la section 7.3, l'avènement des modèles à base de segments permet d'éviter d'avoir à modéliser les déplacements locaux. Par exemple, l'extraction d'un bisegment tel que ('*mechanical translation*','*traduction automatique*') dispense de modéliser, pour ce bloc particulier, le changement de position relative entre le nom et l'adjectif quand on passe de l'anglais au français. On attend donc des modèles de réordonnancement qu'ils modélisent les restructurations syntaxiques majeures, qui donnent lieu à des déplacements importants. Les modèles décrits dans cette section (à l'exception du modèle hiérarchique) sont en fait peu adaptés à cette nouvelle tâche et ne sont utiles qu'en tant qu'ils définissent l'espace de recherche de la traduction (voir la section 7.5). Leur amélioration constitue donc un enjeu important pour la qualité des systèmes, surtout pour des systèmes de traduction mettant en jeu des paires de langues très différentes et fait toujours l'objet d'une recherche active.

Dans la suite de la discussion, un réordonnancement sera donc toujours associé à un couple formé d'une segmentation conjointe de la cible et de la source (notée σ) et d'une permutation (notée π) exprimant les différences d'ordre entre les segments correspondants. Ces deux informations définissent un *alignement* entre les segments source et cible, que nous noterons dans la suite $\mathbf{a} = (\sigma, \pi)$, par cohérence avec la section 7.3.1.

7.4.2. *Évaluer les permutations*

Même si l'on impose des contraintes sur les réordonnancements autorisés, leur nombre augmente de manière exponentielle avec la longueur de la phrase à réordonner. Il est donc nécessaire de valuer les réordonnancements et d'intégrer ces valuations pendant la recherche de la meilleure traduction (voir ci-dessous la section 7.5).

Rappelons que dans les systèmes de traduction à base de segments, chaque candidat traduction en langue cible est construit en aggrégeant des fragments $\mathbf{e}_1 \dots \mathbf{e}_l$ qui sont tels qu'il existe une permutation π de $[1 : l]$ et un ensemble de fragments $\{\mathbf{f}_1, \dots, \mathbf{f}_l\}$ tels que :

1) $\forall k \in [1\!:\!l], \pi(i_k) = k$ ou, de manière équivalente, $\pi^{-1}(k) = i_k$

2) $\mathbf{f}_{i_1}\mathbf{f}_{i_2}...\mathbf{f}_{i_l} = \mathbf{f}$

3) $\forall k \in [1\!:\!K], (\mathbf{f}_{i_k}\!:\!\mathbf{e}_k)$ est dans la table de traduction

Chaque segment en langue source et cible définit un intervalle ; dans la discussion qui suit nous notons $\mathbf{e}_i.g$ l'indice de début de cet intervalle, et $\mathbf{e}_i.d$ l'indice de fin de cet intervalle. Ces notations sont illustrées sur la figure 7.12. Sur cette figure la phrase anglaise est composée des segments $\mathbf{e}_1$ ='*she*', $\mathbf{e}_2$ ='*bravely*', $\mathbf{e}_3$ ='*seized that*', etc ; chacun de ces segments est en correspondance avec un segment source à travers π : ainsi $\pi(1) = 3$, dénotant l'appariemment de $\mathbf{e}_1$ =*she* avec $\mathbf{f}_3$ =*elle*, $\pi(2) = 6$ puisque '*bravely*' est la traduction de '*avec courage*', etc. Le segment source $\mathbf{f}_2$ ('*menaces de mort*') a comme frontières gauche et droite respectivement $\mathbf{f}_2.g = 4$ et $\mathbf{f}_2.d = 6$, correspondant respectivement au quatrième ('*menaces*') et au sixième ('*mort*') mot source.

source segmentée :
'[*en dépit de*]$_1$ [*menaces de mort*]$_2$ [*elle*]$_3$ [*a saisi sa*]$_4$ [*chance*]$_5$ [*avec courage*]$_6$'
source réordonnée et segmentée :
'[*elle*]$_3$ [*avec courage*]$_6$ [*a saisi sa*]$_4$ [*chance*]$_5$ [*en dépit de*]$_1$ [*menaces de mort*]$_2$'
cible segmentée :
'[*she*]$_1$ [*bravely*]$_2$ [*seized that*]$_3$ [*opportunity*]$_4$ [*despite*]$_5$ [*threats to her life*]$_6$'

Figure 7.12 – Identification des segments et des intervalles correspondants

7.4.2.1. *Modélisation du langage cible*

Une manière d'évaluer la plausibilité des permutations d'une séquence est d'utiliser des modèles stochastiques de la langue (voir la section 7.2.1 et le chapitre A) : chaque hypothèse de traduction, $e_1 \dots e_J$, reçoit une probabilité $\mathrm{P}(e_1 \dots e_J)$, qui exprime globalement la qualité syntaxique de la recombinaison de ces différents segments et donc en particulier la plausibilité de leur ordonnancement.

Toutefois, les modèles stochastiques de langue de type n-gramme, qui restent les modèles prédominants en traduction statistique, ne fournissent que des évaluations médiocres des réordonnancements possibles. Cette faiblesse, attendue si l'on se rappelle le caractère très local des contraintes syntaxiques capturées par un modèle n-gramme, est clairement mise en évidence dans l'étude de [ZAS 09]. L'expérience réalisée par ces auteurs consiste à permuter aléatoirement les mots d'une phrase initialement correcte et à utiliser les valuations de phrases produites par modèle statistique de langue pour rétablir l'ordre initial des mots. Une des conclusions de cette étude est que ces modèles de langue échouent à reconstruire l'ordre initial des phrases et qu'il est souvent possible de trouver des réordonnancements qui sont meilleurs, du point de vue de ces modèles, que les phrases non-permutées.

7.4.2.2. *Distorsion*

Un critère simple consiste à valuer chaque déplacement d'un facteur numérique qui dépend des distances, mesurées dans la langue source, entre deux segments qui sont adjacents dans la langue cible : on parle dans ce cas de mesure de *distorsion*. La valuation du réordonnancement induit par $\mathbf{a} = (\sigma, \pi)$ s'écrit alors comme la somme des valuations associées à chaque concaténation entre $\mathbf{e}_k$ et $\mathbf{e}_{k+1}$. Si chaque valuation correspond à la distance δ (mesurée en nombre de mots) entre la frontière droite de $\mathbf{f}_{i_k}$ et la frontière gauche[30] de $\mathbf{f}_{i_{k+1}}$, on obtient simplement :

$$c((\mathbf{f}_{i_k}\!:\!\mathbf{e}_k),(\mathbf{f}_{i_{k+1}}\!:\!\mathbf{e}_{k+1})) = \mid \mathbf{f}_{i_k}.d - \mathbf{f}_{i_{k+1}}.g + 1 \mid$$

En poursuivant l'exemple de la figure 7.12, le coût du réordonnancement associé au couple de segments $(\mathbf{f}_6, \mathbf{e}_2) = (\text{'}\textit{avec courage}\text{'} : \text{'}\textit{bravely}\text{'})$ et $(\mathbf{f}_4, \mathbf{e}_3) = (\text{'}\textit{a saisi sa}\text{'} : \text{'}\textit{seized that}\text{'})$ vaut $\mid 13 - 8 + 1 \mid \; = 6$.

Le coût d'une séquence complète de l segments s'en déduit selon :

$$c(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{k=1}^{l-1} c((\mathbf{f}_{i_k}\!:\!\mathbf{e}_k),(\mathbf{f}_{i_{k+1}}\!:\!\mathbf{e}_{k+1})) \tag{7.21}$$

Une valuation de ce type est utilisée, par exemple dans les systèmes Pharaoh [KOE 04] et Moses [KOE 07b]. Cette manière de procéder est très grossière, puisqu'elle ne prend en considération aucune information de nature syntaxique, ni même les propriétés morpho-syntaxiques ou l'identité des mots considérés, dont la propension à être impliqués dans des réordonnancements est très variable. Elle présente toutefois l'avantage de fournir des valuations faciles à calculer, et qui sont *additives*, c'est-à-dire qu'elles se décomposent sous la forme d'une somme de valuations locales qui n'impliquent que deux bisegments adjacents, comme dans l'"équation (7.21). Cette propriété s'avère essentielle pour construire des algorithmes de décodage efficaces (voir la section 7.5.2).

30. On pourrait aussi bien utiliser la distance entre les mots placés au centre de ces segments.

7.4.2.3. *Réordonnancement lexicalisé*

Une modélisation plus fine des réordonnancements dans le cadre des systèmes à base de segments a été proposée dans [TIL 04] et reprise sous des formes diverses dans de nombreux travaux. L'idée générale est toujours d'évaluer le coût de chaque concaténation, en intégrant cette fois d'autres informations dans la fonction c. Une grande variété de modèles rentrant globalement dans ce cadre ont été proposés. Il est possible de faire dépendre c de la distance (en source) entre deux segments adjacents en cible (comme précédemment) ou plus simplement, des positionnements relatifs des segments source et cible, qui définissent des classes de valeur de δ. Les configurations suivantes sont généralement distinguées, pour un couple de segments adjacents en cible :

1) les deux segments correspondants sont également adjacents en source et l'ordre est respecté (formellement $\pi(k+1) = \pi(k)+1$), auquel cas on dit que la concaténation a une *orientation monotone* (à droite pour $\mathbf{e}_k$ et à gauche pour $\mathbf{e}_{k+1}$).

2) les deux segments sont également adjacents en source et l'ordre est inversé (formellement $\pi(k+1) = \pi(k)-1$), auquel cas on dit qu'on a affaire à une *orientation inversée* (à droite pour $\mathbf{e}_k$ et à gauche pour $\mathbf{e}_{k+1}$).

3) les deux segments ne sont plus adjacents en cible mais l'ordre est préservé (formellement $\pi(k+1) > \pi(k)+1$).

4) les deux segments ne sont plus adjacents en cible et, de surcroît, l'ordre est inversé (formellement $\pi(k+1) < \pi(k)-1$).

Toutes ces configurations ne sont pas nécessairement distinguées : dans le système Moses [KOE 07b], il est possible de distinguer l'orientation *monotone* des trois autres, ou bien encore de distinguer *monotone*, *inversion*, et les deux autres, à l'instar de la proposition de [TIL 04]. Dans le modèle de réordonnancement décrit dans [KUM 05], les contraintes impliquent que les seules configurations possibles sont *monotone* ou *inversion*, etc.

Il est, d'autre part, possible de faire dépendre c de l'identité des segments source, voire de l'identité des segments source et cible. Par exemple, le modèle proposé par [ALO 06] définit c comme une fonction de la distorsion et des segments source uniquement. Par contraste, le modèle proposé dans [KOE 07b] ne distingue que quelques orientations (monotone, inversion, etc) mais fait dépendre c de la valeur de l'orientation et des bisegments impliqués dans la concaténation.

Dans toutes ces approches, la fonction de score c s'appuie sur des modèles probabilistes conditionnels, qui prédisent la valeur de l'empan en source entre deux segments adjacents en cible, conditionnellement aux segments impliqués. Par exemple, dans l'approche implémentée dans [KOE 07b], on décompose le coût d'une concaténation en deux termes, l'un pour le bisegment de gauche (c_g) et l'autre pour le bisegment de

droite (c_d) :

$$c((\mathbf{f}_{i_k}{:}\mathbf{e}_k),(\mathbf{f}_{i_{k+1}}{:}\mathbf{e}_{k+1}),k) = c_g((\mathbf{f}_{i_k}{:}\mathbf{e}_k),k) + c_d((\mathbf{f}_{i_{k+1}}{:}\mathbf{e}_{k+1}),k) \qquad (7.22)$$

Dans cette formulation, c_g est calculé comme $-\log(\mathrm{P}(o_g|(\mathbf{f}_{i_k}{:}\mathbf{e}_k)))$, avec o_g l'orientation relative du bisegment de gauche par rapport au bisegment de droite (monotone, inversée, etc). Ce terme modélise donc la propension du bisegment $(\mathbf{f}_{i_k} : \mathbf{e}_k)$ à se positionner par rapport aux bisegments adjacents ; c_d s'écrit de manière similaire, à ceci près qu'on modélise maintenant le comportement de $(\mathbf{f}_{i_{k+1}} : \mathbf{e}_{k+1})$ par rapport au bisegment qui précède. Ces probabilités peuvent être estimées par la méthode du maximum de vraisemblance sur des corpus parallèles alignés, sous la forme de rapports de comptes.

Cette modélisation des coûts de réordonnancement présente l'avantage, déjà mentionné pour les valuations fondées sur la distorsion, de rester additive : le coût global de réordonnancement d'une hypothèse de traduction se calcule comme une somme de termes selon :

$$c(\mathbf{f},\mathbf{e},\mathbf{a}) = \sum_{k=1}^{K-1} c((\mathbf{f}_{i_k}{:}\mathbf{e}_k),(\mathbf{f}_{i_{k+1}}{:}\mathbf{e}_{k+1}))$$

Mentionnons pour finir qu'il est naturellement possible d'utiliser d'autres familles de modèles pour construire ces évaluations, à la façon par exemple de [ZEN 06], qui utilise des modèles exponentiels entraînés discriminativement, ou encore [XIO 06], qui a toutefois recours à des contraintes hiérarchiques plus complexes.

7.5. Traduire : un problème de recherche

Dans les sections précédentes, nous nous commes focalisés sur la construction de modèles statistiques à partir de ressources textuelles multilingues (modèles de traduction et de réordonnancement) ou monolingues (modèles de langue). Dans cette section, nous expliquons comment ces modèles sont utilisés pour produire des traductions nouvelles. La première question qui se pose est naturellement celle de la combinaison de ces différents modèles, qui sont chacun en charge d'évaluer un aspect particulier des traductions qui sont produites. Cette question est abordée à la section 7.5.1. Nous poserons ensuite le problème de la recherche d'une bonne traduction dans toute sa généralité et discuterons sa complexité théorique. Nous présenterons enfin quelques approches heuristiques pour résoudre ce problème : celles qui utilisent une recherche en meilleur d'abord, d'une part (Section 7.5.4.1), et celles qui utilisent une recherche locale (Section 7.5.4.2) d'autre part.

7.5.1. *Combiner les modèles*

7.5.1.1. *Position du problème*

Rappelons que dans les systèmes de traduction à base de segments, le processus de traduction consiste à combiner les décisions suivantes :

– segmenter la phrase cible $\mathbf{f}$ en segments de longueur variable $\mathbf{f}_1 \ldots \mathbf{f}_l$

– pour chaque segment $\mathbf{f}_k$, choisir un équivalent en langue cible $\mathbf{e}_k$

– réarranger les segments $\mathbf{e}_1 \ldots \mathbf{e}_l$ de façon à obtenir la traduction $\mathbf{e}$

Ces décisions sont naturellement interdépendantes et ne sont pas dans la pratique effectuées dans l'ordre suggéré ci-dessous, mais plutôt de façon simultanée.

Dans sa version la plus élémentaire, un système de traduction statistique fonde ces décisions en s'appuyant sur les trois modèles que nous avons présentés ci-dessus et qui chacun produisent des grandeurs numériques :

– un modèle de traduction, qui évalue numériquement la qualité d'un appariemment entre une phrase source $\mathbf{f}$ et une phrase cible $\mathbf{e}$, et délivre un coût $c_t(\mathbf{f}, \mathbf{e}, \mathbf{a})$;

– un modèle de réordonnancement, qui évalue la plausibilité du réordonnancement induit par cet appariemment et délivre un coût $c_r(\mathbf{f}, \mathbf{e}, \mathbf{a})$;

– un modèle de langage, qui évalue la qualité de la phrase cible ainsi formée et délivre un coût $c_l(\mathbf{e})$ (généralement $-\log(\Pr(\mathbf{e}))$).

La meilleure traduction est celle qui réalise le meilleur compromis entre les évaluations de ces différents modèles. Pour élaborer ce compromis, qui est réalisé lors du décodage, il est essentiel de quantifier l'importance de chacun des modèles dans l'évaluation globale de l'appariemment entre $\mathbf{f}$ et $\mathbf{e}$. Cette étape est cruciale, car les coûts calculés par les différents modèles ne se comparent pas directement entre eux : ils n'ont pas nécessairement le même domaine, ni la même dynamique. Un moyen simple de combiner ces valeurs est de former leur combinaison linéaire et de valuer chaque hypothèse de traduction par :

$$c(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \lambda_t c_t(\mathbf{f}, \mathbf{e}, \mathbf{a}) + \lambda_r c_r(\mathbf{f}, \mathbf{e}, \mathbf{a}) + \lambda_l c_l(\mathbf{e}) \tag{7.23}$$

Une fois ce principe posé, il est facile de l'étendre en rajoutant des fonctions de coût supplémentaires : par exemple une évaluation de la différence de longueur entre source et cible, des coûts évaluant l'adéquation des segments source figurant dans l'hypothèse avec des constituants syntaxiques, ou bien encore plusieurs modèles de langue, plusieurs modèles de traduction, etc. Une formulation générale, connue sous le nom de *modèle log-linéaire*[31], est alors la suivante (on note $c_m()$ les fonctions de

31. La terminologie est ici de nouveau trompeuse : s'il est vrai que certains coûts dérivent de log-probabilités (typiquement le coût du modèle de langage), dans la pratique de nombreux

coût et λ_m les paramètres associés) :

$$c(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{m=1}^{M} \lambda_m c_m(\mathbf{f}, \mathbf{e}, \mathbf{a}) \tag{7.24}$$

Sous cet angle, la meilleure traduction de la phrase source $\mathbf{f}$ est "simplement" la phrase $\mathbf{e}$ pour laquelle le coût $c(\mathbf{f}, \mathbf{e}, \mathbf{a})$ est minimal ; la recherche de cet optimum définit le programme d'optimisation que doit résoudre le décodeur. Avant d'étudier plus en détail ce problème, il reste une question à résoudre, celle du réglage des paramètres $\boldsymbol{\lambda} = \{\lambda_m, m = 1 \dots M\}$.

7.5.1.2. *Réglage des coefficients par minimisation de l'erreur de traduction*

Le réglage du vecteur de coefficients $\boldsymbol{\lambda}$ est effectué, comme pour le reste des paramètres, par optimisation numérique sur un corpus de développement $\mathbf{D}$ associant un ensemble de phrases source $\mathbf{F}$ et des traductions de référence $\mathbf{E}$. Pour cette étape, il est indispensable de savoir évaluer les variations des performances du système en fonction des variations des paramètres. Le plus souvent, cette évaluation repose sur des métriques automatiques (voir la section 7.6) calculées en comparant les traductions de ce corpus avec une (ou plusieurs) traductions de référence. Pour chaque valeur de $\boldsymbol{\lambda}$, la qualité du système sur $\mathbf{D}$ est évaluée par $L(\boldsymbol{\lambda}, \mathbf{D})$. On cherche alors la valeur des coefficients qui permette de traduire au mieux ce corpus, soit formellement les coefficients qui résolvent :

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}}\, L(\boldsymbol{\lambda}, \mathbf{D}) \tag{7.25}$$

Ce programme d'optimisation est non trivial, dans la mesure où les coefficients λ_k influencent de manière très indirecte la fonction objectif définie par (7.25) : en changeant les réglages, on change les traductions produites, ce qui modifie la mesure de qualité du système. En particulier, la fonction objectif de ce programme n'est pas différentiable en $\boldsymbol{\lambda}$ ce qui interdit d'utiliser les techniques classiques d'optimisation numérique. Lorsque l'on a peu de paramètres à régler, il est naturellement possible de faire une recherche exhaustive ; lorsque (comme c'est le cas en pratique) le nombre de coefficients est de l'ordre de la dizaine ou plus, il faut utiliser des algorithmes d'optimisation plus sophistiqués.

La méthode standard (*Minimum Error Rate Training* souvent abrégée par *MERT*) permettant de réaliser cette optimisation est présentée dans [OCH 03a, ZAI 09] et diverses variantes et améliorations de cette méthode sont proposées dans [CER 08,

---

coûts ne sont pas dérivées de tels modèles : on a simplement affaire à une combinaison *linéaire* de différentes fonctions de coût.

MOO 08, FOS 09, MAC 08]. La remarque principale sur laquelle s'appuie cet algorithme est que la fonction objectif du programme (7.25) est *constante par morceau* : de petits changements des coefficients laissent les meilleures hypothèses de traduction (et la fonction objectif) inchangées, jusqu'à un point où une traduction change, ce qui fournit une nouvelle valeur pour la fonction objectif. Une conséquence est qu'il est possible de calculer exactement l'optimum le long de n'importe quelle direction, ce qui permet de mettre en œuvre des techniques de recherche d'un optimum telles que celle de Powell [POW 64]. Cette méthode est loin d'être parfaite et conduit, en particulier, à des valeurs des coefficients qui sont très variables (d'une expérience à l'autre ou d'un corpus à l'autre) et qui sont, en conséquence, peu robustes à des petits changements des conditions expérimentales.

Plus embarrassant peut-être, le programme d'optimisation spécifié par (7.25) conduit souvent, dès que le nombre de paramètres dépasse les quelques dizaines, à des solutions qui sur-évaluent la performance en apprentissage et généralisent mal à des données de test. L'utilisation d'un critère pénalisé ou l'optimisation de fonctions objectif alternatives (inspirées des programmes de maximisation de la marge), comme dans [WAT 07, CHI 08], permet en théorie de limiter ces problèmes.

7.5.2. *Le problème du décodage*

Le rôle du module de décodage est de construire une traduction pour n'importe quelle phrase source. Dans le cadre de la traduction automatique, la meilleure hypothèse de traduction est naturellement celle qui est la mieux évaluée par les différents modèles : traduire consiste alors à chercher, parmi toutes les phrases qui sont susceptibles de s'aligner avec $\mathbf{f}$, la phrase $\mathbf{e}^*$ qui maximise la fonction de coût $c(\mathbf{f}, \mathbf{e}, \mathbf{a})$ définie à l'équation (7.24). Traduire est donc essentiellement un problème de recherche. Pour apprécier la difficulté de ce problème, il convient de rappeler quel est l'ensemble des candidats $\mathbf{e}$ qui doit être évalué. Il s'agit de toutes les phrases que l'on peut construire en :

– segmentant $\mathbf{f}$ de toutes les manières possibles, il y a 2^{J-1} manières de faire ;

– traduisant chaque segment $\mathbf{f}_k$ de toutes les manières qu'autorise le modèle de traduction ;

– permutant les traductions partielles $\mathbf{e}_k$ de toutes les manières possibles, il y en a $k!$ pour une segmentation en k segments. L'espace de recherche est donc particulièrement grand.

La recherche de la meilleure hypothèse de traduction implique donc la résolution d'un problème d'optimisation combinatoire, qui, sauf hypothèse simplificatrice sur la fonction de coût ou sur l'espace de recherche, ne peut être résolu (effectivement) de manière exacte. Une démonstration de la complexité théorique de ce problème est donnée dans [KNI 99], qui prouve que, même sous des hypothèses relativement

simplistes, ce problème de recherche est NP-difficile. Cette démonstration repose sur le fait que l'exploration de tous les réordonancements possibles rend ce problème formellement analogue[32] à celui du problème du voyageur de commerce, un problème combinatoire notoirement difficile [ZAS 09].

Face à cette difficulté, deux stratégies de contournement sont possibles, qui sont présentées dans les sections qui suivent :

– restreindre l'espace de recherche de manière à pouvoir utiliser des stratégies de résolution exacte ;

– adopter des techniques de recherche heuristique et s'accomoder de solutions approximatives.

Avant de débuter cette discussion, introduisons quelques notations : dans la suite, h dénote une *hypothèse de traduction en cours de construction*, à laquelle sont associés :

– des coûts, correspondant aux différents modèles, notés ici respectivement $h.c_t$ (le coût du modèle de traduction), $h.c_s$ (celui du modèle de langage), etc. Comme pour une hypothèse complète, le coût global de h se déduit selon (7.24) ;

– un préfixe de phrase cible, noté $h.\mathbf{e}$, obtenu par concaténation de $h.t$ segments $\mathbf{e}_1 \dots \mathbf{e}_t$

– une couverture, notée $h.span$, qui enregistre les fragments de la phrase source $\mathbf{f}$ dont la traduction figure déjà dans h. Formellement, $h.span$ est une union de t intervalles disjoints de $[1:J]$.

7.5.3. *Algorithmes de recherche exacte*

Pour les modèles de traduction à base de segments et les fonctions d'évaluation qui leurs sont associées, la recherche peut être réalisée de manière exacte *lorsque l'on restreint l'ensemble des réordonnancements possibles*. Nous considérons tout d'abord le cas où il n'y a pas du tout de réordonnancement, puis le cas de réordonnancements locaux (cf. la section 7.4.1).

7.5.3.1. *Traduire sans réordonnancement*

Considérons l'exemple jouet correspondant au modèle de traduction B représenté dans le tableau 7.3, dans lequel on introduit les notations suivantes : si $b = B[i]$ est un bisegment, alors $b.\mathbf{f}$ (resp. $b.\mathbf{e}$) est le segment source (resp. cible), $b.c_t$ est le coût du modèle de traduction[33] (non représenté dans la table), $b.c_s$ le coût interne du modèle

32. C'est-à-dire précisément : il existe une réduction polynomiale de l'un vers l'autre.

33. Nous considérons les évaluations des bisegments comme des coûts : plus une association est vraisemblable, plus son coût sera faible. Ces coûts correspondent donc à l'opposé du log des probabilités calculées en (7.19) et (7.20).

de langage (voir supra), $b.J$ (resp. $b.I$) la longueur du segment source (resp. cible), $b.\mathbf{e}[i]$ est le $i^{\text{ème}}$ mot du segment cible.

$b.\mathbf{f}$	$b.\mathbf{e}$	$b.I$	$b.J$	$b.e[1]$	$b.c_s$
'*this*'	'*ce*'	1	1	'*ce*'	0
'*this*'	'*cet*'	1	1	'*cet*'	0
'*this*'	'*cette*'	1	1	'*cette*'	0
'*this small*'	'*ce petit*'	2	2	'*ce*'	$-\log(\text{P}(\text{'}petit\text{'}\vert\text{'}ce\text{'}))$
'*this small*'	'*cette petite*'	2	2	'*cette*'	$-\log(\text{P}(\text{'}petite\text{'}\vert\text{'}cette\text{'}))$
'*small*'	'*petit*'	1	1	'*petit*'	0
'*small*'	'*petite*'	1	1	'*petite*'	0
'*small*'	'*petits*'	1	1	'*petits*'	0
'*country*'	'*pays*'	1	1	'*pays*'	0
'*country*'	'*nation*'	1	1	'*nation*'	0
'*country*'	'*campagne*'	1	1	'*campagne*'	0

Tableau 7.3 – Une table de traduction simplifiée

L'ensemble des hypothèses de traduction se représente sous la forme d'un graphe étiqueté orienté acyclique, ou, de manière équivalente, sous la forme d'un automate fini (voir la figure 7.13). Ce graphe contient un sommet par préfixe de $\mathbf{f}$ et comprend un arc étiqueté par $B[i].\mathbf{f}$ entre les sommets k et $k+l$ pour tous les bisegments vérifiant $B[i].\mathbf{f} = \mathbf{f}[k:k+l-1]$[34]. On dit alors que les mots source de l'intervalle $[k:k+l]$ *sont couverts* par le bisegment x.

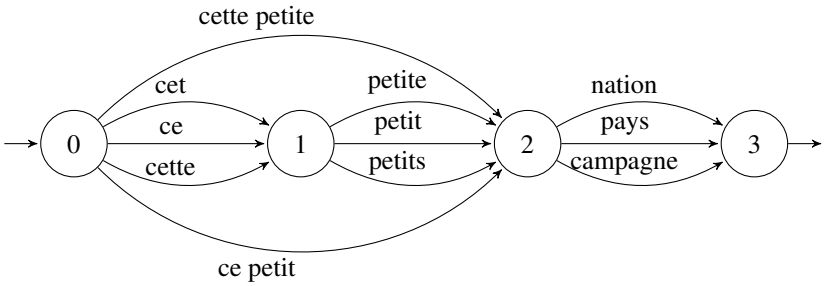


Figure 7.13 – L'espace de recherche du décodage monotone sans modèle de langue cible . La phrase source est '*this small country*'

34. Il est d'usage de ne pas les prendre tous, mais d'opérer dès cette étape un filtrage basé sur les scores $B[i].c_t$, de manière à ne garder que quelques dizaines de bisegments par segment source.

Si l'on n'utilise que des fonctions de coût non-contextuelles et additives, c'est-à-dire qui se décomposent comme des sommes de valuation élémentaires associées aux bisegments, il est facile de distribuer les coûts des différents modèles sur les arcs de ce graphe. Ainsi, par exemple, l'arc joignant les nœuds 0 et 2 traduit le segment '*this small*' et porte un coût c qui intègre :

– la contribution de ce bisegment au coût total du modèle de traduction, soit $\lambda_t.B[3].c_t$;

– un coût fixe lié à l'ajout d'un segment supplémentaire, qui favorise les hypothèses utilisant des segments longs.

En procédant de cette manière, la valuation totale d'un chemin complet entre source (le nœud 0 de la Figure 7.13) et le puits (le nœud 3) dans le graphe, qui représente une hypothèse de segmentation+traduction particulière, correspond au coût $c(\mathbf{f}, \mathbf{e}, \mathbf{a})$. La recherche de la meilleure traduction se résout de manière exacte en temps polynomial en utilisant des algorithmes de plus courts chemins, ce qui conduit à une complexité quadratique en la longueur de la phrase source. La traduction se déroule de la gauche vers la droite, en construisant des hypothèses de traduction pour des préfixes de longueur croissante de $\mathbf{f}$: les traductions du premier mot, puis des deux premiers mots etc. À chaque indice j de $[1 : J]$ est donc associé un ensemble $\mathcal{H}_j$ d'hypothèses de traduction, chacune nantie d'un score partiel qui correspond à la somme des coûts accumulés jusqu'au sommet i. En l'absence d'autre valuation, à chaque itération de l'algorithme, il n'est nécessaire de développer que la meilleure des hypothèses de $\mathcal{H}_j$ en abandonnant toutes les autres, en application du principe de *recombinaison* des hypothèses. Selon ce principe, qui garantit l'efficacité des algorithmes de plus courts chemins utilisant la programmation dynamique, de deux hypothèses partielles parvenant au même état du graphe de recherche, seule la meilleure doit être conservée, l'autre ne pouvant (dans le futur) produire une meilleure solution. Ainsi, par exemple, au sommet 2 du graphe représenté Figure 7.13, sont associées les hypothèses suivantes {['*ce*']['*petit*']; ['*ce*']['*petite*']; ['*ce*']['*petits*']; ['*ce petit*']; ['*cette petite*']...} : seule la meilleure de ces hypothèses figurera dans le meilleur chemin complet.

La prise en compte d'un modèle de langage bigramme (en langue cible) complique un peu ce calcul, en particulier parce que le coût syntaxique c_s de chaque segment cible dépend de son contexte gauche. Regardons en effet comment ce coût se calcule. Pour les bisegments dont la cible contient plus de deux mots, il faut tout d'abord intégrer un coût syntaxique "interne", qui est simplement l'opposé de la log-vraisemblance bigramme en cible (voir la table 7.3 supra), qui se calcule selon $B[i].c_s = -\sum_{t=2}^{T[x].I} \log\left(\mathrm{P}(B[i].e[t]|B[i].e[t-1])\right)$. Mais il existe également un coût "externe", qui correspond au score bigramme du mot $B[i].e[1]$ *étant donné le mot qui le précède dans l'hypothèse courante* : il n'est donc plus possible de le porter directement sur l'arc correspondant à i. La solution est de construire une partition des hypothèses de $\mathcal{H}_j$ en fonction du dernier mot cible w qu'elles contiennent, donnant lieu à des sous-ensembles $\mathcal{H}_{j,w_1} \ldots \mathcal{H}_{j,w_n}$. Au sein de chaque ensemble $\mathcal{H}_{j,w}$, seule

la meilleure hypothèse est conservée et développée, en considérant tous les biseg-ments $B[i]$ qui couvrent les positions $j+1$ et suivantes le long d'un arc qui intègre, en plus du coût du modèle de traduction, un coût bigramme $-\log \mathrm{P}(B[i].e[1]|w)$. Une implantation simple de cette approche est de construire le graphe de recherche en y intégrant un nœud par préfixe de $\mathbf{f}$ et par dernier mot cible, comme sur la figure 7.14 : par exemple, le nœud (2c) correspond à toutes les hypothèses traduisant le préfixe '*this small*' et se terminant par '*petite*'.

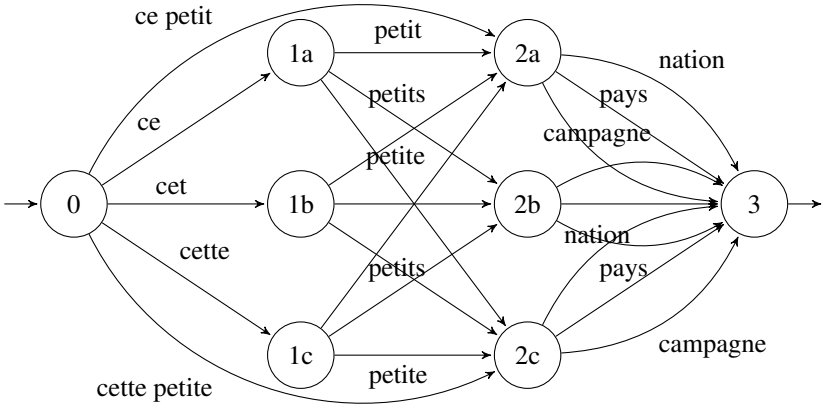


Figure 7.14 – L'espace de recherche du décodage monotone avec modèle de langue cible bigramme. La phrase source est '*this small country*'. Tous les arcs ne sont pas complètement représentés.

Cette approche se généralise à des modèles de langue cible d'ordre supérieur sans autre difficulté, sinon la multiplication des sous-ensembles d'hypothèses qu'il faut considérer à chaque instant. On aura en effet un ensemble d'hypothèses pour chaque histoire possible du modèle de langage, un nombre qui croît exponentiellement avec l'ordre du modèle de langage (voir la section 7.2.3). Lorsque l'on utilise de « gros » modèles de traduction[35] couplés à des modèles de langue d'ordre 3 ou 4, explorer tous ces ensembles d'hypothèses devient prohibitif, ce qui impose d'utiliser des techniques d'élagage (voir ci-dessous), au risque de ne plus trouver la solution optimale.

35. Le corpus Europarl [KOE 05] contient aujourd'hui de l'ordre d'un million et demi de phrases alignées ; le modèle de traduction appris par les techniques standard sur ce corpus contient plusieurs dizaines de millions d'entrées.

7.5.3.2. *Traduire avec des réordonnancements locaux*

Considérons maintenant comment cette stratégie de recherche doit être amendée pour prendre en compte des réordonnancements locaux, en considérant simplement le cas le plus simple, celui dans lequel deux segments source adjacents peuvent être inversés en cible (cf. la section 7.4.1). Dans ce cas, la seule modification à porter à l'algorithme précédent consiste à modifier la définition du graphe sous-jacent à la recherche (représenté, dans le cas monotone, à la figure 7.13). Pour explorer les réordonnancements locaux, il suffit en effet de procéder en deux étapes[36] consistant à :

1) construire l'ensemble des permutations admises de la phrase source sous la forme d'un graphe de mot valué G_r

2) explorer G_r de la même manière que précédemment, à ceci près que :
   a) il existe un ensemble d'hypothèses partielles pour chaque sommet de G_r ;
   b) le calcul du coût total doit intégrer un terme évaluant les réordonnancements.

La première de ces étapes, la construction de G_r, est sans difficulté : il suffit de considérer toutes les segmentations de $\mathbf{f}$ qui sont induites par T, et pour chaque paire de segments adjacents, considérer la possibilité qu'ils soient ou non permutés en cible. Les valuations des arcs de ce graphe correspondent au coût associé au déplacement correspondant. Toujours pour la phrase $\mathbf{f}$ = '*this small nation ...*', le graphe correspondant est partiellement représenté à la figure 7.15.

En remplaçant chaque arc de G_r étiqueté par le segment $\mathbf{f}$ par des arcs étiquetés par des traductions possibles de $\mathbf{f}$, on obtient exactement l'ensemble de toutes les hypothèses de traduction. Il reste alors à l'explorer comme précédemment, en construisant un ensemble d'hypothèses par état de G_r et par historique du modèle de langage.

L'implémentation la plus générique des techniques décrites dans cette section repose sur des algorithmes de manipulation de transducteurs finis pondérés : c'est la voie suivie notamment par [BAN 02, CAS 04] pour des modèles de mots, ainsi que, pour des modèles de segments par [KUM 03, KUM 06]. Cette approche présente l'avantage de n'utiliser que des algorithmes éprouvés, pour lesquels il existes des implantations efficaces et de donner lieu à de multiples variantes couramment utilisées dans le domaine des technologies vocales [MOH 02], telles que notamment l'élagage d'une partie de l'espace de recherche (voir ci-dessous), la génération de listes des n meilleures hypothèses de traduction ou de graphes de mots[37], la traduction des graphes de mots en langue source (par exemple issus de systèmes de reconnaissance vocale), etc.

---

36. Nous adoptons ce point de vue pour simplifier l'exposé : il est naturellement possible d'implanter cette idée en développant le graphe de recherche dynamiquement en même temps qu'on le parcourt.

37. Formellement, d'automates pondérés représentant un ensemble d'hypothèses de traduction.

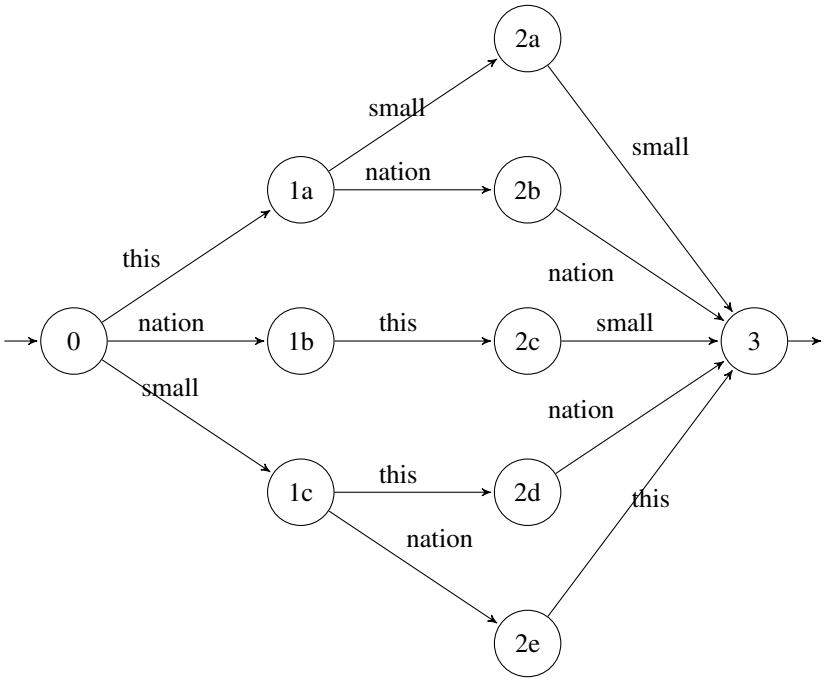


Figure 7.15 – L'ensemble partiel des permutations locales en langue source de '*this small nation*'

7.5.4. *Algorithmes de recherche heuristique*

Les techniques de recherche exactes sont limitées par la taille de l'espace des réordonnancements qui est construit et évalué durant la recherche. Pour considérer des réordonnancements à plus longue distance, il faut renoncer aux techniques de recherche exacte et utiliser des stratégies de recherche heuristique. Quatre approches principales sont discutées dans la littérature :

– l'utilisation de techniques de recherche de type *meilleur d'abord*, qui s'inspirent principalement de l'algorithme A^* [PEA 84] ;

– l'utilisation de techniques d'heuristiques gloutonnes de *recherche locale* : il s'agit ici de construire une première hypothèse de traduction, puis de chercher à l'améliorer en explorant des voisinages de cette solution ;

– l'utilisation de techniques de décodage monotone (voir supra), appliqués à sur des ensembles de permutation de taille combinatoirement grande : cette stratégie peut s'appliquer en source (génération de permutations de la source conjoint à un décodage monotone), ou bien en cible (génération d'une traduction monotone et exploration des permutations possibles des hypothèses de traduction). C'est, par exemple l'approche

suivie par [CRE 07]. La difficulté principale est la génération des permutations autorisées, puisque la recherche elle-même est effectuée selon les principes généraux décrits ci-dessus ;

– la transformation du problème de décodage en un problème combinatoire connu et utilisation de solveurs génériques pour ce problème ; [GER 01] applique cette technique en transformant le problème de la traduction (avec des modèles de mots) en un problème de programmation linéaire en nombres entiers ; plus récemment, [ZAS 09] utilise une réduction vers le problème du voyageur de commerce.

Nous détaillons dans la suite les deux premières de ces familles de techniques. Notons en préambule que, presque indépendamment[38] de la stratégie de recherche utilisée, il existe une technique éprouvée pour améliorer les résultats de la recherche, consistant à procéder par raffinement progressif de l'espace de recherche (on parle de stratégie *multi-passe*). La première passe, plutôt que produire une solution unique, a pour fonction de sélectionner une petite sous-partie de l'espace de recherche, sous la forme d'une liste des n-meilleures solutions, ou encore d'un graphe de mots. Une fois cette sélection opérée, il est possible d'utiliser des modèles plus sophistiqués pour réévaluer, lors d'une nouvelle recherche, les hypothèses de traduction qui restent en compétition et effectuer une sélection mieux informée de la meilleure traduction. Les stratégies de recherche multi-passe [KUM 04, TRO 08], en particulier celles qui impliquent la combinaison de plusieurs systèmes [ROS 07, MAT 08], constituent donc un domaine de recherche encore actif.

7.5.4.1. *Recherche 'meilleur d'abord'*

Le principe général des algorithmes de recherche 'meilleur d'abord' est d'explorer une partie de l'espace des réordonnancements permis tout en construisant des traductions partielles de portions arbitraires de la phrase source. Contrairement aux techniques de décodage monotone présentées ci-dessus, la phrase source est donc traduite *dans un ordre quelconque*. Chaque hypothèse de traduction h doit donc mémoriser la partie de la phrase source $h.span$ qui a déjà été traduite, afin qu'un même fragment ne soit pas traduit plusieurs fois. Conceptuellement, l'algorithme effectue les opérations suivantes de manière itérative, jusqu'à épuisement du stock des hypothèses de traduction :

1) choisir une hypothèse h de coût partiel $h.c$ *prometteuse* dans le stock d'hypothèses disponible, h couvre $h.span$

2) choisir un fragment de la phrase source pas encore traduit par h

3) le traduire, donnant lieu à une nouvelle hypothèse partielle h' de coût $h'.c$, qui couvre une portion plus grande de la source $h'.span$

38. En réalité, toutes les stratégies de recherche listées ci-dessus ne se prêtent pas aussi bien à la recherche des n meilleures solutions.

Cette approche se généralise sans difficulté réelle à la production de listes des n-meilleures hypothèses et de graphes de mots, ce qui explique en partie sa popularité.

L'efficacité de cette approche dépend du processus de sélection de l'hypothèse qui doit être développée à un instant donné. Cette sélection implique de pouvoir comparer des hypothèses de traduction *couvrant des portions différentes et de longueur variable de la phrase source*. L'approche la plus commune est l'Algorithme 5, souvent présentée sous le nom de décodage multi-pile[39] (*multi-stack decoding*). Il résoud partiellement cette difficulté en subdivisant l'ensemble des hypothèses partielles selon la longueur de la zone couverte de la phrase source. Pour une phrase source de longueur J, l'algorithme maintient donc J files de priorité différentes. Si deux hypothèses appartenant à la même file sont ainsi plus directement comparables, leurs coûts partiels ne permettent pas de les comparer de manière complètement équitable : en effet, les hypothèses ayant déjà couvert des parties *faciles à traduire* de la cible auront souvent un coût inférieur à celui d'hypothèses couvrant des parties *difficiles*. Toute comparaison équitable doit alors également prendre en compte du coût (futur) de traduction des parties non-encore couvertes, ou plutôt, d'une approximation de ce coût, puisque le calculer de manière exacte demanderait de conduire la recherche à son terme. Nous noterons $h.g$ cette approximation, qui se décompose en une approximation du coût de traduction $h.g_t$, du coût du modèle de langage $h.g_s$, etc. Trois éléments supplémentaires s'avèrent nécessaires pour décrire formellement cette stratégie de recherche :

– une politique de *recombinaison* des hypothèses partageant un même futur, instanciée ci-dessous dans la fonction `AjouteRecombine` ;

– une politique *d'élagage* de ces files, dont la taille croît combinatoirement avec la longueur de la zone couverte ; cet élagage est réalisé ci-dessous par la fonction `Elagage` ;

– une politique de limitation des réordonnancements, sans laquelle le nombre de développements possibles d'une hypothèse rend la recherche trop coûteuse ; cette politique fait partie des contraintes sur lesquelles s'appuie la fonction `Developpe` de l'Algorithme 5.

Formellement l'algorithme de décodage multi-pile pour des modèles de traduction à base de segments [KOE 04, QUI 07] est décrit dans l'algorithme 5.

7.5.4.1.1. Développement d'une hypothèse

La fonction `Developpe` étend une hypothèse de traduction h en la complétant par la traduction d'un bisegment $b = (f, e)$ couvrant un intervalle $[j_1 : j_2]$ non-encore

39. Improprement, puisque l'implémentation la plus naturelle de cette stratégie a recour à des files de priorité, plutôt qu'à des piles.

Algorithme 5: Recherche en meilleur d'abord

pour $j \in [1:J]$ **faire**
  $\mathcal{H}_j \leftarrow \emptyset$
fin
$\mathcal{H}_0 \leftarrow \{h_0\}$;
pour $j \in [0:J-1]$ **faire**
  pour $h \in \mathcal{H}_i$ **faire**
    pour $h' \in \texttt{Developpe}(h)$ **faire**
      $k \leftarrow |h'.span|$;
      $\texttt{AjouteRecombine}(h', \mathcal{H}_k)$;
      $\texttt{Elague}(\mathcal{H}_k)$;
    fin
  fin
fin
$\mathbf{e}_c = \{\text{argmin}_{h \in \mathcal{H}_J} h.c\}.\mathbf{e}$;
retourner *(*$\mathbf{e}_c$*)*

traduit de $\mathbf{f}$. Des contraintes particulières sur les réordonnancements possibles interviennent ici pour contraindre ce choix : par exemple en imposant une distance maximale entre le mot déjà couvert le plus à droite de $\mathbf{f}$ et j_1. L'hypothèse h' résultant correspond à un préfixe de traduction $h'.\mathbf{e} = h.\mathbf{e}.e$ et couvre $h'.span = h.span \cup [j_1 : j_2]$. Son coût actualisé se déduit de celui de h :

$$h'.c = h.c + \lambda_t c_t(f, e) + \lambda_r c_r(f, e, j_1, h) + \lambda_s(h.\mathbf{e}, e)$$

Reste à calculer le *coût anticipé* $h.g$ associé à h. Comme $h.c$, il intègre plusieurs termes [QUI 07] :

– le premier correspond au coût anticipé du modèle de traduction, qui peut se calculer en prenant le coût de la meilleure traduction possible de chaque portion non-couverte de la phrase source ;

– le second correspond à une estimation du coût du modèle de langue, et se calcule à partir des coûts syntaxiques internes des segments utilisés pour produire l'approximation précédente ;

– le troisième correspond à un coût anticipé de réordonnancement et se déduit, par exemple, de la forme particulière de la portion de la source restant à couvrir.

De la qualité de cette approximation dépend la précision de la recherche : comme pour l'algorithme A^*, il importe qu'elle sous-estime (c'est le cas de l'approximation présentée) le coût réel, de manière à ne pas éliminer de manière injustifiée des hypothèses prometteuses dont on aurait surestimé le coût futur.

7.5.4.1.2. Gestion de la file d'hypothèses

La file d'hypothèses est gérée par la fonction `AjouteRecombine`$(h', \mathcal{H})$, qui effectue deux opérations. La première consiste à rechercher s'il existe déjà dans la file une hypothèse *équivalente*, c'est-à-dire une hypothèse h'' (i) qui couvre la même partie de la source, (ii) qui partage avec h' un suffixe commun de longueur égale à l'ordre du modèle de langage. Si tel est le cas, toutes les hypothèses dérivant de h' et h'' subiront les mêmes incréments de leurs coûts : il suffit alors de conserver seulement la meilleure des deux. Si on ne trouve pas d'hypothèse équivalente, cette fonction se limite à insérer h' dans la file.

7.5.4.1.3. Élagage

L'élagage vise à éviter que les files de priorité $\mathcal{H}_j$ ne contiennent "trop" d'hypothèses, ce qui rendrait leur énumération de plus en plus coûteuse. Il existe deux politiques d'élagage principales, qui donnent lieu à deux manières légèrement différentes d'instancier la fonction `Elague` :

– l'une contrôle la *largeur du faisceau de recherche* (on parle en anglais de *beam pruning*) et consiste à ne conserver que les hypothèses dont le coût (incluant le coût anticipé) n'est pas trop éloigné du coût de la meilleure hypothèse h_b de $\mathcal{H}_|$: on supprime alors de $\mathcal{H}_|$ toutes les hypothèses h telles que $(h.c+h.g) > (1+\alpha)(h_b.c+h_b.g)$, où α est un paramètre (positif) qui détermine la largeur du faisceau,

– l'autre consiste à assigner *a priori* une taille maximale aux files $\mathcal{H}_j$: lorsque cette taille est dépassée, les hypothèses les moins prometteuses sont supprimées et ne seront pas développées. Cette stratégie est connue sous le nom de *élagage par histogramme* (*histogram pruning*). Cette seconde politique est plus grossière, mais permet de déterminer à l'avance l'empreinte mémoire nécessaire à l'exécution de l'algorithme de recherche.

7.5.4.2. *Recherche gloutone et exploration locale*

Les algorithmes de recherche locale mettent en œuvre un principe d'exploration *glouton* de l'espace de recherche. Partant d'une traduction initiale facile à obtenir, par exemple en effectuant un décodage monotone, l'idée est d'explorer le voisinage de cette solution pour y rechercher des hypothèses ayant un coût moindre. Si c'est le cas, on met à jour la meilleure traduction courante et on continue la recherche en explorant son voisinage ; sinon, la recherche s'arrête sur un optimum local, c'est-à-dire tel qu'il n'y a pas de meilleure traduction au voisinage de cet optimum. Un des intérêts de cette technique est qu'elle ne demande de manipuler que des hypothèses de traduction complètes, qui sont plus donc plus faciles à comparer et que l'on peut évaluer globalement. Cette idée très simple se formalise par l'Algorithme 6.

Cette stratégie de recherche peut s'implémenter de multiples manières, dépendant de choix particuliers de (i) la manière dont la solution initiale est construite et de (ii) la manière dont les voisinages sont définis. Le second point est particulièrement critique,

Algorithme 6: Recherche locale

```
h_c ← TraductionSimple(f) ;
stop = faux ;
tant que (stop = faux) faire
    h_v.e = {}; h_v.c = +∞ ;
    pour (h ∈ Voisins(h_c)) faire
        si (h.c < c_v) alors
            h_v ← h; c_v ← h.c ;
        fin
    fin
    si (h.c < h_v.c) alors
        h_c ← h;
    sinon
        stop = vrai ;
    fin
fin
retourner (h_c.e)
```

puisque le compromis entre qualité de l'approximation et vitesse de l'exploration dépend de la définition du voisinage. L'approche proposée dans[GER 01, GER 03] dans le cadre de modèles de traduction à base de mots, considère ainsi que le voisinage d'une traduction contient toutes les traductions obtenues en modifiant la traduction d'un ou deux mots, en insérant ou supprimant un mot, en "fusionnant" deux mots ou encore en déplaçant un groupe de mot. [LAN 07] développe une approche similaire, dans le cadre de modèles à base de segments : le voisinage d'une traduction est obtenu en considérant la possibilité de déplacer un segment en langue cible, de changer la traduction d'un ou deux segments source, ou encore de modifier (par fusion ou fission) la segmentation en langue source sous-jacente. Les auteurs de ce travail montrent que cette technique permet d'améliorer les solutions initialement produites par le système Pharaoh [KOE 04] dans près de 40% des cas. La définition restrictive des voisinages utilisée par ces auteurs, en particulier pour ce qui concerne les déplacements de segments, est naturellement liée à l'impossibilité d'inclure toutes les permutations possibles des mots qu'elle contient ; [EIS 06] montre toutefois qu'il est possible, en utilisant des techniques de programmation dynamique, d'évaluer des voisinages contenant un nombre combinatoire de permutations (toutes les permutations ITG (introduites à la section 7.4.1 page 298)) en temps polynomial.

L'intérêt principal de ces approches réside dans leur grande simplicité de mise en œuvre en particulier pour ce qui concerne le réglage du compromis entre qualité de la solution produite et temps de recherche : pour avoir de meilleures solutions, il suffit

simplement de chercher plus longtemps ou d'augmenter la taille des voisinages. Notons, pour finir, que ces techniques peuvent enfin être aisément améliorées en considérant plusieurs points de départ différents pour la recherche (*multiple restarts*), en se donnant la possibilité de refuser certaines améliorations pour éviter de descendre "trop vite" dans des minimums locaux à la manière des algorithmes de recuit simulé, en effectuant la recherche de manière non-déterministe etc.

7.5.5. *Décoder, un problème de recherche ?*

Les techniques utilisées pour combiner les différents modèles probabilistes et construire des nouvelles traductions s'appuient sur de longues années de travail sur les algorithmes de recherche heuristique menées dans les domaines de l'intelligence artificielle, de la reconnaissance des formes (en particulier en reconnaissance automatique de la parole), et plus récemment en apprentissage automatique. Ces techniques sont relativement matures, et permettent de réaliser efficacement la tâche qui leur est confiée, que l'on mesure cette efficacité en temps de traitement ou en termes d'erreurs de recherche. Ces techniques ne font donc plus l'objet de recherches très actives, sauf pour ce qui concerne (i) l'étape de réglage des paramètres (voir la section 7.5.1), qui reste extrêmement coûteuse en temps et est susceptible de fournir des résultats très aléatoires et (ii) les stratégies de recherche multi-passe et la combinaison de systèmes. On se reportera aux quelques références données ci-dessus pour se faire une idée des développements récents en la matière.

7.6. Évaluer la traduction automatique

La traduction automatique, comme toute autre technologie, doit pouvoir être évaluée[40]. Pour dépasser cette évidence, il convient de réaliser qu'il existe de multiples contextes d'utilisation de la traduction automatique, qui donnent lieu à des besoins en évaluation très différents, et de multiples points de vue sur la question. La traduction automatique n'est pas très différente, de ce point de vue, d'autres applications du traitement automatique des langues et les réflexions générales sur ces questions valent également pour ce domaine [GAL 95, KIN 96, CHA 08]. Un utilisateur pourra donc vouloir juger un système de traduction automatique selon la qualité intrinsèque de la traduction produite dans tel ou tel domaine, mais également selon le temps nécessaire pour produire des traductions, selon la facilité à le faire évoluer ou à apprendre de ses erreurs, selon la facilité avec laquelle il s'interface avec d'autres logiciels (par exemple de traitement de textes ou bien encore de navigation sur Internet), ou bien

40. Si possible, d'une manière plus scientifique que celle qui consiste à demander au service de traduction automatique de Google de traduire répétitivement des passages de la bible, puis à s'esclaffer de la piètre qualité du résultat produit [ECO 07].

encore vouloir comparer deux systèmes selon ces différents critères. Mais cette question est également primordiale pour les concepteurs / développeurs de systèmes, ainsi que pour les chercheurs en traduction automatique, car ces activités demandent en permanence de pouvoir diagnostiquer les faiblesses de leurs systèmes, de pouvoir mesurer les apports qualitatifs et quantitatifs d'une innovation, et, de plus en plus, de savoir convaincre les financeurs des progrès effectivement réalisés. Il existe donc sur la question de l'évaluation automatique des traductions une littérature abondante. Ce domaine de la recherche reste très actif, et continue de susciter de nouvelles propositions, aussi bien en ce qui concerne les évaluations subjectives que les métriques automatiques (voir par exemple [CAL 10]. Résumer ici l'ensemble de ces recherches excède donc largement notre propos et nous nous en tiendrons à une présentation très sommaire des travaux conduits sur ce thème important.

7.6.1. *Les évaluations subjectives*

On désigne sous le terme d'*évaluations subjectives* les évaluations qui impliquent le recours à des jugements humains. Les évaluations humaines de systèmes de traduction consistent à confier à des juges le soin d'analyser les traductions automatiques et de les noter selon un certain nombre de critères subjectifs. Les critères qui sont le plus typiquement utilisés sont l'intelligibilité, la fluidité, la fidélité, l'adéquation ou encore l'informativité. Concernant, par exemple, l'intelligibilité, on demandera aux juges d'évaluer sur une échelle numérique (par exemple de 1 à 5) à quel point la traduction est facile à comprendre pour le lecteur. On notera que certains critères demandent des juges bilingues, du moins qui soient capables d'évaluer à quel point le texte traduit rend *fidèlement* compte des d'informations qui sont présentes dans le texte source. À l'inverse, certains jugements, comme ceux portant sur la fluidité du texte généré, peuvent être effectués par des juges monolingues qui ne connaissent que la langue cible. Ces jugements sont plus faciles à recueillir. Sur cette base, de multiples protocoles d'évaluation ont été proposés, qui diffèrent principalement selon :

– les unités de traductions qui sont évaluées (des passages ? des phrases ? des fragments de phrases ?) ;

– les critères qui sont évalués ;

– les échelles qui sont utilisées.

Une alternative à l'utilisation d'échelles de scores fixes consiste à demander à des juges humains d'ordonner entre elles les sorties de différents systèmes de traduction automatique. Cette approche conduit donc à des obtenir des *scores relatifs* qui permettent simplement de comparer différents systèmes, ou différentes variantes d'un même système. Elle contourne donc le biais introduit par l'utilisation d'échelles de notations fixes, qui rendent souvent les jugements humains difficiles à réconcilier entre eux. C'est, par exemple, l'approche qui a été retenue par des campagnes d'évaluation récentes des systèmes de traduction [CAL 08, CAL 09b].

Une troisième approche permettant d'évaluer, cette fois de manière moins directe, la qualité d'un système de traduction consiste à concevoir des protocoles dans lesquels un humain doit accomplir une tâche avec l'aide d'un système de traduction automatique : par exemple, répondre à des questions factuelles portant sur le texte à traduire, ou bien engager une négociation avec un autre utilisateur, ou bien trouver une information, ou bien encore... traduire un document. La traduction sera alors d'autant meilleure qu'elle permettra que la tâche soit accomplie rapidement et avec succès. C'est l'approche pour laquelle plaident avec vigueur les auteurs de [BLA 07].

Le choix d'un protocole parmi ces manières très variées de réaliser des évaluations subjectives dépend au final des buts qui sont assignés à l'évaluation. Tous les protocoles partagent toutefois certains travers : ils impliquent des jugements humains, ce qui implique une certaine variabilité des résultats, des incohérences ou des désaccords entre jugements, etc ; ils sont également longs et coûteux à collecter[41], ce qui les rend inadaptés, par exemple, en phase de développement. Ces deux caractéristiques des évaluations subjectives ont suscité, par contraste, le développement de mesures automatiques de la qualité d'un système, que nous abordons plus longuement dans la section suivante.

7.6.1.1. *Les métriques automatiques*

La manière la plus simple d'envisager de mesurer automatiquement la qualité des traductions est de comparer une traduction produite automatiquement à une traduction réalisée par un humain : disposant d'un côté d'une traduction automatique, de l'autre d'une traduction dite « de référence », l'intuition commande que la première sera d'autant meilleure qu'elle ressemblera le plus à la seconde. La même démarche est au fondement des métriques qui sont utilisées dans d'autres domaines d'application du traitement des langues, comme par exemple en reconnaissance automatique de la parole. Elle présente l'avantage de ne mobiliser des experts humains qu'à un seul moment, lors de l'étape de production des traductions de référence et de pouvoir être réalisée pour un coût modique[42].

De nouveau par analogie avec la reconnaissance vocale, une première mesure simpliste de la qualité d'une traduction est ainsi donnée par la métrique WER (pour *Word*

41. Ce point est à nuancer : le déploiement de systèmes de traduction en ligne a suscité une activité de collecte à bas coût de jugements subjectifs des traductions, prenant la forme de notes que les internautes peuvent attribuer aux traductions réalisées. Voir également la proposition récente de [CAL 09a], qui propose de tirer partie de la facilité avec laquelle on peut aujourd'hui accéder à des services en ligne sur Internet.

42. En particulier, l'effort demandé ne dépend pas du nombre de systèmes à comparer, ce qui est un élément très important lorsqu'il s'agit de comparer un grand nombre de systèmes entre eux.

Error Rate), obtenue en calculant la distance d'édition[43] entre une hypothèse de traduction et une référence.

La mise en œuvre de cette approche se heurte toutefois à plusieurs obstacles. Elle ne permet pas de prendre en compte des changements d'ordre entre les deux énoncés. Plus grave, il est possible qu'une traduction très éloignée de la référence soit également correcte ou qu'à l'inverse, une traduction très similaire contienne des contresens majeurs : qu'on pense aux conséquences possibles de l'oubli d'une négation. Dit autrement, cette mesure est linguistiquement très naïve, car elle présuppose que des ressemblances de surface impliquent des ressemblances de signification. Elle ne prend pas non plus en compte le fait que les mêmes significations peuvent être exprimées de manières extrêmement variées et diverses.

En dépit de ces obstacles, les métriques automatiques se sont progressivement installées dans le paysage et constituent indispensable pour orienter au quotidien le développement de systèmes. La proposition la plus populaire est le score $BLEU_k$ et ses multiples variantes, qui a été proposé originellement par Papineni et ses co-auteurs dans [PAP 01].

7.6.2. *La métrique* $BLEU_k$

Comme la métrique WER, la métrique $BLEU_k$ (pour *Billingual Evaluation Understudy*) se calcule par comparaison entre une hypothèse et une ou plusieurs références. Formellement, soient $\mathbf{e}$ une hypothèse de traduction et $\{\mathbf{e}_1,...\mathbf{e}_l\}$ un ensemble de l traductions de référence d'une phrase source $\mathbf{f}$, on note $c_n(\mathbf{e})$ le nombre de n-grammes apparaissant dans $\mathbf{e}$ et $m_n(\mathbf{e})$ le nombre de n-grammes de $\mathbf{e}$ qui apparaissent également dans *au moins une* des références. La *précision n-gramme* $p_n(\mathbf{e})$ est simplement le rapport entre ces deux grandeurs[44]. Le score $BLEU_k$ est alors essentiellement défini comme la moyenne *arithmétique* des précisions n-gramme, pour n variant entre 1 et k, à deux détails près :

– si, pour une valeur de n, $c_n(\mathbf{e})$ est nul, alors la moyenne arithmétique $\sqrt[k]{\prod_{n=1}^{k} p_n(\mathbf{e})}$ sera nulle, et l'hypothèse $\mathbf{e}$ aura un score $BLEU_k$ égal à zéro, quelles que soient les valeurs des autres précisions n-grammes. Calculer le score $BLEU_k$ d'une phrase isolée n'a donc aucun sens. Pour éviter de telles situations, la précision n-gramme est calculée sur de grands ensembles (quelques centaines, voire quelques

43. La distance d'édition entre deux séquences mesure le nombre minimal d'opérations d'édition pour passer de l'une à l'autre. Les opérations d'édition correspondent à l'insertion, à la suppression d'un élément d'une séquence, ou à une substitution entre deux éléments [WAG 74].

44. Enfin, presque : p_n est en fait la précision n-gram *modifiée*, qui se calcule comme $\max(1, \frac{c_n}{m_n})$, afin ne pas récompenser (à tort) un système qui produirait *plusieurs fois* un mot ou un n-gram correct.

milliers) d'hypothèses $\mathbf{E} = \{\mathbf{e}^{(1)} \dots \mathbf{e}^{N}\}$ par :

$$p_n(\mathbf{E}) = \frac{\sum_{i=1}^{N} m_n(\mathbf{e}^{(i)})}{\sum_{i=1}^{N} c_n(\mathbf{e}^{(i)})}$$

– il existe un moyen simple d'obtenir de bonnes précisions n-grammes, consistant à produire des hypothèses très courtes : en produisant moins de mots, on a moins de chance de se tromper. Il est donc nécessaire de corriger les valeurs de la précision moyenne en intégrant un facteur (BP pour *brevity penalty* dans la formule (7.26)) qui pénalise les hypothèses qui seraient trop courtes par rapport aux références. Ce facteur est également une moyenne établie sur l'ensemble des hypothèses proposées par le système[45] de termes $BP(\mathbf{e})$; chacun de ces termes est défini par ($\mathbf{e}_i$ désigne la meilleure référence pour $\mathbf{e}$) :

$$BP(\mathbf{e}) = \begin{cases} 1 \text{ si } |\mathbf{e}| > |\mathbf{e}_i| \\ \exp\left(1 - \frac{|\mathbf{e}_i|}{|\mathbf{e}|}\right) \text{ sinon} \end{cases}$$

Au final, la mesure $BLEU_k$ de l'ensemble d'hypothèses $\mathbf{E}$ est donc définie par[46] :

$$BLEU_k(\mathbf{E}) = BP \times \exp\left(\sum_{n=1}^{k} \frac{1}{k} \log(p_n(\mathbf{E}))\right) \tag{7.26}$$

Par construction, le score $BLEU_k$ est donc compris entre 0 et 1, cette dernière valeur étant atteinte lorsque l'hypothèse est égale à une des traductions de référence. Un exemple de calcul du score $BLEU_k$ est détaillé dans la figure 7.16.

Ainsi, la famille de mesures $BLEU_k$ introduit deux innovations qui pallient, dans une certaine mesure, les problèmes évoqués plus haut :

– il devient possible d'utiliser plusieurs références dans le calcul du score ;

– la mesure de similarité est moins rigide qu'une métrique basée sur les distances d'édition entre séquences de mots et autorise des variations dans l'ordre des mots, mais plus précise qu'une métrique qui ne prendrait en compte que la similarité entre des mots isolés.

Cette mesure présente, de surcroît, le mérite d'être extrêmement simple à calculer, puisqu'elle n'implique que des comparaisons triviales entre séquences de mots.

---

45. Il existe plusieurs manières légèrement différentes de calculer ce facteur selon que l'on utilise la longueur de la référence la plus longue, la plus courte, celle dont la longueur est la plus proche de l'hypothèse, etc.

46. Il existe des variantes qui pondèrent différemment chacune des précisions n-grammes, mais dans la pratique, on leur attribue le plus souvent un poids identique.

$\mathbf{f}$	'*I think the suggestion is worth looking into .*'
$\mathbf{e}_1$	'*je pense que la suggestion vaut la peine que l' on s' y intéresse .*'
$\mathbf{e}_2$	'*à mon avis , la proposition est digne d' être prise en considération .*'
$\mathbf{e}$	'*je pense que la proposition est la peine de réfléchir .*'
$c_1(\mathbf{e})$	9 : '*je*', '*pense*', '*que*', '*la*', '*proposition*', '*est*', '*la*', '*peine*', '*.*'
$c_2(\mathbf{e})$	6 : '*je pense*', '*pense que*', '*que la*', '*la proposition*', '*proposition est*', '*la peine*'
$c_3(\mathbf{e})$	3 : '*je pense que*', '*pense que la*', '*la proposition est*'
$c_4(\mathbf{e})$	1 : '*je pense que la*'
BP	$\exp(1 - 14/11) \approx 0.76$
$BLEU_4(\mathbf{e})$	$BP \times \exp\left(0.25 \times (\log(9/11) + \log(6/10) + \log(3/9) + \log(1/8))\right)$ ≈ 0.29

Figure 7.16 – Détail du calcul du score $BLEU$

Les valeurs numériques produites par cette métrique et ses avatars restent toutefois difficiles à interpréter et constituent des mesures très grossières de la qualité d'un système de traduction. Leur adoption par une large partie de la communauté résulte de l'observation répétée de corrélations statistiques significatives entre le score $BLEU$ au niveau d'un corpus et le jugement humain. Il est ainsi plus ou moins accepté que (i) toute amélioration de $BLEU_k$ au-delà d'une certaine valeur se traduirait par une amélioration des jugements humains et que (ii) pour un corpus donné, le système ayant le meilleur score $BLEU_k$ serait également le meilleur du point de vue des jugements humains. Les dernières évaluations internationales conduites dans le cadre des campagnes WMT[47] [CAL 08, CAL 09b] et NIST[48] laissent penser que ces deux affirmations restent relativement fondées, du moins tant que l'on considère plusieurs références et que l'on utilise des corpus d'évaluation suffisamment grands.

7.6.3. *Les alternatives à* $BLEU_k$

Les critiques du score $BLEU_k$ sont toutefois nombreuses : en effet, cette métrique ne permet pas d'évaluer la qualité de phrases isolées ; elle ne se fonde que sur des statistiques calculées phrase par phrase et ne peut donc pas évaluer la qualité de la traduction d'un texte ; elle ne s'appuie que sur des ressemblances de surface entre chaînes de caractères ; elle est insensible au contenu informationnel des mots correctement traduits ; elle favorise les systèmes de traduction statistiques au détriment de systèmes utilisant d'autres techniques, en particulier des systèmes à base de règles, etc (voir, par exemple, [CAL 06]). De nouvelles propositions de métriques continuent

47. http://www.statmt.org/wmt07, http://www.statmt.org/wmt08, etc.
48. http://www.itl.nist.gov/iad/mig/tests/mt/

donc d'émerger, qui fournissent souvent de meilleures prédictions des jugements humains (voir de nouveau [CAL 09b]). Ces métriques s'appuient en particulier sur les progrès en traitement automatique des langues et sur des ressources linguistiques plus riches, qui permettent de mettre en œuvre des mesures de similarité plus fines. Ainsi, METEOR [BAN 05] utilise des outils de lemmatisation et des ressources sémantiques permettant de détecter les cas de synonymie. Cette manière de procéder demande toutefois de disposer d'outils robustes, c'est-à-dire qui soient à même de traiter les hypothèses de traduction produites automatiquement, qui sont parfois très éloignées de phrases grammaticalement correctes.

Une autre proposition de métrique intéressante est la famille TER (pour *Translation Edit Rate*) [SNO 06]. HTER (pour *Human TER*) mesure la qualité d'une traduction par l'effort qui serait nécessaire à un expert pour la transformer en une traduction correcte. HTER mesure donc le coût d'une post-édition, ce qui conduit à une métrique plus facile à interpréter que $BLEU_k$, qui plus est capable de produire des scores pour des phrases isolées. Pour autant que l'on dispose d'une traduction de référence, cette post-édition peut-être effectuée par des correcteurs monolingues, soit à coût moindre que la production de plusieurs références. Enfin, en théorie, cette proposition a le mérite de remédier à un autre biais de $BLEU_k$ et toutes les métriques apparentées. En effet la comparaison entre une hypothèse et des références n'a de sens que si le système est capable de prédire ces références : pour cela, il faut au moins que son vocabulaire contienne tous les mots des références. À défaut, la métrique fonctionne comme un enseignant qui sanctionnerait un élève pour ne pas savoir une leçon qui n'a pas encore été enseignée. À l'inverse, la métrique TER mesure l'écart à des traductions correctes, qui sont par construction proches de ce que les systèmes savent réellement produire. Dans la pratique, pour éviter d'avoir à effectuer cette étape de post-édition, on s'en remet le plus souvent à la métrique TER, qui remplace les références post-éditées par des références construites a priori.

Formellement, le TER se calcule comme le nombre minimal d'opération d'édition nécessaires pour transformer une sortie automatique en la plus proche des traductions de référence ; les opérations autorisées sont les opérations "classiques" des distances d'édition (substitution, suppression, insertion de mots), auxquelles s'ajoute une nouvelle opération consistant à déplacer un mot ou un groupe de mot. En notant $\Delta(\mathbf{e}, \mathbf{e}_i)$ le nombre minimal d'opérations pour transformer $\mathbf{e}$ en $\mathbf{e}_i$, la mesure TER de l'hypothèse $\mathbf{e}$ s'exprime par :

$$TER(\mathbf{e}) = \min_{\mathbf{e}_1 \ldots \mathbf{e}_l} \frac{\Delta(\mathbf{e}, \mathbf{e}_i)}{|\mathbf{e}_i|} \tag{7.27}$$

Le calcul du TER est illustré en figure 7.17.

Référence : '*saudi arabia denied this week information published in the american new york times*'
Hypothèse : '*this week the saudis denied information published in the new york times*'
$\Delta(\mathbf{e}, \mathbf{e}') = 4$: déplacer('*this week*'), substituer('*saudi*', '*the*'), substituer('*arabia*','*saudis*'), supprimer('*american*')

Figure 7.17 – Détail du calcul du score TER (d'après [SNO 06])

Le calcul de cette mesure présente toutefois une difficulté majeure puisque l'évaluation des termes $\Delta(\mathbf{e}, \mathbf{e}_i)$ implique la résolution d'un problème d'optimisation NP-difficile[49] : calculer le TER est donc algorithmiquement bien plus coûteux que le score $BLEU_e$t seuls des outils de calcul approchés existent[50]. Il n'empêche que cette métrique continue d'être très utilisée et que des variantes existent, qui, notamment (TERp [SNO 09]) enrichissent la définition des coûts des diverses opérations d'édition en intégrant des connaissances linguistiques : ainsi, on diminuera le coût de substitution de deux synomymes ou de deux mots morphologiquement apparentés.

7.6.4. *Évaluer : un problème non résolu*

En résumé, la mise en place de mesures d'évaluation de la qualité des traductions repose le plus souvent sur la disponibilité d'un ensemble de traductions de référence avec lesquelles les hypothèses de traduction seront comparées. Par principe, ces métriques s'avèrent inadaptées pour mesurer la qualité globale de la traduction de documents. De surcroît, les mesures de comparaison utilisées sont le plus souvent grossières : concevoir des mesures de comparaison entre phrases qui rendent compte de leur proximité sémantique, tout en étant rapides à calculer et robustes à des entrées très agrammaticales constitue un défi qui, en dépit de recherches actives, reste ouvert. L'activité de développement de systèmes de traduction automatique dépendant crucialement de ces mesures, elles restent, faute de mieux, très utilisées. En parallèle, le recours à des protocoles d'évaluation qui impliquent des jugements humains, tout en se dispensant de la production de traductions de référence, fournit une alternative de plus en plus crédible pour comparer des systèmes et est utilisé en pratique dans les principales campagnes d'évaluation.

49. La difficulté provient naturellement de l'opération de déplacement de blocs, puisque, lorsque cette opération est interdite, le calcul de la distance d'édition entre deux séquences se résoud en temps et en espace de manière polynomiale par programmation dynamique.

50. http://http://www.cs.umd.edu/~snover/tercom/.

7.7. L'état de l'art et ses développements

Dans cette section, nous évoquons rapidement quelques-uns des développements les plus récents en traduction statistique. Nous avons choisi ici de discuter d'un petit nombre de travaux de recherche qui mettent clairement en évidence certaines des limitations des modèles de traduction à base de segments. Ils mentionnent diverses manières de remédier à ces limitations, en espérant que les quelques références données ci-dessous pourront utilement servir de point d'entrée à la très riche littérature portant sur les variantes et extensions du modèle standard.

7.7.1. *Utilisation du contexte en source*

Les modèles de segments, comme les modèles de mots introduits en section 7.2.2, ne prennent pas en compte le voisinage d'une phrase en source pour choisir sa traduction. En effet, les scores associés aux segments sont calculés une fois pour toutes lors de l'apprentissage (voir la section 7.3.3) et ne *varient pas en fonction des phrases à traduire*. Lorsqu'un segment source possède plusieurs traductions possibles, elles seront départagées principalement par le modèle de langage en cible (cf. la discussion de la section 7.1). Intuitivement, cette situation n'est pas très satisfaisante : si, au voisinage d'un mot tel que '*voler*', se trouvent des formes comme '*avion*', '*voyage*', ou encore '*aéroport*', alors la traduction de ce mot par '*fly*' devrait s'en trouver renforcée et la traduction '*steal*' devrait devenir moins probable. Effectuer de telles inférences ne pose aucune difficulté, puisqu'au moment de la traduction, la phrase source est intégralement connue, comme le sont également les phrases voisines. L'utilisation de la connaissance des voisins d'un segment pour améliorer sa traduction est un principe bien connu en traduction par l'exemple et a tout naturellement été transposé de multiples façons dans le cadre des modèles probabilistes. Dans la suite de la discussion, nous distinguerons entre le contexte local ou *micro-contexte* d'un segment, qui se réduit le plus souvent aux quelques mots voisins, au mieux à la phrase ; et le contexte étendu ou *macro-contexte*, qui comprend les autres phrases du document à traduire, voire d'autres documents similaires.

7.7.1.1. *Exploiter le micro-contexte*

Le travail de [CAR 05] propose d'aborder ce problème en s'inspirant de techniques de *désambiguïsation lexicale*[51] : les expériences préliminaires décrites dans ce travail montrent toutefois que désambiguïser le sens de quelques mots isolés avant de les traduire n'apporte aucune amélioration significative. Dans [CAR 07], ce travail est étendu à la désambiguïsation de tous les segments : pour chaque phrase à traduire, des

51. La tâche de désambiguïsation lexicale est entendue en traitement automatique des langues comme la construction d'un appariemment entre une occurrence d'un lemme dans son contexte et un sens du lemme, choisi dans une liste finie de sens possibles.

techniques de désambiguïsation lexicale (ici, une combinaison de classifieurs prenant en compte des propriétés du micro-contexte) sont utilisées pour évaluer en contexte les segments d'une table de traduction, qui est ainsi adaptée à la phrase source courante. Cette proposition n'est pas très éloignée de celle de [ITT 07], qui propose de définir la probabilité d'un bisegment $(f : \mathbf{e})$[52] avec un modèle exponentiel en écrivant $\mathrm{P}(\mathbf{e}|f) \propto \exp \theta^T F(\mathbf{f}, \mathbf{e})$. Utiliser dans le modèle des fonctions caractéristiques qui examinent les mots voisins en source de f permet de faire dépendre $\mathrm{P}(\mathbf{e}|f)$ de son contexte, une extension qu'étudie également [MAU 09] dans un cadre plus orthodoxe . La même idée est poursuivie, avec des techniques différentes, dans [STR 07, GIM 08] etc. Ainsi, l'approche proposée par [STR 07] consiste à ajouter de nouvelles caractéristiques dynamiques dans la table des segments. Ainsi, si $c(\mathbf{f})$ dénote le contexte (au sens large) en source de la phrase $\mathbf{f}$, on ajoutera une nouvelle valuation au bisegments(correspondant par exemple à $\log(\mathrm{P}(\mathbf{e}|\mathbf{f}, c(\mathbf{f}))$. Pour contourner les difficultés liées à l'estimation de tels modèles, le travail pré-cité utilise des techniques d'apprentissage à base d'exemples ; celui de [GIM 08] poursuit cette même idée avec d'autres techniques et étudie en particulier le bénéfice qu'apporte l'ajout de caractéristiques testant des configurations syntaxiques. Le travail de [BAN 07] pousse cette logique à son terme et étudie le comportement d'un système fondé sur un modèle de traduction encore plus riche, mais qui n'intègre aucun modèle de réordonnancement, sinon celui qui est opéré par le truchement du modèle de langage (voir, sur ce sujet, la discussion de la section 7.4.2). L'idée, conceptuellement très simple, consiste à apprendre pour chaque mot e du vocabulaire en langue cible un classifieur binaire décidant de la présence ou de l'absence de ce mot dans une traduction d'une phrase source $\mathbf{f}$. Dans un cadre probabiliste, les auteurs proposent d'apprendre des modèles de régression logistique pour définir $\mathrm{P}(e|\mathbf{f})$, *sans chercher à localiser le ou les mots source qui sont traduits par* e. C'est une différence majeure avec les modèles précédents, dans lesquels on s'intéresse à la traduction d'un mot ou d'un segment source particulier et dans lesquels le contexte se réduit aux quelques mots voisins de ce segment. Ici le contexte pris en compte comprend l'intégralité de la phrase source, ce qui implique que *l'entraînement de tels classifieurs ne demande même pas d'alignements mot à mot*. Pour traduire une nouvelle phrase $\mathbf{f}$, on doit d'abord décider, pour chaque mot cible possible, s'il est dans la traduction : les auteurs proposent de retenir tous les mots e tels que $\mathrm{P}(e|\mathbf{f})$ dépasse un seuil. Il suffira ensuite de laisser un modèle de langage cible choisir la meilleure permutation de l'ensemble des mots retenus. Cette approche pêche par de nombreux aspects (elle reste à base de mots, elle n'utilise pas de modèles de réordonnancements) et ne se compare pas favorablement, en termes de performances, avec les approches classiques. Les résultats obtenus sont néanmoins intéressants et montrent une amélioration des choix lexicaux par rapport aux modèles de segments.

52. Dans ce modèle, les segments source ne contiennent qu'un seul mot

La conclusion provisoire de ces travaux est que la prise en compte du micro-contexte pour évaluer dynamiquement les segments apporte quasi-systématiquement des améliorations des performances, mais que celles-ci restent relativement modestes au regard du surcoût computationnel lié à l'entraînement et à l'utilisation de ces modèles.

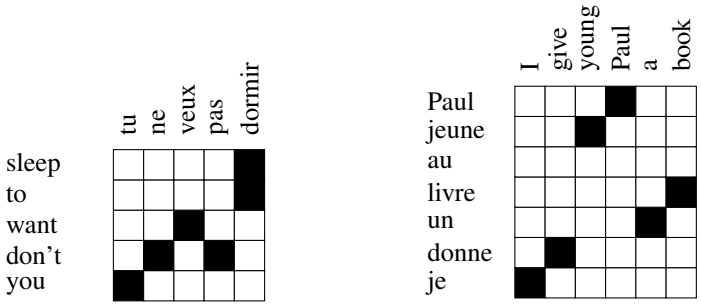
7.7.1.2. *Macro-contexte*

La prise en compte du macro-contexte a donné lieu comparativement à moins de travaux, ne serait-ce que parce que, comme discuté plus haut, les protocoles (et les corpus) de tests les plus couramment utilisés ne considèrent que la traduction de phrases isolées. Plusieurs auteurs ont toutefois cherché à prendre en compte, dans les modèles de traduction comme dans les modèles de langue, des informations portant sur l'ensemble du document à traduire, voire sur des documents voisins. Le problème ici s'énonce le plus souvent comme un problème *d'adaptation* d'un modèle de traduction généraliste à un genre, un registre, ou à une thématique particulière, problème qui rejoint la question bien connue de l'adaptation de modèles statistiques de langage en reconnaissance automatique de la parole. On dispose pour ce faire d'une batterie de techniques (voir par exemple [DEM 98, BEL 01]) qui mélangent fouille de texte classique et modélisation du langage (construction dynamique d'un corpus d'apprentissage, utilisation de techniques de classification non-supervisées, de modèles de mélange, prise en compte de termes d'*a priori* dans les modèles exponentiels, etc). Ces techniques sont en partie directement utilisables pour adapter les modèles statistiques de langage ; concernant le modèle de traduction, il a fallu les ... adapter au cadre de la traduction automatique, qui présente à la fois des aspects simplificateurs (par exemple, on dispose de l'intégralité du document source à traduire) et d'autres qui rendent l'adaptation plus compliquée (les collections à modéliser sont parallèles). On se reportera, par exemple, à [FOS 07, BER 09] et aux références citées dans ces articles pour une présentation plus complète de ce thème de recherche.

7.7.2. *Les modèles hiérarchiques*

Les modèles à base de segments fondent leurs décisions sur des analyses statistiques de gros corpus de textes parallèles. La distribution remarquable des occurrences de mots dans les textes, aggravée par l'imprécision des algorithmes d'alignement, fait que les fréquences d'occurrences de l'immense majorité des bisegments sont très faibles, sans parler de toutes celles que l'on n'observe pas dans les corpus d'entraînement. Une conséquence est la relativement mauvaise capacité de généralisation de ces systèmes, qui demandent de très gros volumes de données pour produire des résultats raisonnables. L'introduction de segments dits « hiérarchiques » par [CHI 05] est une tentative de remédier à cette limitation en autorisant les segments à comporter

des “trous”, une idée déjà en germe dans [SIM 05]. L'idée principale de cette généralisation des modèles à base de segments consiste à extraire des matrices d'alignement (voir la section 7.3.3.2) les règles d'une grammaire hors-contexte synchrone (ces grammaires ont été présentées à la section 7.4.1).



(a) (b)

Figure 7.18 – Extraction de segments hiérarchiques

Considérons, à titre d'exemple, la matrice reproduite en figure 7.18.(a) : comme l'anglais '*don't*' s'aligne avec les deux mots français '*ne*' et '*pas*' qui entourent '*veux*', l'algorithme d'extraction de segments présenté à la section 7.3.3.2 n'identifie que les deux ('*veux*','*want*') et ('*ne veux pas*','*don't want*'). Par construction, la “bonne” généralisation linguistique, consistant à repérer que la négation en français se forme en encapsulant un noyau verbal entre les deux adverbes de négations, est manquée. La conséquence est que, même si l'on a appris à traduire correctement une forme verbale, on ne saura pas nécessairement le traduire dans un contexte de négation.

L'avancée proposée dans [CHI 05] et qui a ensuite été reprise et affinée dans de nombreux travaux, en particulier [ZOL 06, HUA 07, LOP 08, IGL 09], consiste à extraire en lieu des phrases des règles d'une grammaire hors contexte synchrone (voir la section 7.4.1). Dans le cas de l'exemple précédent, on extraira ainsi les deux productions $X \rightarrow$ veux ; want et $X \rightarrow$ ne X pas ; don't X : dans la seconde de ces règles, le sous-bloc ('*veux*','*want*') est abstrait et remplacé par une variable. De même dans l'exemple de droite, l'extraction de segments hiérarchiques permet d'extraire une production telle que $X \rightarrow$ donne X_1 à X_2 ; give $X_2 X_1$, qui rend compte de l'inversion de l'ordre des compléments entre le français '*donne*' et la version ditransitive de l'anglais '*give*'. Ce second exemple met en évidence l'intérêt des règles hiérarchiques pour modéliser les déplacements : la règle précédente pourra être appliquée pour permuter les compléments de toute occurrence de '*donne*', et ce quel que soit le nombre de mots qu'ils comportent.

Cette approche pose plusieurs problèmes difficiles : d'une part cette procédure extrait beaucoup plus de règles que l'algorithme décrit à la section 7.3.3.1 n'extrait de bisegments, puisqu'il faut *a priori* considérer toutes les façons possibles d'abstraire un ou plusieurs sous-blocs. Cela signifie en particulier qu'à partir de l'exemple précédent, on extrait de nombreuses règles synchrones qui sont soit linguistiquement absurdes, soit sans aucune capacité de généralisation. C'est le cas par exemple de la règle synchrone suivante dans laquelle le déterminant est généralisé :

$$X \rightarrow X \text{ livre au jeune Paul ; young Paul } X \text{ book.}$$

Il devient nécessaire de mettre en place des stratégies pour contrôler le nombre de règles extraites, pour filtrer les moins pertinentes, pour enfin associer des probabilités à ces différentes règles. D'autre part, l'utilisation d'une grammaire synchrone en traduction conduit à remplacer les algorithmes de recherche présentés supra par des algorithmes qui sont très semblables à des algorithmes d'analyse de grammaires hors-contexte, dont la complexité est moins bonne[53], et qui surtout construisent la phrase cible de façon ascendante (et non plus de gauche vers la droite). Ceci complique fortement l'interaction avec le modèle de langage pendant l'étape de décodage et demande donc de développer des stratégies de recherche plus élaborées.

7.7.3. *Traduire avec des ressources linguistiques*

Une limitation des modèles à base de segments découle de ce qui constitue une de leur principale force, à savoir leur capacité à s'appliquer pratiquement à n'importe quelle paire de langages source et cible, ce qui a pour corrélat immédiat leur extrême naïveté linguistique. Les unités modélisées correspondent à formes graphiques, c'est-à-dire à des chaînes de caractères non-analysées ; les segments sont des regroupements « plats » et n'ont, contrairement à ce que la terminologie (anglaise) suggère, aucune pertinence syntaxique. De ce point de vue, ces modèles constituent une claire régression par rapport aux systèmes de traduction à base de règles, dont les règles de transfert et de génération s'appuient (au moins) sur une analyse lexicale, morphologique et une analyse de la structure syntaxique des phrases source et cible. Une conséquence directe de cette limitation est la propension, maintes fois révélée, des systèmes de traduction statistiques à produire des contre-sens, des traductions erronées ainsi que des phrases cible très peu grammaticales [VIL 06]. Tous les aspects de la construction d'un système de traduction souffrent de cette limitation et les approches visant à intégrer plus de connaissances linguistiques dans les modèles de traduction ont, dans la pratique, concerné toutes les étapes de modélisation présentées dans ce chapitre.

---

53. Le plus simple de ces algorithmes, CYK [YOU 67], a une complexité cubique en la longueur de la phrase source. Pour pouvoir travailler avec des grammaires non normalisées, on utilise le plus souvent la généralisation proposée dans [CHA 98].

Les efforts les plus significatifs portent, en particulier, sur l'utilisation de connaissances morphologiques et syntaxiques aussi bien en source qu'en cible, même si le premier de ces deux axes de recherche a un peu pâti de la relative simplicité morphologique des langues les plus souvent utilisées en source (chinois, arabe) et en cible (anglais). À l'inverse, les travaux visant à introduire plus de syntaxe ont largement bénéficié de la diffusion d'outils d'analyse syntaxique robustes (analyseurs en constituants et en dépendances), qui sont maintenant disponibles pour de multiples langues.

7.7.3.1. *Dictionnaires et terminologies bilingues*

L'idée la plus naturelle pour introduire des connaissances linguistiques consiste à exploiter des dictionnaires et des terminologies bilingues, qui contiennent des appariemments valides entre mots et termes en source et en cible. D'un point de vue technique, l'utilisation de telles ressources est assez immédiate : pour l'alignement, il suffira d'utiliser ces ressources pour améliorer les estimations des différents modèles, avec un effet très positif sur les paramètres associés aux mots rares (voir par exemple [BRO 93a, NIE 04]). Pour la construction de modèles de traduction, il suffit de rajouter les entrées du dictionnaire à la table des segments, en complétant le vecteur de scores associé à un bisegment de la façon suivante : pour chaque mot ou terme présent dans le dictionnaire on rajoute au vecteur de scores usuel décrit à la section 7.3.3.3 une nouvelle composante valant 1 ; si un appariemment n'apparaît que dans le dictionnaire, les statistiques associées seront mises à 0. Le bénéfice attendu de cette stratégie est naturellement de diminuer le nombre de mots qui sont inconnus du système et qui sont le plus souvent copiés *verbatim* dans l'énoncé cible. Notons toutefois que cette stratégie n'est pas toujours gagnante, dans la mesure où elle peut conduire à augmenter l'ambiguïté du système, en injectant dans la table de bisegments des traductions qui n'existent pas dans le corpus d'apprentissage et qui sont peut-être inutiles pour le genre ou le registre des textes à traduire. On trouvera, par exemple dans [SCH 08], une description de cette approche et des améliorations des performances qu'elle permet d'obtenir.

7.7.3.2. *Analyse morphologique en traduction automatique statistique*

L'utilisation de formes graphiques non-analysées dans les modèles de traduction statistique pose trois types de problèmes relativement distincts. Un premier problème provient du fait que des formes morphologiquement apparentées, par exemples des formes conjuguées d'un même verbe, ou plus généralement des formes fléchies d'un même lemme, sont vues par les modèles comme des entités complètement différentes. Une conséquence est que les informations statistiques recueillies sur une forme d'une famille morphologique ne peuvent se transmettre aux autres mots de la même famille. Savoir que '*cat*' est une traduction de '*chat*' n'aide en rien à traduire '*cats*' en '*chats*'. Cette situation est particulièrement dommageable pour les langues qui ont des systèmes flexionnels riches (les langues latines, l'allemand, le russe, etc) : la probabilité que le corpus d'apprentissage couvre (avec assez de représentants) toutes les formes d'un lemme diminue avec le nombre de formes possibles : les formes inconnues sont

plus nombreuses ; les probabilités (pour les différents modèles) des formes connues sont moins bien estimées. [NIE 04] propose de remédier à ce problème en rendant explicites les relations entre formes, ce qui permet de généraliser le modèle de traduction ; les auteurs proposent de représenter une forme par un vecteur de propriétés contenant par exemple, en plus de la forme elle-même, le lemme, et ses traits morphologiques. Le même type de représentation est sous-jacente aux modèles "factorisés" [KOE 07a] : dans cette approche, on apprend un système complet de traduction statistique pour *chacune des composantes du vecteur de propriétés*. Lors de la traduction d'une phrase inconnue, la traduction procède en trois étapes : construction des vecteurs de propriétés, traduction de chacune des composantes, puis *génération* de la phrase cible à partir des vecteurs de propriétés.

Un second problème découle de la divergence entre les systèmes morphologiques en cible et en source, en particulier des divergences portant sur la notion de mot. Ces situations sont particulièrement problématiques lors de l'étape d'alignement, qui rappelons-le, implique de construire des correspondances dans les deux directions avant de symétriser l'alignement (cf. la section 7.3.1.4), et qui donc, présuppose implicitement que les phrases source et cible sont de même longueur. Pour se ramener à cette situation, deux options sont possibles qui dépendent de la direction de traduction. On pourra soit utiliser des techniques pour segmenter les formes complexes en unités plus petites, ce qui, également, aura un effet positif sur les capacités de généralisation des modèles : voir, par exemple, [KOE 03a] pour un travail sur la segmentation de l'allemand, [OFL 07] pour une étude de différentes stratégies de segmentation pour la traduction entre l'anglais et le turc ; ou encore [CHU 09] pour des travaux récents visant à apprendre à segmenter des formes complexes dans un contexte d'alignement automatique. L'autre option est naturellement de construire des regroupements de mots dans la langue dans laquelle les phrases sont les plus longues, voir par exemple [NIE 04].

Un troisième problème découle du fait que les informations morphologiquement marquées par différentes langues varient en granularité (cf. la section 7.2.2). Cette situation complique la traduction depuis les langues qui marquent peu de distinctions (comme l'anglais), vers celles qui en marquent beaucoup (comme le français), puisque de chaque forme source peut potentiellement se projeter dans de multiples formes cibles, impliquant de mettre en œuvre de nouvelles stratégies de désambiguïsation : qu'on pense à une forme anglaise comme '*think*', qui se traduira en français '*penser*', '*pense*', '*penses*', '*pensons*', etc. Ce problème est discuté, par exemple, dans [UEF 03] qui, pour traduire de l'anglais en espagnol, construit côté source des complexes verbaux intégrant certains pronoms clitiques (le sujet, les pronoms réflexifs), modaux et auxiliaires, afin de faciliter alignement et traduction. Il est également possible d'envisager cette désambiguïsation sous la forme d'un post-traitement en langue cible, comme le proposent [MIN 07] ou encore [GIS 08].

7.7.3.3. *Modélisation des congruences syntaxiques*

Les travaux qui portent sur l'utilisation de connaissances syntaxiques dans les systèmes de traduction partagent une hypothèse commune : celle de l'existence d'un isomorphisme entre les structures syntaxiques en langue source et cible. Sous cette hypothèse, on s'attend à ce que les appariements mot-à-mot soient cohérents avec les structures syntaxiques, permettant d'inférer des appariements entre constituants, ce qui implique, en particulier, que les syntagmes source ont tendance à être "déplacés" en bloc dans la cible. On s'attend également à trouver dans la phrase cible globalement les mêmes relations de dépendance que dans la phrase source ; plus encore, on peut s'attendre que si a est un dépendant de b en source, alors l'équivalent de traduction de a soit dépendant de l'équivalent de traduction de b. Par exemple, si a est sujet de b en source, alors le syntagme apparié en cible avec a doit être sujet du verbe qui traduit b dans la phrase cible. Cette hypothèse, dont la validité est explorée dans [FOX 02, HWA 02], peut être exploitée à différents moments de la construction d'un système de traduction statistique. Avec l'avènement d'une génération d'outils robustes et performants pour produire des analyses morpho-syntaxiques et syntaxiques, disponibles aujourd'hui pour de nombreuses langues, il est aujourd'hui possible de tester la validité de ces hypothèses et de construire des modèles de traduction plus informés linguistiquement que les modèles à base de segments.

Par exemple, plusieurs auteurs ont proposé d'améliorer des alignements par prise en compte des étiquettes morpho-syntaxiques [TOU 02, NEY 04], ou des informations grammaticales telles que des frontières de syntagmes ou des dépendances syntaxiques (par exemple [LIN 03, ZHA 04, CHE 06, MA 08]). L'hypothèse d'un isomorphisme syntaxique entre phrases source et cible semble toutefois contraindre trop fortement les alignements et a pour l'instant conduit à des gains de performances mitigés (voir en particulier l'analyse de [WEL 06]). Les travaux décrits dans [KOE 03b, CHI 08] s'intéressent à améliorer une autre étape de la fabrication du modèle de traduction, à savoir l'extraction et l'évaluation des segments.

Il est également naturel de vouloir contraindre les modèles de réordonnancement en intégrant des informations syntaxiques, et la littérature sur ce thème est donc abondante. On retiendra en particulier les travaux décrits dans [COL 05], qui montrent comment, en utilisant quelques transformations linguistiquement motivées, il est possible de fortement réduire les divergences entre les structures syntaxiques de phrases allemandes et anglaises, et donc de simplifier la tâche de l'alignement et de la traduction. Il est également possible d'apprendre des règles de réordonnancement en observant des alignements de structures syntaxiques en source et ou en cible (voir par exemple [XIA 04, CRE 08]). Cette idée rejoint celle proposée dans [CHE 08] d'utiliser des contraintes syntaxiques pour limiter l'exploration de l'espace de recherche de la traduction.

Il est enfin possible de se concentrer sur l'amélioration des modèles de langue en langue cible, conduisant à utiliser, par exemple, des grammaires hors-contexte

stochastiques ou encore des grammaires en dépendance stochastiques (par exemple [YAM 01, CHA 03, SHE 08, POS 08]). La question est plus délicate qu'il n'y parait car un modèle de langue cible sert principalement à départager des hypothèses (partielles) de traduction et à guider la recherche. En particulier, ces modèles doivent pouvoir évaluer toute hypothèse, voire toute hypothèse partielle. Or, les grammaires stochastiques utilisées en cible ne sont appropriées pour évaluer des hypothèses partielles ; par ailleurs, elles fournissent de mauvaises évaluations lorsqu'on leur soumet des hypothèses très agrammaticales, ce qui est souvent le cas avec les sorties des systèmes de traduction.

Les recherches les plus actives se concentrent toutefois aujourd'hui sur le développement de modèles de traduction modélisant non-plus des appariements entre des segments arbitraires de la source et de la cible, mais entre des unités correspondant à des constituants syntaxiques dans au moins une des langues. Cela implique notamment de probabiliser, d'estimer et d'utiliser pour le modèle de traduction des mécanismes formels plus expressifs que les transducteurs finis afin de pouvoir mettre en relation des sous-arbres, ou des sous-parties de graphes de dépendances. De nombreux mécanismes ont été proposés et étudiés dans la littérature récente, chacun correspondant à un mécanisme de transduction particulier (grammaire synchrone, transducteur d'arbre, etc). Ainsi, [YAM 01, CHA 03] présente un modèle de traduction convertissant des arbres syntaxiques (en langue cible) en des séquences de mots (en langue source) ; dans le cadre du "canal bruité", ce modèle est utilisé pour définir le terme $\mathrm{P}(\mathbf{f}|\mathbf{e})$ dans l'équation (7.2). À la suite de [ALS 00], le travail de [DIN 05] propose un modèle de transducteur stochastique pour mettre en relation des graphes de dépendance dans les deux langues, une approche qui demande d'analyser les deux "côtés" des corpus bilingues pendant l'apprentissage. Par contraste, l'approche de [QUI 05] ne demande qu'un analyseur en dépendances *dans la langue source* : l'idée principale consiste à construire et estimer des modèles manipulant des unités de traduction bilingues appelées *treelets*. Ces unités sont linguistiquement plus motivées que des segments car elles sont obtenues en projetant le graphe de dépendance de la phrase source sur la phrase cible et en extrayant des fragments du graphe ainsi construit. Des travaux récents, tels que [CAR 09, DEN 09], proposent enfin d'utiliser des formalismes dérivés des grammaires d'arbres adjoints synchrones, un formalisme introduit dans [SHI 90]. En plus de ces travaux, signalons également plusieurs travaux qui présentent des analyses plus théoriques de ces différents systèmes formels [EIS 03, MEL 04, SAT 05, GRA 04, KNI 08].

Le surcoût computationnel lié à l'estimation et à l'inférence avec ces modèles, dont l'utilisation demande, comme pour les modèles hiérarchiques, de mettre en œuvre un analyseur syntaxique en source et/ou en cible n'a pour l'instant pas été compensé par des améliorations de la traduction à la mesure des efforts engagés. Parmi les raisons qui expliquent cet échec relatif, mentionnons que les paires de langues qui ont a ce jour reçu le plus d'attention (`chinois:anglais`, `arabe:anglais`) restent suffisamment proches d'un point de vue syntaxique pour que les modèles simples à base

de segments donnent des résultats difficiles à améliorer. Une autre raison est le bruit introduit par les erreurs d'analyse : aucun analyseur n'est aujourd'hui suffisamment robuste pour analyser sans erreur les phrases sources ; la situation est plus problématique pour le côté cible, puisqu'il faudrait pouvoir analyser toutes les hypothèses de traduction, qui sont souvent très loin d'être grammaticales. Mentionnons finalement que l'augmentation de la quantité de données d'entraînement disponibles, qui sont plus facilement exploitées par des modèles plus simples, a dans les années récentes suffisamment permis d'améliorer les modèles à base de segments pour qu'on puisse se dispenser d'utiliser ces modèles syntaxiques plus complexes.

7.8. Ressources utiles

Les évolutions de la recherche dans le domaine de la traduction automatique par approche statistique sont permanentes et rapides. Un point d'entrée dans l'abondante littérature sont les campagnes d'évaluations liées au domaine. Parmi les plus citées :

– l'évaluation internationale organisée par le NIST (*National Institute of Standards and Technology*) organisée depuis 2002. La dernière campagne (2009) portait sur la traduction de l' arabe, du chinois et de l'urdu vers l'anglais (voir http://www.itl.nist.gov/iad/mig/tests/mt/)

– les évaluations organisées dans le cadre des projets européens *EuroMatrix* et *EuroMatrix+* qui portent sur des langues de l'Union Européenne. La plus récente a eu lieu en 2010 (voir http://www.statmt.org/wmt10).

– les évaluations internationales organisées dans le cadre des ateliers IWSLT (*International Workshop on Spoken Language Translation*), qui sont plus focalisées sur la traduction de parole, voir http://mastarpj.nict.go.jp/IWSLT2009/.

Les deux premières évaluations s'attaquent à la tâche de traduction automatique de textes écrits, alors que IWSLT s'intéresse à des transcriptions manuelles de dialogues téléphoniques dans le domaine du tourisme. Pour *WMT* la plupart des données sont librement accessibles via les sites Internet, alors que pour les évaluations *NIST* et *IWSLT*, les ressources nécessaires sont distribuées respectivement par le *LDC (Linguistic Data Consortium)*[54] et le *National Institute of Information and Communication Technology* (au Japon). Enfin, *IWLST* a pour particularité de proposer une tâche plus restreinte que les deux autres et les ressources associées sont des corpus de tailles comparativement petites. Néanmoins, les campagnes d'évaluations ne constituent pas un accès des plus pédagogiques et scientifiques au domaine. Elles donnent cependant des informations sur les travaux et sur les laboratoires les plus en pointe ainsi que sur les verrous scientifiques et technologiques actuels.

54. http://www.ldc.upenn.edu

7.8.0.4. *Bases bibliographiques et ressources en ligne*

Il existe deux ressources bibliographiques très riches dans le domaine :

– http://www.mt-archive.info/ : liste la plupart des articles sur la traduction automatique, avec un accès en ligne pour ceux qui sont librement disponibles sous forme électronique ;

– http://aclweb.org/anthology-new/ : regroupe toutes les publications des conférences et revues liées à l'*Association for Computational Linguistics*. Les thématiques abordés sont donc très variées mais de très nombreuses publications, notamment ces dernières années, s'intéressent à la traduction automatique statistique.

7.8.0.5. *Les corpus parallèles*

En plus des corpus déjà cités en début de section, de nombreux corpus parallèles sont rassemblés au sein d'OPUS (*open source parallel corpus*) [TIE 09]. Les données peuvent être consultées ou téléchargées à partir du site http://www.let.rug.nl/~tiedeman/OPUS/. Les corpus proposés proviennent du Web et couvrent plusieurs domaines comme des textes officiels (la Constitution européenne), des documents médicaux, des sous-titres de films et des documentations informatiques. Il existe également des corpus parallèles construits à partir de bulletins météorologiques, les corpus Météo du RALI : http://www-rali.iro.umontreal.ca/Meteo/index.fr.html.

7.8.0.6. *Outils pour la traduction automatique statistique*

Le système Moses est distribué sous la licence *LGPL* (http://www.statmt.org/moses/), il est donc *open-source*. Cette distribution est fréquemment mise à jour. Elle inclut les outils nécessaires afin de construire un système à base de segments, ainsi que des modèles plus récents comme les modèles hiérarchiques et les modèles factorisés. Le système Joshua (http://www.cs.jhu.edu/~ccb/joshua/index.html) propose également un système pour mettre en œuvre les modèles hiérarchiques présentés à la section 7.7.2.

Hormis les systèmes complets, notons également la libre disponibilité de différents outils d'alignement comme *GIZA++* (http://code.google.com/p/giza-pp/) et les outils de Berkeley (http://code.google.com/p/berkeleyaligner/). Pour manipuler de manière générale les corpus parallèles et les alignements mot-à-mot, il existe par exemple la boite à outils Uplug http://sourceforge.net/projects/uplug/.

Enfin, le développement d'un système de traduction automatique nécessite la construction et la manipulation de modèles statistiques du langage. Deux boites à outils sont couramment utilisés : SRI-LM http://www.speech.sri.com/projects/srilm/ et IRST-LM http://sourceforge.net/projects/irstlm/.

7.8.0.7. *Évaluation de la traduction automatique*

Afin d'évaluer automatiquement une traduction automatique, plusieurs outils existent permettant l'usage de différentes métriques (voir la section 7.6) :

– `ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl` permet de calculer le score $BLEU_k$. Ce paquetage est diffusé dans le cadre des campagnes d'évaluation du NIST ;

– `http://http://www.cs.umd.edu/~snover/tercom/` contient un paquetage du calcul des scores TEP et TERp ;

– `http://www-2.cs.cmu.edu/~alavie/METEOR/` contient les paquetages permettant de calculer le score METEOR et ses variantes récentes.

7.9. Conclusion

Nous avons, dans ce chapitre, présenté les principaux modèles et algorithmes qui sont aujourd'hui utilisés dans les systèmes de traduction automatique statistique de l'état de l'art. En détaillant les modèles de traduction à base de segments, nous avons introduit les outils nécessaires à réaliser les alignements, qui constituent une l'étape préalable à la construction, puis à l'estimation des tables de segments. Nous avons également discuté des problèmes liés aux réordonnancements et passé en revue les principaux outils statistiques impliqués dans la modélisation de ces phénomènes. Nous avons ensuite montré comment ces deux types de modélisation sont intégrés au sein des algorithmes de recherche heuristique qui sont au cœur des outils de traduction. En dressant un rapide panorama des principales recherches en cours, et des verrous technologiques auxquelles elles se confrontent, nous avons évoqué les évolutions possibles à court terme de ces outils, qui restent, par bien des aspects, extrêmement frustres et imparfaits.

En premier lieu, en dépit de leur simplicité conceptuelle, le développement de tels systèmes reste extrêmement complexes et computationnellement coûteux : il s'agit de traiter des millions de phrases, d'ajuster itérativement des centaines de millions de paramètres numériques, de construire, par optimisation sur des corpus de tests, des règles de décision probabilistes combinant de multiples modèles, toutes opérations qui demandent des heures, parfois des jours de calculs, et qui donc ne peuvent être aisément répétées, sauf à disposer de moyens de calcul colossaux. Ces systèmes pêchent donc par leur lourdeur et leur manque d'adaptabilité. Ces systèmes sont également fragilisés par leur dépendance aux ressources parallèles, qui existent dans des quantités inégales selon les couples de langue et les genres, registres et domaines des documents à traduire. Nous l'avons évoqué, les recherches les plus actives visent à compenser cette rareté des données par une intégration de modèles linguistiques de plus en plus réalistes et/ou par une hybridation avec des systèmes par règles, en particulier pour ce qui concerne la morphologie et la syntaxe. Les freins à l'adoption de

tels modèles sont toutefois sérieux : en particulier le surcoût de complexité algorithmique/computationnel qu'ils induisent aussi bien en apprentissage qu'en inférence est rarement payé d'une amélioration proportionnelle de la qualité des traductions obtenues. Pour ces quelques raisons, il est probable que les technologies vont continuer de progresser à un rythme relativement lent.

Il n'en reste que la traduction automatique, en particulier statistique, s'est imposée comme une technologie de base d'un Internet de plus en plus polyglotte. Malgré ses imperfections, elle a permis de donner accès, au plus grand nombre, à des traductions automatiques d'une qualité suffisante pour de nombreux usages. Même si la question des limites de ces technologies est encore relativement ouverte, il est très improbable que des traducteurs automatiques remplacent jamais complètement des traducteurs humains *dans les tâches qui sont aujourd'hui les leurs*, il est vraisemblable que ces technologies vont continuer de diffuser dans les environnements de traduction professionnels, pour assister l'humain lorsque les tâches de traduction le permettent.

Mais il est tout aussi assuré que la diffusion de la traduction automatique auprès du grand public va susciter de nouveaux usages, qui restent à inventer et viendront orienter les développements technologiques de demain. En effet, le schéma de traduction, dans lequel un document en langue source A, produit par un locuteur/scripteur ignorant tout de la langue B, est traduit d'un bloc par un spécialiste de la traduction de A vers B, n'est pas la seule possible. Entre la connaissance parfaite d'une langue et son ignorance totale, il existe toute une palette de situations intermédiaires de bilinguisme partiel. Ces situations se multiplient lorsque l'on considère des couples de langues historiquement apparentées (par ex. les langues romanes), dont l'intercompréhension est grandement facilitée par de nombreuses racines et constructions partagées. Dans de tels contextes, pourront s'imposer des modes d'écriture/lecture/écoute mutli-lingues dans lesquels des dispositifs de traduction automatique viendraient, en fonction des besoins et des connaissances des locuteurs, faciliter la production ou la réception de messages linguistiques.

7.10. Remerciements

Ce chapitre doit beaucoup aux échanges avec nos collègues des équipes du LIMSI/CNRS qui travaillent sur le thème de la traduction automatique, en particulier G. Adda, J.M Crego, H. Bonnaud-Maynard, T. Lavergne et J.-L. Gauvain avec une mention spéciale à A. Max et G. Wisniewski dont les commentaires sur une première version de ce chapitre nous ont été très précieux.

7.11. Bibliographie

[AHO 69] AHO A. V., ULLMAN J., « Syntax Directed Translations and the Pushdown Assembler », *Journal of computer and system sciences*, vol. 3, p. 37–56, 1969. 28

[ALO 06] AL-ONAIZAN Y., PAPINENI K., « Distortion Models for Statistical Machine Translation », *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, p. 529–536, 2006. 7.4, 7.4.2.3

[ALS 00] ALSHAWI H., DOUGLAS S., BANGALORE S., « Learning Dependency Translation Models as Collections of Finite-State Head Transducers », *Computational Linguistics*, vol. 26, n°1, p. 45–60, 2000. 7.7.3.3

[BAN 02] BANGALORE S., RICCARDI G., « Stochastic Finite-State Models for Spoken Language Machine Translation », *Machine Translation*, vol. 17, p. 165–184, 2002. 7.5.3.2

[BAN 05] BANERJEE S., LAVIE A., « METEOR : An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments », *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation*, Ann Arbor, Michigan, p. 65–72, 2005. 7.6.3

[BAN 07] BANGALORE S., HAFFNER P., KANTHAK S., « Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction », *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, p. 152–159, 2007. 7.7.1.1

[BEL 01] BELLAGARDA J. R., « An overview of statistical language model adaptation », *Proceedings of the ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, Sophia Antipolis, France, p. 165–174, 2001. 7.7.1.2

[BER 96] BERGER A., BROWN P. F., PIETRA S. D., PIETRA V. J. D., KEHLER A. S., MERCER R. L., « Language translation apparatus and method of using context-based translation models », U.S. Patent 5510981, 1996. 25

[BER 09] BERTOLDI N., FEDERICO M., « Domain Adaptation for Statistical Machine Translation with Monolingual Resources », *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, p. 182–189, 2009. 7.7.1.2

[BIR 09] BIRCH A., BLUNSOM P., OSBORNE M., « A Quantitative Analysis of Reordering Phenomena », *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, Association for Computational Linguistics, p. 197–205, 2009. 7.4

[BLA 07] BLANCHON H., BOITET C., « Pour l'évaluation externe des systèmes de TA par des méthodes fondées sur la tâche », *Traitement automatique des langues*, vol. 48, n°1, p. 33–65, 2007. 7.6.1

[BLU 06] BLUNSOM P., COHN T., « Discriminative Word Alignment with Conditional Random Fields », *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, p. 65–72, 2006. 7.3.2

[BRA 07] BRANTS T., POPAT A. C., XU P., OCH F. J., DEAN J., « Large Language Models in Machine Translation », *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, p. 858–867, 2007. 7.3.1.1.2

[BRO 90] BROWN P. F., COCKE J., PIETRA S. D., PIETRA V. J. D., JELINEK F., LAFFERTY J. D., MERCER R. L., ROOSSIN P. S., « A Statistical Approach to Machine Translation », *Computational Linguistics*, vol. 16, n°2, p. 79–85, 1990. 7.1.1, 5, 7.2.2.2, 7.2.2.3, 7.3.1, 7.3.1, 7.3.1.2, 7.3.1.3, 7.3.1.3.2

[BRO 92] BROWN P. F., DESOUZA P. V., MERCER R. L., PIETRA V. J. D., LAI J. C., « Class-based n-gram models of natural language », *Comput. Linguist.*, vol. 18, n°4, p. 467–479, MIT Press, 1992. 7.3.1.3.2

[BRO 93a] BROWN P. F., PIETRA S. A. D., PIETRA V. J. D., GOLDSMITH M. J., HAJIC J., MERCER R. L., MOHANTY S., « But Dictionaries Are Data Too », *Proceedings of the ARPA workshop on Human Language Technologies (HLT'93)*, Plainsboro, New Jersey, p. 202–206, 1993. 7.7.3.1

[BRO 93b] BROWN P. F., PIETRA S. A. D., PIETRA V. J. D., MERCER R. L., « The Mathematics of Statistical Machine Translation : Parameter Estimation », *Computational Linguistics*, vol. 19, n°2, p. 263–311, 1993. 7.2.1, 7.2.2.2, 7.3.1.1.2, 7.3.2

[CAL 06] CALLISON-BURCH C., OSBORNE M., KOEHN P., « Re-evaluating the role of BLEU in machine translation research », *Proceedings of European Chapter of the Association for Computational Linguistics (EACL)*, Genoa, Italy, p. 249–256, 2006. 7.6.3

[CAL 08] CALLISON-BURCH C., FORDYCE C., KOEHN P., MONZ C., SCHROEDER J., « Further Meta-Evaluation of Machine Translation », *Proceedings of the Third Workshop on Statistical Machine Translation*, Columbus, Ohio, p. 70–106, 2008. 7.6.1, 7.6.2

[CAL 09a] CALLISON-BURCH C., « Fast, Cheap, and Creative : Evaluating Translation Quality Using Amazon's Mechanical Turk », *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, p. 286–295, 2009. 41

[CAL 09b] CALLISON-BURCH C., KOEHN P., MONZ C., SCHROEDER J., « Findings of the 2009 Workshop on Statistical Machine Translation », *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, p. 1–28, 2009. 7.6.1, 7.6.2, 7.6.3

[CAL 10] CALLISON-BURCH C., KOEHN P., MONZ C., PETERSON K., PRZYBOCKI M., ZAIDAN O., « Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation », *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, Uppsala, Sweden, p. 17–53, 2010. 7.6

[CAR 03] CARL M., WAY A., Eds., *Recent Advances in Example-Based Machine Translation*, vol. 21 de *Text, Speech and Language Technology*, Springer, Verlag, 2003. 7.1.1

[CAR 05] CARPUAT M., WU D., « Word Sense Disambiguation vs. Statistical Machine Translation », *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, p. 387–394, 2005. 7.7.1.1

[CAR 07] CARPUAT M., WU D., « Improving Statistical Machine Translation Using Word Sense Disambiguation », *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, p. 61–72, 2007. 7.7.1.1

[CAR 09] CARRERAS X., COLLINS M., « Non-Projective Parsing for Statistical Machine Translation », *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, Association for Computational Linguistics, p. 200–209, August 2009. 7.7.3.3

[CAS 04] CASACUBERTA F., VIDAL E., « Machine Translation with Inferred Stochastic Finite-State transducers », *Computational Linguistics*, vol. 30, n°3, p. 205–225, 2004. 7.5.3.2

[CER 08] CER D., JURAFSKY D., MANNING C. D., « Regularization and Search for Minimum Error Rate Training », *Proceedings of the Third Workshop on Statistical Machine Translation*, Columbus, Ohio, p. 26–34, 2008. 7.5.1.2

[CHA 98] CHAPPELIER J.-C., RAJMAN M., « A generalized CYK algorithm for parsing stochastic CFG », *Proc. of 1st Workshop on Tabulation in Parsing and Deduction (TAPD'98)*, Paris (France), p. 133–137, apr 1998. 53

[CHA 03] CHARNIAK E., KNIGHT K., YAMADA K., « Syntax-based Language Models for Statistical Machine Translation », *Proceedings of the Machine Translation Summit IX*, New Orleans, USA, 2003. 7.7.3.3

[CHA 08] CHAUDIRON S., CHOUKRI K., Eds., *L'évaluation des technologies de traitement de la langue*, Hermès, 2008. 7.6

[CHE 78] CHEVALIER M., DANSEREAU J., POULIN G., TAUM-Météo : description du système, Rapport, Groupe TAUM, Université de Montréal, Montréal, Canada, 1978. 7.1.1

[CHE 06] CHERRY C., LIN D., « A Comparison of Syntactically Motivated Word Alignment Spaces », *Proceedings of EACL'06*, Genoa, Italy, p. 145–152, 2006. 7.7.3.3

[CHE 08] CHERRY C., « Cohesive Phrase-Based Decoding for Statistical Machine Translation », *Proceedings of ACL-08 : HLT*, Columbus, Ohio, Association for Computational Linguistics, p. 72–80, June 2008. 7.7.3.3

[CHI 05] CHIANG D., « A Hierarchical Phrase-Based Model for Statistical Machine Translation », *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, p. 263–270, 2005. 7.7.2, 7.7.2

[CHI 08] CHIANG D., MARTON Y., RESNIK P., « Online Large-Margin Training of Syntactic and Structural Translation Features », *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, p. 224–233, 2008. 7.5.1.2, 7.7.3.3

[CHU 09] CHUNG T., GILDEA D., « Unsupervised Tokenization for Machine Translation », *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, Association for Computational Linguistics, p. 718–726, August 2009. 7.7.3.2

[COL 05] COLLINS M., KOEHN P., KUCEROVA I., « Clause Restructuring for Statistical Machine Translation », *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, p. 531–540, 2005. 7.7.3.3

[CRE 07] CREGO J. M., NO J. B. M., « Improving SMT by coupling reordering and decoding », *Machine Translation*, vol. 20, n°3, p. 199-215, 2007. 7.4.1.4, 7.5.4

[CRE 08] CREGO J. M., HABASH N., « Using Shallow Syntax Information to Improve Word Alignment and Reordering for SMT », *Proceedings of the Third Workshop on Statistical Machine Translation*, Columbus, Ohio, Association for Computational Linguistics, p. 53–61, June 2008. 7.7.3.3

[DEM 77] DEMPSTER A. P., LAIRD N. M., RUBIN D. B., « Maximum-likelihood from incomplete data via the EM algorithm », *Journal of the Royal Statistical Society*, vol. 39, n°1, p. 1–38, 1977. 7.3.1.1.2

[DEM 98] DEMORI R., FEDERICO M., « Language Model Adaptation », PONTING K., Ed., *Computational Models of Speech Pattern Processing*, Springer Verlag, p. 280–303, 1998. 7.7.1.2

[DEN 06] DENERO J., GILLICK D., ZHANG J., KLEIN D., « Why Generative Phrase Models Underperform Surface Heuristics », *Proceedings of the ACL workshop on Statistical Machine Translation*, New York City, NY, p. 31–38, 2006. 7.3.3

[DEN 09] DENEEFE S., KNIGHT K., « Synchronous Tree Adjoining Machine Translation », *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, Association for Computational Linguistics, p. 727–736, August 2009. 7.7.3.3

[DIN 05] DING Y., PALMER M., « Machine Translation Using Probabilistic Synchronous Dependency Insertion Grammars », *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, p. 541–548, 2005. 7.7.3.3

[ECO 07] ECO U., *Dire presque la même chose : Expériences de traduction*, Grasset, 2007. 40

[EIS 03] EISNER J., « Learning Non-Isomorphic Tree Mappings for Machine Translation », *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), Companion Volume*, Sapporo, p. 205–208, 2003. 7.7.3.3

[EIS 06] EISNER J., TROMBLE R. W., « Local Search with Very Large-Scale Neighborhoods for Optimal Permutations in Machine Translation », *Proceedings of the HLT-NAACL Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, New York, p. 57–75, June 2006. 7.4.1.5, 7.5.4.2

[FOS 07] FOSTER G., KUHN R., « Mixture-Model Adaptation for SMT », *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, p. 128–135, 2007. 7.7.1.2

[FOS 09] FOSTER G., KUHN R., « Stabilizing Minimum Error Rate Training », *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, p. 242–249, March 2009. 7.5.1.2

[FOX 02] FOX H. J., « Phrasal cohesion and statistical machine translation », *EMNLP '02 : Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, Morristown, NJ, USA, Association for Computational Linguistics, p. 304–3111, 2002. 7.7.3.3

[FRA 07] FRASER A., MARCU D., « Measuring Word Alignment Quality for Statistical Machine Translation », *Computational Linguistics*, vol. 33, n°3, p. 293–303, MIT Press, 2007.

7.3.2

[FUC 96] FUCHS C., *Les ambiguïtés du français*, Ophrys, Paris, France, 1996. 7.2.2.1

[GAL 95] GALLIERS J. R., JONES K. S., *Evaluating Natural Language Processing Systems*, Lecture Notes in Artificial Intelligence, Springer, Berlin - Heidelberg - New York, 1995. 7.6

[GER 01] GERMANN U., JAHR M., KNIGHT K., MARCU D., YAMADA K., « Fast Decoding and Optimal Decoding for Machine Translation », *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01)*, Toulouse, France, p. 228–235, 2001. 7.5.4, 7.5.4.2

[GER 03] GERMANN U., « Greedy decoding for statistical machine translation in almost linear time », *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, p. 1–8, 2003. 7.5.4.2

[GIM 08] GIMPEL K., SMITH N. A., « Rich Source-Side Context for Statistical Machine Translation », *Proceedings of the Third Workshop on Statistical Machine Translation*, Columbus, Ohio, p. 9–17, June 2008. 7.7.1.1

[GIS 08] DE GISPERT A., MARIÑO J. B., « On the impact of morphology in English to Spanish statistical MT », *Speech Commununication*, vol. 50, n°11-12, p. 1034–1046, Elsevier Science Publishers B. V., 2008. 7.7.3.2

[GRA 04] GRAEHL J., KNIGHT K., « Training Tree Transducers », SUSAN DUMAIS D. M., ROUKOS S., Eds., *HLT-NAACL 2004 : Main Proceedings*, Boston, Massachusetts, USA, Association for Computational Linguistics, p. 105–112, May 2 - May 7 2004. 7.7.3.3

[HAG 09] HAGHIGHI A., BLITZER J., DENERO J., KLEIN D., « Better Word Alignments with Supervised ITG Models », *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Suntec, Singapore, p. 923–931, 2009. 7.4.1.5

[HUA 07] HUANG L., CHIANG D., « Forest Rescoring : Faster Decoding with Integrated Language Models », *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, p. 144–151, 2007. 7.7.2

[HUA 09] HUANG L., ZHANG H., GILDEA D., KNIGHT K., « Binarization of Synchronous Context-Free Grammars », *Computational Linguistics*, vol. 35, n°4, 2009. 7.4.1.5

[HUT 92] HUTCHINS W. J., SOMERS H. L., *An Introduction to Machine Translation*, Academic Press, 1992. 7.1.1

[HUT 01] HUTCHINS J., « Machine translation over fifty years », *Histoire, Epistémologie, Langage.*, vol. 23, n°1, p. 3–31, 2001, Numéro spécial : Le traitement automatique des langues (edité par Jacqueline Léon). 7.1.1

[HUT 03] HUTCHINS J., « Has machine translation improved ? some historical comparisons », *Proceedings of the MT Summit IX : proceedings of the Ninth Machine Translation Summit*, New Orleans, LO, p. 181–188, 2003. 7.1.1

[HWA 02] HWA R., RESNIK P., WEINBERG A., KOLAK O., « Evaluating translational correspondence using annotation projection », *ACL '02 : Proceedings of the 40th Annual Meeting*

on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics, p. 392–399, 2002. 7.7.3.3

[IGL 09] IGLESIAS G., DE GISPERT A., R. BANGA E., BYRNE W., « Hierarchical Phrase-Based Translation with Weighted Finite State Transducers », *Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, Association for Computational Linguistics, p. 433–441, June 2009. 7.7.2

[ITT 07] ITTYCHERIAH A., ROUKOS S., « Direct Translation Model 2 », *Human Language Technologies 2007 : The Conference of the North American Chapter of the Association for Computational Linguistics ; Proceedings of the Main Conference*, Rochester, New York, p. 57–64, 2007. 7.7.1.1

[JEL 97] JELINEK F., *Statistical Methods for Speech Recognition*, The MIT Press, Cambridge, MA, 1997. 7.2.1, A.4.3.2.2

[KAN 05] KANTHAK S., VILAR D., MATUSOV E., ZENS R., NEY H., « Novel Reordering Approaches in Phrase-Based Statistical Machine Translation », *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Ann Arbor, Michigan, p. 167–174, 2005. 7.4.1.2, 7.4.1.5

[KIN 96] KING M., « Evaluating natural language processing systems », *Communication of the ACM*, vol. 39, n°1, p. 73–79, ACM, 1996. 7.6

[KNI 99] KNIGHT K., « Decoding Complexity in Word-Replacement Translation Models », *Computational Linguistics*, vol. 25, n°4, p. 607–615, 1999. 7.5.2

[KNI 08] KNIGHT K., « Capturing Practical Natural Language Transformations », *Machine Translation*, vol. 22, n°2, p. 121–133, 2008. 7.7.3.3

[KOE 03a] KOEHN P., KNIGHT K., « Empirical methods for compound splitting », *EACL '03 : Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, Morristown, NJ, USA, Association for Computational Linguistics, p. 187–193, 2003. 7.7.3.2

[KOE 03b] KOEHN P., OCH F. J., MARCU D., « Statistical Phrase-Based Translation », *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistic*, Edmondton, Canada, p. 127–133, 2003. 7.2.3, 7.3, 7.3.1.4, 7.3.3.3, 7.7.3.3

[KOE 04] KOEHN P., « Pharaoh : a beam search decoder for phrase-based statistical machine translation models », E.FREDERKING R., B.TAYLOR K., Eds., *Machine translation : from real users to research : Proceedings of the 6th conference of the Association for Machine Translation in the Americas*, Lecture Notes in Computer Science 3265, Springer Verlag, p. 115–124, 2004. 7.4.2.2, 7.5.4.1, 7.5.4.2

[KOE 05] KOEHN P., « Europarl : A Parallel Corpus for Statistical Machine Translation », *2nd Workshop on EBMT of MT-Summit X*, Phuket, Thailand, p. 79–86, 2005. 35

[KOE 07a] KOEHN P., HOANG H., « Factored Translation Models », *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, p. 868–876, 2007. 7.7.3.2

[KOE 07b] 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., HERBST E., « Moses : Open Source Toolkit for Statistical Machine Translation », *Proc. Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, Prague, Czech Republic, 2007. 7.4.1.3, 7.4.2.2, 7.4.2.3

[KUM 03] KUMAR S., BYRNE W., « A weighted finite state transducer implementation of the alignment template model for statistical machine translation », *NAACL '03 : Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Morristown, NJ, USA, p. 63–70, 2003. 7.5.3.2

[KUM 04] KUMAR S., BYRNE W., « Minimum Bayes-Risk Decoding for Statistical Machine Translation », SUSAN DUMAIS D. M., ROUKOS S., Eds., *HLT-NAACL 2004 : Main Proceedings*, Boston, Massachusetts, USA, Association for Computational Linguistics, p. 169–176, May 2 - May 7 2004. 7.5.4

[KUM 05] KUMAR S., BYRNE W., « Local Phrase Reordering Models for Statistical Machine Translation », *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada, p. 161–168, 2005. 7.4.1.1, 7.4.2.3

[KUM 06] KUMAR S., DENG Y., BYRNE W., « A weighted finite state transducer translation template model for statistical machine translation », *Natural Language Engineering*, vol. 12, n°1, p. 35–75, 2006. 7.5.3.2

[LAN 07] LANGLAIS P., PATRY A., GOTTI F., « A greedy decoder for phrase-based statistical machine translation », *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI'07)*, Skövde (Sweden), p. 104–113, 2007. 7.5.4.2

[LEW 68] LEWIS II P. M., STEARNS R. E., « Syntax-Directed Transduction », *Journal of the ACM*, vol. 15, n°3, p. 465–488, 1968. 28

[LIN 03] LIN D., CHERRY C., « Word alignment with cohesion constraint », *NAACL '03 : Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Morristown, NJ, USA, Association for Computational Linguistics, p. 49–51, 2003. 7.7.3.3

[LOP 08] LOPEZ A., « Tera-Scale Translation Models via Pattern Matching », *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, UK, p. 505–512, 2008. 7.7.2

[LOP 09] LOPEZ A., « Translation as Weighted Deduction », *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, Athens, Greece, p. 532–540, March 2009. 25, 7.4.1.3

[MA 08] MA Y., OZDOWSKA S., SUN Y., WAY A., « Improving word alignment using syntactic dependencies », *SSST '08 : Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation*, Morristown, NJ, USA, Association for Computational Linguistics, p. 69–77, 2008. 7.7.3.3

[MAC 08] MACHEREY W., OCH F., THAYER I., USZKOREIT J., « Lattice-based Minimum Error Rate Training for Statistical Machine Translation », *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, p. 725–734, 2008. 7.5.1.2

[MAN 99] MANNING C. D., SCHÜTZE H., *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, MA, 1999. 7.1.1, 7.2.1, A.1, A.5

[MAR 02] MARCU D., WONG D., « A Phrase-Based,Joint Probability Model for Statistical Machine Translation », *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, p. 133–139, July 2002. 7.3

[MAT 08] MATUSOV E., LEUSCH G., BANCHS R. E., BERTOLDI N., DECHELOTTE D., FEDERICO M., KOLSS M., LEE Y.-S., MARIÑO J., PAULIK M., ROUKOS S., SCHWENK H., NEY H., « System Combination for Machine Translation of Spoken and Written Language », *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, n°7, p. 1222-1237, 2008. 7.5.4

[MAU 09] MAUSER A., HASAN S., NEY H., « Extending Statistical Machine Translation with Discriminative and Trigger-Based Lexicon Models », *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, p. 210–218, 2009. 7.7.1.1

[MEL 04] MELAMED I. D., « Statistical machine translation by parsing », *ACL '04 : Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, Association for Computational Linguistics, page653, 2004. 7.7.3.3

[MIN 07] MINKOV E., TOUTANOVA K., SUZUKI H., « Generating Complex Morphology for Machine Translation », *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, Association for Computational Linguistics, p. 128–135, June 2007. 7.7.3.2

[MOH 02] MOHRI M., PEREIRA F. C. N., RILEY M., « Weighted Finite-State Transducers in Speech Recognition », *Computer Speech and Language*, vol. 16, n°1, p. 69–88, 2002. 7.5.3.2

[MOO 08] MOORE R. C., QUIRK C., « Random Restarts in Minimum Error Rate Training for Statistical Machine Translation », *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, UK, p. 585–592, 2008. 7.5.1.2

[NAG 04] NAGAO M., « A framework of a mechanical translation between Japanese and English by analogy principle », ELITHORN A., BANERJI R., Eds., *Artificial and Human Intelligence*, Elsevier Science Publishers, 2004. 7.1.1

[NEY 04] NEY H., POPOVIC M., « Improving Word Alignment Quality using Morpho-syntactic Information », *Proceedings of Coling 2004*, Geneva, Switzerland, COLING, p. 310–314, Aug 23–Aug 27 2004. 7.7.3.3

[NIE 04] NIESSEN S., NEY H., « Statistical Machine Translation with Scarce Resources Using Morpho-syntactic Information », *Computational Linguistics*, vol. 30, n°2, p. 181–204, MIT Press, 2004. 7.7.3.1, 7.7.3.2

[NIE 08] NIEHUES J., VOGEL S., « Discriminative Word Alignment via Alignment Matrix Modeling », *Proceedings of the Third Workshop on Statistical Machine Translation*, Columbus, Ohio, p. 18–25, 2008. 7.3.2

[NIE 09] NIE J., HIRST G., *Cross-Language Information Retrieval*, Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers, 2009. 3

[OCH 03a] OCH F. J., « Minimum Error Rate Training in Statistical Machine Translation », *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, p. 160–167, 2003. 7.5.1.2

[OCH 03b] OCH F. J., NEY H., « A Systematic Comparison of Various Statistical Alignment Models », *Computational Linguistics*, vol. 29, n°1, p. 19-51, 2003. 7.3.1, 7.3.1.2.1, 20, 7.3.1.3.2, 7.3.1.4, 7.3.2

[OCH 04] OCH F. J., NEY H., « The Alignment Template Approach to Statistical Machine Translation », *Computational Linguistics*, vol. 30, n°4, p. 417–449, MIT Press, 2004. 7.2.3

[OFL 07] OFLAZER K., EL-KAHLOUT I. D., « Exploring different representational units in English-to-Turkish statistical machine translation », *StatMT '07 : Proceedings of the Second Workshop on Statistical Machine Translation*, Morristown, NJ, USA, Association for Computational Linguistics, p. 25–32, 2007. 7.7.3.2

[PAP 01] PAPINENI K., ROUKOS S., WARD T., ZHU W.-J., Bleu : a method for automatic evaluation of machine translation, Rapport n°RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, 2001. 7.6.1.1

[PEA 84] PEARL J., *Heuristics : Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, 1984. 7.5.4

[POS 08] POST M., GILDEA D., « Parsers as language models for statistical machine translation », *Proceedings of AMTA*, 2008. 7.7.3.3

[POW 64] POWELL M., « An efficient method for finding the minimum of a function of several variables without calculating derivatives », *Computer Journal*, vol. 7, p. 152–162, 1964. 7.5.1.2

[QUI 05] QUIRK C., MENEZES A., CHERRY C., « Dependency treelet translation : syntactically informed phrasal SMT », *ACL '05 : Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, Association for Computational Linguistics, p. 271–279, 2005. 7.7.3.3

[QUI 07] QUIRK R. C. M. . C., « Faster beam-search decoding for phrasal statistical machine translation », *Proceedings of the Machine Translation Summit XI*, Copenhagen, Denmark, p. 321–327, 2007. 7.5.4.1, 7.5.4.1.1

[ROS 07] ROSTI A.-V., MATSOUKAS S., SCHWARTZ R., « Improved Word-Level System Combination for Machine Translation », *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, Association for Computational Linguistics, p. 312–319, 2007. 7.5.4

[SAE 09] SAERS M., WU D., « Improving Phrase-Based Translation via Word Alignments from Stochastic Inversion Transduction Grammars », *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation (SSST-3) at NAACL HLT 2009*, Boulder,

Colorado, Association for Computational Linguistics, p. 28–36, June 2009. 7.4.1.5

[SAT 05] SATTA G., PESERICO E., « Some Computational Complexity Results for Synchronous Context-Free Grammars », *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada, Association for Computational Linguistics, p. 803–810, October 2005. 7.7.3.3

[SCH 08] SCHWENK H., FOUET J.-B., SENELLART J., « First steps towards a general purpose French/English statistical machine translation system », *StatMT'08 : Proceedings of the Third Workshop on Statistical Machine Translation*, Morristown, NJ, USA, Association for Computational Linguistics, p. 119–122, 2008. 7.7.3.1

[SHE 08] SHEN L., XU J., WEISCHEDEL R., « A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model », *Proceedings of ACL-08 : HLT*, Columbus, Ohio, p. 577–585, June 2008. 7.7.3.3

[SHI 90] SHIEBER S. M., SCHABES Y., « Synchronous tree-adjoining grammars », *Proceedings of the 13th conference on Computational linguistics*, Morristown, NJ, USA, p. 253–258, 1990. 7.7.3.3

[SIM 05] SIMARD M., CANCEDDA N., CAVESTRO B., DYMETMAN M., GAUSSIER E., GOUTTE C., YAMADA K., LANGLAIS P., MAUSER A., « Translating with Non-contiguous Phrases », *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada, p. 755–762, 2005. 7.7.2

[SNO 06] SNOVER M., DORR B., SCHWARTZ R., MICCIULLA L., MAKHOUL J., « A Study of Translation Edit Rate with Targeted Human Annotation », *Proceedings of the conference of the Association for Machine Translation in the America (AMTA)*, p. 223–231, 2006. 7.6.3, 7.17

[SNO 09] SNOVER M., MADNANI N., DORR B., SCHWARTZ R., « Fluency, Adequacy, or HTER ? Exploring Different Human Judgments with a Tunable MT Metric », *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, Association for Computational Linguistics, p. 259–268, March 2009. 7.6.3

[STR 07] STROPPA N., VAN DEN BOSCH A., WAY A., « Exploiting source similarity for SMT using context-informed features », WAY A., GAWRONSKA B., Eds., *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI'07)*, Skövde, Sweden, p. 231–240, 2007. 7.7.1.1

[TIE 09] TIEDEMANN J., « News from OPUS – A Collection of Multilingual Parallel Corpora with Tools and Interfaces », NICOLOV N., ANGELOVA G., MITKOV R., Eds., *Recent Advances in Natural Language Processing V*, vol. 309 de *Current Issues in Linguistic Theory*, Amsterdam & Philadelphia, John Benjamins, p. 227–248, 2009. 7.8.0.5

[TIL 03] TILLMANN C., NEY H., « Word Reordering and a Dynamic Programming Beam Search Algorithm for Statistical Machine Translation », *Computational Linguistics*, vol. 29, n°1, p. 97–133, 2003. 25

[TIL 04] TILLMAN C., « A Unigram Orientation Model for Statistical Machine Translation », DUMAIS S., MARCU D., ROUKOS S., Eds., *HLT-NAACL 2004 : Short Papers*, Boston,

Massachusetts, USA, p. 101–104, 2004. 7.4.2.3, 7.4.2.3

[TOU 02] TOUTANOVA K., ILHAN H. T., MANNING C. D., « Extensions to HMM-based statistical word alignment models », *EMNLP '02 : Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, Morristown, NJ, USA, Association for Computational Linguistics, p. 87–94, 2002. 7.7.3.3

[TRO 08] TROMBLE R., KUMAR S., OCH F., MACHEREY W., « Lattice Minimum Bayes-Risk Decoding for Statistical Machine Translation », *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, p. 620–629, 2008. 7.5.4

[UDU 06] UDUPA R., MAJI H. K., « Computational Complexity of Statistical Machine Translation », *Proceedings of the Meeting of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, p. 25–32, 2006. 7.3.1.3.1

[UEF 03] UEFFING N., NEY H., « Using POS information for statistical machine translation into morphologically rich languages », *EACL '03 : Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, Morristown, NJ, USA, Association for Computational Linguistics, p. 347–354, 2003. 7.7.3.2

[VEN 03] VENUGOPAL A., VOGEL S., WAIBEL A., « Effective Phrase Translation Extraction from Alignment Models. », *ACL*, p. 319-326, 2003. 7.3

[VIL 06] VILAR D., XU J., LUIS FERNANDO D., NEY H., « Error analysis of statistical machine translation output », *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, Genoa, Italy, 2006. 7.7.3

[VOG 96] VOGEL S., NEY H., TILLMANN C., « HMM-based word alignment in statistical translation », *Proceedings of the 16th conference on Computational linguistics*, Morristown, NJ, USA, p. 836–841, 1996. 7.3.1, 7.3.1.2.1, 7.3.1.2.1, 7.3.1.2.1

[VOG 05] VOGEL S., « PESA : Phrase Pair Extraction as Sentence Splitting », *Proceedings of the tenth Machine Translation Summit*, Phuket, Thailand, 2005. 7.3

[WAG 74] WAGNER R. A., FISCHER M. J., « The String-to-String Correction Problem », *Journal of the ACM (JACM)*, vol. 21, n°1, p. 168–173, ACM Press, 1974. 43

[WAT 07] WATANABE T., SUZUKI J., TSUKADA H., ISOZAKI H., « Online Large-Margin Training for Statistical Machine Translation », *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, Association for Computational Linguistics, p. 764–773, June 2007. 7.5.1.2

[WEL 06] WELLINGTON B., WAXMONSKY S., MELAMED I. D., « Empirical Lower Bounds on the Complexity of Translational Equivalence », *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, Association for Computational Linguistics, p. 977–984, July 2006. 7.4.1.5, 7.7.3.3

[WU 97] WU D., « Stochastic Inversion Transduction Grammar and Bilingual Parsing of Parallel Corpora », *Computational Linguistics*, vol. 23, n°3, p. 377–404, 1997. 7.4.1.5, 7.4.1.5

[XIA 04] XIA F., MCCORD M., « Improving a Statistical MT System with Automatically Learned Rewrite Patterns », *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, p. 508–514, 2004. 7.4.1.4, 7.7.3.3

[XIO 06] XIONG D., LIU Q., LIN S., « Maximum entropy based phrase reordering model for statistical machine translation », *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Morristown, NJ, USA, p. 521–528, 2006. 7.4.2.3

[YAM 01] YAMADA K., KNIGHT K., « A Syntax-based Statistical Translation Model », *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, p. 523–530, 2001. 7.7.3.3

[YOU 67] YOUNGER D. H., « Recognition and Parsing of context-free languages in time n^3 », *Information and Control*, vol. 10, n°2, p. 189–208, 1967. 53

[ZAI 09] ZAIDAN O. F., « Z-MERT : A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems », *The Prague Bulletin of Mathematical Linguistics*, vol. 91, p. 79–88, 2009. 7.5.1.2

[ZAS 09] ZASLAVSKIY M., DYMETMAN M., CANCEDDA N., « Phrase-Based Statistical Machine Translation as a Traveling Salesman Problem », *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Suntec, Singapore, p. 333–341, 2009. 7.4.2.1, 7.5.2, 7.5.4

[ZEN 02] ZENS R., OCH F. J., NEY H., « Phrase-based statistical machine translation », JARKE M., KOEHLER J., LAKEMEYER G., Eds., *KI-2002 : Advances in artificial intelligence*, vol. 2479 de *LNAI*, Springer Verlag, p. 18-32, 2002. 7.2.3, 7.3

[ZEN 03] ZENS R., NEY H., « A Comparative Study on Reordering Constraints in Statistical Machine Translation », *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, p. 144–151, 2003. 7.4.1.5, 29

[ZEN 06] ZENS R., NEY H., « Discriminative reordering models for statistical Machine Translation », *HLT-NAACL : Proc. of the Workshop on Statistical Machine Translation*, New York, NY, p. 55–63, 2006. 7.4.2.3

[ZHA 03] ZHANG Y., VOGEL S., WAIBEL A., « Integrated Phrase Segmentation and Alignment Algorithm for Statistical Machine Translation », *Proceedings of International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE'03)*, Beijing, China, October 2003. 7.3

[ZHA 04] ZHANG H., GILDEA D., « Syntax-Based Alignment : Supervised or Unsupervised ? », *Proceedings of Coling 2004*, Geneva, Switzerland, p. 418–424, 2004. 7.7.3.3

[ZHA 06] ZHANG H., GILDEA D., « Efficient Search for Inversion Transduction Grammar », *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, Association for Computational Linguistics, p. 224–231, July 2006. 7.4.1.5

[ZOL 06] ZOLLMANN A., VENUGOPAL A., « Syntax Augmented Machine Translation via Chart Parsing », *Proceedings on the Workshop on Statistical Machine Translation*, New

York City, p. 138–141, 2006. 7.7.2