Published at RFIA 2006.

Transcription et traduction de débats parlementaires

Speech translation of parliamentary sessions

D. Déchelotte H. Schwenk J.-L. Gauvain

LIMSI/CNRS
Bâtiment 508, Université de Paris-Sud, 91403 Orsay (France)
dechelot@limsi.fr

Résumé

Cet article présente un système complet de traduction automatique de la parole non-contrainte. Une approche statistique est utilisée aussi bien pour la reconnaissance de la parole que pour la traduction. Les modèles, algorithmes et optimisations utilisés dans le système de traduction statistique sont décrits en détail. Des résultats sont présentés pour la transcription et la traduction des débats du Parlement européen, de l'anglais vers l'espagnol et inversement. Ils suggèrent que les modèles stochastiques de traduction sont adaptés à la traduction de la parole, de part leur relative robustesse constatée face aux erreurs introduites par la reconnaissance automatique.

Mots Clef

Reconnaissance de la parole continue, traduction automatique par méthodes statistiques.

Abstract

The article presents a complete system for translation of unconstrained speech. A statistical approach is used both for speech recognition and automatic translation. The models, algorithms and optimizations used in the translation system are described in detail. Results are presented for the transcription and translation of parliamentary sessions at the European Parliament, from English to Spanish and the other way around. These results suggest that stochastic translation models fit the task of speech translation, exhibiting robustness in the presence of errors introduced by automatic transcription.

Keywords

Continuous speech recognition, automatic translation by statistical methods.

1 Introduction

Le domaine de la traduction automatique de texte et celui de la reconnaissance de la parole se sont développés essentiellement en parallèle, utilisant des algorithmes et techniques propres à chaque problématique sans qu'il y ait beaucoup d'interactions. Tous les systèmes de reconnaissance de la parole à grand vocabulaire utilisent aujourd'hui des approches statistiques, notamment des modèles de Markov cachés pour la modélisation acoustique [8] et les modèles de langage n-grammes. Les paramètres de ces modèles statistiques sont entraînés sur des corpus de parole annotés de plusieurs centaines d'heures et des corpus de plusieurs centaines de millions de mots de textes représentatifs. La traduction automatique de texte, par ailleurs, étant considérée plutôt comme une tâche linguistique, d'autres approches étaient habituellement utilisées. On peut par exemple citer des systèmes basés sur un ensemble de règles de traduction [9], ou l'approche *interligua* où la phrase est d'abord traduite dans une « langue » artificielle intermédiaire [27]. Ces approches font l'objet d'un grand nombre de publications et sont à l'origine de plusieurs produits commerciaux.

Les approches statistiques de la traduction automatique [2] présentent l'avantage de n'interdire *a priori* aucun « évènement », aussi improbable soit-il. En effet, la traduction de la parole nous confronte à plusieurs problèmes difficiles à traiter avec les autres méthodes de traduction automatique. Il est notamment possible d'observer des phénomènes linguistiques dans un discours oral, surtout spontané, qui ne sont pas connus dans du texte écrit, comme les reprises, les corrections et hésitations ou même des erreurs importantes de syntaxe et sémantique. Les erreurs commises par le système de reconnaissance de la parole compliquent aussi la traduction. Finalement, une étroite intégration entre la reconnaissance et la traduction ne peut être que facilitée lorsque les mêmes approches sont utilisées dans les deux modules.

Dans cet article, un système complet de traduction de la pa-

role par approche statistique est décrit. La tâche considérée est la transcription et la traduction des débats du Parlement européen pour la paire de langues espagnol/anglais dans les deux sens. La section suivante résume les fondements mathématiques de l'approche statistique pour la traduction. Les sections 3.2 et 3.3 décrivent respectivement le système de reconnaissance de la parole continue et les détails des algorithmes de traduction. La section 4 présente finalement des résultats comparatifs.

2 L'approche statistique en traduction automatique

Il pourrait paraître surprenant au premier abord de vouloir traiter un processus linguistique comme la traduction par des méthodes statistiques. Mais la traduction d'un texte nécessite la prise de décisions, comme par exemple le choix d'un mot, d'une locution ou d'une phrase par rapport à un autre, et la prise en considération de dépendances souvent faibles ou vagues, ou qui ont un effet additif, ce dont l'approche statistique rend compte. En outre, le traitement statistique permet de garantir que pour toute phrase source, une phrase traduite sera générée (même si la syntaxe de cette phrase n'est pas correcte, elle permettra très probablement de capter le sens de la phrase originale). On peut donc résumer la traduction statistique comme la combinaison d'une modélisation linguistique et d'une *prise de décision statistique*.

2.1 Notion d'alignement

Parmi les nombreux modèles statistiques existants, la quasi-totalité d'entre eux introduisent une variable cachée $\mathcal{A}$, appelée *alignement*, qui décrit une correspondance entre les mots d'une phrase et ceux de sa traduction (parmi les traductions possibles). La figure 1 montre un exemple d'un tel alignement. Les alignements de groupes de mots à d'autres groupes de mots sont *a priori* autorisés, de même que l'alignement à un mot spécial appelé NUL utilisé lorsqu'un ou plusieurs mots d'une phrase n'ont pas de correspondance dans l'autre phrase (formellement, il y a un mot NUL dans chacune des langues).

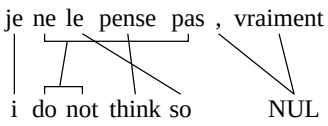

FIG. 1 – Exemple d'alignement entre deux phrases

Un modèle statistique de traduction évalue par la quantité $\Pr(\mathbf{f}|\mathbf{e})$ la probabilité qu'un traducteur humain produise la phrase $\mathbf{f} = f_1 \dots f_J$ pour traduire la phrase $\mathbf{e} = e_1 \dots e_I$. On introduit l'ensemble des alignements $\mathcal{A}$, puis en pratique seul l'alignement le plus probable est considéré :

$$\Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathcal{A}} \Pr(\mathbf{f}, \mathcal{A}|\mathbf{e}) \approx \max_{\mathcal{A}} \Pr(\mathbf{f}, \mathcal{A}|\mathbf{e}) \quad (1)$$

2.2 Traduction automatique de la parole continue

La traduction automatique de la parole continue présente plusieurs difficultés supplémentaires par rapport à la traduction automatique de texte, dont voici une liste non-exhaustive.

– La ponctuation est susceptible d'améliorer la qualité de la traduction en indiquant des sous-segments que le système peut choisir de traduire indépendamment. Cependant, un système de reconnaissance ne restitue en général pas la ponctuation dans la parole transcrite, ce qui prive la traduction automatique d'indices syntaxiques et sémantiques, et, partant, dégrade les performances.
– La segmentation de la parole continue en tours de parole, puis en phrases et éventuellement en fragments de phrase est un problème ouvert. Lorsque le locuteur fait une pause dans la phrase, le système de reconnaissance risque de reconnaître plusieurs segments et le système de traduction devra gérer des phrases incomplètes (tous les systèmes de traduction font à ce jour de la traduction hors-contexte).
– La parole est sujette aux erreurs de grammaire, aux répétitions (« je, je ne le pense pas ») et aux réparations (« il serait…, il faudrait tout d'abord… »), que le système de traduction doit traiter, ou bien en les traduisant mot-à-mot, ou bien en les détectant et en les ignorant.
– Enfin, la transcription automatique est susceptible de commettre des erreurs de reconnaissance. Ces erreurs entraînent la suppression de mots éventuellement porteurs de sens et l'insertion de mots courts et fréquents ou de mots sans lien sur le plan sémantique avec le sujet de la phrase.

La traduction de la parole continue est ainsi un problème sensiblement différent de la traduction de texte.

2.3 Le modèle IBM-4

De nombreuses décompositions de la probabilité $\Pr(\mathbf{f}, \mathcal{A}|\mathbf{e})$ en sous-modèles existent. Les modèles à base de groupes de mots [15, 23], par exemple, obtiennent à ce jour de très bons résultats. Nous avons opté pour ce travail pour le modèle « IBM-4 » [3, 17].

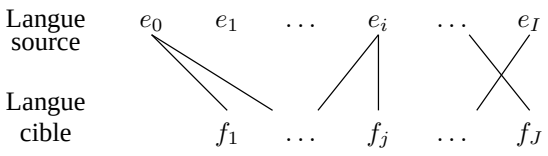

FIG. 2 – Exemple d'alignement autorisé par IBM-4

IBM-4 modélise la génération d'une phrase *cible* $\mathbf{f} = f_1 \dots f_J$ par une phrase *source* $\mathbf{e} = e_1 \dots e_I$. L'alignement $\mathcal{A}$ entre les phrases source et cible n'est pas aussi général que celui de la figure 1 ; la figure 2 donne un exemple d'alignement autorisé par IBM-4. Il est de la forme $\mathcal{A} =$

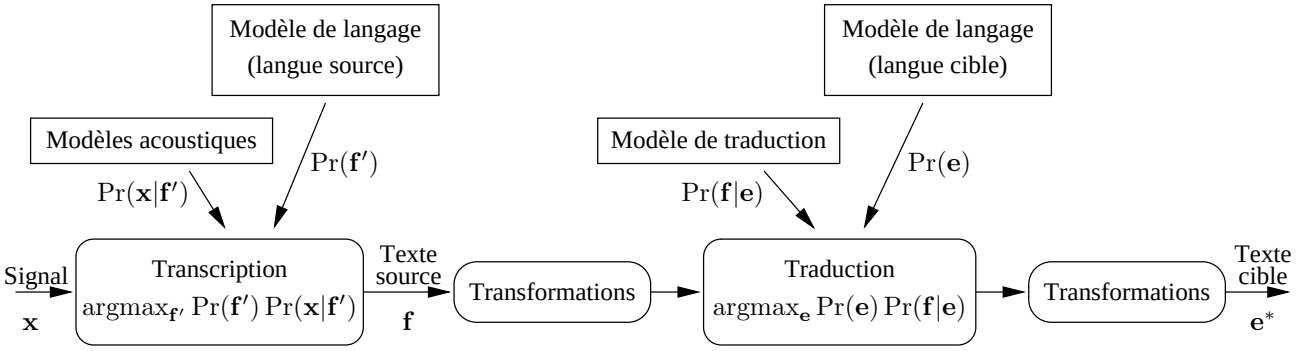

FIG. 3 – Architecture du système de traduction de la parole

$a_1 \dots a_J$, où, pour tout j de l'intervalle $[1, J]$, $a_j \in [0, I]$. $a_j = i > 0$ signifie que le mot cible f_j est aligné, ou a été produit par, e_i, tandis que $a_j = 0$ signifie que f_j est apparu « spontanément » au cours de la génération de $\mathbf{f}$. Ainsi, un alignement de cette forme autorise l'alignement de plusieurs f_j à un seul e_i, mais pas l'inverse : un mot cible f_j est aligné à 0 ou 1 mot source.

La probabilité $\Pr(\mathbf{f}, \mathcal{A}|\mathbf{e})$ introduite à l'équation 1 est ensuite décomposée en produit de quatre sous-modèles :

1. Le modèle de fertilité $n(\phi|e)$: pour chaque position source $i \in [1, I]$, ϕ_i est le nombre de mots cible alignés à e_i, soit
$$\phi_i = \text{Card}\,\{j|a_j = i\} \tag{2}$$
2. Le modèle de traduction lexicale $t(f|e)$, qui donne la probabilité que e produise f comme traduction.
3. Le modèle de distorsion, qui comprend la loi $p_{=1}(j|j')$ (pour positionner le premier mot cible généré par e_{a_j} en fonction de la position des derniers mots cibles produits) et la loi $p_{>1}(j|j')$ (utilisé lorsque $\phi_{a_j} > 1$ pour positionner les mots cible suivants générés par le même e_{a_j}).
4. Le modèle caractérisant les mots alignés à e_0, qui comprend une probabilité p_1 de génération spontanée d'un mot cible aligné à e_0 pour la génération de chaque mot cible aligné à un mot source, et la loi $t(f|e_0)$ pour déterminer quel mot cible sera ainsi généré.

IBM-4 modélise la production de la phrase cible par la phrase source en trois temps. Dans un premier temps, la fertilité ϕ_i de chaque mot source e_i est déterminée selon la loi $n(\phi|e)$. Dans un deuxième temps, les mots source e_i dont la fertilité est non nulle produisent chacun respectivement ϕ_i mots cible, dont le choix suit la loi $t(f|e)$ et qui sont réordonnés suivant les lois de distorsion $p_{=1}(j|j')$ et $p_{>1}(j|j')$. $\sum_{i=1}^{I} \phi_i$ mots cibles sont ainsi produits. Enfin dans le troisième et dernier temps, pour chaque mot cible déjà produit, une décision binaire d'introduire (ou non) un mot cible aligné à e_0 est prise, avec une probabilité p_1 d'introduire un tel mot (selon la loi $t(f|e_0)$). Ces mots « spontanés » alignés au mot NUL e_0 ne portent pas de sens ; en pratique, ils sont insérés de façon à respecter la grammaire de la langue cible.

Le lecteur est invité à se reporter à [3] pour une description détaillée du modèle ainsi que pour l'entraînement de ses paramètres. Les paramètres du système de traduction décrit dans cet article ont été entraînés à l'aide du programme libre Giza++ [15], qui met en œuvre les algorithmes de [3].

3 Description du système

3.1 Architecture générale

La figure 3 représente l'architecture du système de traduction de la parole. Reconnaissance de la parole et traduction sont effectuées séquentiellement. Un exemple de transformation intervenant entre la reconnaissance et la traduction est la reconstitution des sigles épelés : le moteur de reconnaissance produit, par exemple, « `S. N. C. F.` », qu'il faut transformer en « `SNCF` » pour que le traducteur puisse correctement le traduire. La suppression des mots incomplets et des évènements prosodiques comme les bruits de respiration sont d'autres exemples de transformations précédant la traduction.

3.2 Reconnaissance automatique de la parole continue

Le système de reconnaissance de la parole utilisé dans ce travail a la même structure de base que le système développé pour la reconnaissance d'émissions radio- et télévisées [7].

Il repose sur deux composants principaux : un segmenteur audio et un décodeur lexical. La segmentation audio est effectuée de manière itérative, par un algorithme de segmentation-agglomération utilisant des mélanges de gaussiennes. Le résultat est un ensemble de segments acoustiquement homogènes correspondant aux tours de parole des locuteurs intervenant dans le document. Le décodeur lexical utilise des modèles de Markov cachés avec des densités de probabilité continues (sommes pondérées de gaussiennes) pour les modèles acoustiques, et des statistiques n-grammes obtenues sur de grands corpus de

textes pour le modèle linguistique. Les modèles de Markov cachés représentent des allophones contextuels avec une structure gauche-droite à états liés. Ils modélisent des séquences de trames centisecondes représentées par 39 composantes : 12 coefficients cepstraux (PLP) et le logarithme de l'énergie à court-terme, avec leurs dérivées d'ordre 1 et 2.

Le décodage s'effectue en deux passes : une première passe de génération rapide d'hypothèses initiales, suivie par une adaptation non-supervisée au locuteur (MLLR contrainte et non-contrainte), et une deuxième passe qui utilisent ces modèles adaptés. Le résultat de ce processus est un treillis de mots avec les scores acoustiques et linguistiques correspondants. Ce treillis est ensuite ré-évalué par un modèle de langage neuronal [20].

3.3 Moteur de traduction

On suppose maintenant qu'il faille trouver la phrase cible $\mathbf{e}$, traduction telle que prédite par le modèle IBM-4 d'une une phrase source $\mathbf{f}$ donnée. La loi $\Pr(\mathbf{e}|\mathbf{f})$, qui représente la traduction directe de $\mathbf{f}$ vers $\mathbf{e}$, ne permet pas seule de déterminer une phrase $\mathbf{e}$ qui soit correcte syntaxiquement et grammaticalement. La relation de Bayes[1] est donc utilisée, afin de faire apparaître un modèle de langage de la langue cible $\Pr(\mathbf{e})$:

$$\underset{\mathbf{e}}{\operatorname{argmax}} \Pr(\mathbf{e}|\mathbf{f}) = \underset{\mathbf{e}}{\operatorname{argmax}} \Pr(\mathbf{e}) \Pr(\mathbf{f}|\mathbf{e}) \qquad (3)$$

Cette transformation, inspirée de la reconnaissance automatique de la parole, reprend le principe « Source/Canal de transmission » et modifie le rôle du traducteur, qui doit maintenant retrouver la phrase $\mathbf{e}$ qui a *produit* la phrase $\mathbf{f}$ observée. Notons qu'en pratique, le décodeur pondère les différentes sources d'information que sont le modèle de langage $\Pr(\mathbf{e})$ et les quatre sous-modèle de traduction qui composent le modèle de traduction $\Pr(\mathbf{f}|\mathbf{e})$ (modèle log-linéaire [16]). On a ainsi un modèle à cinq fonctions caractéristiques h_i :

$$\underset{\mathbf{e}}{\operatorname{argmax}} \Pr(\mathbf{e}|\mathbf{f}) \approx \underset{\mathbf{e}}{\operatorname{argmax}} \prod_{i=1}^{5} h_i(\mathbf{e}, \mathbf{f})^{\lambda_i} \qquad (4)$$

où $h_1(\mathbf{e}, \mathbf{f}) = \Pr(\mathbf{e})$ est le modèle de langage cible et $h_2 \ldots h_5$ les sous-modèles de fertilité, traduction lexicale, distorsion et de génération spontanée détaillés à la section 2.3. Les λ_i sont optimisés sur l'ensemble de développement.

Pour retrouver la phrase $\mathbf{e}$ qui a produit la phrase $\mathbf{f}$, le traducteur va progressivement ajouter des mots cibles e_i qui vont chacun « expliquer », ou *couvrir*, un ou plusieurs mots f_j de la phrase source à traduire. L'objectif du décodeur est de progressivement couvrir toutes les positions source. Les positions source sont couvertes dans n'importe quel ordre, tandis que les mots cible sont ajoutés dans l'ordre de lecture, de gauche à droite.

Dans la suite de cet article, le *coût* d'une décision est égal à l'opposé du logarithme de la probabilité de cette décision. À titre d'exemple, le coût de la traduction du mot e en f est $-\log(t(f|e))$.

Stratégie de recherche. La recherche de la meilleure traduction $\mathbf{e}^*$ parmi l'ensemble des phrases cible $\mathbf{e}$ est réalisée en suivant une stratégie de type « A* ». Le décodeur maintient une liste d'hypothèses partielles qui évolue comme suit [25] :

1. La liste d'hypothèses partielles est initialisée avec une hypothèse vide (aucune position source couverte, aucun mot cible produit).
2. Le décodeur sélectionne la meilleure hypothèse partielle et l'enlève de la liste.
3. Si cette hypothèse couvre toutes les positions source, il s'agit de la meilleure traduction complète et l'algorithme est terminé.
4. Sinon, le décodeur étend l'hypothèse en couvrant une position source de plus. Ce faisant, il génère autant de nouvelles hypothèses partielles qu'il y a de mots non encore traduits et qu'il y a de traductions possibles pour ces mots.
5. Les nouvelles hypothèses sont incorporées à la liste et triées par coût croissant (c'est-à-dire probabilité décroissante), puis l'on revient à l'étape 2.

Une hypothèse contient juste l'information nécessaire pour l'étendre et de quoi reconstruire la phrase finale lorsque le décodage est terminé. Ainsi, elle ne contient que les informations suivantes :
– un pointeur arrière sur l'hypothèse qu'elle étend ;
– les deux derniers mots cible produits, car le modèle de langage cible est trigramme. Les différents sous-modèles de traduction cités à la section 2.3 nécessitent les informations suivantes :
– la liste des mots cible produits à la dernière extension ;
– la dernière position source couverte ;
– le nombre de mots source actuellement alignés au dernier mot cible produit ;
– la somme des positions de ces mots source (nécessaire pour le calcul de $p_{=1}(j|j')$).

Étendre une hypothèse partielle signifie couvrir une position source supplémentaire. Ceci est généralement accompli en ajoutant un mot cible supplémentaire, mais peut éventuellement ne produire aucun mot cible, ou en produire plusieurs. Le décodeur dispose des quatre « opérateurs » suivants pour étendre une hypothèse :
– l'opérateur *Ajout* qui traduit un mot source supplémentaire un ajoutant un mot cible ;
– l'opérateur *NZFertAjout* qui ajoute un ou plusieurs mots cible de fertilité nulle (alignés à aucun mot source et

[1] Dans l'équation 3, le terme $\Pr(\mathbf{f})$ au dénominateur est supprimé car il n'a pas d'influence sur l'opération $\operatorname{argmax}_{\mathbf{e}}$.

donc ne couvrant aucune position source) avant d'ajouter un mot cible qui traduise un mot source supplémentaire ;
– l'opérateur *Extension* qui aligne un mot source supplémentaire au dernier mot cible produit : aucun mot cible n'est donc ajouté avec cet opérateur mais une position source est bien couverte ;
– l'opérateur *CompletementNul* est particulier : il complète l'hypothèse partielle en alignant toutes les positions source non encore couvertes au mot cible NUL (aucun mot cible produit).

Les opérateurs *Ajout* et *NZFertAjout* peuvent être utilisés pour couvrir n'importe quelle position source non-couverte, au contraire de l'opérateur *Extension* qui ne peut couvrir qu'un mot source à droite du dernier mot source couvert. L'opérateur *CompletementNul* ne peut être utilisé que si au moins la moitié des positions source est déjà couverte (en effet, IBM-4 implique que $\phi_0 \leq \sum_{i=1}^{I} \phi_i = J - \phi_0$).

Organisation des hypothèses en plusieurs files. Un décodeur de type « A* » peut regrouper les hypothèses partielles en une ou plusieurs files. N'utiliser qu'une seule file a certes l'avantage d'une gestion plus simple des hypothèses mais présente le risque de faire des erreurs de décodage, c'est-à-dire de produire une phrase $\mathbf{e}$ qui ne soit pas $\mathbf{e}^*$. En effet, l'optimalité de la recherche n'est garantie qu'avec une capacité de stockage et un temps de calcul infinis, or en pratique, un élagage sur le nombre maximal d'hypothèses partielles traitées ou en attente est nécessaire. En mettant en compétition dans une même file des hypothèses qui ont traduit un nombre différent de mots, un décodeur à file unique prend le risque que de nombreuses hypothèses peu prometteuses n'ayant traduit que peu de mots finissent par provoquer l'élagage de bonnes hypothèses partielles qui ont déjà traduit de nombreux mots.

C'est pourquoi le traducteur décrit dans cet article utilise plusieurs files et plus exactement une par sous-ensemble de positions source, soit 2^J files, où J est le nombre de mots dans la phrase source. Ce nombre élevé de files (et augmentant de façon exponentielle avec le nombre de mots à traduire) permet de réduire leur taille et de ne mettre en compétition que des hypothèses partielles qui ont couvert exactement les mêmes positions source. Limiter la taille des files à une valeur finie revient à faire un élagage de l'espace de recherche. D'après nos expériences, une taille de file de 20 hypothèses partielles permet d'obtenir des résultats marginalement moins bons que pour une taille beaucoup plus importante (1000), alors que le temps d'exécution augmente de façon quasi-linéaire avec cette taille. La figure 4 illustre le mécanisme des files d'hypothèses partielles dans le décodeur.

Heuristiques. Disposer de plusieurs files d'hypothèses complique la recherche de la meilleure hypothèse partielle à étendre à chaque itération. Il est possible de choisir l'hypothèse de moindre coût parmi toutes les hypothèses partielles de toutes les files. De cette façon, une mauvaise hypothèse H_{mvs} n'ayant couvert qu'une position source sera étendue avant une bonne hypothèse ayant couvert de nombreuses positions. Toutefois, à l'inverse d'un décodeur à file unique, les hypothèses produites par H_{mvs} ne pourront provoquer l'élagage des bonnes hypothèses, puisque les multiples files assurent une comparaison juste entre les hypothèses. Cette gestion, qui n'utilise aucune heuristique d'estimation du coût futur, correspond à l'algorithme de Dijkstra [4]. En pratique, elle conduit à un décodage lent et parfois sous-optimal, malgré les files multiples.

Une heuristique donne une estimation du coût futur de complètement d'une hypothèse, et une heuristique *admissible* donne toujours un *minorant* de ce coût futur. L'optimalité d'une recherche de type « A* » n'étant assurée que si les heuristiques sont admissibles, une telle heuristique a été développée en s'inspirant de [14].

Une heuristique dépendant uniquement de l'ensemble des positions source non couvertes a été choisie car elle est ainsi particulièrement adaptée à une organisation des hypothèses en files indicées par le sous-ensemble des positions source couvertes. En effet, toutes les hypothèses partielles d'une même file vont partager le même coût futur estimé, calculé une seule fois à la création de la file, et ce coût va permettre de mieux comparer les files d'hypothèses entre elles, en favorisant les files qui ont traduit plus de mots.

L'approche utilisée consiste à évaluer un coût minimal de couverture de chaque position source, noté $h_{\text{opt}}(j)$, et d'obtenir le coût heuristique pour compléter une hypothèse $H_{\text{partielle}}$ en sommant sur les positions non-couvertes :

$$h_{\text{opt}}(H_{\text{partielle}}) = \sum_{j \text{ non couvert par } H_{\text{partielle}}} h_{\text{opt}}(j) \qquad (5)$$

Plusieurs composantes ont été successivement ajoutées à $h_{\text{opt}}(j)$. La première et la plus naturelle prend simplement en compte la probabilité de traduction $t(f|e)$; elle est notée $h^T(j)$:

$$h^T(j) = \min_e -\log t(f_j|e) = -\log \max_e t(f_j|e) \qquad (6)$$

Il est possible d'y incorporer la loi de fertilité $n(\phi|e)$ lorsque le mot e qui produit f n'est pas e_0 ; la nouvelle heuristique est notée $h^{TF}(j)$:

$$h^{TF}(j) = -\log \max \left\{ t(f_j|e_0), \max_{e \neq e_0, \phi} t(f_j|e) \sqrt[\phi]{n(\phi|e)} \right\} \qquad (7)$$

La racine $\phi^{\text{è}}$ est nécessaire pour ne pas comptabiliser plusieurs fois la probabilité $n(\phi|e)$ dans le cas où plusieurs mots source f_j sont alignés au même mot cible e.

Enfin, le modèle de distorsion est pris en compte en ajoutant le terme suivant, lui indépendant de la position j :

$$h^D = -\log \max \left\{ \max_{j,j'} p_{=1}(j|j'), \max_{j>j'} p_{>1}(j|j') \right\} \qquad (8)$$

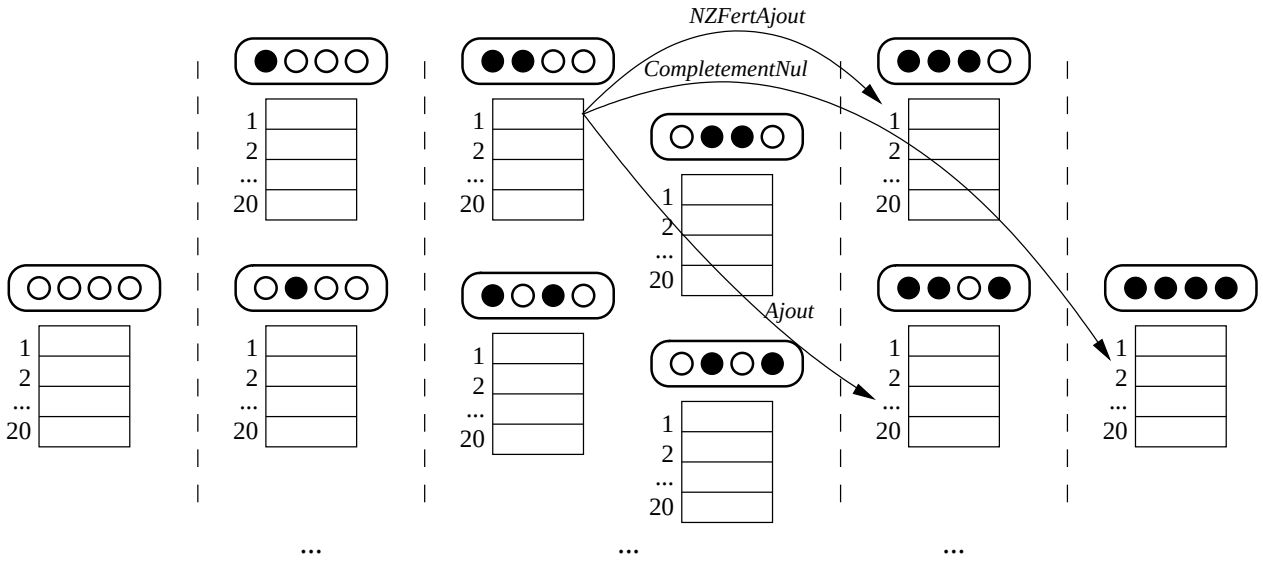

FIG. 4 – Les files d'hypothèses partielles dans le traducteur, pour une phrase source de quatre mots. La figure montre un exemple d'expansion de la meilleure hypothèse de la file `1100`. Les opérateurs *Ajout* et *NZFertAjout* ont été utilisés pour produire des hypothèses dans les files `1110` et `1101`. De plus, l'opérateur *CompletementNul* a pu être utilisé pour couvrir la totalité des positions source vacantes avec le mot spécial NUL, sans produire de « vrai » mot cible.

Compte tenu de la forme de la loi $p_{=1}$ (qui dépend de la différence $j - j'$, où j' est une moyenne d'indices de mots couverts), il est difficile sinon impossible de faire mieux que cette heuristique très générale pour estimer le coût de distorsion.

L'heuristique finalement retenue vaut :

$$h_{\text{opt}}(j) = h^{TF}(j) + h^{D} \tag{9}$$

et est calculée en début de décodage pour chacune des J positions source. Le gain de temps pour le décodage compense largement le temps passé à évaluer ces quantités. En revanche, faute de moyen efficace pour calculer pour chaque mot cible e son plus faible coût de modèle de langage $\max_{e',e''} \Pr(e|e'',e')$, celui-ci n'est pas inclus dans l'heuristique retenue.

4 Évaluation expérimentale

4.1 Score BLEU

L'évaluation automatique de la qualité de la traduction reste un problème ouvert. L'évaluation manuelle de la qualité d'une traduction est coûteuse, car elle nécessite de demander à des personnes bilingues et familières du domaine de la tâche d'évaluer la qualité de nombreuses traductions selon plusieurs critères (qualité de la langue, fidélité de la traduction, quantité d'information transmise). Le score BLEU [18] a été conçu pour permettre une évaluation rapide et automatique de la sortie des systèmes de traduction. Papineni *et al.* montre en effet que le score BLEU est fortement corrélé avec le jugement humain. BLEU est aujourd'hui, avec le score NIST [13], le score de référence utilisé dans la communauté de la traduction automatique par méthodes statistiques.

Pour évaluer la qualité d'une traduction candidate $\mathcal{C}$, le score BLEU utilise des précisions de n-grammes notées p_n, évaluées à partir d'une ou de plusieurs traductions de référence. p_n est le rapport du nombre de n-grammes de $\mathcal{C}$ qui se trouvent dans une des traductions de référence sur le nombre de n-grammes de $\mathcal{C}$. On a donc :

$$p_n = \frac{\sum_{n\text{-gramme}\in\mathcal{C}} \textit{Compte}_{\text{ref}}(n\textit{-gramme})}{\sum_{n\text{-gramme}\in\mathcal{C}} \textit{Compte}(n\textit{-gramme})} \tag{10}$$

$\textit{Compte}_{\text{ref}}(n\textit{-gramme})$ compte le nombre maximum de fois que le n-gramme apparaît dans au moins une des références, sans bien sûr dépasser $\textit{Compte}(n\textit{-gramme})$, c'est-à-dire le nombre de fois que le n-gramme apparaît dans la phrase candidate. Signalons que le dénominateur vaut $|\mathcal{C}| - n + 1$, où $|\mathcal{C}|$ est la taille de $\mathcal{C}$.

La moyenne géométrique des précisions pour les unigrammes, bigrammes, trigrammes et quadrigrammes est calculée. Cette moyenne est finalement multipliée par une pénalité de *brièveté* PB de façon à éviter que les systèmes augmentent artificiellement leurs taux de précision en produisant des phrases délibéremment trop courtes. Si $|\mathcal{R}|$ est, parmi les références, la taille de la référence la plus proche de $|\mathcal{C}|$, la pénalité de brièveté vaut 1 si $|\mathcal{C}| \geq |\mathcal{R}|$, et $e^{(1-|\mathcal{R}|/|\mathcal{C}|)}$ sinon. Le score BLEU vaut alors :

$$\text{BLEU} = \text{PB} \cdot \exp\left(\sum_{n=1}^{4} \frac{1}{4} \log p_n\right) \tag{11}$$

Le score BLEU évolue donc entre 0 et 1, et la traduction est d'autant meilleure que son score BLEU est élevé. En

pratique, ce sera toujours le score BLEU multiplié par 100 (pour « % BLEU ») qui sera donné.

4.2 Corpus

		Espagnol	Anglais
Entr. (TF)	Paires de phrases	1 207 740	
	Nombre de mots	31 360 260	30 049 355
	Vocabulaire	139 587	93 995
	Singletons	48 631	33 891
Dev. (Verb)	Phrases	2 643	1 750
	Mots	20 289	23 407
	Vocabulaire	2 932	2 566
	Mots hors-voc	46	59
Test (Verb)	Phrases	1 073	792
	Mots	18 896	19 306
	Vocabulaire	3 302	2 772
	Mots hors-voc	145	44

FIG. 5 – Statistiques sur les corpus d'entraînement, de développement et de test. « Verb » signifie « Verbatim », transcription manuelle et exacte du signal ; « TF » signifie « Texte Final », version ponctuée et corrigée des hésitations, reprises, agrammaticalités, etc.

Ce travail a été réalisé dans le cadre du projet européen TC-STAR [21]. Ce projet, financé par la Commission Européenne, a pour objectif d'améliorer significativement la traduction automatique de la parole et de réduire l'écart de performance entre la traduction manuelle et la traduction automatique. La tâche considérée est la parole conversationnelle non contrainte dans les trois langues suivantes : l'anglais européen, l'espagnol européen et le mandarin (chinois).

La paire de langues anglais/espagnol a été retenue pour ce travail et les données utilisées sont les transcriptions des débats des députés au parlement européen (données « EPPS » [22, 10]). Il s'agit des transcriptions des députés s'exprimant dans leur langue natale et des interprètes traduisant dans chacune des langues officielles.

Le tableau 5 rassemble des informations factuelles sur le corpus utilisé, à savoir les données d'entraînement, de développement et d'évaluation de la première campagne d'évaluation [12] du projet TC-STAR.

4.3 Résultats

Après avoir adapté les coefficients des modèles de langage et de traduction (équation 4) sur l'ensemble de développement, nous avons traduit d'une part la transcription manuelle, d'autre part la transcription automatique, et ce pour les deux sens de traduction. Les résultats sont portés dans le tableau 6.

	Scores BLEU (%)		WER
	Verbatim	ASR	(ASR)
Anglais → espagnol	29,6	27,5	10,6%
Espagnol → anglais	34,8	31,0	11,6%

FIG. 6 – Résultats sur l'ensemble de test. Les scores BLEU obtenus en traduisant la transcription manuelle (Verbatim) et automatique (ASR) sont donnés dans les deux sens de traduction, ainsi que le taux d'erreur par mot (WER) de la transcription automatique.

On observe que les performances sont meilleures pour la traduction vers l'anglais que pour la traduction vers l'espagnol. Ceci s'explique par la plus grande variété morphologique de l'espagnol, qu'il est difficile de produire avec exactitude en partant de l'anglais.

Par ailleurs, et conformément aux attentes, la présence d'erreurs de reconnaissance dégrade les performances de la traduction. Toutefois, cette dégradation n'est pas un effondrement. En effet, dans le sens « espagnol → anglais », le score BLEU passe de 34,8 à 31,0, soit une diminution relative d'un peu moins de 11%. Pour le sens « anglais → espagnol », la diminution du score BLEU de 29,6 à 27,5 représente une perte relative d'un peu plus de 7%. Une comparaison directe du taux d'erreur par mot de la reconnaissance automatique et de la baisse relative du score BLEU n'est pas possible, puisque BLEU est un score qui aggrège plusieurs calculs de précisions, et que le processus même de traduction interdit toute transposition simpliste du taux d'erreur dans une langue à une autre langue. Cependant, on peut retenir que la chute du score BLEU est du même ordre que le taux WER, et même légèrement moindre en pourcentage, ce qui est un résultat encourageant pour cette approche (statistique) de la traduction automatique.

5 Conclusion

Dans cet article, le travail mené dans le cadre du projet TC-STAR en vue d'améliorer la traduction automatique de la parole non-contrainte a été décrit. Le module de reconnaissance de la parole et celui de traduction automatique utilisent tout deux des approches statistiques. Alors que l'utilité de cette approche n'est guère contestée pour la reconnaissance de la parole, elle reste à étayer pour la traduction automatique. Les résultats obtenus à l'issu de ce travail suggèrent que les modèles stochastiques de traduction se comportent de façon plutôt satisfaisante face aux erreurs introduites par la reconnaissance automatique, puisque la dégradation relative de leurs performances est inférieure au taux d'erreur par mot du système de reconnaissance.

Des conclusions similaires ont été obtenues dans le cadre d'autres projets de recherche sur la traduction automatique de la parole, bien que des tâches plus limitées aient été considérées. Des exemples sont le projet allemand Verbmobil [24] (planification de rendez-vous et de voyages tou-

ristiques), ou les évaluations IWSLT [1, 6] (phrases typiques rencontrés dans des guides touristiques).

Les travaux futurs poursuivront donc l'approche statistique pour le système de traduction et s'attacheront en particulier à utiliser une interface plus riche avec le système de reconnaissance, par exemple en utilisant ses n meilleures hypothèses : [26, 19, 11, 5] montrent qu'il est ainsi possible d'améliorer la traduction. De façon complémentaire, le décodeur de traduction a été modifié de façon à produire un treillis de mots et des travaux sont en cours afin de rescorer les treillis produits avec des modèles de traduction et de langage différents. À titre d'exemple, un gain d'environ 1 point BLEU a été observé sur les données de développement en rescorant les treillis de traduction (obtenus avec un décodage trigramme) avec un modèle de langage cible quadrigramme, alors que le décodage quadrigramme direct aurait présenté un coût prohibitif en terme de temps de calcul.

Remerciements

Les auteurs remercient G. Adda, E. Bilinski, O. Galibert and L. Lamel pour leur participation dans le développement des systèmes de reconnaissance sur lesquels ce travail est basé.

Ce travail a été partiellement financé par l'Union Européenne, via la subvention FP6-506738 (Projet TC-STAR, `http://www.tc-star.org/`).

Références

[1] Yasuhiro Akiba, Marcello Federico, Noriko Kando, Hiromi Nakaiwa, Michael Paul, and Jun'ichi Tsujii. Overview of the iwslt04 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 1–12, 2004.

[2] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Frederik Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2) :79–85, 1990.

[3] Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation : Parameter estimation. *Computational Linguistics*, 19(2) :263–311, 1993.

[4] Edsger. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1 :269–271, 1959.

[5] Daniel Déchelotte, Holger Schwenk, Jean-Luc Gauvain, Olivier Galibert, and Lori Lamel. Investigating translation of parliament speeches. In *Proceedings of ASRU 2005*, Décembre 2005.

[6] Matthias Eck and Chiori Hori. Overview of the IWSLT 2005 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, 2005.

[7] Jean-Luc Gauvain, Lori Lamel, and Gilles Adda. The LIMSI broadcast news transcription system. *Speech Communication*, 37(2) :98–108, 2002.

[8] J.L. Gauvain and L.F. Lamel. Systèmes de reconnaissance, de compréhension et de dialogue. In J. Mariani, editor, *Reconnaissance de la parole Traitement automatique du langage parlé*, volume 2, pages 47–83. Hermes Lavoisier, 2002.

[9] H. Kaji. An efficient execution method for rule-based machine translation. In *COLING-88 : Proceedings of the 12th International Conference on Computational Linguistics*, volume 2, pages 824–829, Budapest, Hungary, 1988.

[10] Philipp Koehn. Europarl : A multilingual corpus for evaluation of machine translation. Non publié.

[11] E. Matusov, S. Kanthak, and Hermann Ney. On the integration of speech recognition and statistical machine translation. In *Proceedings of Interspeech 2005*, pages 3177–3180, 2005.

[12] H. Ney, V. Steinbiss, R. Zens, E. Matusov, J. Gonzalez, Y.-S. Lee, S. Roukos, M. Federico, M. Kolss, and R. Banchs. Slt progress report. In *TC-STAR deliverable D5*, 2005.

[13] NIST. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. `http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf`.

[14] Franz J. Och, Nicola Ueffing, and Hermann Ney. An efficient a* search algorithm for statistical machine translation. In *Data-Driven Machine Translation Workshop*, pages 55–62, Toulouse, France, July 2001.

[15] Franz Josef Och and Hermann Ney. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong, China, October 2000.

[16] Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 295–302, University of Pennsylvania, 2002.

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

[18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, University of Pennsylvania, 2002.

[19] H.V. Quan, M. Federico, and M. Cettolo. Integrated n-best re-ranking for spoken language translation. In *Proceedings of Interspeech 2005*, pages 3181–3184, 2005.

[20] Holger Schwenk and Jean-Luc Gauvain. Building continuous space language models for transcribing european languages. In *Eurospeech*, pages 737–740, 2005.

[21] TC-STAR. Technology and corpora for speech to speech translation. `http://www.tc-star.org/`.

[22] EUROPARL. Site du parlement européen. `http://www.europarl.eu.int/`.

[23] S. Vogel, Y. Zhang, F. Huang, A. Venugopal, B. Zhao, A. Tribble, M. Eck, and A. Waibel. The cmu statistical machine translation system. In *Proceedings of Machine Translation Summit IX*, 2003.

[24] W. Wahlster. *Verbmobil : Foundations of Speech-to-Speech Translation*. Springer verlag, 2000.

[25] Ye-Yi Wang and Alex Waibel. Decoding algorithm in statistical machine translation. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 366–372, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

[26] Ruiqiang Zhang, Genichiro Kikui, Hirofumi Yamamoto, Tar Watanabe, Frank Soong, and Wai Kit Lo. A unified approach in speech-to-speech translation : Integrating features of speech recognition and machine translation. In *Cooling*, 2004.

[27] B. Zhou, Y. Gao, J. Sorensen, Z. Diao, and M. Picheny. Statistical natural language generation for speech-to-speech machine translation. In *Proceedings of ICSLP-2002*, 2002.