REPUBLIQUE TUNISIENNE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DE SFAX

FACULTE DES SCIENCES ECONOMIQUES ET DE GESTION DE SFAX

THÈSE

POUR L'OBTENTION DU GRADE DE DOCTEUR EN INFORMATIQUE

Présentée et soutenue publiquement le 10/03/2017 par

ACHRAF OTHMAN

TRADUCTION AUTOMATIQUE EN LANGUE DES SIGNES BASÉE SUR UNE APPROCHE STATISTIQUE

Membres du JURY

Mr. Faiez GARGOURI	**Prof. d'Ens. Sup.**	**Univ. de Sfax**	**Président**
Mr. Mohamed JEMNI	**Prof. d'Ens. Sup.**	**Univ. de Tunis**	**Directeur de thèse**
Mr. Mounir ZRIGUI	**Maître de Conf.**	**Univ. de Monastir**	**Rapporteur**
Mme. Chiraz LATIRI	**Maître de Conf.**	**Univ. de Manouba**	**Rapporteur**
Mr. Kais HADDAR	**Maître de Conf.**	**Univ. de Sfax**	**Examinateur**

ANNÉE UNIVERSITAIRE 2016/2017

Remerciements

Merci ...

À Dieu

À mes parents

À mes sœurs

À mon épouse et ma petite Maram

À toute la grande famille

À mes chers amis

pour leurs soutiens et amour

Merci ...

À mon directeur de thèse Prof. Mohamed Jemni

pour son encadrement et son encouragement infini

Merci

À mes collègues du lab. LaTICE

À mes collègues de l'ENSIT (ex ESSTT)

À mes collègues de l'ISMAIK

pour leurs discussions

Merci

Aux honorables membres de jury

d'avoir accepté d'évaluer mon travail

Table des matières

Introduction Générale 1
Chapitre 1 : État de l'art 4
1. Introduction 4
2. Les langues des signes 4
2.1. Les gestes dans les langues des signes 4
2.2. Composantes de la langue des signes 5
2.3. Les différentes classes de signes 6
3. Les théories linguistiques 7
3.1. Espace de signation 7
3.2. Principe d'iconicité 7
3.3. Les transferts de grande iconicité 8
4. Les recherches sur les langues des signes 8
5. Les ressources linguistiques en SL 9
5.1. La langue des signes Américaine 10
5.2. La langue des signes Allemande 10
5.3. La langue des signes Arabe-Unifiée 11
5.4. La langue des signes Britannique 11
5.5. La langue des signes Coréenne 11
5.6. La langue des signes Chinoise et Taïwanaise 12
5.7. La langue des signes Espagnole 12
5.8. La langue des signes Française 12
5.9. La langue des signes Irlandaise 13
5.10. Récapitulatif et problématiques 13
6. Traitement automatique des langues des signes 14
6.1. Interprétation 14
6.2. Synthèse 3D 14
6.2.1. Le projet "ViSiCAST" 15
6.2.2. Le projet "eSIGN" 15
6.2.3. Le projet "TESSA" 16
6.2.4. Le projet de "Matt Huenerfauth" 16
6.2.5. Le projet "Vcom3D" 17
6.2.6. Le projet "WebSign" 18
6.3. Discussion et problématiques 18
6.4. Traduction automatique 18
6.4.1. Le système de traduction "Albuquerque Weather" 19
6.4.2. Le système de traduction "TEAM" 19
6.4.3. Le système de traduction du projet "ViSiCAST" 20
6.4.4. Le système de traduction "ASL Workbench" 20
6.4.5. Le système de traduction "SASL" 21
6.4.6. Le système de traduction "Spanish SL" 22
6.4.7. Le système de traduction "MaTrEx" 22
6.5. Discussion et problématiques générales 23
7. Conclusion 23
Chapitre 2 : Systèmes de transcription des langues des signes 24
1. Introduction 24
2. Problématiques 24
3. Les conflits et ses origines 25
3.1. L'universalité 26
3.2. Les langues vocales 26
3.3. L'iconicité 27
3.4. Les standards d'écriture 27
4. Structure de la langue des signes 28
4.1. Configuration 28
4.2. Orientation 29
4.3. Emplacement 29

4.4. Mouvement 29
4.5. Gestes non manuels 30
5. Les systèmes de transcription des langues des signes 30
5.1. Un système de notation 32
5.2. Un système d'annotation 32
5.3. Système de notation de "Stokoe" 33
5.4. Système de notation "SignWriting" 33
5.5. Système de notation "HamNoSys" 35
5.6. Système d'annotation « SiGML » 36
5.7. Système de notation "Movement-Hold" 36
5.8. Système d'annotation « SML » 37
5.9. Système d'annotation textuelle "Gloss" 38
6. Discussion et problématiques 39
7. Motivation et contributions 40
8. Visualisation de la langue des signes 42
8.1. Visualisation à base de vidéo 42
8.1.1. Avantages 42
8.1.2. Inconvénients 42
8.2. Visualisation via des agents conversationnels 43
8.2.1. Avantages 43
8.2.2. Inconvénients 43
9. Conclusion 44
Chapitre 3 : Modélisation d'un système de transcription en XML pour l'ASL 45
1. Introduction 45
2. Les gloses dans la langue du milieu 45
3. La représentation en XML 46
4. Convention d'annotation en XML 46
4.1. Structure de base 48
4.2. Les mots composés 51
4.3. Les nombres 52
4.4. Les pronoms personnels 52
4.5. Préfixes et suffixes 54
4.6. Aspects 54
4.7. Réduplication 55
4.8. Verbes multiples et exhaustives 55
4.9. Les expressions faciales et les gestes non-manuels 55
4.10. Les classificateurs 56
4.11. Exemples de phrases complètes 58
5. Schéma de validation et visualisation 59
5.1. Validation 59
5.2. Représentation pour le web 59
6. Discussion et perspectives 60
7. API de création de XML-Gloss 60
7.1. Description JDOM 60
7.2. API d'annotation de gloses 61
8. Evaluation 62
8.1. Exemple 1 63
8.2. Exemple 2 64
9. Conclusion 65
Chapitre 4 : Génération de discours en ASL à partir des règles de dépendances 66
1. Introduction 66
2. Motivation et contributions 66
3. Eléments de grammaire 66
3.1. Noyau verbal 67
3.2. Classificateurs 68
3.3. Pronoms 69
3.4. Le temps 70
3.5. Construction d'un énoncé 70
3.6. Types de phrases 71
3.7. Négation 71

3.8. Coordination et subordination 72
3.9. Les émotions 73
4. Discussions 73
5. Description de l'approche proposée 74
5.1. Architecture du système proposé 76
5.2. Aperçu du fonctionnement du système 77
6. Implémentation 78
6.1. Prétraitement et analyse lexicale 78
6.2. Analyse grammaticale 79
6.3. Analyse syntaxique 80
6.4. Analyse de dépendances 83
6.5. Matrice d'adjacence 84
6.6. Lemmatisation et mise en forme 86
6.7. Reconnaissance des entités nommées 86
6.8. Résolution des coréférences 86
6.9. Génération de la transcription XML-Gloss 87
7. Evaluation 88
8. Conclusion 88
Chapitre 5 : Architecture du système de traduction automatique statistique ASL-Anglais 89
1. Introduction 89
2. Principes de la traduction automatique statistique 89
2.1. Principes de la traduction automatique statistique 89
2.2. Traduction automatique sous-phrastique 90
2.3. Implémentation 94
3. Architecture du traducteur automatique statistiques 94
4. Discussions, contributions et démarche 98
5. Génération d'un corpus parallèle Anglais-ASL 98
6. Implémentation et expérimentation 100
7. Evaluation 101
7.1. Score BLEU 102
7.2. Score METEOR 102
7.3. Score F-Mesure 103
7.4. Score NIST 103
8. Conclusion et discussions 104
Chapitre 6 : Construction d'une mémoire de traductions lexicales "Anglais-ASL" 105
1. Introduction 105
2. Génération d'une mémoire de traduction lexicale 105
2.1. Définition 105
2.2. Traduction lexicale 106
2.3. Estimation de la distribution probabiliste 106
2.4. Alignement 107
3. Algorithmes d'apprentissage et implémentation 109
3.1. Modèle d'alignement IBM 1 109
3.2. Détermination de l'incertitude 116
3.3. Application de l'algorithme EM sur le modèle IBM 2 117
3.4. Application de l'algorithme EM sur le modèle IBM 3 120
3.5. Les modèles IBM 4 et 5 122
4. Alignement 122
5. Evaluation 123
6. Conclusion 124
Chapitre 7 : Traduction automatique statistique Anglais-ASL 125
1. Introduction 125
2. Mémoire de traduction à partir des segments 125
2.1. Modèle Standard 125
2.2. Formulation mathématique 126
2.3. Approche 128
2.4. Algorithme d'extraction 129
2.5. Phase d'apprentissage 130
3. Modélisation du langage 130

4. Décodage 132
4.1. Approche 132
4.2. Exemple 132
5. Evaluation 134
6. Conclusion 135
Conclusion Générale 136
Bibliographie 138
Annexe A : Configuration des mains en ASL 1
Annexe B : Schéma XSD du modèle d'annotation en gloses 2
Annexe C : Feuille de styles XSLT du système de transcription en gloses 4
Annexe D : Liste des relations de dépendances grammaticales 8
Annexe E : Hiérarchie des relations de dépendances grammaticales 12
Annexe F : Calcul de la distance de Jaro-Winkler 14

Table des Figures

Figure 1. Paramètres de formation d'un signe 5
Figure 2. Description paramétrique du signe "CAN" 6
Figure 3. Représentation schématique de l'espace de signation 7
Figure 4. Interface d'annotation et de consultation du "ASL Lexicon Video Dataset ASLLVD" 10
Figure 5. Exemples d'agents conversationnels pour la synthèse en langues des signes 14
Figure 6. (a) Phase d'enregistrement (b) Visualisation des données générés par les capteurs de mouvements en 3D 17
Figure 7. Architecture du système de traduction « VisiCAST » 20
Figure 8. Architecture du système de traduction "Spanish SL" 22
Figure 9. Signe "STUDENT" dans les langues des signes Américaines, Italiennes et Thaïlandaise 26
Figure 10. Le "fingerspelling" de la phrase "I hope you have a great day" en ASL 28
Figure 11. L'emplacement des mains dans l'espace du signeur 29
Figure 12. Exemple de gestes non-manuels dans les signes "SICK", "HAPPY" et "NOT-LIKE" 30
Figure 13. Représentation d'une phrase en utilisant la notation HamNoSys 32
Figure 14. Notation du signe "DONT-KNOW" en "Stokoe" 33
Figure 15. Notation du signe "DEAF" en "SignWriting" 34
Figure 16. Le fichier SWML pour le signe "DRINK" 34
Figure 17. Notation du signe "HISTORY" en HamNoSys 35
Figure 18. Notation du signe "GOING-TO" en SiGML et en HamNoSys 36
Figure 19. Notation du signe "WEEK" en MH Model 37
Figure 20. Rotation d'articulation du coude en utilisant l'annotation SML 38
Figure 21. Architecture d'un système de traduction vers ou à partir des langues des signes 41
Figure 22. Transcription en gloss de la phrase "His aunt lived in Turkey. There had been no contact with the aunt. She died and left something to him in her will" 47
Figure 23. Transcription en ASL de la phrase "*His aunt lived in Turkey. There had been no contact with the aunt. She died and left something to him in her will*" 48
Figure 24. Diagramme de classes du modèle de transcription en gloses 49
Figure 25. Le signe composé "SAME-AS" (à gauche) et le signe "THANK-YOU" (à droite) en ASL 51
Figure 26. Exemples de signes en ASL en utilisant le préfixe "TWO" 52
Figure 27. Le signe "PRO-1" et "PRO" en ASL 52
Figure 28. Les deux signes {AGE0}-FIFTEEN et {AGE0}-FOUR en ASL 54
Figure 29. Le suffixe "ER" en ASL encadré en rouge lors de la signation des deux signes "TEACHER" et "DRIVER" 54
Figure 30. Les deux signes "MONDAY" et "MONDAY[WEEKLY]" en ASL 55
Figure 31. Les trois façons de signer le mot "GIVE" 55
Figure 32. Transcription de la phrase "that boy, the girl pushed" en gloses 56
Figure 33. Rendu d'une transcription en gloses en utilisant une feuille de style XSLT 59
Figure 34. Architecture d'un système de traduction automatique vers la langue des signes 60
Figure 35. Transcription XML-Gloss de la phrase "I don't like chocolate" 63
Figure 36. Transcription en XML-Gloss de la phrase "i have going to the conference for five years" 65
Figure 37. Deux formes du verbe "ASK" en ASL. Celui à gauche correspond à la phrase "I ASK YOU" et celui à droite correspond à la phrase 'YOU ASK ME'. 67
Figure 38. La relation R-locus dans la phrase "*Kate asked Bob a question*" 68
Figure 39. Exemple de classificateur "flat" et "thick" du signe "BOOK" 69
Figure 40. Les relations de dépendances "*IN*" et "*FROM*" dans deux phrases en anglais 69
Figure 41. Relation de dépendance temporelle dans la phrase "*Last week, the meeting started at ten and ended at a quarter to three*" 70
Figure 42. Transcription en glose de la phrase "*John doesn't buy the house*" 72
Figure 43. Geste manuel pour exprimer la négation dans la langue des signes de la Jordanie 72
Figure 44. La relation de coréférence dans la phrase "*Mary sais she would help me*" 73
Figure 45. Principe de traduction automatique selon le triangle de Vauquois 76
Figure 46. Architecture du système de traduction proposé 76
Figure 47. Aperçu du fonctionnement du système de traduction proposé 78
Figure 48. Arbre syntaxique de la phrase "*Kate gave chocolate for each boy, yesterday*" 83
Figure 49. Les relations de dépendances de "*Kate gave chocolate for each boy, yesterday*" 84

Figure 50. Graphe de dépendance de la phrase "*Kate gave chocolate for each boy, yesterday*" 84
Figure 51. Matrice d'adjacence du graphe de dépendance de la phrase "*Kate gave chocolate for each boy, yesterday*" 85
Figure 52. Exemple de coréférence 87
Figure 53. Transcription en gloses de la phrase "*Kate gave chocolate for each boy, yesterday.*" 88
Figure 54. Modèle de traduction à base de statistiques 90
Figure 55. Alignement bi-directionnel au niveau des segments entre deux phrases 92
Figure 56. Deux phrases en relation de traduction dont les segments sont alignées 92
Figure 57. L'architecture d'un décodeur classique 93
Figure 58. Architecture globale du système de traduction automatique en deux sens 95
Figure 59. Aperçu de la taille du corpus Anglais-ASL pour apprentissage, raffinement et évaluation (gauche) Nombre de phrases (droite) Nombre de jetons 100
Figure 60. Diagramme d'alignement de la phrase 'YOU BLUE CAR' en ASL 108
Figure 61. Diagramme d'alignement de la phrase 'i love you' et la phrase 'PRO-1ST YOU LOVE' 108
Figure 62. Diagramme d'alignement de la phrase 'D'ont you like candy ?' 108
Figure 63. Evolution de la distribution probabiliste du jeton 'YOU' durant 100 itérations 113
Figure 64. Convergence des traductions lexicales les plus probables vers 1 calculée à partir du modèle d'alignement IBM1 113
Figure 65. Convergence des traductions lexicales les plus probables vers 1 calculée à partir du modèle d'alignement IBM1 en utilisant la distance de Jaro-Winkler 115
Figure 66. Evolution de la distribution probabiliste du jeton 'YOU' durant 20 itérations (en rouge) sans distance de Jaro-Winkler (en bleu) avec distance de Jaro-Winkler 116
Figure 67. Exemple de traduction en utilisant le modèle d'IBM 2 118
Figure 68. Les différentes étapes de traduction du modèle IBM 3 120
Figure 69. Evolution de la distribution probabiliste de la traduction lexicale (WHAT,∅) et sa convergence vers 1 121
Figure 70. Matrice d'alignement de la phrase 'I finished reading the book.' (les lignes) et sa traduction 'PRO-1st FINISH BOOK' (les colonnes) 123
Figure 71. Deux matrices d'alignement de la phrase 'I finished reading the book.' et sa traduction 'PRO-1st FINISH BOOK' (en haut) de l'anglais vers l'ASL (en base) de l'ASL vers l'anglais (à droite) couplage entre les deux alignements 123
Figure 72. Alignement à base de segments de la phrase en anglais 'Did you see the new clothing store they put in ?' et sa traduction en ASL 'WHAT-DO YOU SEE NEW CLOTHES STORE THEY PUT' 126
Figure 73. Alignement à base de segments (avec croisement) de la phrase en anglais 'It is nice. It is next to the grocery store' et sa traduction en ASL 'IT NICE FOOD STORE NEXT TO' 126
Figure 74. Calcul de la distance de réorganisation d'une phrase source vers une phrase cible 127
Figure 75. Matrice d'alignement de la phrase en anglais 'Did you see the new clothing store they put in ?' et sa traduction en ASL 'WHAT-DO YOU SEE NEW CLOTHES STORE THEY PUT IN' 128
Figure 76. Les différentes étapes de traductions de la phrase 'NAME YOU WHAT' 133
Figure 77. Grille des différentes traductions possibles de la phrase 'NAME YOU WHAT' (à droite) les traductions possibles (à gauche) score en logarithmique 133
Figure 78. Variation des résultats d'évaluation (bleu) score BLEU (cyan) Modèle 4-gram (magenta) Modèle 3-gram (vert) Modèle 2-gram (rouge) Modèle 1-gram (à gauche) sans utiliser la distance Jaro-Winkler (à droite) en utilisant la distance Jaro-Winkler 135

Liste des tableaux

Tableau 1. Les différents attributs de la balise "token"....50
Tableau 2. Les pronoms personnels dans l'ASL....53
Tableau 3. Liste des expressions faciales....56
Tableau 4. Exemples d'utilisation de gloses pour représenter des énoncés en ASL....58
Tableau 5. Les principales méthodes pour la création d'un fichier XML-Gloss....61
Tableau 6. Liste des catégories grammaticales en Anglais....80
Tableau 7. Extrait d'une table de traduction construite sur le corpus parallèle Europarl....92
Tableau 8. Statistiques des données extraites à partir du corpus Gutenberg....99
Tableau 9. Statistiques du corpus après la phase de nettoyage....100
Tableau 10. Nombre hypothétique des différentes traductions possibles du mot 'REVIEW'....106
Tableau 11. Les traductions lexicales pour 3 jetons composant la phrase 'YOU BLUE CAR'....107
Tableau 12. Application de l'algorithme EM sur le modèle IBM1....113
Tableau 13. Application de l'algorithme EM sur le modèle IBM1 en utilisant la distance de Jaro-Winkler....115
Tableau 14. Evolution de la distribution probabiliste de P(e,f) et de l'incertitude au cours de l'apprentissage en utilisant le modèle IBM 1....117
Tableau 15. Les probabilités d'alignement possibles du segment 'NEXT TO'....126
Tableau 16. Variation du score BLEU selon la taille du corpus d'évaluation....134
Tableau 17. Variation du score BLEU selon la taille du corpus d'évaluation en utilisant l'approche des chaînes similaires....135

Liste des abréviations

AER	:	*Alignment Error Rate*
ASL	:	*American Sign Language*
BLEU	:	*BiLangual Evaluation Understudy*
DGS	:	*Deutsche Gebärdensprache*
EM	:	*Expectation Maximization*
LV	:	Langue Vocale (*Spoken Language)*
MERT	:	*Minimum Error Rate Training*
METEOR	:	*Metric for Evaluation of Translation with Explicit ORdering*
MIRA	:	*Margin Infused Relaxed Algorithm*
NIST	:	*National Institute of Standards and Technology*
PBMT	:	*Phrase-Based Machine Translation*
SL	:	*Sign Language*
SML	:	*Sign Modelling Language*
SMT	:	*Statistical Machine Translation*
VoiceXML	:	*Voice eXtensible Markup Language*
WER	:	*Word Error Rate*
XML-Gloss	:	*eXtensible Markup Language for GLOSS transcription system*

Introduction Générale

Échanger, collaborer, construire ensemble, tout serait plus simple si l'on pouvait parler toutes les langues. Ou, à défaut, disposer d'un outil de communication le permettant. La communication est l'essence même de l'interaction humaine. L'un des moyens les plus efficaces de communication entre êtres humains s'effectue à travers l'utilisation des langues, qui sont elles-mêmes nées de l'interaction humaine. La communication et la langue ont donc une relation d'interdépendance. Cependant, ce moyen de communication naturel peut constituer un obstacle dans les cas où les langues se diffèrent. Nul ne peut ignorer l'évidente barrière de communication entre des langues de modalités différentes, à savoir les langues vocales (LV) et les langues des signes "*Sign Langage*" (SL).

En effet, les langues des signes, utilisées par les communautés sourdes, qui n'entendent pas ou entendent mal, sont des langues visio-gestuelles, dont le message est transmis par les gestes et reçu par le canal visuel. Les langues vocales, quant à elles, ont un caractère audio-phonatoire. Leur message est émis via le canal phonatoire et reçu grâce au canal auditif. Les canaux utilisés par les SL sont donc différents de ceux utilisés habituellement et les distinguent des autres langues. Les connaissances linguistiques acquises après des années de recherches et d'études sur diverses langues vocales sont difficilement transposables aux SL qui deviennent ainsi un nouvel objet d'étude linguistique, et ce seulement à partir des années 60. Ce champ de recherche est donc davantage récent, ce qui explique que les connaissances sur le fonctionnement linguistique des SL soient en constante évolution.

En outre, les langues des signes sont, comme toute langue, des systèmes nécessitant chacun un apprentissage particulier, preuve si besoin est, qu'il s'agit bien de "langues naturelles" et non d'un moyen universel et primitif de communication. Il n'existe pas en effet une langue des signes universelle, mais plusieurs, telles la Langue des Signes Américaine (*American Sign Language* abrégé en ASL), la Langue des Signes Française (LSF), la Langue des Signes Britannique (*British Sign Language* abrégé en BSL), etc. Autant qu'il existe de communautés différentes de sourds, existent des langues des signes leur étant appropriées, chacune d'elles ayant son histoire, ses unités signifiantes et son propre lexique.

Aujourd'hui, l'interprétation en langues des signes suscite de plus en plus l'intérêt de la communauté scientifique, cela permettant à la communauté sourde d'accéder aux informations de la vie quotidienne, et d'utiliser même des outils de communication. D'où les obstacles que vivent les personnes atteintes de surdité au quotidien pourront être dépassés et idéalement brisés.

Le Traitement automatique du langage naturel dit (TALN) représente l'ensemble des recherches et développements visant à modéliser et reproduire, à l'aide de machines, la capacité humaine, mais aussi à réaliser et assimiler des énoncés linguistiques dans des buts de communication. Il sera donc question de langage humain. Les champs du traitement automatique du langage couvrent de très nombreuses disciplines de recherche pouvant mettre en œuvre diverses compétences tels que : la traduction automatique, le

résumé automatique, l'annotation sémantique, etc. Spécialement, la langue des signes, et plus précisément la Langue des Signes Américaine notée « ASL », sera celle utilisée tout au long de cette thèse. Cette dernière est une langue naturelle, comme nous l'avons précisé antérieurement, dotée d'un lexique et d'une grammaire que plusieurs linguistes, experts et interprètes de l'ASL ont définie. Quant au système TALS, il s'agit d'une discipline de traitement automatique de la langue des signes servant à définir des théories, des approches et également des outils afin de coder, noter, extraire, interpréter et traduire la langue des signes, soit entre une langue parlée et une LS, soit entre des LS.

Cette thèse relève donc du traitement automatique des langues et s'intéresse, en particulier, au cas des langues des signes. Exotiques par leur mise en œuvre de gestes et non de sons, celles-ci sortent du cadre phonologique traditionnel et ne disposent pas d'une écriture phonétique internationale, comme les langues vocales, pour transcrire leurs réalisations. Également exotiques par leur multi-linéarité, elles mettent à profit leur canal gestuel afin de véhiculer plusieurs éléments d'informations en même temps, tandis que l'appareil vocal ne permet la production que d'un son à la fois. La modélisation des langues gestuelles pour des applications en informatique pose donc naturellement des questions distinctes de celles posées par les langues vocales.

La transcription, est l'opération qui consiste à substituer à chaque phonème (on parle alors de transcription phonologique) ou à chaque son (transcription phonétique) d'une langue un graphème ou un groupe de graphèmes d'un système d'écriture. Elle dépend donc de la langue cible, un unique phonème pouvant correspondre à différents graphèmes suivant la langue considérée. Plus simplement, il s'agit de l'écriture, dans un système donné, de mots ou phrases prononcés. La transcription vise également à être sans perte, de sorte qu'il devrait idéalement toujours être possible, en connaissant les règles de transcription, de reconstituer la prononciation originale à partir de celle-ci. Ainsi, les tentatives pour représenter et décrire les gestes humains ont conduit à l'élaboration de systèmes de notation et de transcriptions gestuelles, ainsi qu'au développement de langages informatiques dédiés à l'animation des personnages virtuels de communication. Tous ces systèmes visent à décrire des mouvements plus ou moins codifiés et structurés, suivant qu'ils s'appuient ou non sur des unités gestuelles bien identifiées, et des règles syntaxiques ou sémantiques. Les objectifs de ces représentations sont multiples : elles permettent de fixer des phénomènes gestuels ayant un déroulement dans le temps, mais aussi de les explorer, les transmettre et les faire évoluer. Elles favorisent en particulier le stockage, l'échange et la communication d'informations à distance. D'où l'objectif de ce mémoire est de supprimer les freins qu'il pourrait y avoir suite à l'utilisation de l'informatique comme support de l'écriture de la langue des signes, afin de ne pas induire des dégradations supplémentaires de l'information véhiculée. Il sera laissé aux linguistes le soin d'améliorer le passage de l'information Visio-gestuelle à l'information écrite, qui constitue un parallèle indissociable à ce travail.

Dans cette optique, cette thèse porte sur la transcription des langues des signes et la traduction automatique par méthodes statistiques, et se concentre plus particulièrement sur la traduction automatique d'un texte en anglais vers l'ASL et inversement. Le travail effectué au cours du doctorat s'articule autour de quatre axes :

1. Le développement et l'amélioration des systèmes de transcriptions ;

2. L'étude des spécificités de la traduction de la langue des signes transcrite manuellement ou bien automatiquement, avec l'accent porté sur la transcription automatique ;
3. La génération de corpus parallèles artificiels en utilisant des approches linguistiques.
4. La traduction automatique en utilisant une approche statistique.

Plus précisément, cette thèse s'organise comme suit :

- Le premier chapitre présentera d'abord la langue des signes, objet de notre modélisation : sa structure, les bases de sa théorie et les recherches qu'elle a subit dans le temps. Nous situerons ensuite les différents traitements automatiques, à savoir la génération automatique, en utilisant des signeurs virtuels 3D et des traducteurs automatiques de SL.
- Le deuxième chapitre dressera nos réflexions diverses façonnant la problématique, une étude des travaux existants sur la transcription de SL en la matière, proposant une analyse critique et détaillée de chaque modèle, afin de dégager le système le plus adapté à notre travail.
- Le troisième chapitre constituera le cœur de notre travail, en ce sens où nous y exposerons le modèle élaboré durant ce mémoire. Il fera état d'une représentation d'un système d'annotation de la Langue des Signes Américaine en décrivant quelques résultats qui visent la validation de notre solution.
- Le quatrième chapitre portera sur la génération de discours en langue des signes Américaine à partir des règles de dépendances linguistiques extraites à partir des règles grammaticales de la langue Anglais d'une phrase donnée.
- Le cinquième chapitre présentera l'architecture globale du système de traduction automatique statistique en l'ASL et l'anglais en deux sens.
- Le sixième chapitre, suite à la présentation du système de traduction, décrira la construction d'une mémoire de traductions lexicales à partir du discours artificiel.
- Le septième chapitre abordera la mise en place du traducteur avec l'expérimentation et l'interprétation des résultats obtenus.
- Enfin, La dernière partie de ce rapport présentera une conclusion générale.

Ce travail s'adapte au cadre du projet WebSign[1] du laboratoire de recherche LaTICE qui met en œuvre un outil de communication pour les malentendants à travers le web, en utilisant un personnage virtuel 3D.

[1] Site Web du projet WebSign [http://www.latice.rnu.tn/websign]

Chapitre 1 : État de l'art

1. Introduction

Les langues des signes (SL) sont les langues naturelles des personnes sourdes. Nous écrivons le terme « langues des signes » au pluriel, car contrairement à ce que pense la majorité des personnes, il n'existe pas une unique langue des signes, internationale, mais une langue des signes par pays. L'objectif de ce chapitre est de situer le travail de recherche rapporté dans cette thèse. Il présente dans la première partie la SL dans le contexte d'une langue de gestes de communication. Nous proposons dans une première partie une présentation globale des langues de signes puis nous nous intéresserons aux théories linguistiques et aux recherches entamées. Dans une seconde partie, nous aborderons les différents traitements automatiques effectués sur la langue des signes.

2. Les langues des signes

Les langues des signes abrégées en "SL" sont des langues Visio-gestuelles. Elles sont considérées par la communauté parlante comme des langues standards propres aux sourds, dont le message est transmis par les gestes et reçu par le canal visuel. Les langues vocales abrégées en "LV", quant à elles, sont des langues audio-phonatoires. Leur message est émis via le canal phonatoire et reçu à travers le canal auditif. Les canaux utilisés par les SL sont donc différents de ceux utilisés habituellement et distinguent les SL des autres langues. Les SL assurent toutes les fonctions remplies par les autres langues naturelles vocales. Elles sont, pour les sourds, le seul mode linguistique véritablement approprié, le seul moyen qui leur permet un développement cognitif et psychologique d'une façon équivalente à ce qu'il en est d'une LV pour un non sourd.

La SL est une langue à part entière, comparable à l'anglais, au français ou à n'importe quelle langue du monde, qui possède toutes les caractéristiques des langues naturelles et peut s'analyser comme toute langue. Donc la langue des signes possède un système de communication et de représentation du monde ayant des traits bien définis et une structure très précise. Ainsi, elle possède la flexibilité requise pour créer tout nouveau vocabulaire et toute nouvelle structure grammaticale nécessaire et une pleine capacité d'abstraction et d'expression. Elle se transmet d'une génération à l'autre et il n'existe pas seulement une langue des signes universelle mais plusieurs chacune d'elles s'étant développées dans un contexte unique.

2.1. Les gestes dans les langues des signes

"Geste" n'est pas un simple synonyme de "signe". Un geste est un ensemble de paramètres (mouvements, posture, forme de la main, etc.) à partir desquels un signe est construit. Un geste est un signe si l'ensemble de ces paramètres suffit à créer du sens. Les différentes composantes d'un geste sont illustrées dans la "Figure 1".

Le geste est une trajectoire de différentes composantes manuelles et non-manuelles. Les composantes non-manuelles sont tout ce qui concerne le corps et le visage

sans tenir compte des mains telles que la position du buste, la position des épaules, direction du regard, etc. Les gestes manuels couvrent les aspects visuels des mains qui sont divisés en deux parties : main dominée et main dominante. Tous ses paramètres seront décrits dans ce qui suit.

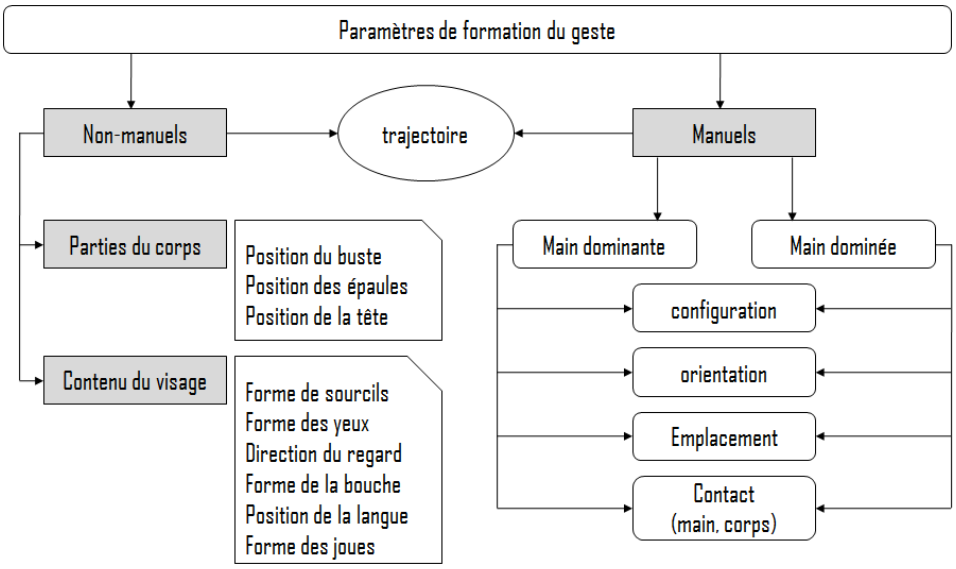


Figure 1. Paramètres de formation d'un signe

2.2. Composantes de la langue des signes

Chaque phrase en SL est constituée d'une suite de gestes des mains que l'on appelle "signes" [1] et qui sont agencés suivant une syntaxe régie par une logique spatiale et temporelle. Chaque signe de SL est caractérisé par cinq paramètres : la configuration, l'orientation, l'emplacement, le mouvement et les expressions du visage [2]. Ils se produisent simultanément et ils permettent de distinguer les signes les uns des autres. Chacune de ces caractéristiques est discutée ci-dessous :

- La configuration : désigne la forme que la main doit adopter pour la production du signe (par exemple "main plate" ou "poing fermé") ;
- L'emplacement : désigne l'endroit où se situe la main au début de la production du signe ;
- L'orientation : représente à la fois l'orientation de la paume de la main par rapport au signeur[2] (vers le haut, le bas, l'intérieur - le corps du signeur - ou l'extérieur) et l'orientation de la main montrant la direction des doigts ;
- Le mouvement décrit le déplacement de la main, dont sa vitesse et sa direction ;
- Les expressions du visage : regroupent les mouvements des sourcils, du front, des joues, des yeux et de la bouche du signeur durant la production du signe.

On distingue aussi la main dominante (la main droite pour les droitiers, la gauche pour les gauchers) de la main dominée. Lors de la production d'un signe, si une seule main est requise, on utilisera la main dominante (sauf si celle-ci est indisponible). Si les deux

[2] Signeur: les personnes qui emploient la langue des signes, définition prise du rapport "A la découverte de la surdité", disponible en ligne : http://goo.gl/AmiBcy

mains sont nécessaires, deux cas se présentent : soit les deux mains vont se déplacer en symétrie, soit les deux mains vont se comporter différemment. La main dominée désignera alors souvent un référent fixe, un repère, tandis que la main dominante sera en mouvement par rapport à ce repère. Tous ces paramètres sont illustrés ci-dessous par la Figure 2 avec le signe "**CAN**" (en langue des signes Américaine) [3].

Figure 2. Description paramétrique du signe "CAN"

La langue des signes comporte de nombreuses particularités grammaticales [4]:

- Les temps du verbe n'existent pas : on précise seulement le moment où se déroule l'action. Pour cela, on les situe sur une ligne de temps qui est perpendiculaire au corps : le futur se trouvant devant et le passé derrière (dans la plupart des langues des signes).
- Les classificateurs qui sont des signes génériques représentant des classes d'objets de taille, forme ou épaisseur similaires et qu'on retrouve dans certaines langues orales comme le chinois.
- La localisation des personnages, des choses dans l'espace du signeur et auxquels il peut faire référence, par exemple par la direction du regard.
- L'ordre d'exécution des signes est moins important que leur arrangement visuel dans l'espace. Généralement, les compléments circonstanciels de temps et de lieu viennent en premier et le verbe à la fin.

2.3. Les différentes classes de signes

En langue des signes, on peut distinguer différentes classes de signes, chacune correspondant à un usage particulier et ayant son importance dans la formation des phrases en SL :

- Les signes standards : correspondent à des mots (nom, verbe, adjectif...) ayant un sens bien établi.
- Spécificateurs de forme et de taille : permettent de décrire un objet, un animal, une scène. C'est à l'aide de la forme des mains, de leur orientation et de leur mouvement que le signeur décrit la forme et les dimensions d'un objet.
- Classificateurs : sont similaires aux spécificateurs dans le sens où ils représentent également un objet (ou personne, animal, etc.). Un classificateur peut ensuite être utilisé pour représenter un objet déjà cité dans le début de phrase.

3. Les théories linguistiques

Actuellement, la plupart des théories linguistiques sur les SL sont issues des études sur les LV, et reposent sur une approche générativiste qui suppose que toutes les langues sont descriptibles par un unique système [5]. Comme la langue des signes est une langue visio-gestuelle dont le fonctionnement est très différent de celui des langues vocales, les linguistes s'attachent à en définir les contours, en prenant plus ou moins de distance avec les théories linguistiques qui ont été proposées pour les langues vocales. Pour produire un signe, on a besoin seulement d'une personne dont son environnement sera un espace réservé pour placer les différents éléments cités dans le discours. Cet espace est nommé "Espace de signation" [6]. Aussi, les signeurs utilisent les mêmes mots avec différents signes selon le contexte et l'état, on parle de l'iconicité et les transferts.

3.1. Espace de signation

L'une des principales propriétés des SL est l'utilisation intensive de l'espace de signation. Cet espace permet de structurer le discours. Les entités du discours (personnes, objets, évènements, notions abstraites, etc.) sont localisées dans cet espace. Ce dernier se situe face au signeur, et forme un volume d'une profondeur, d'une largeur et d'une hauteur qui égale à la longueur des bras du locuteur, (voir Figure 3) [5]. Ensuite, des structures spatio-temporelles sont utilisées pour élaborer des énoncés.

Par exemple, les relations spatiales entre les entités sont généralement établies sans utiliser de signe lexical dédié. C'est l'utilisation de proformes[3] mises en relation dans l'espace qui permet d'exprimer de telles relations. L'ordre d'apparition des signes dans l'énoncé est moins important que leur arrangement dans l'espace, mais on peut dégager des principes généraux au niveau de l'organisation temporelle : on signe du plus général, le contexte, vers le plus précis, l'action ; les entités les plus statiques sont signées avant les entités les plus mobiles.

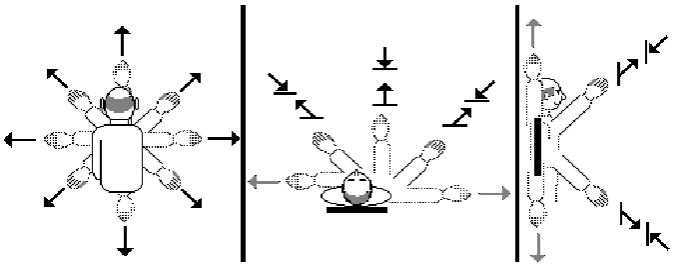


Figure 3. Représentation schématique de l'espace de signation

3.2. Principe d'iconicité

Cuxac [7] a mis en place le mécanisme d'iconicité au cœur du système linguistique de la SL qui permet au signeur de choisir entre deux stratégies discursives issues d'une expérience de l'environnement. Au départ, la première stratégie était construite sous une forme iconique (montrant visuellement ce qui est exprimé). Ensuite, les signes évoluent

[3] Proforme : configuration de la main qui représente un point de vue sur une entité précédemment introduite dans le discours.

vers une seconde stratégie dont la forme est dite "générique", perdant de son intention d'iconicité (l'intention disparait, mais l'iconicité peut être conservée dans le signe).

Ces signes génériques sont largement appelés "signes standards", actuellement appelés "signes-mots" par Cuxac [8]. Ainsi, la langue est potentiellement exprimable suivant deux manières, selon qu'on vise à dire "sans montrer", ou "en montrant". La première solution consiste à employer des signes-mots (faisant partie du lexique), plus ou moins iconiques. Quant à la seconde, elle met en œuvre des structures dites de « grande iconicité » qui montrent à l'interlocuteur ce dont il est question dans la conversation.

Ces deux stratégies coexistent et le va-et-vient entre les deux est constant. C'est par ces deux grands axes qu'une grammaire de la SL peut être établie. Ils font naître des différences structurales.

Cuxac tente de démontrer que ces structures d'iconicité sont bien des éléments linguistiques et constituent même la démonstration la plus subtile et la plus convaincante de ce qu'est une langue. En effet, ces structures sont quasiment identiques d'une SL à l'autre, ce qui permet une intercompréhension rapide entre des signeurs de nationalités différentes.

3.3. Les transferts de grande iconicité

L'utilisation de multiples articulateurs[4] (mains, bras, épaules, buste, tête, visage et yeux) qui s'articulent de façon simultanée au sein d'un espace d'un point de vue linguistique permet d'élaborer des structures linguistiques complexes, mettant en œuvre l'iconicité à différents niveaux du discours. Cuxac [7] a proposé une catégorisation de ces structures linguistiques de grande iconicité. Il a distingué trois types de structures qu'il appelle "transferts" :

- le transfert de taille et de forme, qui est utilisé pour décrire la taille et la forme d'une entité,
- le transfert situationnel, qui est utilisé pour montrer le positionnement et le déplacement d'une entité par rapport à un repère locatif stable,
- le transfert personnel, où le signeur incarne une des entités du discours, ceci met en évidence la capacité du locuteur à entrer dans la peau des protagonistes de l'énoncé (personne, animal, objet).

4. Les recherches sur les langues des signes

Il y a actuellement des recherches sur les SL dans de nombreux domaines : histoire, sociologie, anthropologie et les recherches en linguistique qui sont considérablement développées. La diffusion d'informations et les progrès technologiques en SL nécessitent, comme pour toute autre langue, de développer des outils informatiques qui leurs sont

[4] Articulateur : toute partie mobile du corps sur laquelle on peut agir volontairement et qui est fonctionnelle dans la production de la parole, ce terme étant employé dans son sens premier, celui de langage incarné [source: http://m.antoniotti.free.fr/condvoc.htm].

dédiés, et par conséquent de mener des recherches en traitement automatique des langues des signes (TALS).

L'étude des langues des signes est en train de sortir de son cadre spécifique et confidentiel pour rejoindre les recherches sur les langues, aussi bien du point de vue linguistique qu'informatique. Les colloques spécifiques se multiplient et en même temps, les recherches sur les langues de signes sont présentées dans les colloques non thématiques. L'étude de la langue des signes, révélateur de faits de langues, intéresse au-delà du linguiste spécialisé. En informatique de l'image, les recherches en analyse vidéo et en synthèse de personnages investissent le champ de la langue des signes.

Avec l'Internet, les outils de traitement automatique des langues font partie de la vie et du développement de celles-ci, et les langues des signes doivent pouvoir être utilisées sur Internet pour communiquer, pour se documenter, et pour apprendre ces langues. Ceci crée des besoins d'applications performantes, robustes et respectueuses des spécificités de la langue des signes. Ces propriétés ne pourront être atteintes que si ces applications s'appuient sur des modélisations élaborées et des formalismes rigoureux des langues des signes.

D'un point de vue linguistique, la formalisation [9] et l'informatisation des langues des signes sont particulièrement intéressantes : le médium différent devrait permettre de dégager les caractères communs aux langues humaines mais aussi les spécificités propres aux langues des signes, par exemple en mettant à disposition des corpus, des outils d'annotation et d'analyse plus ou moins automatisées. Ainsi, ce domaine couvre les aspects de reconnaissance et de synthèse de la SL, ainsi que des conceptions de grammaire (pour la génération de langue) et de traduction, mais aussi des connaissances dans d'autres domaines, principalement linguistique. Plusieurs équipes de recherche travaillent sur l'un et/ou l'autre aspect. Notre étude se situe donc dans le domaine du traitement automatique des langues des signes abrégé en TALS, dans l'aspect de la mise au point d'un modèle de la ASL : notre but est en effet de traiter automatiquement l'ASL par un système de transcription ou d'annotation, afin d'animer un agent conversationnel.

5. Les ressources linguistiques en SL

Les ressources linguistiques sont une collection de données qui sert au traitement automatique des langues des signes comme la traduction automatique, l'extraction des connaissances, etc. Généralement, la collecte des ressources se fait à travers des projets nationaux ou internationaux en spécifiant les caractéristiques du corpus résultant :

- Les objectifs du corpus ;
- Taux des données (en mots, phrases ou heures) ;
- Politique d'accessibilité aux ressources ;
- Nombre des participants (sourds, interprètes ou autres) ;
- Système de transcription (Gloss, HamNoSys, etc.) ;
- Lemmatisation ;
- Outils d'annotation ;
- Les mouvements et les postules selon le modèle de Liddel [10] ;
- La qualité des vidéos enregistrées.

Malgré les initiatives pour la collecte des ressources, mais jusqu'à aujourd'hui on ne trouve pas de grand corpus qui seront utiles pour le traitement automatique. Cette limite est due au coût élevé de la production des signes par signeurs [11].

5.1. La langue des signes Américaine

La langue des signes Américaine abrégée en ASL est la langue des signes la plus chanceuse en termes de ressource. Plusieurs travaux ont été réalisés et d'autres qui sont en cours. Certaines se focalisent sur des signes uniques et d'autres sur des phrases et des discours complets. A partir de l'année 2000, le "National Center for Sign Language and Gesture Ressources", de l'Université de Boston, a commencé la publication des séquences de vidéo en ASL annotées en gloses avec l'outil SignStream [12] du projet "American Sign Language Linguistic Research Project [5]" [13] . Son objectif est de fournir le plus grand corpus pour les linguistes. Son avantage majeur est qu'il a tenu comptes des expressions faciales lors des annotations [14].

Figure 4. Interface d'annotation et de consultation du "ASL Lexicon Video Dataset ASLLVD"

De la même université, un autre corpus [15] a été réalisé. Il contient plus que 3000 signes signés par 6 personnes sourdes en utilisant trois caméras de haute définition annotés avec SignStream et accessible en ligne [16]. Ensuite, chaque séquence a subi des traitements d'images pour extraire les mouvements et les configurations des mains et les expressions faciales tel que le regard, les mimiques, etc. (voir Figure 4).

5.2. La langue des signes Allemande

Plusieurs projets ont été menés pour construire des ressources en langue des signes allemandes (Deutsche Gebärdensprache DGS). Ebbinghaus et Heßmann ont publié le premier corpus en DGS appelé "Berlin Corpus" [17]. Il contient essentiellement des interviews en langue des signes réalisés par 90 signeurs. Le corpus a été annoté par l'outil FileMaker[6] en utilisant la transcription en gloses et en HamNoSys [18]. Cette initiative

[5] Le site web du "Amercian Sign Language Linguistic Research Project" [http://www.bu.edu/asllrp]
[6] Filemaker: logiciel de bases des données [http://www.filemaker.com]

était le point de départ pour définir la première grammaire (phonologie, morphologie, morphosyntaxique and syntaxique) pour la DGS.

En 2009, le projet "DGS-Korpus"[7] [19] a été démarré visant à enregistrer 1159 heures en DGS signés par 330 sourds. Les annotations ont été faites par l'outil iLex [20]. L'objectif de ce projet est de fournir les variations lexicales, morphologiques, phonologiques et sémantiques. A partir de l'année 2015, ce corpus est accessible en ligne. Un autre corpus en DGS est disponible en ligne nommé RWTH-Phoenix [21]. Les séquences vidéo sont extraites des émissions télé qui contiennent un interprète traduisant les programmes en langue des signes. Toutes les séquences enregistrées ont été transcrites en gloses et annotées en utilisant l'outil ELAN [22].

5.3. La langue des signes Arabe-Unifiée

Jusqu'à maintenant, la langue des signes Arabe-Unifiée[8] n'est pas encore confirmée pour tous les pays Arabes. Elle a été adoptée par l'Organisation Arabe pour l'Éducation, la Culture et les Sciences abrégé en "ALECSO" en 2001 pour qu'elle soit enseignée dans différents pays Arabes comme Égypte, Qatar et la Tunisie. Le premier travail indexé et publié pour le traitement automatique en utilisant un corpus est celui de Abdelfattah en 2005 [23]. Les autres travaux sont limités à la création des dictionnaires[9].

5.4. La langue des signes Britannique

En 2007, Le projet "British Sign Language Corpus Project"[10] [24] a démarré. Son objectif est de créer un corpus compréhensible par les traducteurs automatiques pour la langue des signes Britannique. Les séquences vidéo ont été rassemblées par des signataires sourds et non-sourds à travers le Royaume-Uni. Plus que 180 heures ont été enregistrés par 240 personnes. L'accès aux ressources est sous permission et dans le cadre de la recherche seulement. Aussi, l'inconvénient majeur de cette ressource est qu'il n'existe ni d'annotation ni de transcription. En outre, les données seront limitées pour quelques situations, en se concentrant principalement sur les données de conversation et d'entretien. Une partie de la base de données est annotée et étiquetée en utilisant le logiciel ELAN [22], des descriptions de métadonnées sont enregistrés à l'aide de l'outil IMDI [25].

5.5. La langue des signes Coréenne

Le seul travail qu'on trouve dans la littérature est le travail de Hong [26]. Le corpus contient des vidéos en la langue des signes Coréenne (KSL) animé par 35 personnes de différents âges. Le corpus se compose d'une conversation spontanée, de la signature dans un but didactique, ainsi que des réponses. Les données de conversation spontanée

[7] DGS Korpus: le projet est réalisé par "Institute for German Sign Language and Communication of the Deaf" de l'Université d'Hambourg et financé par "the German Federal Government and Federal States" et "Federal Ministry of Education and Research" [http://www.sign-lang.uni-hamburg.de/dgs-korpus/]
[8] Portail Web pour apprendre la langue des signes Arabe-Unifiée [http://www.menasy.com]
[9] Dictionnaire de la langue des signes Arabe-Unifiée [http://www.menasy.com]
[10] Site web officiel du "British Sign Language Corpus Project" [http://www.bslcorpusproject.org]

impliquent au moins deux informateurs parlant les uns aux autres autour des signes de la vie quotidienne. Toutes les séquences vidéo ont été annotées par l'outil iLex [20] pour mettre en valeur les verbes d'agrément, qui sont divisés en quatre catégories :

- Les verbes qui apparaissent uniquement avec deux arguments animés [27];
- Les verbes qui apparaissent avec deux arguments animés ou bien avec un argument animé et un argument concret inanimé [28];
- Les verbes qui apparaissent avec deux arguments animés ou avec un argument animé et un seul paramètre abstrait inanimé ;
- Les verbes qui apparaissent toujours avec un argument animé et un seul paramètre inanimé [29].

D'après Miyamoto [30] [31], la langue des signes Coréenne, y inclut la notion des verbes d'agrément, a été inspiré de la langue des signes Américaine qui disposait d'une grammaire et d'un lexique bien clair, ainsi que des structures lexicales, syntaxiques et morphologiques.

5.6. La langue des signes Chinoise et Taïwanaise

Afin de développer un traducteur automatique à base de statistiques entre les deux langues de signes Chinoise et Taïwanaise, Chiu and Cheng ont publié un corpus contenant plus que 2000 phrases entre ces deux langues de signes [32]. Les phrases en ont été annotées en gloses pour la facilité du traitement automatique.

5.7. La langue des signes Espagnole

Pour la langue des signes Espagnole, il existe un seul corpus parallèle [33] pour deux domaines spécifiques : le service de renouvellement des documents identifiés et le service des permis de conduite. Ce corpus a été réalisé en collaboration avec les offices gouvernementaux. Plus que 5000 phrases ont été interprétées et annotées en gloses en utilisant l'outil ELAN [22]. L'objectif de ce corpus est de développer des logiciels de génération de la voix à partir de la langue des signes Espagnole et vice versa.

5.8. La langue des signes Française

Le projet "LS-COLIN" [34] propose un corpus annoté pour la langue des signes Française (LSF). L'objectif principal de ce projet est de montrer les structures iconiques et leurs importances dans l'espace de signation. Le nombre total de séquences vidéo était 39 vidéos de durée moyenne d'une minute annotées en gloses et effectuées par des sourds. Toutes les vidéos ont été prises par trois caméras.

En 2007, un autre projet nommé "CREAGEST"[11] [35] a été lancé pour la création d'un corpus pour apprendre la LSF pour les enfants sourds. Les interprètes ont traduit les textes dans les histoires de bandes dessinées en LSF. Seulement 60 minutes ont été enregistrées sans annotation, ce qui a limité l'utilisation à cette ressource pour les enfants sourds. En deuxième partie, une extension du projet a été établie afin de produire un

[11] Descriptif du projet "CREAGEST" par Christophe Cuxac disponible en ligne http://goo.gl/hVFfIU

corpus de gestualité naturelle, gestualité co-verbale d'adultes non-sourds mis en relation ou non avec des adultes sourds pratiquant la LSF. En troisième partie, qui en cours, Les deux corpus s'attacheront d'une part afin d'élaborer une plateforme collaborative sur un serveur web et à mettre en place des moyens de pérennisation des méthodes, des collectes et des collaborations. Et d'autre part, à adapter des outils d'aide à la transcription et à la fouille de corpus.

5.9. La langue des signes Irlandaise

Le corpus de la langue des signes Irlandaise "ATIS" a été développé dans le "the Center for Deaf Studies" à Dublin [36]. Il contient une collection de vidéos réalisée par 40 personnes sourdes. Les participants sont âgés entre 18 et 65 ans et ils ont raconté des histoires personnels et des contes pour les enfants. Toutes les séquences ont été annotées en utilisant ELAN avec une transcription en gloses. Le corpus était spécifique pour le domaine de l'information sur le transport aérien. Le même corpus a été étendu vers les langues des signes Allemande et Sud-Africaines.

5.10. Récapitulatif et problématiques

Les ressources linguistiques pour les langues des signes diffèrent d'une langue à une autre. On remarque bien que les corpus à base de vidéos sont obligés de passer par un outil d'annotation pour mieux expliquer le contenu et la sémantique dans certain cas, puisque les images ne sont pas satisfaisantes pour la reconnaissance et l'interprétation. Certaines ressources ont été développées pour traduire automatiquement un texte en langue des signes en utilisant des approches statistiques ou des règles de transfert ou à base d'exemples. Aussi, d'autres ressources ont été publiées afin de fournir aux linguistiques tous ce qu'ils ont besoin pour définir les structures lexicales, syntaxiques, morphologiques et même sémantiques pour le traitement automatique des langues des signes.

On remarque aussi que certaines langues avaient plusieurs ressources en différents formats et accessible en ligne, et d'autres se limitent à des dictionnaires qui ne fournit pas une grande utilité, d'où les outils de traduction ou de transcription sont très limités. Les problématiques qu'on peut les extraire après avoir étudié différents corpus en différentes langues des signes sont :

- Absence d'une grande base de données textuelle (en gloses par exemple) pour les traitements automatiques en langue de signes.
- Presque tous les corpus vidéo sont spécialisés pour un domaine spécifique puisque la généralisation n'est pas simple.
- Le coût élevé pour la création des corpus vidéo annotés[12].

Aussi, on remarque que les outils d'annotation ne sont pas nombreux et ils ne tiennent pas compte de toutes les spécificités techniques et linguistiques de la langue des signes puisqu'elles sont dédiées à l'annotation en général ou à une seule langue de signes.

[12] Tous les corpus mentionnés dans ce chapitre sont fondés par des NGO ou EU pour la recherche scientifique.

On se qui suit dans ce chapitre, une étude sur les différents outils et approches pour le traitement automatique des langues des signes, allant de l'interprétation jusqu'à la synthèse en utilisant des agents conversationnels. Ceci, nous permet de bien clarifier les problématiques spécifiques aux langues des signes par rapport aux langues naturelles.

6. Traitement automatique des langues des signes

Après avoir étudié différents corpus, on remarque que la plupart des études porte sur la modélisation informatique des structures linguistiques de la SL qui sont menées dans un premier lieu à la génération des systèmes de transcriptions basés sur une décomposition linéaire des signes gestuels en différents paramètres caractéristiques (configuration, orientation, emplacement, mouvement et mimique faciales) et sur une écriture en forme signifiante par l'intermédiaire de symboles. Tous ces systèmes seront décrits ultérieurement dans le chapitre suivant. Ainsi, dans un deuxième lieu, elles sont menées dans le contexte d'une application donnée typiquement par la génération automatique d'énoncés réalisés par un système de synthèse (voir Figure 5).

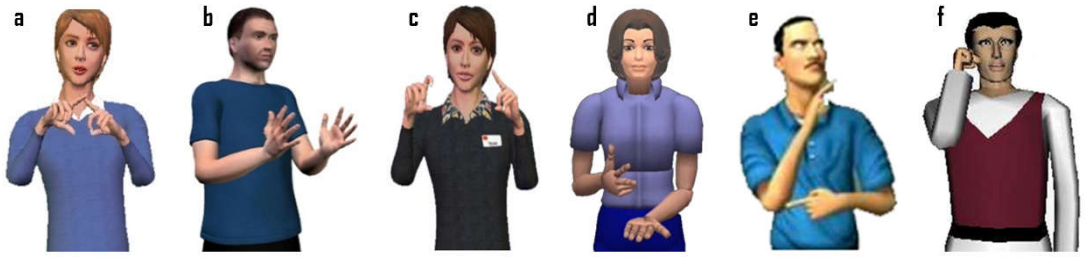


Figure 5. Exemples d'agents conversationnels pour la synthèse en langues des signes

6.1. Interprétation

L'interprétation en langue des signes, comme toute interprétation est un processus complexe, faisant intervenir de nombreuses tâches cognitives en parallèle (écoute, compréhension, recherche d'équivalents, reformulation, contrôle, etc.) [37]. Elle a pour objet d'être un outil d'accessibilité et de communication pour le public sourd. Elle est également un outil d'intégration. L'interprétation en SL existait déjà bien avant sa création officielle.

On pouvait en trouver partout : chez les particuliers, aux services d'accompagnement à la vie sociale, dans les services d'interprètes, etc. Mais il s'agissait seulement d'interprètes non sourds, de sourds sachant assez bien lire, et non de professionnels de la traduction dont la langue primaire est la langue des signes.

6.2. Synthèse 3D

L'interprétation des SL reste toujours limitée par la présence d'un interprète expert pour une langue de signes. Un système de synthèse peut pallier à ce problème. La synthèse des langues des signes est un système d'interprétation à distance sous forme d'animation à deux dimensions ou à trois dimensions. Deux approches sont explorées pour la génération d'animation en langue de signes : l'une est basée sur la vidéo et l'autre sur l'animation de personnages de synthèse ou agent conversationnel (dit avatar dans certains ouvrages ou publications).

Plusieurs systèmes développés propres à la langue des signes, seront décrits dans les sections suivantes. Certains agents conversationnels sont valables pour n'importe quelle langue des signes, on les appelle des avatars multicommunautaires, et d'autres sont spécifique à une seule langue des signes afin de mettre en valeur une ou plusieurs caractéristique comme les verbes d'agrément ou autres. Aussi, les objectifs de cette synthèse à trois dimensions sont multiples, on peut citer par exemple, l'apprentissage à distance, la traduction automatique, etc.

6.2.1. Le projet "ViSiCAST"

Le projet ViSiCAST (Virtual Signing Animation, Capture, Storage, and Transmission) est un projet Européen qui a débuté en janvier 2000 [38] dont le but est de faciliter l'accès à des services pour les personnes sourdes via l'utilisation d'agent conversationnel dans le cadre des prévisions météorologiques [36], des transactions postales, et d'applications multimédia et Web. ViSiCAST a développé un agent conversationnel pour la génération des langues des signes de différents pays Européennes. La synthèse est construite en utilisant la technique de SignAnim[13]. Les objectifs du projet ViSiCAST sont :

- Intégration dans les télévisions : les sous-titrages ont été interprétés avec un agent conversationnel.
- Plugin pour le Web et les contenus Multimédia : les contenus textuelles sont interprétés en langue des signes ce qui facilite la lecture et la compréhension pour les personnes sourdes.
- Logiciel de traduction face-to-face : A travers une interface de traduction à base de règles de dépendances, une personne non-sourde peut communiquer avec une personne sourde.

Dû aux plusieurs problèmes rencontrés lors de l'implémentation, le projet s'est arrêté. Les problèmes étaient le manque de données (absence d'un grand corpus en différentes langues des signes Européennes) et l'approche de traduction à partir des règles de dépendances, extraites d'un analyseur lexicale "CMU Link Parser Grammar[14]" [39] [40], n'était pas efficace pour l'appliquer sur le système de transcription HamNoSys [18]. Aussi, la génération des interprétations de l'agent conversationnel à partir des images GIF n'a pas été acceptée par la communauté des sourds à cause de la qualité. D'où, une extension du projet a été développée pour construire un avatar à trois dimensions pour la synthèse de la langue des signes (voir l'agent conversationnel 'a' de la Figure 5).

6.2.2. Le projet "eSIGN"

eSign[15] [41] (Essential Sign Language Information on Government Networks) est une extension du projet européen VisiCAST pour le traitement automatique des langues de signes Européennes. Son objectif et de fournir un logiciel pour la synthèse en 3D en

[13] SignAnim: agent conversationnel généré à partir des images aux format GIF http://goo.gl/kZFSJO
[14] CMU Link Parser Grammar: Analyseur lexical pour la langue Anglaise [http://www.link.cs.cmu.edu/link]
[15] Site Web du projet eSign [http://www.sign-lang.uni-hamburg.de/esign]

utilisant des avatars signants. Il est principalement mené par l'Université East Anglia[16] et l'Université d'Hambourg[17]. Plus précisément, il s'agit d'intégrer un avatar signant sur des sites web, en tant que substitution à des vidéos en langue des signes. Ce système est testé sur les sites intranet des gouvernements Allemands, Néerlandais et Anglais. Les techniques de contrôle et d'animation de l'avatar du projet ViSiCAST et de l'avatar (b) de la Figure 5 ont repris la notation SiGML [42] tout en essayant de l'améliorer [43].

6.2.3. Le projet "TESSA"

Une autre extension du projet VisiCAST est le projet TESSA[18] (Text and Sign Support Assistant) [44] qui est une application développée par l'Université East Anglia en partenariat avec La Poste Anglaise. Il allie un système de reconnaissance vocale à un agent conversation (voir l'avatar (c) de la Figure 5) pour constituer un intermédiaire entre l'employé de "La Poste Anglaise" parlant la langue Anglaise et le client sourd connaissant que la langue des signes Britannique (abrégé en BSL).

Les signes réalisés par l'avatar sont issus d'un enregistrement préalable des mouvements réellement effectués par une personne signant en BSL. Les enregistrements sont effectués au moyen d'un système de capture de mouvements placé sur la personne. Ce système fonctionne pour le BSL, mais peut facilement être transposé aux autres SL. Le système de capture de mouvement utilisé pour ce projet, prend en compte séparément les mouvements des mains, du buste, ainsi que les mouvements faciaux. Les mouvements enregistrés lors de la capture peuvent être reproduits par un squelette en trois dimensions.

Celui-ci est ensuite recouvert d'un maillage texturé qui en suit les mouvements. Pour cette application, 115 petites phrases ont été enregistrées qui couvre 90% des transactions de La Poste (éléments variables tels que les nombres ou les jours de la semaine). La concaténation de plusieurs phrases permet au système de générer environ 370 énoncés. L'avantage de ce projet est que l'animation est fluide et proche de l'interprétation humaine grâce à la base de données composée des mouvements capturés par des systèmes de capture de mouvement. Aussi, le système TESSA comprend une partie de reconnaissance de la parole vers un texte, qui sera ensuite traduit en langue des signes [45]. L'évaluation du système a été effectuée par des sourds de naissance [46].

6.2.4. Le projet de "Matt Huenerfauth"

Matt Huenerfauth[19], chercheur en traitement automatique de l'ASL, a évalué les agents conversationnels qui ont été créés dans "Virtual Human Testbed[20]" (voir agent conversationnel (d) de la Figure 5) ou avec le système d'animation faciale Greta[21] [47]. L'objectif du projet de Matt est de fournir des ressources linguistiques de haute qualité pour la traduction automatique vers l'ASL. Au début, Il a commencé par faire des études

[16] Site web de l'université East Anglia [http://www.uea.ac.uk]
[17] Site Web de l'Université de Hambourg [http://www.uni-hamburg.de]
[18] Site Web du projet TESSA [http://www.visicast.cmp.uea.ac.uk/Tessa.htm]
[19] Page personnel de Matt Huenerfauth [http://eniac.cs.qc.cuny.edu/matt]
[20] Site de Virtual Test Bed [http://vtb.engr.sc.edu]
[21] Site Web du SV Greta [http://www.tsi.enst.fr/~pelachau/Greta]

exhaustives sur les approches et les méthodologies sur la modélisation de l'ASL pour la traduction automatique [48]. Ensuite, il a proposé une nouvelle approche multicanaux pour la représentation de l'ASL en tenant compte du regard, des animations faciales et des deux mains (dominante et dominée) [49]. Puis, Matt s'est intéressé à la modélisation des prédicats classificateurs [50] des verbes d'actions et leurs représentations spatiales.

En 2012, Il a publié le premier corpus de données capturés à partir des capteurs de mouvements (Motion Capture) en utilisant trois caméras (voir Figure 6) [51]. Ce corpus s'est focalisé sur les pronoms personnels et les verbes directionnels à partir des discours spontanés signés par un ensemble de personnes sourdes. Les annotations de ce corpus sont basées sur les conventions d'annotation de SignStream [13].

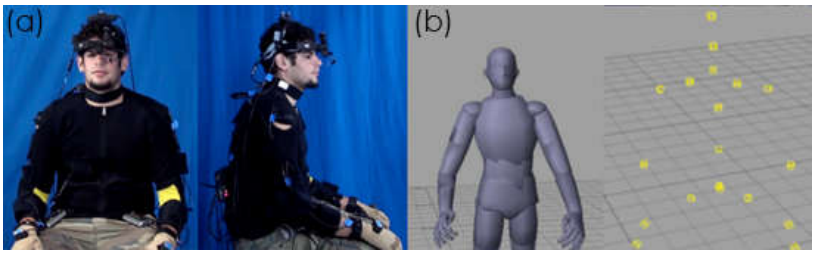


Figure 6. (a) Phase d'enregistrement (b) Visualisation des données générés par les capteurs de mouvements en 3D

Les approches proposées par Matt sont prometteuses et peuvent être appliquées sur d'autres langues de signes s'ils ont une modélisation claire et définie par une grammaire.

6.2.5. Le projet "Vcom3D"

La société Vcom3D[22] propose des outils de génération d'ASL. Il s'agit principalement de dictionnaires et d'un outil d'édition d'ASL. L'éditeur propose la mise en forme d'un message à partir d'un lexique représenté sous forme d'animations prédéfinis à l'avance. Les animations sont au nombre de 2500 et sont présentées dans leur forme de base (non fléchie) et qui sont animés par l'agent conversation (e) de la Figure 5. Les exemples de productions réalisés avec cet éditeur sont peu satisfaisants du point de vue des expressions et de la fluidité du mouvement.

Cependant la vidéo "The Forest – A Story in ASL"[23] sort totalement du lot et présente une animation remarquable par la richesse de son contenu linguistique. En effet, il semble qu'aucun point ne fasse défaut, qu'il s'agisse de la qualité des postures, des mouvements, des expressions du visage ou encore des variations prosodiques (accélérations, ralentissements, pauses, amplifications). De plus, des mécanismes linguistiques complexes sont présents tels que les transferts ou les modificateurs, grâce à une liaison directe avec des capteurs de mouvements installés sur un sourd de naissance. Enfin, l'utilisation du jeu du regard est totalement maitrisée. Bien que ce projet présente beaucoup de détails des avatars mais le manque de publications nous empêchent de savoir les techniques utilisées. On trouve seulement un brevet [52].

[22] Site Web du Vcom3D [http://www.vcom3d.com]
[23] Visible sur YouTube "The Forest - A Story in ASL" [http://www.youtube.com/watch?v=oUclQ10BsH8]

6.2.6. Le projet "WebSign"

WebSign[24] [53] est un projet évolué au sein du Laboratoire de recherche LaTICE de l'Université de Tunis. L'objectif principal de ce projet est d'établir un système de communication pour l'amélioration de l'accessibilité des personnes sourdes à l'information. Cet outil permet d'augmenter l'autonomie des sourds et n'oblige pas les non sourds à acquérir des compétences particulières pour communiquer avec eux.

Ce projet est multicommunautaire, il offre la possibilité d'inscrire un signe sous plusieurs langues. En effet, il intègre une interface interactive qui permet la création des dictionnaires. La synthèse est basée sur un avatar virtuel (voir Figure 5 (f)) [54]. WebSign intègre un système de transcription SML (Sign Modelling Language). Le noyau de synthèse leur a permis de développer plusieurs applications [55] [56].

6.3. Discussion et problématiques

Nous avons présenté dans ce qui précède quelques systèmes de synthèse vers la langue des signes. Tous les travaux présentent une qualité remarquable, en particulier la technologie des avatars qui est une solution flexible pour différentes langues des signes.

Tous ces moteurs de rendus en SL nécessitent une étape de traduction avant l'interprétation puisque l'utilisateur fournit en entrée une phrase en langue naturelle. Et cette entrée doit être traitée et traduite vers un système d'annotation pour la synthétiser en utilisant un avatar signant. Aussi, on remarque que la qualité de l'interprétation ou de la synthèse dépend du dictionnaire, qui fait toujours recours à des capteurs de mouvement afin d'avoir les positions des articulations dans l'espace de l'agent conversationnel. De plus, on remarque que différent systèmes de transcription ont été utilisés afin de construire la traduction soit pour l'avatar soit pour un affichage textuelle.

Une autre problématique se pose est l'automatisation de la traduction sans l'intervention d'un expert ou d'un développeur. C'est pour ça, nous présenterons par la suite quelques traducteurs automatiques vers la langue des signes existants.

6.4. Traduction automatique

Selon Bian Harris [57], la traduction automatique est : "*La traduction est une opération linguistique de premier niveau, c'est-à-dire une opération pratiquée par le sujet parlant sans même qu'il en soit conscient. La traduction se place ainsi au même niveau que la prononciation. Par contre, si on ne fait pas la traduction, mais par contre on en parle, lorsqu'on analyse en tant que linguiste, ce qu'est la traduction, on atteint alors un niveau égal à celui de la phonologie par rapport à la prononciation.*"

Par analogie à la langue des signes qui est différente à une langue vocale ou phonologique, la traduction est une opération de transfert vers un espace de signation en tenant compte des relations sémantiques entre les signes soit en action soit en mouvement, dans ce cas on atteint un niveau syntaxique et sémantique à la fois. La

[24] Site Web du projet WebSign [http://www.latice.rnu.tn/websign]

traduction automatique couvre plusieurs disciplines outre que le traitement automatique des langues surtout lorsqu'elle se base sur les statistiques ou sur les exemples.

Cependant, les différentes approches ne sont pas applicables pour les langues des signes, d'une part à cause de l'absence d'une grammaire de ces langues, soit à cause de manque des corpus de données pour l'extraction des connaissances afin d'utiliser des approches sémantiques. Dans ce cadre, nous allons présenter dans ce qui suit différents traducteurs automatiques vers les langues des signes. Certains traducteurs seront décrit de point de vue architecture puisqu'on a parlé du projet dans ce qui précède et d'autres seront explorés et expliqués brièvement.

6.4.1. Le système de traduction "Albuquerque Weather"

Grieve-Smith [58] décrit un système qui traduit un texte Anglais en langue des signes Américaine (ASL) dans le domaine de la prévision météorologique. Le choix du domaine permet une limite de variation de la structure de la phrase et du vocabulaire, en réduisant ainsi le nombre de règles. La phase d'analyse consiste à des représentations sémantiques plutôt que des structures syntaxiques.

Du point de vue du format de données des langues des signes, Grieve-Smith a proposé une composition d'orthographe littérale en se basant sur une technique de lustrage codée qui utilise les lettres de l'alphabet pour représenter des caractéristiques phonétiques, comme la structure de la main, le mouvement et l'emplacement.

La phase de transfert prend la forme d'une simple table de recherche où les représentations sémantiques sont mappées sur un arbre d'analyse en ASL. Cette table de recherche se compose d'un analyseur lexical où chaque élément comporte une traduction idiomatique en ASL en utilisant un système de transcription appelé "Newkirk notation" utilise l'alphabet latin pour écrire des signes[25].

L'évaluation de ce système s'est limitée sur un domaine spécifique. Aussi, la qualité de la traduction n'a pas été mesurée. Selon les auteurs, les règles de transferts sont valables que l'ASL. Et aucun agent conversationnel n'a été utilisé pour l'interprétation.

6.4.2. Le système de traduction "TEAM"

Le projet TEAM [59] a été développé en utilisant une approche de transfert pour traduire l'anglais en ASL aussi. Les données en entrée sont analysées et distinguées en informations syntaxiques et morphologiques.

Par la suite, on retrouve une représentation intermédiaire, abrégée en RI, produite sous forme de notation en gloses avec quelques paramètres incorporés, comme des expressions de visage. Au cours de la première phase étant la génération de la RI et l'analyse de l'ordre des mots en anglais, l'ordre des signes sera détecté. Puis, L'arbre grammaticale selon le "Synchronous Tree-Adjoining Grammar (STAG)" [60] est utilisé pour mapper l'information de l'anglais en ASL.

[25] Le système d'annotation "Newkirk notation" a été publié en 1986 selon les citations des travaux existants, mais tous les liens web mentionnés dans citations ne sont plus disponibles en ligne.

Dans la deuxième phase du processus de traduction, les renseignements de l'IR sont alimentés dans le signe synthétisé. Dans ce projet, l'accent est mis sur la modélisation fine des mouvements incluant les propriétés aspectuelles et les compléments de manière, grâce à une approche basée sur la notation LMA (Laban Movement Analysis) [61].

6.4.3. Le système de traduction du projet "ViSiCAST"

VisiCAST [38] a proposé une architecture permettant de passer de l'anglais écrit à la langue des signes Britannique (BSL). Ici encore, il s'agit d'une approche par transfert, au niveau sémantique [45]. L'énoncé en anglais est représenté sous forme d'un ensemble de variables et de prédicats sémantiques dans une structure DRS (*Discourse Représentation Structure* en anglais) [62] permettant de représenter des entités.

Après une conversion en une représentation de type HPSG (*Head Driven Phrase Structure Grammar* en anglais) [63], puis en une description des articulations à l'aide d'un langage de spécification dédié, les structurations proposées par HamNoSys [18] seront reprises par SiGML (Signing Gesture Markup Language) [42] pour la synthèse en trois dimensions.

Cette approche permet une plus grande richesse au niveau du type d'énoncé. L'utilisation d'un niveau sémantique permet d'éviter une grande influence de la syntaxe de l'Anglais écrit sur l'énoncé en BSL généré et de rester centré sur des énoncés de nature classique. L'architecture du système est décrite dans Figure 7.

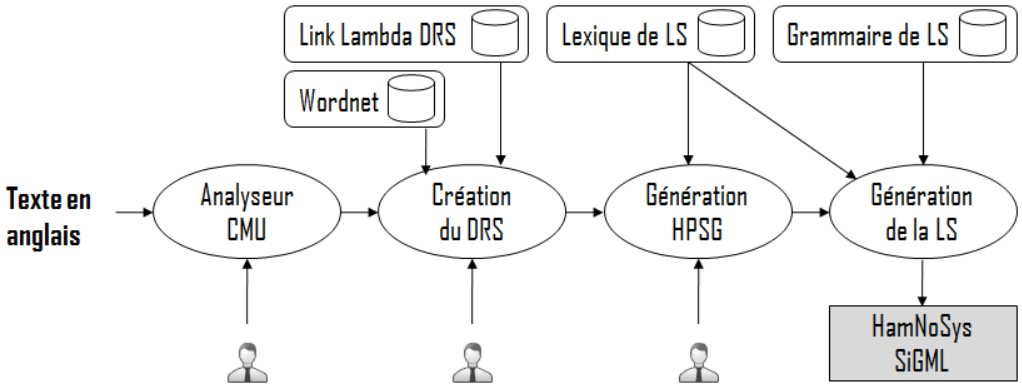


Figure 7. Architecture du système de traduction « VisiCAST »

6.4.4. Le système de traduction "ASL Workbench"

Le projet "ASL Workbench" décrit par Speers [64] a pour fonction de traduire l'anglais en langue des signes Américaine (ASL). Il incorpore la recherche moderne des modèles phonologiques de l'ASL. Dans ce projet, le modèle de notation adopté est celui de Mouvement-Hold [65].

Ce procédé divise et classifie les signes en une séquence de segments en fonction de leurs représentations phonologiques, où chaque phonème comprend une série de caractéristiques de spécification de son articulation. Aussi, ce système utilise une grammaire lexicale-fonctionnelle abrégé en LFG (*Lexical Functional Grammar* en anglais) [66] pour l'analyse du texte anglais dans une structure fonctionnelle. Cette analyse permet d'ajouter des labels fonctionnels aux constituants des arbres syntaxiques (tels que "sujet", "objet"), ou encore des caractéristiques linguistiques sur le texte (tel que "voix passive").

Dans ce projet, les règles de transfert sont faites à la main pour convertir une f-structure anglaise à une f-structure ASL.

Quant aux règles de LFG, elles permettent de produire la traduction en ASL. Lorsque le système rencontre des difficultés dans le choix lexical ou d'autres tâches de traduction, il demande à l'utilisateur du système des conseils. Le système enregistre également un modèle de discours très simpliste de l'entrée de l'anglais (composé d'une liste plate des éléments du discours et de leurs emplacements spatiaux qui sont spécifiés par l'opérateur humain). Ainsi, le système "Workbench ASL" a été conçu pour être un outil d'aide aux traducteurs professionnels dans la conversion de l'anglais à l'ASL.

En effet, le système ne peut fonctionner correctement sans intervention humaine. Il ne fait aucune tentative pour résoudre les référents du discours dans un texte anglais. Aussi, l'évaluation de ce système n'a pas été décrite.

6.4.5. Le système de traduction "SASL"

Le projet TEAM décrit ci-dessus est la base d'une partie des travaux de traduction effectués sur la langue des signes de l'Afrique du Sud abrégé en SASL au sein de l'Université de Stellenbosch [67]. Cette variante du système diffère de l'approche inter linguistique du projet TEAM en utilisant davantage de méthodologies de transfert dépendant de la langue [60]. Les arbres de grammaire de SASL et ses règles de transfert ont été construits manuellement à partir d'un ensemble de prototypes de phrases. La base de données est constituée de mots, de phrases et des listes d'annotation en SASL.

Dans la première phase d'analyse, le framework GNApp[26] a été intégré pour plus d'accessibilité. Ce framework prend en entrée les phrases en langue Anglaise qui peuvent être saisies et annotées à l'aide d'icônes en SignWriting[27]. Cette annotation est introduite dans un analyseur de phrase qui crée une arborescence d'analyse syntaxique de l'entrée étiquetée. A ce stade, l'entrée de l'analyseur syntaxique est examinée pour les objets (dans ce cas, les "objets" sont limités aux personnes) pour lesquels il peut être nécessaire d'attribuer un emplacement dans l'espace de signature de SASL.

Chaque nouvel objet est alors attribué à un emplacement dans l'espace de signation, soit en tant que lieu d'origine, soit en tant que point de référence. Une analyse plus approfondie est mise en œuvre pour incorporer les gestes non manuels dans le système. Les arbres d'entrée STAG sont étiquetés pour le contenu émotionnel ou expressif en utilisant WordNet [68]. Ces balises prendront la forme de la parole au niveau des gloses. Une autre étape de l'analyse identifie le type de la phrase, par exemple, une question ou une affirmation, pour mieux interpréter les gestes non-manuels possibles et appropriées pour la traduction en SASL.

D'après les auteurs de ce système, l'approche de traduction est proche de l'interprétation humaine puisqu'elle tient compte des relations Sujet-Objet, mais

[26] Site web du framework GNApp [http://www.napsa.org.za/gnapp.htm]
[27] SignWriting est un système d'annotation graphique pour les LS [http:// www.signwriting.org]

l'évaluation de ce système n'a pas été effectuée. Aussi, le nombre de phrases testés n'a pas été mentionné.

6.4.6. Le système de traduction "Spanish SL"

Le paradigme traditionnel de transfert de traduction automatique, texte en texte ou texte en avatar, a été étendu avec [69] pour inclure une composante de reconnaissance vocale. Dans les domaines restreints des gares, des vols et des informations météorologiques, ce système peut traduire un discours en espagnol à la langue de signes Espagnole abrégée en SSL.

En dehors du système traduction automatique, un dispositif de reconnaissance de la parole, IBM ViaVoice Edition Standard[28], est utilisé pour convertir la parole en texte Espagnol. Dans la méthodologie basée sur le transfert, l'analyseur Phoenix v3.0 [70] développé au Centre de recherche sur le langage parlé et à l'Université du Colorado, utilise une grammaire hors-contexte pour analyser l'entrée en trames sémantiques dans la phase d'analyse. L'ensemble des règles de grammaire doit être préalablement créé et chargé par le concepteur de système.

L'architecture du système est illustrée par la Figure 8. Dans la phase de transfert, les règles de grammaire sont compilées en réseaux de transition récursifs (RTN) [71] et en algorithmes de recherches d'analyse graphique pour une meilleure entrée sémantique. Le résultat de la phase de transfert est une séquence des unités d'analyse représentant le SSL où chaque unité est un concept sémantique. La phase de génération est la phase de séquence des gestes. Les règles sont limitées à des types de phrases et au contexte du domaine choisi. La sortie finale du système est une synthèse via un agent conversationnel qui anime le squelette à partir des gestes déjà sauvegardés à l'avance.

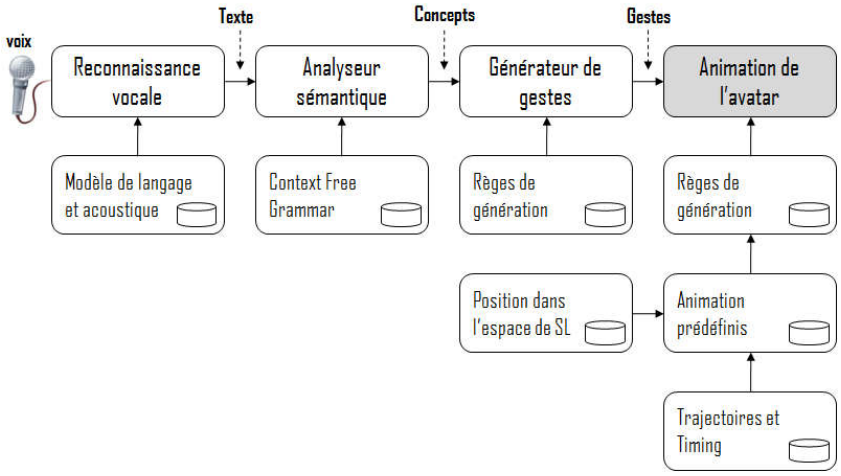


Figure 8. Architecture du système de traduction "Spanish SL"

6.4.7. Le système de traduction "MaTrEx"

Sara Morissey était la première qui a proposé une approche statistique pour la traduction automatique d'une transcription en langue des signes Irlandaise (ISL) vers un

[28] Site web de l'IBM ViaVoice [http://www-306.ibm.com/software/pervasive/embedded viavoice]

texte Irlandais [72] (le sens inverse des approches proposés précédemment). Le projet nommé "MaTrEx" prend comme mémoire de traduction un corpus en gloses annotés manuellement avec l'outil ELAN [22]. Par exemple, la phrase "*Is flight BA a round trip*?" sera transcrite en "IX-FLIGHT FLIGHT BA ROUND TRIP IX-FLIGHT palm-up".

Toutes les phrases du corpus sont couplées avec une transcription en gloses. Ensuite, pour chaque nouvelle entrée en ISL, le système cherche la traduction la plus probable par phrases ou sous-phrases en Irlandais [73]. Puis, une reformulation a été effectuée pour corriger la structure. Les résultats étaient encourageantes, selon le score BLEU [74] calculé, ce qui a mené à l'appliquer sur quatre paires de langues : ISL vers l'anglais, ISL vers l'allemand, DGS vers l'allemand et DGS vers l'anglais. Mais, la problématique posée par ce système est la taille du corpus qui n'a pas dépassé mille phrases en total.

6.5. Discussion et problématiques générales

Un aperçu de chaque système et de ses caractéristiques ont été présenté dans les sections précédentes. Différentes approches ont été développées allant des transferts directes vers la langue des signes jusqu'à des approches statistiques qui minimisent l'intervention des experts ou des interprètes. La problématique principale de la traduction automatique est l'absence d'un score pour se comparer et le manque de ressources linguistiques pour le traitement automatique.

Aussi, les systèmes d'annotation et de notation utilisée lors des transferts ne sont pas efficaces puisqu'ils ont été proposés par des linguistiques voulant à modéliser la langue des signes sans tenir compte des aspects de traitement informatique. Pour l'évaluation, la seule méthode est de faire appel aux sourds de naissance pour noter la qualité de synthèse générée à partir d'un agent conversationnel.

7. Conclusion

Selon l'étude, le traitement automatique des langues des signes couvrent plusieurs disciplines du traitement linguistiques allant de la création des corpus jusqu'à la traduction et la synthèse. Toutes les disciplines posent plusieurs problématiques dû au manque de publications et de références de ce domaine, puisque la première initiative pour la modélisation des langues de signes était dans les années 60 avec Stokoe [75] [76].

D'autres axes de recherches sont en cours de développement concernant le traitement automatique des langues des signes outre qu'on a étudié tel que la reconnaissance des gestes et des mouvements à partir des vidéos [77] ou à partir des séquences 3D animés par des agents conversationnels [78].

Cette thèse démarrera à partir des problématiques existantes afin de proposer une nouvelle approche de modélisation et de traduction vers et à partir des langues des signes. Dans le deuxième chapitre, nous commençons par présenter les différents systèmes de transcription existants pour les langues des signes et proposer en troisième chapitre notre système pour la traduction automatique basé sur l'annotation en gloses.

Chapitre 2 :

Systèmes de transcription des langues des signes

1. Introduction

Les langues des signes (SL) représentent la première langue des personnes sourdes. Elles complètent les langues naturelles et elles sont communiquées par une modalité Visio-gestuelle. Ce chapitre portera sur les concepts, les structures et les notations basiques de la SL. Dans un premier temps, nous décrirons notre problématique, nos objectifs ainsi que la façon dont s'organisent nos recherches pour mener à bien cet objectif et aboutir à des résultats satisfaisants. Dans un second temps, nous investirons les conceptions erronées communes aux SL et explorerons la façon dont celles-ci sont nées.

Les raisons de la naissance de ces idées fausses seront expliquées. Puis, dans un troisième temps, nous mettrons l'accent sur la structure linguistique des SL. Enfin, un aperçu des notations et des annotations des SL pouvant être utilisées pour faire progresser l'étude des SL seront données. Les méthodologies adoptées seront examinées selon leurs avantages et inconvénients respectifs.

2. Problématiques

Les SL représentent une langue Visio-gestuelle composée de fonctions manuelles et non manuelles [79]. Les fonctions manuelles font référence à une configuration à la main. Les fonctions non manuelles font référence aux expressions faciales et aux mouvements corporels. Ils utilisent une combinaison de mouvements des mains et d'autres encore, à savoir, le mouvement des lèvres, l'expression faciale et le langage corporel, et cela afin de communiquer et transmettre des informations. Ces langages sont développés à cause des difficultés de communication des personnes sourdes face à d'autres personnes sourdes mais aussi face à des personnes non sourdes. Ils constituent un langage de communication gestuelle, complexe et riche. Ils sont des langages quasi-complets et naturels avec leurs morphologies[29], phonologie[30] et syntaxe[31] spécifiques.

En 1960, les travaux de Stokoe [75] avaient montré que les SL sont des langages réels et indépendants et ont ainsi fait valoir que les SL constituent la langue maternelle des personnes sourdes. Avant ses travaux, les SL n'étaient nullement considérées comme des langues réelles. Elles étaient considérées comme un ensemble de gestes sans signification. Puis, une SL ne pouvait pas être utilisée dans l'éducation des enfants sourds.

[29] Morphologie se réfère à la manière dont les parties de signes sont disposés à faire une phrase.
[30] Phonologie se réfère à la manière dont les phonèmes sont combinés.
[31] Syntaxe est une règle qui régit la façon dont les signes sont disposés pour former une phrase.

D'autres méthodes étaient utilisées tel que l'oralisme, la communication totale et la méthode de culture bilingue :

- Oralisme : L'oralisme est la pratique de communication par la lecture labiale (des lèvres) [80] et les paroles uniquement. Il n'y a pas de SL utilisée ici. Cette méthode inclut que les personnes sourdes doivent pratiquer les sons de la parole.
- Communication totale : La communication totale[32] inclut l'utilisation de toute forme de communication : langue des signes, fingerspelling, lecture labiale et images. La communication totale a été critiquée pour sa pauvre qualité en SL. La méthode a été considérée inefficace car la signation et les paroles ont produit une langue des signes très faibles.
- Méthode de culture bilingue : La méthode de culture bilingue inclut l'enseignement du SL en tant que première langue, et l'anglais (par exemple) en tant que seconde. Dans cette méthode, les SL sont utilisées pour enseigner les langages parlés. Donc les étudiants apprennent le SL et le langage parlé en parallèle.

D'où notre objectif est la création d'une forme écrite de la SL qui répondrait aux attentes des programmeurs en traitement automatique des langues des signes :

- De manière indirecte, par son usage dans les systèmes de transcription (pour prendre des notes, utiliser des documents en SL), elle pourrait aider à réduire l'illettrisme chez les sourds grâce à une forme écrite correspondant à leur langue.
- Elle rapprocherait les sourds des non sourds en permettant aux sourds d'utiliser des dispositifs usuels de transfert différé (ex : livres, ordinateurs, etc.) au lieu de dépendre exclusivement de vidéos.
- Elle agrandirait les communautés linguistiques grâce à ses dispositifs de transfert différé permettant un plus grand accès aux connaissances disponibles, et facilitant des contacts multimédias à faible bande passante.

On pourrait croire que les avancées de ces dernières décennies dans le domaine de l'informatique ont facilité le stockage de la connaissance sous forme de synthèses vidéos ou de personnages virtuels, avec notamment des encodages numériques donnant des fichiers plus compacts, et de vastes espaces de stockage.

3. Les conflits et ses origines

La section précédente a démontré que les SL représentent des langages naturels nécessitant d'avoir une forme écrite. La section qui suit étudiera les conflits communs aux SL. Depuis la découverte de Stokoe [76], ces derniers ont été reconnus en tant que véritable langage, possédant une complexité grammaticale et expressive. Les écoles enseignent les SL et des recherches sur le SL sont réalisées dans de nombreux pays.

Cependant, beaucoup de conceptions erronées persistent encore, même au sein de la communauté sourde elle-même. Ces conceptions erronées seront énoncées dans les paragraphes ci-dessous.

[32] Définition extraite du site [http://www.handsandvoices.org/comcon/articles/pdfs/totalcomm.pdf]

3.1. L'universalité

Les SL ne sont pas universelles. Si cela était le cas, la barrière de communication entre les communautés sourdes serait absente. Il existe une multitude des langues des signes qui ont été développées indépendamment aux quatre coins du monde. Par exemple, ils sont au nombre de 25 en Afrique. En outre, l'ethnologue des langues mondiales répertorie 121 langues des signes.

La plupart de ces langues sont distinctes. A cause de la diversité de culture et de milieu, les SL varient d'un pays à un autre. Par conséquent, il existe une langue des signes Thaïlandaise (TSL), une langue des signes Américaine (ASL), une langue des signes danoise (DSL), une langue des signes Italienne (LIS, *Lingua dei Segni Italiana*), etc., comme on a vu dans le chapitre précédent. Les signes peuvent être iconiques, mais restent différents. Prenons par exemple le signe "STUDENT". En ASL, les mains en "open-B"[33] sont déplacées vers la gauche pour décrire la taille de l'enfant [81]. En LIS, la main représente la forme "OpenB" au milieu du corps et effectue un mouvement vers l'intérieur. En TSL, la main prend la forme "Flat-C"[34] en haut du visage du signeur et effectue un mouvement au milieu du visage. Les différents signes du signe "STUDENT" sont illustrés par la Figure 9 [79].

Par ailleurs, il pourrait y avoir quelques variations des SL dans un même pays. En Afrique du Sud par exemple, la variation du lexique diffère selon les régions, mais la structure grammaticale reste la même.

Figure 9. Signe "STUDENT" dans les langues des signes Américaines, Italiennes et Thaïlandaise

3.2. Les langues vocales

Les SL ne dépendent pas des langues vocales LVs. Elles ont une grammaire riche et complexe. Elles ne sont pas non plus des épellations des LVs. Dans la SL, l'épellation est surtout utilisée pour les noms de personnes, les lieux géographiques et pour les termes scientifiques, empruntés à la LV. Les SL tirent parti des fonctionnalités uniques du support visuel, et sont donc Visio-gestuelles.

En revanche, les langues vocales sont orales, auditives et linéaires. Au sein des LVs, la plupart des mots et des objets référents ont une relation arbitraire. Par exemple, la taille, la prononciation ou encore l'épellation du mot voiture ("*car*" en anglais), ne sont

[33] Le "Open B" (ou "Open Hand"): Les quatre doigts sont étendus ensemble et le pouce est prolongé pour faire une main à plat avec un espace entre le pouce et le tranchant de la main. Cette forme est également connue comme "extended-B", "open-hand" et "closed-5".
[34] La liste complète des configurations des mains de l'ASL est décrite dans l'annexe E.

pas reliées à l'objet de référence. En contraste, les SL sont hautement iconiques : la plupart des signes ressemblent à l'objet de référence. L'espace grammatical complexe des SL diffère de celui des LVs. Par exemple, en anglais le verbe se place avant l'objet : "*I am reading a book*". Dans certains SL, le verbe se place après l'objet : "I BOOK READ".

Cependant, Il existe des SL qui peuvent être considérées comme une représentation visuelle des LVs en utilisant la même grammaire et la même morphologie, dans ce cas, ces signes sont utilisés dans un ordre de mots exact. En ASL par exemple, les différents ordres des mots doit être défini à l'avance lors du discours, puisque tous ces ordres suivants sont permises : SVO, SOV, OSV, OVS, VOS, VSO, tel que S est le sujet, O est l'objet et V est le verbe [82].

Par analogie aux langues vocales, la phonologie en langue des signes existe aussi, comme les grammaires. Dans les LV, la phonologie, qui l'étude des plus petites entités de langue, est l'ensemble des voyelles et consonnes, et en langues des signes, c'est l'ensemble des mouvements et des postules selon Liddel et al. [83].

3.3. L'iconicité

La section précédente a indiqué que la SL est une langue iconique. L'iconicité ne se réfère pas à pointer sur un objet (par exemple, montrant une voiture à proximité) ou à agir sur des événements réels (par exemple, sauter de haut en bas). Elle est une ressemblance entre un terme linguistique et sa signification. Elle est présentée à la fois dans les SL et les LVs, mais plus fréquemment au sein des SL.

Cela pourrait être dû au fait que dans le monde réel, il y a davantage d'images visuelles que de sons pouvant se rapporter clairement à un concept. Par exemple, une table peut être caractérisée par sa forme. La modalité Visio-gestuelle fournit un certain nombre de moyens pour rendre les images visuelles plus aisément reconnaissables. L'iconicité n'est pas toujours présente (certains signes n'ont aucune iconicité de ce qu'ils représentent).

La formation des signes au sein n'est pas seulement déterminée par leur ressemblance avec un objet ou une action. Celle-ci peut se transformer en des symboles arbitraires au cours du temps [69].

3.4. Les standards d'écriture

Les SL n'étaient pas considérées comme de véritables langues car elles ne disposaient pas d'une forme écrite. Il n'existe actuellement aucune forme écrite de SL acceptée universellement, mais il y a des moyens pour transcrire ce qu'une personne signe. De nombreuses notations SL ont été proposées. Certaines d'entre elles sont devenues largement utilisées par les chercheurs SL et d'autres sont utilisées en tant qu'outils pédagogiques. La SL peut être écrite à l'aide de glose[35] ou encore à l'aide de notations. La glose utilise des étiquettes de la langue parlée afin de traduire des signes.

[35] La glose est une explication qui éclaircit le sens d'un texte [http://www.linternaute.com]

Par exemple, la phrase an Anglais "*What is your name ?*" sera transcrite en "NAME YOU WHAT" puisque la personne sourde ou l'interprète ne signera pas la phrase mot-à-mot, mais plutôt la reconstruire selon les sujets, les objets et l'action généré par le verbe. Cette méthode est à l'origine de notre étude. En ce qui concerne la notation, celle-ci utilise des symboles spéciaux pour décrire les paramètres physiques de SL.

Parmi les exemples de notations SL, nous trouvons la notation de "Stokoe", le système de notation Hambourg "HamNoSys", le système "SignWriting" et d'autres. Toutes ces notations seront discutées dans ce qui suit.

Les méthodologies sont très importantes pour les traitements automatiques des langues des signes, sans elle, aucun processus de traitement ne peut être effectué. Mais, ce qu'on va remarquer que certaines notations sont graphiques et d'autres sont textuelles.

4. Structure de la langue des signes

Cette section étudie la structure ainsi que les phonèmes des SL. Dans les LVs, un phonème est le plus petit élément de la parole qui distingue un sens. De même, un signe dans la SL est composé d'un phonème. Les phonèmes de SL sont divisés en cinq composantes : la configuration, l'orientation, l'emplacement, le mouvement et les expressions du visage [10].

Ces dernières se produisent simultanément et distinguent les signes les uns des autres, tel que les phonèmes dans les langues parlées distinguent les mots les uns des autres. Chacun des phonèmes de la langue des signes sera discuté dans les paragraphes ci-dessous.

4.1. Configuration

La configuration correspond à la forme de la main définie par les doigts. C'est le paramètre le plus évident d'un signe. Chaque configuration a une appellation donnée selon qu'elle fasse partie de la dactylologie (alphabet : A, B, C…), des chiffres (1, 2, 3….). L'alphabet est prolongé par des modifications de la configuration à la main tels que : bent-B[36] ou à flat-O[37]. Les signes sont basés sur la première lettre d'un mot anglais en ASL (voir la liste complète des configurations dans l'annexe A).

Figure 10. Le "fingerspelling" de la phrase "I hope you have a great day" en ASL

[36] Les doigts sont étendus et se pencha à la base
[37] Quatre doigts sont courbés directement à la base avec l'index et la majeure touche le pouce

Cela montre le rôle significatif de la configuration à distinguer les signes les uns des autres. En outre, certains signes peuvent être similaires et ne diffèrent que par la forme de la main. L'importance de la configuration peut être vue dans le "fingerspelling" (voir Figure 10) dans lequel le signe est inconnu et les lettres du mot à signer sont précisées à sa place. Compte tenu de l'importance de la configuration dans un signe, il est essentiel qu'elle soit correctement rendue.

4.2. Orientation

L'orientation de la main est définie par deux axes de la main, soit horizontal, soit vertical [84]. Il existe 6 directions de la paume de la main. Elles peuvent varier durant l'exécution des signes. Ces différentes orientations sont définies comme suit :

- Vers le haut : c'est-à-dire la paume est orientée vers le haut et le dos de la main est orienté vers le bas.
- Vers le bas : c'est-à-dire la paume est orientée vers le bas et le dos de la main est orienté vers le haut.
- Vers la droite : lorsque la tranche de la main est orientée vers le locuteur.
- Vers la gauche : lorsque la tranche de la main est orientée vers le locuteur.
- Vers soi : lorsque la paume est orientée vers le locuteur, on parle de « supination ».
- Devant soi : lorsque le dos de la main est orienté vers le locuteur, on parle de « pronation ».

4.3. Emplacement

L'emplacement est la position de la main par rapport au corps. La plage d'emplacement dans la SL est limitée par l'espace accessible par la main. Sa localisation est presque toujours au-dessus de la taille et dans la zone en face du signataire et de chaque côté de la tête. Dans la plupart des cas, l'emplacement de la signation contribue à sa signification. Par exemple, la Figure 11 est une interprétation de la phrase " Viens, c'est l'heure du bain" qui sera transcrite en "HOUR NOW BATH COME".

Figure 11. L'emplacement des mains dans l'espace du signeur

Dans cet exemple, l'emplacement des mains est au milieu du signeur sans faire référence à des objets.

4.4. Mouvement

Le mouvement de la main est un paramètre clé de la langue des signes. Il correspond à la trajectoire décrite par la main (ligne, arc de cercle, etc.). Ce paramètre est réalisé par les doigts, les mains, les poignets, les bras, séparément ou ensemble.

De nombreuses significations des signes peuvent être décrites par le mouvement. Ce mouvement peut être utilisé afin de ressembler à l'acteur ou à l'action. En ASL, les signes "HELP-ME" et "HELP-YOU" sont similaires mais diffèrent en mouvement. Dans le signe "HELP-ME", le mouvement vers le signataire indique que le signataire est l'offreur de l'aide. Dans le signe "HELP-YOU", le mouvement s'éloigne du signataire. Aussi, la taille d'un référent peut être décrite par la taille du mouvement. De même, la vitesse et le rythme du référent peuvent être indiqués par un mouvement. Par exemple, le mouvement sera lent pour une action lente, et rapide pour une action plus rapide. Le taux d'apparition ou "de la pluralité" peut être indiqué par un mouvement répétitif.

4.5. Gestes non manuels

Les SL n'impliquent pas seulement les mouvements des mains, mais également les mouvements d'autres parties du corps, par exemple les épaules, la tête ou encore les mimiques du visage, des yeux, de la bouche, des joues, des sourcils. Ceux-ci renseignent très souvent l'interlocuteur sur la vitesse de l'action, la taille d'un objet, ou la quantité.

Nous appellerons ces mouvements "les gestes non manuels" abrégés en GNM. Des modifications fines sont à l'origine de différences sémantiques importantes. Ainsi, par exemple, selon que les joues sont plus ou moins gonflées et les yeux plus ou moins ouverts, on exprimera telle ou telle nuance qualifiante (voir Figure 12).

Ces expressions jouent de multiples rôles dans les énonces en SL : désambigüisation de signes, modes de discours (interrogation, négation, conditionnel, etc.). Par exemple, les questions "yes/no" nécessitent une forte hausse de sourcils et que la tête bascule en avant. Et, pour les questions "Wh-question", comme "who?", "what?", "when?", "where?", "why?", "which?" et "how ?", utilisent des expressions faciales interrogatives illustrées par un abaissement des sourcils.

Tout changement ou absence de gestes non manuels peut exprimer un sens entièrement différent et même dans certain cas opposé.

Figure 12. Exemple de gestes non-manuels dans les signes "SICK", "HAPPY" et "NOT-LIKE"

5. Les systèmes de transcription des langues des signes

Compte tenu de la structure de la SL et de l'importance des phonèmes dans la section précédente, les langues parlées sont principalement séquentielles. Autrement dit, la majorité des phonèmes produisent un mot. Les mots structurent une phrase et les phrases donnent naissance à un texte. Ceux-ci se produisent donc en une seule séquence, l'un après l'autre. Par conséquent, les langues parlées peuvent être écrites sous forme séquentielle.

Cependant, la langue des signes est constituée de deux types de signes : manuels et non-manuels, tel que nous l'avons indiqué dans ce qui précède. Ces signes sont définis simultanément par le corps, les mains, ou encore le visage. Ainsi, contrairement aux signes traditionnels d'écriture, un système de transcription de SL doit être en mesure de représenter l'aspect non séquentiel des signes.

Ces systèmes de SL sont importants puisque les gestes de SL exécutés par un avatar dépendent d'eux. Il existe deux types de système de transcription pour la SL : les systèmes de notation et les systèmes d'annotation. De manière assez générale, deux courants semblent caractériser les divers systèmes.

D'une part les systèmes qui tentent d'utiliser un symbolisme existant et d'autre part ceux qui en inventent un propre, par exemple :

- notation HamNoSys [18] : système créé pour la représentation écrite d'un très grand nombre de signes.
- notation Jouison [85] : la finalité de ce système est de représenter tous les comportements de communication d'un locuteur de langue des signes.
- notation Laban [86] : ce système de transcription est originellement produit pour consigner les mouvements d'un danseur plutôt que ceux d'un locuteur de langue signée. Toutefois, une adaptation pour la transcription des signes indiens de l'ethnie Nakota/Assiniboine a été réalisée par Farnell [87]. Le symbolisme utilisé est inventé et n'a aucun lien direct avec un quelconque autre système ;
- notation Liddell et Johnson [83] : ce système isole dans une cellule chaque constituant du signe proposé sous la forme d'une grille ;
- Signfont de Newkirk [88] : c'est un système graphématique composé de symboles de mouvements pour fournir aux langues signées un système d'écriture ;
- SignWriting de Sutton[38] [89] : ce système de transcription permet plusieurs niveaux de représentation allant de la description phonétique à l'écriture sténographique.

Les systèmes de notation et d'annotation doivent avoir les spécificités suivantes :

- avoir la capacité de transcrire aussi bien des signes isolés que des phrases entières ;
- séparer sur deux lignes la représentation des configurations manuelles de chaque main lorsque la production du signe en implique deux ;
- contenir une représentation de l'expression faciale et de manière plus générale de tous les comportements dits non manuels ;
- fournir une représentation des propriétés prosodiques du signe, et en particulier des modalités détaillées du mouvement ;
- utiliser des symboles, qu'ils soient iconiques ou alphabétiques, aussi mnémotechniques que possible ;
- offrir deux possibilités descriptives dont l'une serait interne et concernerait en somme le mode articulatoire et l'autre serait externe et appartiendrait davantage au mode de perception des signes.

[38] Site Web de la communauté de SignWriting [www.signwriting.org]

5.1. Un système de notation

La notation [90] est définie comme étant l'écriture des mots ou des signes d'un texte qui a été signé ou bien parlé, ou encore comme un système actuel de symboles utilisés à cette fin. L'un des objectifs majeurs des systèmes de notation est de permettre au lecteur de reproduire des symboles graphiques, avec une précision plus ou moins grande selon le degré de détail de la notation, le texte original parlé ou signé.

La Figure 13 est une interprétation de la phrase "AGO THREE-WEEKS I SWITZERLAND GOING-TO" représentée dans le système de notations HamNoSys que nous détaillerons ultérieurement.

Figure 13. Représentation d'une phrase en utilisant la notation HamNoSys

D'une manière générale, la notation est une sorte de grille d'analyse qui permet de décrire aussi fidèlement que possible et dans l'optique que l'on s'est choisie les données observées, tels que la création de scripts aux systèmes d'écriture pour l'analyse phonologique ou l'analyse grammaticale. Ils servent aussi en tant que formes écrites des textes sources qui sont lisibles par machine, par conséquent, susceptibles d'être traitées par ordinateurs.

Ces systèmes de notation de la gestuelle souffrent quant à eux, soit d'un manque de popularité, soit d'un manque de standardisation.

5.2. Un système d'annotation

Au départ, une annotation est une sorte de commentaires supplémentaires, à ajouter à un texte déjà existant et écrit, qu'il s'agisse d'une transcription d'un texte parlé ou d'un fragment de l'écriture conventionnelle pour une langue signée. Les annotations étaient souvent inclues pour aider à la compréhension.

Concernant les linguistes, les annotations sont transformées en des mini commentaires linguistiques qui sont ajoutés aux unités identifiées dans une langue. Elles ajoutent des informations pertinentes à une unité de la langue : phonologique, morphologique, syntaxique et sémantique, en fonction de l'objet de l'analyse.

Ainsi, l'annotation se réfère au type d'annotations automatiques ajoutées aux textes écrits, après qu'elles aient été numérisées puis traitées à l'aide d'ordinateurs. Bien que certains systèmes particuliers ont été inventés et sont utilisés dans certaines petites communautés, nous allons présenter par la suite quelques systèmes qui sont généralement utilisés par les chercheurs.

5.3. Système de notation de "Stokoe"

Le système de notation Stokoe a été développé en 1960 par William Stokoe [75] pour écrire la langue des signes Américaine (ASL), et plus tard adapté pour rédiger les langues des signes Britanniques et Australiennes. Il a été le premier script créé pour l'écriture phonétique d'une langue signée, même si son but premier n'était pas de servir aux sourds à noter leur langue, et qu'il n'a d'ailleurs jamais été utilisé à cette fin.

Au contraire, il a toujours été consacré à la recherche et à l'analyse linguistique. Stokoe fut le premier linguiste à étudier et proposer une description de la langue des signes en termes de phonèmes qu'il dénomma "chérèmes". Il décrit initialement trois facteurs, à savoir la tabulation (tab) se référant à l'emplacement d'un signe, le désignateur (dez) se référant à la forme de la main, et les signassions (sig) se référant au type de mouvement à articuler.

Un exemple de la notation Stokoe pour le signe "DONT-KNOW" est illustré par la Figure 14. Cette notation utilise l'alphabet latin et les chiffres pour représenter la forme de la main. Les icônes Glyphe sont utilisées pour représenter les positions des mains et des mouvements. Le premier symbole indique le lieu d'articulation qui est sur le front. La lettre "B" indique que la main prend la forme B (voir configuration des mains dans l'annexe A) et la lettre T indique l'orientation de la main, étant la paume tournée vers le signataire. La lettre "x" indique la présence d'un contact. Puis, la lettre "ɒ" indique que la paume est tournée vers le bas. Le "T-renversé" indique que la paume de la main est opposée au signataire.

Figure 14. Notation du signe "DONT-KNOW" en "Stokoe"

Cette notation s'écrit de façon linéaire de gauche à droite. Les symboles sont disposés dans un ordre standard : l'emplacement, la forme de la main, l'orientation et le mouvement. Cela diffère de la notation "SignWriting" (à base de caractères iconographiques) qui se note de haut en bas. Ce système de notation est une description graphique où les signes sont des images stockées, ce qui rend difficile la reconnaissance du signe par une machine, à partir de l'image. Aussi, la notation Stokoe ne tient pas compte des gestes non-manuels tels que les expressions faciales et le regard.

5.4. Système de notation "SignWriting"

Une méthode alternative a été développée par Valerie Sutton qui est appelée "SignWriting". Cette approche décrit également la langue des signes phonologiquement, mais, à la différence des autres, elle a été développée en tant que système d'écriture visuel qui permet de lire, d'écrire et de dactylographier (taper à la machine) n'importe quelle langue des signes du monde.

Ce système utilise des symboles visuels pour représenter les configurations, les mouvements et les expressions faciales de toutes les langues des signes. Ces dessins au

trait simple, comme montrés dans la Figure 15, rendent le système plus facile à apprendre, car ils sont plus intuitifs et visuels, puis reliés aux signes eux-mêmes.

Figure 15. Notation du signe "DEAF" en "SignWriting"

Le système "SignWriting" est aujourd'hui enseigné aux enfants sourds et aux adultes du monde entier en tant que version manuscrite de la langue des signes. Le système n'est pas encore largement utilisé, mais son ergonomie et la vitesse à laquelle elle a été adoptée a donné lieu à la publication de livres pour l'apprentissage du système. Le détail linguistique de la langue des signes est encodé dans la ligne des dessins. Ce système a présenté une solution efficace pour exposer plusieurs langues des signes de différentes communautés, comme l'Italie[39], le Brésil[40] ou encore la Tunisie[41].

La nature graphique de "SignWriting" consiste en fait en une notation pratique pour une compréhension et une lecture humaine faciles. Cependant, cette nature graphique rend aussi la notation difficile à analyser avec un ordinateur. La notation devra être adaptée pour utiliser uniquement des caractères ASCII standard ou être codée dans un langage de balisage comme XML afin d'être pratique pour l'animation par ordinateur.

Une telle représentation efficace de symboles graphiques dans un système informatique devrait faciliter cette tâche. À cette fin, "SignWriting Markup Language" abrégé en SWML, un format basé sur XML, a été proposé. Cette notification est devenue plus exploitable par des logiciels comme VSigns qui utilisent des ressources codées en SWML pour générer une animation via un avatar. Un exemple de notation est illustré par la Figure 16.

```
1  <signbox>
2      <seq>01-05-034-01-02-01</seq>
3      <seq>02-08-012-01-01-02</seq>
4      <seq>04-04-001-01-01-01</seq>
5      <sym width="36" height="35" left="-18" top="-18">04-04-001-01-01-01</sym>
6      <sym width="17" height="20" left="-8" top="19">01-05-034-01-02-01</sym>
7      <sym width="26" height="23" left="-10" top="-43">02-08-012-01-01-02</sym>
8  </signbox>
```

Figure 16. Le fichier SWML pour le signe "DRINK"

La Figure 16 montre le fichier SWML correspondant au signe "DRINK", tel qu'il est stocké dans un fichier texte SW généré par l'éditeur SignWriter. Chaque élément d'un signe SW est nommé "symbole". Ces symboles sont codés par une séquence de 6 entiers permettant de retrouver l'équivalent du caractère correspondant.

Bien que les langues signées soient d'une densité sémantique supérieure aux langues écrites, si l'on faisait une comparaison entre les lettres et les symboles, puis entre

[39] Groupe de travail sur SignWriting en Italie : [http://www.signwriting.org/italy]
[40] Groupe de travail sur SignWriting en Brésil : [http://www.signwriting.org/brazil]
[41] Groupe de travail sur SignWriting en Tunisie : [http://www.signwriting.org/tunisia]

les signes SW et les mots, il est ici nécessaire d'utiliser une ligne pour chaque lettre, et en moyenne 8 lignes pour chaque mot. Cet encodage est donc très complexe d'un point de vue informatique, mais aussi d'un point de vue du temps de calcul et de l'espace de stockage nécessaire.

De plus, pour un traitement informatique, on fait recours aux techniques de traitement d'images puisque le fichier SWML ne contient aucune information linguistique, seuls les emplacements dans l'espace sont décrits.

5.5. Système de notation "HamNoSys"

Un autre système de notation explicite et graphique pour la langue des signes est le système de notation de Hambourg "Hambourg Notation System"[42], dit "HamNoSys" [18]. Ce système utilise un jeu de symboles indépendants de la langue qui représentent les caractéristiques phonologiques de la langue des signes. Il est enraciné du système de Stokoe décrit précédemment. Il présente plusieurs détails mais, dans la plupart des cas, cela est seulement une description de la forme manuelle.

Dans la Figure 17, nous montrons les symboles qui dénotent le signe "HISTORY" :

Figure 17. Notation du signe "HISTORY" en HamNoSys

Mais les notations décrites ci-dessus sont purement scripturales et leur iconicité est généralement faible. Bien adaptées pour constituer des bases de données à des fins d'études linguistiques, elles révèlent leurs limites dans la représentation visuelle de la langue des signes car l'essence de cette dernière est spatiale et dynamique et a des caractères difficilement transposables sur le papier. Par conséquent, comme ce système est difficile à transcrire, il ne pouvait être destiné à la communication quotidienne dans la SL, tel que la rédaction de lettres ou de journaux. Il visait à fournir un support écrit pour les chercheurs sur les langues des signes afin de les stocker. Aussi, ce système est indépendant de la langue et applicable à tout SL.

L'inconvénient majeur de cette codification est qu'elle nécessite d'y consacrer une période de formation pour en avoir une bonne maîtrise des bibliothèques ainsi que de l'utilisation du clavier qui se fait via cinq niveaux pour accéder aux symboles. Il existe cependant une interface de saisie par sélection des codes organisés en groupes logiques.

HamNoSys est un système souvent utilisé comme point de départ des systèmes de reconnaissance et de génération automatique (par exemple SiGML décrit ci-dessous) puisque son encodage informatique est relativement aisé. Cette initiative a pour but principal de générer une animation via des agents conversationnels en trois dimensions.

[42] Site web du Système de notation "HamNoSys" de l'université d'Hamburg http://goo.gl/ih6s5T

5.6. Système d'annotation « SiGML »

Le système "Signing Gesture Markup Language" dit SiGML [42] a été initialement développé dans le projet ViSiCAST [38] qui cherche à élaborer un ensemble d'outils servant à la synthèse des gestes de la langue des signes.

Dans le contexte de ce projet, un langage de description SiGML basé sur le langage textuel XML a été développé. Ce langage s'appuie sur la notation HamNoSys. SIGML présente une décomposition de la description d'un signe similaire à celle d'HamNoSys [18]. Celle-ci s'effectue en une structure qui représente dans sa partie manuelle des fonctionnalités, tels que la forme de la main, l'orientation, l'emplacement, et le mouvement. Il s'agit d'une catégorie qui inclut non seulement des changements de localisation, mais aussi de forme et d'orientation.

Ce système contient également des opérateurs permettant la description des signes pouvant être abrégée de diverses manières, par exemple en spécifiant une symétrie entre les mains en signe à deux mains, et en précisant les différentes formes de mouvements répétées dans un signe. SiGML suit aussi HamNoSys en fournissant une composante non-manuelle spécifiant le comportement des articulateurs du visage, tel que la bouche, les yeux, les sourcils, le regard, mais aussi les mouvements non manuels corporels, tels que les clins d'œil, les hochements de tête et l'inclinaison du corps. La Figure 18 expose le signe "GOING-TO" en DGS selon le système d'annotation SiGML.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE sigml SYSTEM "http://www.visicast.cmp.uea.ac.uk/sigml/sigml.dtd">
<sigml>
    <hamgestural_sign gloss="going_to_DGS">
        <sign_manual both_hands="true">
            <handconfig handshape="finger2" thumbpos="out" />
            <handconfig extfidir="uo" palmor="1" />
            <par_motion>
                <directmotion curve="u" direction="o" />
                <tgt_motion>
                    <change_posture />
                    <handconfig extfidir="do" />
                </tgt_motion>

            </par_motion>
        </sign_manual>
    </hamgestural_sign>
</sigml>
```

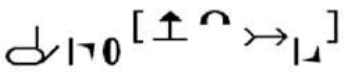
: 

Figure 18. Notation du signe "GOING-TO" en SiGML et en HamNoSys

Cette approche a l'avantage d'avoir à la fois une annotation qui est facilement compréhensible aussi bien par les lecteurs humains que par les ordinateurs. Mais, son inconvénient est la nécessité d'une étape supplémentaire de la traduction.

5.7. Système de notation "Movement-Hold"

Le système de notation "Movement-Hold", abrégé en MH, a été créé en 1989 par Scott Liddell et Robert Johnson [83]. A l'opposé du modèle de "Stokoe" [75] qui utilise des symboles graphiques, ce système utilise des mots anglais pour décrire les actions physiques de signature. Le modèle MH propose la composition d'un signe par une séquence de segments de postures ("*holds*" en anglais) et de mouvements ("*movements*" en anglais), cela étant analogue aux consonnes et voyelles dans les langues vocales :

- Les postures (symbole H) sont définies comme étant les moments pendant lesquels toutes les caractéristiques manuelles demeurent constantes, en particulier la position de la main, la configuration et l'orientation.
- Les mouvements (symbole M) caractérisent au contraire une phase de transition entre deux ensembles de caractéristiques articulatoires.

En conséquence, un mouvement M est relié à deux de ces ensembles, tandis qu'une posture H n'est connectée qu'à un seul. Ainsi, le système MH isole dans une cellule chaque constituant du signe proposé sous forme d'une grille. Entre autres, il préfère les abréviations à l'emploi d'un symbole unique pour représenter les différents aspects du signe, et il réserve une place privilégiée à la caractérisation de la segmentation prosodique du signe.

La notation des configurations manuelles fait davantage état des constituants permettant sa formation. Ces segments contiennent, par conséquent, toutes les informations nécessaires pour décrire un signe tel que les formes, les lieux, les orientations de la main, mais aussi les composants non-manuels tels que les expressions du visage et la configuration du torse. Ce qui suivra dans la Figure 19 est un exemple de notation du MH pour le mot anglais "WEEK" selon l'ASL.

	HOLD	MOVEMENT	HOLD
RIGHT HAND			
handshape	1		1
location	base of left hand		Tip of left hand
orientation	palm down		palm down
non-manual	-		-
LEFT HAND			
handshape	B		
location	front of torso		
orientation	palm		
non-manual	-		

Figure 19. Notation du signe "WEEK" en MH Model

Le signe WEEK débute par une posture (H), avec la main droite (pour les signataires droitiers) dans la base de la main gauche. Ensuite, il fait un mouvement (M) vers l'extrémité de la main gauche et se termine par une posture à ce même endroit. Le changement de signe se fait dans l'emplacement de la main active, et de la base vers l'extrémité de la main passive.

Ce modèle est particulièrement adapté pour décrire les processus phonologiques souvent trouvés dans les études linguistiques.

5.8. Système d'annotation « SML »

Le système « Sign Modeling Language » dit SML [54] est un langage descriptif effectué pour réaliser une approche collaborative servant à écrire des signes dans le cadre du projet Websign[43] (voir projet WebSign du chapitre précédent), ce dernier étant basé sur la technologie des agents conversationnels (animation dans le monde virtuel).

[43] Site web du projet WebSign du Lab. LaTICE de l'Université de Tunis [http://www.latice.rnu.tn/websign]

Pour ce faire, SML est utilisé afin de stocker les descriptions des signes qui peuvent être générés par un humain virtuel ou un avatar au cours du temps. Cette animation peut être utilisée par des systèmes et contextes différents, comme par exemple le chat, l'animation des robots, les personnages virtuels, l'apprentissage en ligne pour les enfants sourds, etc. L'animation, selon SML, est un ensemble de mouvements ou de rotations de groupes d'articulations classées en fonction de leur mouvement au-dessus de l'axe des temps.

Chaque mouvement ou rotation a un intervalle de temps fixe, pendant lequel est faite la rotation de chaque articulation dans le groupe. Ce système d'annotation est complètement indépendant de l'aspect linguistique des langues des signes.

Figure 20. Rotation d'articulation du coude en utilisant l'annotation SML

Dans la Figure 20, l'avatar tourne le coude gauche "l_elbow" d'une rotation de$heading = 80$, $attitude = 50$ et $bank = 50$ en utilisant l'angle d'Euler[44]. La durée de déplacement est de 0,5 secondes. Dans cette annotation, l'avatar revient toujours à la position initiale pour produire un nouveau signe, ce qui diminue son efficacité. Ce mouvement représenté en SML peut être intégrer dans plusieurs signes à la fois sans restriction à langue des signes particulières.

D'où, cette description est très utiles pour la communication inter-machines pour l'animation des agents conversationnels, mais le seul inconvénient est la création des dictionnaires qui prend énormément de temps et les approches automatique ne sont pas encore développées.

5.9. Système d'annotation textuelle "Gloss"

C'est un système servant à annoter, transcrire, décrire des séquences de signes s'appuyant sur des étiquettes (mots), des langues parlées et écrites conventionnellement en caractères majuscules. Donc, une glose [91] est un mot anglais utilisé pour faire référence à un signe en ASL. Les gloses ne sont pas équivalentes à cent pour cent à la signification d'un signe ASL. Les gloses sont utilisées pour référencer rapidement la signification de base du vocabulaire ASL, mais elles ne révèlent pas la structure du signe ASL. D'où, lorsque nous transcrivons l'ASL à l'aide des gloses, nous pouvons profiter de quelques autres moyens de transmission de toutes les informations importantes de l'ASL.

[44] Le mouvement d'un solide par rapport à un référentiel qui intervient 6 paramètres dites angles d'Euler

Par exemple, la phrase en Anglais "What is your name ?" sera annoté "NAME YOU WHAT^{WH}" en glose puisque le signeur ne va pas interpréter la phrase en Anglais mot-à-mot mais plutôt un ensemble de mots avec un ordre différent selon son espace et sa cognition. L'indice "WH" du signe "WHAT" désigne la présence d'une composante non-manuelle qui est une animation facial d'une question de type "wh".

Outre des composantes citées, l'annotation en glose est devenue le système le plus adopté dans les références bibliographiques malgré le rejet au début par la communauté sourde. Tout d'abord, en glose, les relations sémantiques entre les jetons est visible, par exemple dans la phrase "John loves Kate" qui est transcrite en "JOHN^{X} KATE^{Y} LOVE$^{X\rightarrow Y}$".

Ici, on remarque que le verbe d'action "LOVE" nous permet de savoir l'orientation du regard allant de la position "X" vers la position "Y" après placé au début John dans la position "X" et Kate dans la position "Y". Ce système, et grâce à sa simplicité, on le trouve implicitement dans plusieurs projets comme le SignStream [13] et SiGML [42].

6. Discussion et problématiques

La représentation de la langue des signes sous forme écrite est nécessaire pour la génération lorsqu'on emploie des agents conversationnels. Bien que ce problème puisse être résolu en utilisant l'une des notations SL discutées dans ce qui précède, il convient de noter qu'aucune de ces annotations ont été acceptées comme forme écrite standard de la langue des signes.

De plus, chaque annotation est réalisée dans un but et une étude spécifiques. Par exemple, l'annotation ne sera pas similaire pour des études portant sur l'emplacement dans l'espace ou le lien entre la main ou le regard ou l'articulation, etc. Dans notre cas, l'annotation se doit être simple, lisible et économique, dans le but d'un traitement automatique des langues des signes.

Cela veut dire que nous n'entendons pas que le système doit permettre une traduction des transcriptions mais plutôt une visualisation de ce qu'il se passe au niveau des gestes manuels et non-manuels. Nous souhaitons par exemple que le hochement de tête soit indiqué de façon claire et simple, par un seul paramètre. Aussi, si la phrase s'avère négative, nous voulons simplement exprimer qu'il y a une négation, sans entrer dans les détails des façons de signer cette dernière, de même pour les questions "wh" dans la langue Anglaise.

Par ailleurs, les notations décrites ci-dessus, bien adaptées à des fins d'études linguistiques, sont purement scripturales et ne permettent pas de représenter tous les aspects variables des langues des signes, qu'ils proviennent de variations syntaxique-sémantiques, ou de flexions des signes liées à l'iconicité. Ensuite, elles ne permettent pas une traduction aisée de la description linguistique des signes vers la génération de commandes gestuelles destinées à contrôler un personnage virtuel.

Concernant les systèmes d'annotation, ceux-ci ne tiennent pas compte de la dynamique des mouvements ni du contexte. Ainsi, pour le système SiGML, nous avons toujours besoin d'une phase de traduction de la langue parlée vers la notation HamNoSys.

De même, pour le système SWML, la création de signes est difficile et des connaissances de base sont nécessaires pour cette transcription, ainsi que son encodage très complexe à la fois d'un point de vue informatique et de calcul. Enfin, concernant le système SML, il s'agit d'un langage descriptif des articulations regroupées en fonction de leur animation dans le temps indépendamment de la syntaxe, morphologie et la sémantique des signes.

Donc, n'importe quelle système d'annotation ou de notation doit répondre à des objectifs de recherche défini à l'avance puisque la généralisation n'est pas une tâche facile à faire, dû au manque de références et d'études linguistiques de ces langues, et parfois son absence complète pour certain langue de signes dont leurs publications sont restreintes à l'élaboration des dictionnaires.

7. Motivation et contributions

Les études sur les langues des signes sont récentes et innovants et au même temps sont nombreux, allant des études sur l'aspect linguistique, cognitif et grammaticales jusqu'à la création des corpus, la traduction automatique et la synthèse en temps réel. Aussi, on peut classifier ses études par pays, puisque les langues des signes ne sont pas universelles et ne partagent pas les mêmes structures syntaxiques, phonologiques, lexicales, morphologiques et sémantiques.

Ceci est un avantage pour nous, pour faire des recherches sur domaine. Cette thèse vise à développer des ressources linguistiques en langue des signes afin de les traiter pour la traduction automatique, soit d'un texte en Anglais, par exemple, vers la langue des signes Américaine, soit de la langue des signes Américaine vers le texte.

Cette mission n'est pas assez simple puisque les problématiques citées précédemment sont nombreux ou bien plusieurs travaux sont arrivés à des résultats satisfaisants. En effet, pour traitement des SL, on doit toujours passer par une forme écrite pour la représentation et la modélisation.

Malgré l'existence d'outils de notation et d'annotation, chacun présente des inconvénients. Cependant, pour l'annotation textuelle en glose, ayant plus d'avantage pour un traitement automatique, il n'existe pas une représentation XML détaillé de ce système pour le stockage et l'animation via des avatars signants.

D'où, nous avons créé notre propre système d'annotation basé sur la glose, en rapport avec nos besoins spécifiques pour la traduction automatique vers et à partir des langues des signes. Ainsi, dans le cadre de notre travail, nous n'avons pas eu besoin de transcrire l'ASL autrement qu'au moyen des gloses, puisque nous ne nous sommes pas intéressés par la description des gestes (configuration et emplacement) des mains, mais plutôt sur le changement de forme et de configuration dans le contexte.

En effet, nous partons d'une représentation textuelle de la langue des signes Américaine, qui sera notre langue d'étude tout au long de cette thèse, vers une représentation au moyen des gloses, à base du langage "XML". Ce travail sera limité à deux contributions essentielles, la première étant la modélisation d'un système de transcription pour l'annotation de la langue des signes, et en deuxième partie la traduction automatique selon deux approches avec une étude comparative avec d'autres travaux.

On ne s'intéressera pas à la synthèse ou l'interprétation qui peut être une extension de ce travail. Sachant que l'objectif principal des systèmes de notation et d'annotation est de représenter la langue des signes de point de vue mouvement ou linguistique. Il reste que dans certain cas d'illettrisme par exemple, l'interprétation ou la visualisation est le plus compréhensible par des sourds, que nous présenterons dans la section suivante, après la présentation de l'architecture du système de traduction.

D'où, l'architecture complète d'un système de traduction vers ou à partir des langues des signes est illustré dans la Figure 21.

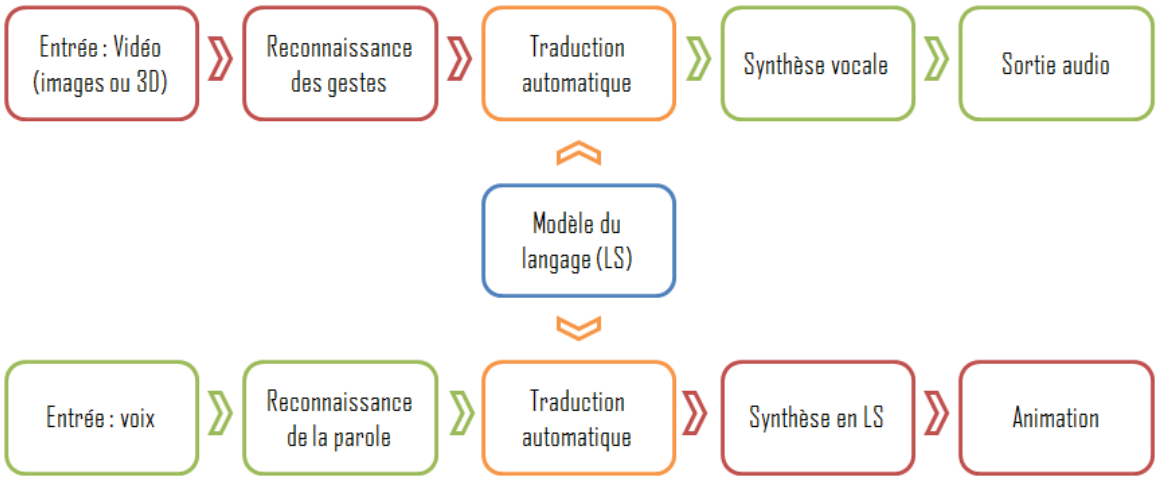


Figure 21. Architecture d'un système de traduction vers ou à partir des langues des signes

Les parties en vert (synthèse vocale, reconnaissance de la voix, etc.) ont une liaison directe avec le traitement de son qui ne seront pas traités dans ce travail et seule la synthèse vocale peut être intégrée en utilisant des outils existant. Pour le reste feront partie de cette thèse soit par la proposition de nouvelles approches et algorithmes soit par un aperçu rapide. Ce qu'on remarque dans l'architecture du système, le modèle de langage de la langue des signes qui joue le rôle principal pour le passage entre les langues vocales et les langues des signes, ce qui prouve la nécessité de modéliser les SL en vue du traitement automatique de celle-ci.

Pour la traduction automatique, plusieurs problématiques et défis ont été cités. Parmi ces défis est la construction d'un corpus en langue des signes, qui généralement difficile à cause du coût élevé de ce type de projet. Ce que nous proposons une nouvelle approche pour la génération automatique d'un corpus Anglais-ASL. Cette génération peut être considérée comme une traduction sauf que son passage par une phase de validation la rend une phase pour la construction.

Cette approche nous permet de faire la traduction en sens inverse, c.-à-d. à partir d'une annotation en SL, on peut déterminer la traduction la plus probable en Anglais. En plus de ça, nous étudierons la classification des émotions à partir d'un texte, afin de rajouter les composantes non-manuel à notre système de transcription. Pour la reconnaissance des gestes à partir des vidéos, on s'intéressera à étudier une approche de reconnaissance à partir des flux de données capturés par des caméras de profondeur. En ce qui suit, dans ce chapitre, nous discutons la visualisation de la langue des signes.

8. Visualisation de la langue des signes

Dans la section précédente, nous avons détaillé les différents systèmes de notation et d'annotation qui nous permettent de décrire la langue des signes. Cette description peut se faire sous forme textuelle (gloses ou autres), ou bien sous forme de symboles graphiques (HamNoSys, SignWriting, etc.). Dans la section qui suit, nous traiterons des méthodes existantes servant à visualiser des langues des signes. La visualisation est une méthode utilisée afin de communiquer un message et employer une communication appropriée pour les sourds. Ces méthodes sont subdivisées comme suit : des méthodes à base de vidéos par lesquelles des signataires natifs enregistrent des séquences en langue des signes, et des méthodes à base d'avatars où un agent conversationnel génère les animations effectuées dans la SL à partir d'un dictionnaire. Les deux sections qui suivent présentent une description de ces méthodes ainsi que leurs avantages et inconvénients respectifs.

8.1. Visualisation à base de vidéo

Un système de visualisation basé sur la vidéo enregistre des signataires humains en exécutant des gestes de SL. Cette approche est utilisée généralement dans les dictionnaires de SL. Des exemples peuvent être trouvés dans [92]. Aussi, beaucoup de chercheurs dans le monde, travaillant sur la langue des signes, utilisent ces techniques de transcription dans les sites web afin de guider les utilisateurs et leur permettre de naviguer facilement par la transcription des menus. Notons par exemple, le site web WebSourd[45] qui donne accès aux informations en langue des signes sous forme de vidéos.

8.1.1. Avantages

Il est plus facile d'enregistrer un grand nombre de gestes en langue des signes par une simple correspondance de la vidéo à une phrase. Pour un mot recherché dans un dictionnaire, l'on affiche la vidéo de l'entrée correspondante, ou encore pour une phrase, il suffit de mettre les vidéos de signes élémentaires en séquences afin de créer une phrase en langue des signes cible. D'où, les signataires responsables, sourds ou interprètes, produisent des séquences naturelles et réalistes contenant des fonctionnalités manuelles et non manuelles qui sont l'avantage majeur de cette approche. De plus, les vidéos peuvent être annotés en utilisant les outils mentionnés précédemment ce qui augmente la compréhension des séquences.

8.1.2. Inconvénients

Les vidéos utilisent une grande quantité d'espace disque pour le stockage [93]. Le type d'équipement utilisé produit des vidéos de haute qualité pour capturer les performances de l'homme réaliste et des expressions faciales. Afin de maintenir cette qualité de transmission, une grande bande passante est exigée [94].

Les vidéos enregistrées sont limitées à une capacité de recherche trouvant la vidéo correcte qui correspond à un texte anglais. Par conséquent, si une vidéo déjà enregistrée

[45] Site Web de la communauté des sourds en France [http://www.websourd.org/]

doit être mise à jour ou modifiée, celle-ci doit être enregistrée à nouveau. D'où un système de visualisation basé sur la vidéo est convenable uniquement lorsqu'il existe un nombre fini de phrases ou bien quand le message transmis est connu à l'avance. La concaténation des signes à partir d'autres vidéos pour créer une nouvelle phrase SL dans une vidéo est très difficile et nécessite un dictionnaire des signes élémentaires par la même personne qui les signe.

Deux défis se posent suite aux difficultés de concaténation de ces vidéos. Le premier défi consiste à s'assurer que les transitions entre les signes dans chaque vidéo soient souples et que les expressions faciales et le paramétrage de l'ampleur ou de la vitesse de certains signes doivent être intégrés de façon appropriée avec les signes correspondants. Le second défi consiste à gérer les changements de configurations des mains selon le contexte qu'on l'appelle "classificateurs prédicats" (discutés dans le chapitre suivant) et leurs trajectoires qui ne sont pas connues à l'avance.

8.2. Visualisation via des agents conversationnels

Un agent conversationnel ou avatar est un personnage virtuel en trois dimensions. Les systèmes de visualisation basés sur l'avatar utilisent des avatars plutôt que des signataires humains. Ils consacrent une procédure plus complexe pour traduire les langues parlées en langue des signes. Pour ce faire, ils se servent d'une annotation textuelle pour l'animation, comme une entrée. Cette visualisation est devenue la solution la plus rapide pour l'évaluation des résultats de la traduction automatique ainsi pour résoudre les problèmes posés par la visualisation en utilisant les vidéos.

8.2.1. Avantages

Tout d'abord, les systèmes de visualisation à base d'avatars sont entièrement contrôlables dans leurs mouvements et leur vitesse d'exécution à travers des commandes simples. Aussi, on peut citer comme avantage :

- Les mouvements peuvent être complètement décrits dans un langage très compact formé de commandes de pilotage. Le volume de données à transmettre est de plusieurs ordres de grandeur, inférieurs au volume de la vidéo correspondante.
- Ils sont totalement modulables (changement du squelette, etc.). La description du personnage lui-même est également de taille réduite.
- Les animations peuvent être employées dans des applications éducatives pour les enfants sourds ou les personnes non-sourdes voulant apprendre la LS.
- L'angle de vue est au choix de l'utilisateur, ce qui améliore la compréhension du signe de plusieurs perspectives.
- L'approche basée sur avatar évite la question de la cohérence en gardant le signataire et l'environnement uniformes au cours de toutes les vidéos générés.

8.2.2. Inconvénients

Deux grands défis sont posés lors de l'utilisation d'avatars. Le premier consiste à modéliser un avatar suffisamment articulé pour effectuer des gestes de SL [63]. Un tel avatar a été développé dans de nombreux projets (discutés dans Chapitre 1). Celui-ci doit

être aussi naturel que possible [64]. Il doit également être en mesure d'effectuer des gestes aussi subtils que des changements d' expressions de visage, tel que sourire, soulever ou froncer les sourcils, etc. Le deuxième défi est d'animer l'avatar. Les mouvements d'avatar doivent être physiquement plausibles et réalistes afin d'être interprétés et compris par les humains.

9. Conclusion

Dans ce chapitre, nous avons démontré que les SL étaient des langues indépendantes et non universelles. Aussi, nous avons profondément étudié la structure de ces derniers. Il a été montré que les phonèmes de la SL ne sont pas linéaires mais simultanés, et se doivent d'être rendus correctement dans un signe. Afin de produire des signes animés en utilisant un avatar, étant l'un des systèmes de visualisation de la SL, une forme écrite de SL est nécessaire. Pour ce faire, divers systèmes de notations et d'annotations de SL ont été étudiés afin de choisir une annotation appropriée pour un système prévu. En conséquence, dans le chapitre suivant, un nouveau système de transcription sera présenté et détaillé commençant par les différents composants systèmes jusqu'à la génération.

Chapitre 3 : Modélisation d'un système de transcription en XML pour l'ASL

1. Introduction

La représentation des SL sous une forme écrite est nécessaire pour la génération de la langue des signes lorsque l'on emploie des agents conversationnels. Bien que ce problème puisse être résolu en utilisant l'une des notations ou annotations discutées dans le chapitre précédent, il convient de noter qu'aucune de ces annotations ont été acceptés en tant que forme écrite standard pour une langue des signes. Dans ce cadre, nous présentons une nouvelle annotation pour la langue des signes Américaine (ASL) basée sur le système d'annotation en glose.

Dans un premier temps, une étude descriptive sur les gloses sera détaillée afin d'éclaircir cette transcription. Dans un deuxième temps, nous décrivons notre système suivant un format XML, pour faciliter l'interprétation des données codées que nous attribuerons. Enfin, nous proposerons notre API permettant de construire une annotation de glose sous format XML valable pour la création de corpus ou pour manipuler un signeur virtuel comme un langage intermédiaire.

2. Les gloses dans la langue du milieu

La langue du milieu est la langue de référence de la communauté linguistique environnante. Une glose[46] se définit comme "un commentaire linguistique ajouté dans le corps d'un texte ou d'un livre, ou dans sa marge, expliquant un mot étranger ou dialectal, un terme rare". Le mot vient du grec ancien $\gamma\lambda\tilde{\omega}\sigma\sigma\alpha/gl\tilde{\text{o}}ssa$, littéralement langue, qui désigne un terme difficile à expliquer. L'explication elle-même est nommée $\gamma\lambda\acute{\omega}\sigma\sigma\eta\mu\alpha/gl\acute{\text{o}}ss\hat{\text{e}}ma$. Actuellement, la glose renvoie à l'explication et non au mot à gloser.

Dans les exemples linguistiques issus d'une langue non familière au lecteur, les gloses permettent de donner une traduction de la langue étrangère, morphème par morphème. L'utilité est d'illustrer, par exemple, les mécanismes de fonctionnement morphologique de la langue traduite grâce aux morphèmes de la langue maîtrisée, par le sujet pour lequel l'illustration est faite. Pour les langues des signes, il s'agit de représenter les signes à l'aide de la langue du milieu, pour nous, l'anglais. En effet, à l'absence de système vulgarisé d'écriture ou de transcription, la plupart des apprenants prennent notes dans la langue vocale dominante, agrémentée de symboles et de dessins, dans le but de donner une description rapide du mouvement.

L'intérêt d'écriture pour l'apprenant se situe au niveau de la prise de notes. Les codes qui émergent sont propres à chacun et n'ont pas vocation à être utilisés pour la diffusion d'information. Du côté des études scientifiques sur les langues des signes [95], la

[46] Définition extraite du site web : [http://fr.wikipedia.org/wiki/Glose]

glose représente la signification d'un signe sous sa forme de citation, non marquée et hors contexte.

Dubuisso et Lacerte précisent également que la glose se note conventionnellement en majuscule et peut être complétée d'éléments permettant la désambigüisation, tel que des marques aspectuelles, des indications sur le regard, sur les actants et les relations sémantiques. D'un point de vue des niveaux linguistiques, la transcription sous forme de gloses relève de l'organisation morphologique, voire morphosyntaxique de la langue.

3. La représentation en XML

XML [96] est un format général de documents orienté texte. Il s'est imposé comme un standard incontournable de l'informatique. Il est aussi bien utilisé pour le stockage de document que pour la transmission de données entre applications. Sa simplicité, sa flexibilité et ses possibilités d'extension ont permis de l'adapter à des multiples domaines. D'où la nécessité d'un format basé sur XML pour représenter nos fichiers permettre également la facilité de comprendre. Aussi, les fichiers XML peuvent être validé par un schéma qu'on le défini après avoir présenté le modèle complet de notre système de transcription.

Au cours de ce chapitre, à partir d'une étude de langue des signes Américaine, notre système de transcription sera défini étape par étape, c.-à-d. à chaque définition d'une composante de l'ASL, on donne sa modélisation en XML. A la fin, on donnera un exemple complet suivi d'un schéma de validation et d'une feuille de style pour la visualisation sur les navigateurs.

4. Convention d'annotation en XML

Les conventions décrites ici reflètent les conventions d'annotation que nous avons utilisées pour la transcription de données. Nous ferons référence à l'ensemble des conventions décrites dans les conventions de Liddell [65]. De même, nous fournirons un ensemble de champs et de valeurs spécifiquement conçus pour l'annotation des données en ASL. Certaines de ces conventions que nous avons adoptées ont été motivées par les limitations de moyens dont nous disposons pour la transcription des données.

Chaque glose est représentée par un mot anglais, écrit en lettres majuscules. Cette notation, bien que simplifiée, ne reflète pas la richesse morphologique des signes ASL seulement, mais aussi la fonction grammaticale très importante des expressions faciales. Les signes non-manuels transmis par le visage peuvent se produire simultanément avec les composantes manuelles. Ils sont représentés par une ligne au-dessus des signes.

Pour notre traduction, elle annote toutes les caractéristiques importantes de la grammaire en SL. Un seul mot Anglais en majuscules identifie un seul signe ASL, par exemple le signe d'un chat sera "**CAT**". Des mots majuscules en Anglais séparés par des tirets représentent également un signe unique c.-à-d. dans le cas où plus d'un mot anglais est nécessaire pour traduire un seul signe. Le signe "**OH-I-SEE**" est une transcription de la phrase en anglais "*Oh I see.*". Il peut exister des conventions conflictuelles. Dans les cas où il y a différentes conventions, nous avons dû faire un choix. Par exemple, il y a des

expressions comme '*up to now*' qui est transcrit parfois en "**UP-TO-NOW**" et parfois en "**SINCE**".

Prenons le cas d'une transcription en ASL de la phrase en anglais "*His aunt lived in Turkey. There had no contact with the aunt. She died and left something to him in her will*", présentée dans la Figure 23 [65].

A première vue, la transcription n'a aucune relation la langue anglaise, mais ce n'est pas vrai. En effet, regarder seulement les signes via des images ou une séquence vidéo n'est pas suffisant, donc, l'annotation en glose devient indispensable. L'annotation en glose est comme suit :

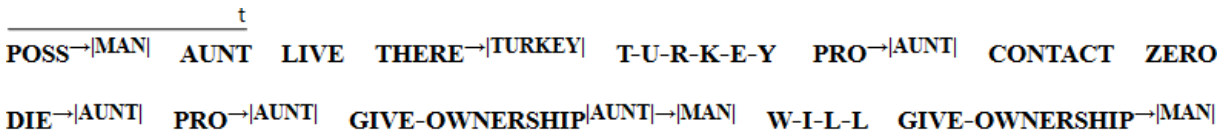


Figure 22. Transcription en gloss de la phrase "His aunt lived in Turkey. There had been no contact with the aunt. She died and left something to him in her will"

Les gloses ici ne sont pas de simples mots en anglais. Les traits d'union entre les lettres majuscules indiquent une séquence de signes des caractères alphabétiques utilisés pour épeler un mot lettre par lettre. Par exemple, si l'interprète souhaite épeler le signe "**TURKEY**", il n'a que le représenter sous la forme suivante "**T-U-R-K-E-Y**". Une glose qui commence par le symbole " # " représente un signe lexical ASL qui a débuté comme une séquence de signes des caractères alphabétiques, surtout dans le cas où le mot est inconnu, comme "**#KATE**".

Dans cette section, on s'intéressera à la grammaire de la langue des signes Américaine et comment annoter en gloses chaque composante de l'ASL. Cette étude nous permet de faire le tour sur tous les aspects morphologiques afin de modéliser un modèle en XML, qui sert à traiter cette langue avec machine. Notre étude couvera les points suivants :

- Les mots composés
- Intégration des nombres
- Les pronoms personnels et les déterminants possessifs
- Préfixes et suffixes
- Aspects
- Réduplication
- Numérotation
- Verbes multiples et exhaustives

Les points cités couvrent l'aspect morphologique. Pour la syntaxe de l'ASL, on étudiera :

- Structure de la phrase
- Les sujets et les clauses principales
- Transitivité

A partir de cette convention d'annotation, on pourra ensuite passer à l'étude de la sémantique entre les sujets et les objets et leurs emplacements dans l'espace du signeur.

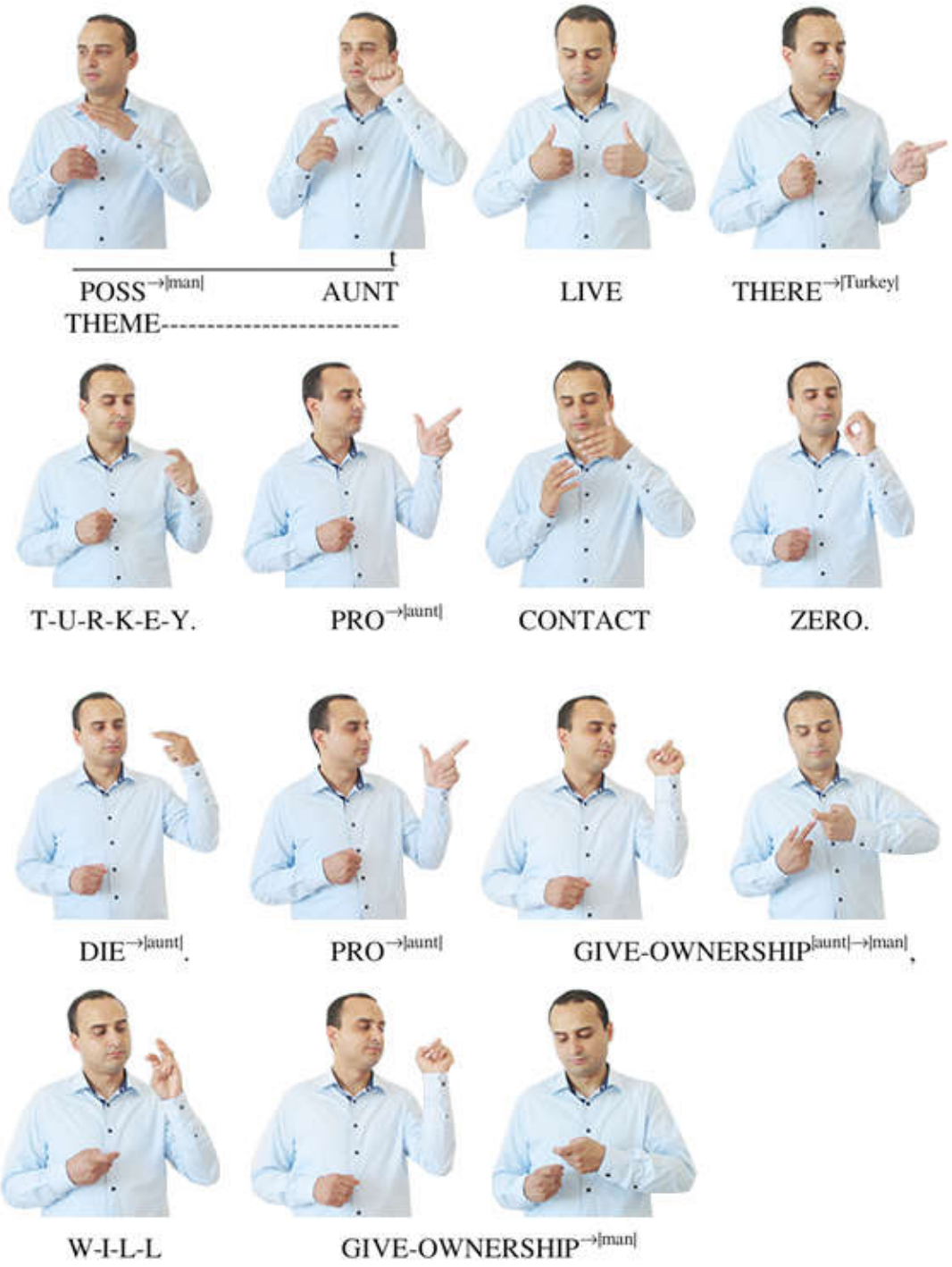


Figure 23. Transcription en ASL de la phrase "*His aunt lived in Turkey. There had been no contact with the aunt. She died and left something to him in her will*"

4.1. Structure de base

Comme on a déjà étudié, le discours en langue des signes, noté "bloc", est une transcription de plusieurs phrases en langue textuelle source comme l'Anglais. Chaque phrase est composée de un ou plusieurs signes, dit "jeton", dont un sous-ensemble peut être accompagné de gestes non-manuels en parallèle.

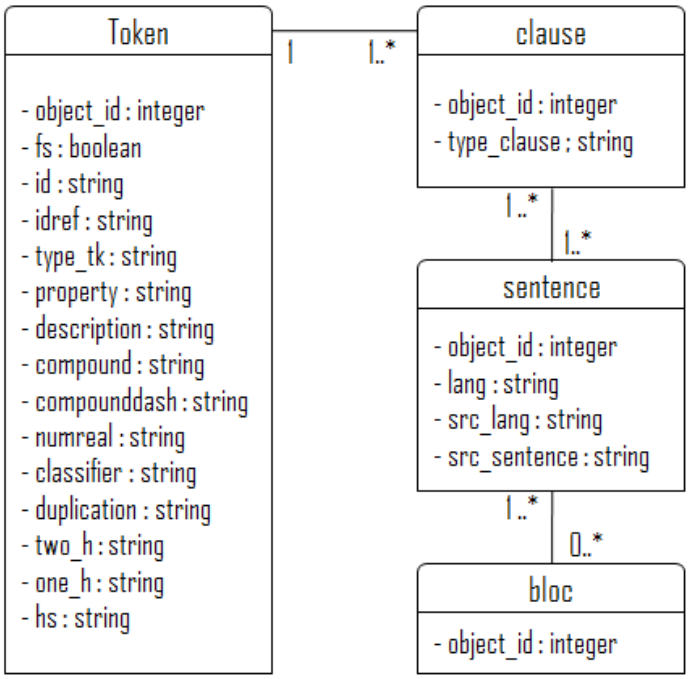


Figure 24. Diagramme de classes du modèle de transcription en gloses

On appellera ce sous-ensemble "clause". Cette clause doit y avoir un type prédéfini à l'avance. Les relations entre les structures composantes de la langue des signes sont représentées par un diagramme des classes dans la Figure 24. Ce diagramme des classes identifie la structure des classes de notre système, y compris les propriétés de chaque objet, il est composé de quatre classes nommées "bloc", "sentence", "clause" et "token". La classe "bloc" regroupe l'ensemble d'objet de type "sentence" qui est défini par :

- `src_lang` : pour indiquer la langue source de l'entrée du système.
- `lang` : pour indiquer la langue cible du système ici c'est la langue des signes.
- `src_sentence` : c'est la phrase qui sera transcrite.

L'ensemble des informations concernant le signe est inclus dans une balise "*token*" qui peut être autonome lorsque le signe est simple ou bien, on inclut d'autres attributs relatifs aux transformations à appliquer sur ce signe. L'exemple (1) est un signe simple ne subissant aucun changement. Pour l'exemple (2), il s'agit d'un signe qui subit un changement au niveau de la génération qui est généralement produite avec une seule main mais se fait avec les deux mains. Le jeton peut avoir plusieurs propriétés qui seront décrites dans ce qui suit comme l'épellation, les signes composés, etc.

```
(1)    <token>HOUSE</token>
(2)    <token two_h="yes">GIVE</token>
```

Lorsque nous voudrons faire varier des éléments non manuels dans la réalisation du signe on ajoute une balise <clause> tout en spécifiant le type de geste non manuel. L'exemple (3) montre le signe de "PEOPLE" avec l'expression du visage "*puff*" pour indiquer un grand nombre de gens :

```
(3)    <clause type_clause="puff">
           <token>PEOPLE</token>
       </clause>
```

Toutes les expressions faciales appliquées sur les clauses seront étudiées plus tard. L'ensemble de "token" et de clauses sont inclus dans une balise <sentence> pour limiter une seule phrase à signer.

```
(4)    <sentence> </sentence>
```

Les trois classes "clause", "sentence" et "bloc" sont des classes simples. Par contre, la classe "token" contient plusieurs propriétés afin de transcrire correctement les discours en SL. Les propriétés sont appliquées sur le jeton directement, entre deux jetons ou bien entre le jeton et sa clause parente. Ces propriétés peuvent être une transcription des composantes non-manuelles par exemple. Le Tableau 1 résume les propriétés étudiées :

Tableau 1. Les différents attributs de la balise "token"

Attribut	Signification
type_tk	Pour décrire les pronoms personnels
id	identifiant unique pour chaque jeton
idref	Référence à un autre jeton
fs	Epellation / Fingerspelling
compound	Mot composé
compounddash	Mot composé avec tiret (*dash*)
property	Indiquer le changement de sens
numeral	Nombre / Numérotation
classifier	Classificateur / Modificateur
description	Signification dans le contexte
duplication	Répétition d'un signe
one_h	Signation avec une seule main
two_h	Signation avec deux mains
hs	Configuration de la main

Une description consiste ainsi en une liste de commandes de construction d'objets suivie de contraintes. Si une ligne contient une référence à un objet d'une autre ligne, cette dernière doit être placée avant la première. Par exemple, dans l'exemple 5, le verbe "**LOVE**" utilise le sujet "**JOHN**" comme point de départ et le sujet "**KATE**" comme point d'arrivée, ces deux derniers étant définis avant. Les descriptions font ainsi apparaître les dépendances entre éléments.

De manière générale, si un élément A dépend d'un élément B, au moins une contrainte sur A fera référence à B.

```
(5)    <sentence lang="asl" src_lang="en" src_sentence="john love kate.">
       <clause type_clause="none">
              <token fs="yes" id="x">JOHN</token>
              <token fs="yes" id="y">KATE</token>
              <token idref="x,y">LOVE</token>
              </clause>
```

```
</sentence>
```

Dans l'exemple 6, nous commençons par produire les signes du pronom possessif "*my*" et de "*daughter*" comme étant le sujet "*topic*" de discours, puis le verbe "*love*" et enfin le nom composé "*ice*" et "*cream*" :

```
(6)    <sentence lang="asl" src_lang="en" src_sentence="As for my daughter, she loves
       ice cream.">
       <clause type_clause="t">
                    <token type_tk="poss1st"/>
                    <token>DAUGHTER</token>
              </clause>
              <clause type_clause="none">
                    <token>LOVE</token>
                    <token compound ="-">
                          <token>ICE</token>  <token>CREAM</token>
                    </token>
              </clause>
       </sentence>
```

Dans ces deux exemples, la structure de base a été décrite avec différents utilisation des propriétés soit pour les clauses soit pour les jetons. Dans le reste de ce chapitre, on étudiera de près chaque composante de l'ASL qui sera notre étude de cas, ainsi qu'une focalisation sur les relations entre les signes afin qu'il soit notre transcription une entrée pour les systèmes de synthèse via les agents conversationnels ou pour les systèmes de traduction automatique, sans oublier la création des corpus pour les langues des signes.

4.2. Les mots composés

Le lexique de la langue des signes Américaine contient un très grand nombre de signes composés construit à partir de signes existants. Par exemple, le signe "**SAME-AS**" est formé par la combinaison des deux signes "**SAME**" et " **AS**" (voir Figure 25 (a)), sachant que dans l'ASL, les deux signes existent déjà dans le dictionnaire. Au niveau interprétation, l'articulation doit être fluide sans posture entre les deux signes.

Un contre-exemple de signe composé est le signe "**THANK-YOU**" qui n'est pas composé des deux signes "**THANK**" et "**YOU**", mais plutôt un signe différent avec sa propre syntaxe (voir Figure 25 (b)).

Figure 25. Le signe composé "SAME-AS" (à gauche) et le signe "THANK-YOU" (à droite) en ASL

La transcription des deux signes "**SAME-AS**" et "**THANK-YOU**" sont comme suit :

```
(7)    <token compound="-"><token>SAME</token><token>AS</token></token>
(8)    <token>THANK-YOU</token>
```

4.3. Les nombres

En glose, les accolades "{ }" indiquent un signe composé de deux morphèmes liés comme {**TWO**}{**O'CLOCK**}. Dans ce cas, le signeur effectue deux signes différents : le premier qui le nombre "**TWO**", le deuxième le signe "**O'CLOCK**". Au cours de l'interprétation, le deuxième signe garde la configuration de la main du premier signe qui est le " **TWO** ". Si le signe est composé du préfixe {**AGE$_0$**} c.-à-d. il est combiné avec un signe numéral.

Par exemple, si une personne âgée de quinze ans, le signe sera {**AGE$_0$**}-**FIFTEEN**. Et dans ce cas, la configuration de la main du mot "**AGE**" sera inchangé lors de la signation du signe "**FIFTEEN**". Dans la Figure 26, on remarque que la configuration de la main reste inchangée pour les deux signes {**TWO**}{**O'CLOCK**} et {**TWO**}{**WEEK**}.

Figure 26. Exemples de signes en ASL en utilisant le préfixe "TWO"

Dans notre système de transcription, le cas de figure de l'utilisation des nombres est effectué en utilisant l'attribut "numeral" de la balise "token". Par exemple:

```
(9)    <token numeral="yes">TWO</token> <token numeral="yes">WEEK</token>
```

4.4. Les pronoms personnels

Le Tableau 2 [65] contient une liste de vingt-six pronoms personnels signés différent. Ces pronoms se classifient en deux catégories selon Meier [97]: 1er personne et les autres. Le pronom personnel de la première personne singulier "PRO-1" fait référence au signeur lors d'une conversation. Le pronom personnel "PRO" fait référence directe à une autre personne dans la scène sauf lui-même (voir Figure 27).

Figure 27. Le signe "PRO-1" et "PRO" en ASL

Tableau 2. Les pronoms personnels dans l'ASL

	1er personne ("*first person*")	**Autres ("not-*first person*")**
Singulier	PRO-1	PRO
	POSS-1 (MY)	POSS
	SELF-ONLY-1	SELF-ONLY
	SELF-ONLY-1!	SELF-ONLY!
Multiple	PRO-FORMAL-1	PRO-FORMAL
	PRO-DUAL-1	PRO-DUAL
	{PRO-MULT-1}{THREE}	{PRO-MULT}{THREE}
	{PRO-MULT-1}{FOUR}	{PRO-MULT}{FOUR}
	{PRO-MULT-1}{FIVE}	{PRO-MULT}{FIVE}
Pluriel	PRO-PL-1 (WE)	PRO-PL
	POSS-PL-1 (OUR)	POSS-PL
	SELF-ONLY-PL-1 (OURSELVES)	SELF-ONLY-PL

Par exemple, la phrase en anglais "*I am satisfied*" sera transcrite en utilisant deux signes seulement, elle est annoté en gloses comme suit : "PRO-1 SATISFIED". Comme le pronom personnel "PRO", il existe aussi les pronoms pour deux : "PRO-DUAL-1" et "PRO-DUAL". Ces pronoms sont utilisés uniquement pour les humains avec les verbes d'actions en gardant la configuration "2" de la main dominante.

Par exemple, la phrase "*two students are going to school*" sera annoté en "PRO-DUAL STUDENT SCHOOL GO". Et de même pour les autres pronoms multiples (trois, quatre et cinq) et les pluriels.

Pour les pronoms personnels possessifs, la même règle sera appliquée. A titre d'exemple, la phrase "*My mother loves me*" sera annoté en gloses : "MOTHER POSS-ONLY-1 LOVE". Selon notre système, les pronoms sont défini par l'attribut "type_tk" de la balise "token" :

```
(10)   <token type_tk="pro1st"/>
```

Voici un exemple complet en XML de la phrase "I don't like chocolate", où on utilise le pronom personnel "PRO-1" :

```
(11)   <sentence lang="asl" src_lang="en" src_sentence="i don't like chocolate.">
       <clause type_clause="neg">
              <token type_tk="pro1"/>
              <token>LIKE</token>
              <token>CHOCOLATE</token>
       </clause>
       </sentence>
```

4.5. Préfixes et suffixes

Le signe {**AGE0**}-**FIFTEEN**, déjà décrit précédemment, contient le préfixe {**AGE0**}, et de même que l'exemple cité dans la Figure 28. Le préfixe consiste à prendre quelques caractéristiques du signe tel que le mouvement, l'orientation etc. lors de la signation du second signe.

Cette forme morphologique est appliquée aussi pour les suffixes comme les préfixes. Dans la Figure 29, le signe "**DRIVER**" et le signe "**TEACHER**" partagent le suffixe qui est le signe "**ER**". Ce suffixe permet de distinguer entre le verbe "**DRIVE**" et le nom "**DRIVER**" par exemple.

L'étude de suffixe et préfixe est utile pour la construction des corpus ou l'annotation en utilisant des outils d'annotation de vidéos, mais, dans notre transcription XML, on ne va pas tenir compte de cette spécificité, puisqu'elle ne joue aucun rôle lors de la traduction.

Figure 28. Les deux signes {AGE0}-FIFTEEN et {AGE0}-FOUR en ASL

Figure 29. Le suffixe "ER" en ASL encadré en rouge lors de la signation des deux signes "TEACHER" et "DRIVER"

4.6. Aspects

Beaucoup de signes dans ASL ont des formes morphologiquement complexes qui expriment des significations aspectuelles selon le contexte. Ces formes complexes ne sont pas produites par l'ajout de préfixes ou suffixes. Au lieu de cela, ils sont généralement créés par des changements dans la forme du panneau lui-même. Ces modifications pourraient découler du processus de duplication [98], un changement interne sous la forme d'un signe [99], ou de certaines combinaisons de ceux-ci.

La Figure 30 illustre deux modes qui pourraient également être compris par moyen d'une analyse du contexte. Le signe "**MONDAY**" est produit à des mouvements répétés vers le haut. Le signe "**MONDAY**[WEEKLY]" garde la même configuration de la main et la position de départ et rajouter une répétition du mouvement en bas du visage.

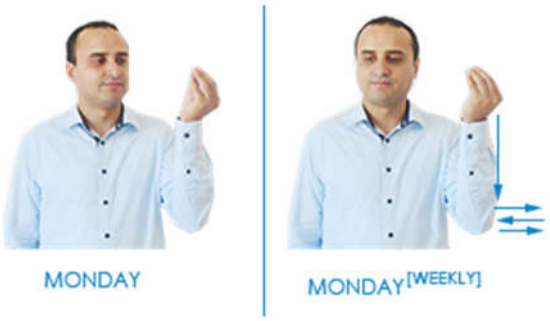


Figure 30. Les deux signes "MONDAY" et "MONDAY[WEEKLY]" en ASL

Cette propriété a été tenu compte dans notre système de transcription comme suit :

```
(12)   <token property="HABITUAL">GO</token>
```

4.7. Réduplication

L'utilisation du symbole " + " indique la répétition (redoublement) du même signe. Le nombre de fois de ce symbole indique le nombre de répétitions. Par exemple, le signe "**DIFFERENT++**" sera répété deux trois et le signe "**GIVE+**" sera répété deux fois. Dans notre système de transcription la valeur par défaut de l'attribut "duplication" est 1. Voici les deux transcriptions des deux signes :

```
(13)   <token duplication="3">DIFFERENT</token>
(14)   <token duplication="2">GIVE</token>
```

4.8. Verbes multiples et exhaustives

Un mot entre crochets à la suite d'un signe indique un changement du sens associé à la morphologie grammaticale. Par exemple le signe "**WAIT**" diffère du signe "**WAIT[DURATIONAL]**", puisque dont le second signe le signeur ralentit la vitesse de signation. Les autres étiquettes sont [**CONTINUATIVE**], [**EVERY**], [**MULTIPLE**], [**HABITUAL**], [**EXHAUSTIVE**], [**EMPHATIC**], [**HABITUAL**], [**INCESSANT**], [**DAILY**] et [**WEEKLY**].

Figure 31. Les trois façons de signer le mot "GIVE"

4.9. Les expressions faciales et les gestes non-manuels

Une expression faciale [81] est une mimique faciale chargée de sens. Le sens peut être l'expression d'une émotion, un indice sémantique ou une intonation dans la Langue des Signes. L'utilisation d'expressions faciales pour marquer les contrastes grammaticales est unique et apparemment universelle pour les langues signées.

Une ligne au-dessus d'un signe ou d'une séquence de signes indique la portée d'une expression du visage ou d'autres comportements non-manuels nommés au-dessus de la ligne. Dans cet exemple, "**t**" indique le sujet de la phrase "*that boy, the girl pushed*" :

t
BOY GIRL PUSH

Figure 32. Transcription de la phrase "that boy, the girl pushed" en gloses

De même que le signe "**t**", on trouve aussi :

- "**neg**" : pour exprimer la négation, la négation en ASL peut être réalisée avec l'élément lexical "**NOT**" qui doit exister en parallèle avec le signe non manuel "**NEG**", ou peut être indiqué par le signe non manuel "**NEG**" sans aucun élément lexical dans une phrase négative ;
- "**cond**" : une expression non-manuelle indiquant une phrase conditionnelle ;
- "**rh**" : les questions rhétoriques sont les questions demandées, sans l'espoir d'avoir une réponse ;
- "**q**" : une expression non manuelle formulant des questions de type "oui / non" ;
- "**wh-q**" : une expression non-manuelle interrogative pour les questions de types "*wh-questions*".

Une liste complète des expressions faciales est détaillée dans le Tableau 3.

4.10. Les classificateurs

En ASL, les classificateurs ou modificateurs sont utilisés afin de montrer le mouvement, l'emplacement et l'apparence souhaitée de chaque signe. Suite à une personne ou une chose indiquée par le signataire, les classificateurs peuvent être utilisés à leur place pour signaler le type, la taille, le nom, le mouvement, ou l'étendue de l'information donnée. Les signes contenant un classificateur sont appelés des "prédicats de classificateurs" [100] ou des "verbes de mouvement, d'orientation, et d'emplacement" [101]. Un prédicat est la partie de la phrase qui a été modifiée, le sujet de la phrase ou un nom ou un autre syntagme nominal dans la phrase [10]. Le mouvement du prédicat présente la façon dont l'entité se déplace réellement.

Tableau 3. Liste des expressions faciales

Glose	Description d'adverbe facial	Signification
puff	les deux joues remplies d'air (FACS[47] AU 13)	une grande quantité, une très grande masse, loin dans le temps ou l'espace.
mm	lèvres serrées ; la lèvre inférieure peut se modifier en une moue légère. (FACS AU 15 + 22B)	normalement, régulièrement avec plaisir

[47] FACS (Facial Action Coding System) est tenté pour décrire l'expression des émotions sur le visage à l'aide de critères formels [http://face-and-emotion.com/dataface/facs/description.jsp].

int	lèvres serrées sont tirées en arrière et les dents sont serrées ensemble (FACS AU 20 + 25 + 31**)**	grand étonnement, degré inhabituellement grand, très loin dans le temps ou dans l'espace
ps	lèvres serrées sont tirées en arrière avec une petite ouverture au centre (FACS AU 23 + 25)	juste manqué, très mince, lisse
th	la mâchoire est relâchée et une partie des lèvres légèrement en saillie pour montrer la langue (FACS AU 19 + 26)	négligence ; sans attention ; incorrecte, faux
sta	lèvres sont légèrement ouvertes ; les dents sont serrées ; la mâchoire s'ouvre et se ferme (FACS UA + 22T 25, 26)	encore et encore, trop, dur
cha	les lèvres sont légèrement ouverte, les dents sont serrées ; la mâchoire tombe soudainement (FACS UA 22T + 25 + 27)	relativement large
cs	les lèvres serrées sont tirées en arrière sur un côté du visage, les dents sont serrées ensemble (FACS AU 20R + 25R + 31)	très proche du présent temps ou lieu ; récemment ;
pah	lèvres serrées s'ouvrent soudain pour former le mot (implicite) "pah" (FACS AU 24, 27)	enfin / exactement
pow	lèvres serrées s'ouvrent soudain pour former le mot "pow" (FACS AU 24, 27, 18 +26)	ce qui signifie toujours en cours

Les classificateurs sont généralement représentés avec les lettres CL. Des informations supplémentaires peuvent soit indiquer la manière dont le classificateur est produit, soit caractériser la signification que le classificateur représente. Les différents types de classificateurs modificateurs [102] sont indiqués ci-dessous :

- **CL:A** : Décrire les objets fixes, tel que les statues, les vases, une bouteille sur une étagère, le placement des objets sur une table, etc.
- **CL:B** : Décrire les surfaces, les murs, les planchers, les étagères, des photos sur un mur, les vêtements, les objets plats, etc.
- **CL:C** : Décrire les objets cylindriques, tel que les tasses, les poubelles, les bâtiments, un gros tuyau, la forme de quelque chose, etc.
- **CL:F** : Décrire les petits objets circulaires comme les pièces, les boutons, les yeux qui vont et viennent, les colliers de perles, la forme de cigare, etc.
- **CL:G** : Décrire les formes minces, tel que les sourcils, la bouche, les moustaches, les becs d'oiseaux, etc.
- **CL:ILY** : Utilisé pour les avions, les hélicoptères, mais pas pour un oiseau qui vole. Les oiseaux qui volent sont exprimés avec les classificateurs (2h) CL:B pour montrer les battements d'ailes, etc.
- **CL:L** : Pour montrer la forme du périmètre des objets ronds et carrées plats comme des cadres, des horloges, des assiettes, la forme d'un lac, etc.

- **CL:O** : Sert à montrer les petits objets cylindriques tel que les tubes, les guidons, les tuyaux, etc.
- **CL:S** : Pour afficher une tête qui va et vient, jeter un coup d'œil par la porte, des animaux étirent / tournent la tête, etc.
- **CL:U** et **V** : Ces deux classificateurs sont utilisés pour décrire des gens assis ou debout. Le classificateur V est essentiellement le même que le classificateur 2 dans ses utilisations. Le classificateur U peut être utilisé pour montrer les objets minces, tel que la langue, un plongeoir, une cravate, etc.
- **CL:1** : Personne debout, repère de temps, l'altitude d'un crayon, un seul individu qui marche, etc.
- **CL:2** : Deux personnes qui marchent, assises ou debout, la langue de serpent, etc.
- **CL:3** : Véhicule, animal, 3 personnes qui marchent ou debout, etc.
- **CL:4** : Quatre personnes, le style des cheveux, des moustaches, des bandes, des barrières, des barres, des dents découvertes, etc.
- **CL:5** : Cinq personnes à pied, la chute des feuilles, le style des cheveux, des bandes, des barres, etc.

4.11. Exemples de phrases complètes

Nous avons détaillé dans les sections précédentes les conventions avec lesquelles nous avons travaillé pour représenter notre système d'annotation en gloses. Ainsi, quelques exemples [65] [10] sont décrits dans le Tableau 4. Ces exemples comprennent seulement deux colonnes de notation :

- la colonne 1 montre les phrases en langue source qui est l'anglais.
- la colonne 2 montre les transcriptions en ASL. Ces mots sont intégrés par les annotations conventionnelles pour les composantes manuelles ou non-manuelles.

Tableau 4. Exemples d'utilisation de gloses pour représenter des énoncés en ASL

Phrase source en Anglais	Transcription ASL en gloses
John loves Kate.	#JOHN$_x$ #KATE$_y$ LOVE$^{\rightarrow x,y}$
I don't like chocolate.	neg PRO1st LIKE CHOCOLATE
What is your name?	wh_q YOUR $^{\rightarrow x}$NAME WHAT
There were too many people and the floor collapsed.	puff PEOPLE MANY FLOOR COLLAPSE
A lot of something.	puff 2h: HS:5 SCADS-OF
A person is walking along normally.	mm WALK CL: 1 'person moving'
There was a huge group of cats.	Int CAT 2h:HS:5 'hoard of'
The jet just missed hitting the tree.	ps J-E-T CL: 'fly over the tree'

Driving carelessly	th DRIVE
Work hard over and over.	sta WORK +++
The book was really thick.	cha BOOK LARGE
Just arrived.	cs ARRIVE

5. Schéma de validation et visualisation

5.1. Validation

Le système de transcription en XML proposé est défini par un schéma XML permettant de définir la structure et le type de contenu de notre document XML, pour utilisation ultérieure. Cette définition permet notamment de vérifier la validité de tout nouveau document, c'est-à-dire respecter les contraintes définis par le schéma. Le schéma XML est présenté dans l'Annexe B ou en le téléchargeant à partir du Web[48].

5.2. Représentation pour le web

Afin de représenter notre fichier XML sous forme des pages web, nous avons fait recours au langage XSL (XML Stylesheet Language) [96] pour faire la mise en page du document. Le processus de transformation d'un document XML en une page HTML a été scindé en deux unités distinctes XSLT et XSL-FO. Le langage XSLT (pour XML Stylesheet Language Transformation) est un langage de transformation de documents XML. Le langage XSL-FO (pour XML Stylesheet Language - Formatting Objets) est un langage de mise en page de document. Dans la première phase, le document XML est transformé en un document XSL-FO à l'aide de feuilles de style XSLT. Dans la seconde phase, le document FO obtenu à la première phase est converti par un processeur FO en une page html.

Une feuille de style XSLT contient des règles qui décrivent des transformations. Ces règles sont appliquées à un document source XML pour obtenir un nouveau document XML résultat. Cette transformation est réalisée par un programme appelé processeur XSLT. La feuille de style est présentée dans l'Annexe C. Ainsi nous allons proposer quelques résultats des représentations des phrases en appliquant notre feuille de style dans la Figure 33:

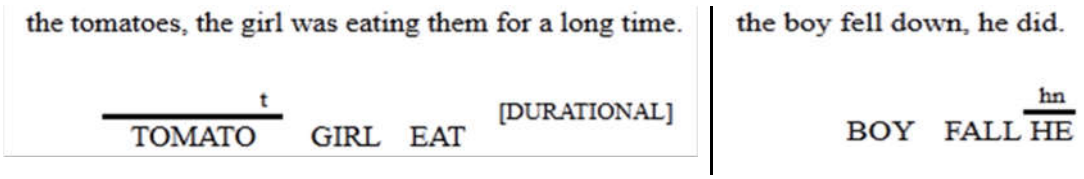


Figure 33. Rendu d'une transcription en gloses en utilisant une feuille de style XSLT

[48] Le schéma XML et la feuille de styles XSLT sont disponibles sur le site : [http://www.achrafothman.net/aslsmt/schemaXML.xsd] et [http://www.achrafothman.net/aslsmt/formexsl.xsl]

6. Discussion et perspectives

Dans les sections et les chapitres précédents, nous avons décrit et expliqué les conventions d'annotation des gloses pour la langue des signes Américaines après avoir étudier l'aspect lexical et morphologique. Ainsi que les différentes caractéristiques du système de transcription proposé à base de XML qui sera nommé XML-Gloss. Ce dernier fournit une définition élaborée d'un XML compatible au système de notation de la langue des signes adapté à des fins d'animation et synthèse via les agents conversationnels.

L'avantage de cette transcription est la conservation de l'aspect sémantique entre les signes au cours d'une conversation et en plus elle est indépendante de la mise en scène et de la géométrie spatiale. D'où, le fichier XML résultant peut servir à une entrée de n'importe quel système de traduction après l'avoir analysé. Par exemple, on peut donner une architecture d'un système de traduction automatique en utilisant notre modèle de transcription (voir Figure 34).

Figure 34. Architecture d'un système de traduction automatique vers la langue des signes

La modélisation spatiale consiste à récupérer les jetons de la transcription XML-Gloss ainsi que les relations de dépendances et les placer dans un espace de signation. Ensuite et à partir d'un dictionnaire, on génère l'animation en utilisant par exemple le langage SML [103]. L'animation via avatar ou agent conversationnel est la phase finale pour l'utilisateur.

Toutes les étapes intermédiaires doivent être transparentes pour l'utilisateur et le passage entre les étapes doit assurer l'alignement des mots à signer, les trajectoires entre les signes, la coarticulation et la fluidité. Puisque toutes ces caractéristiques entrent dans l'évaluation de la qualité de synthèse et dans l'acquisition de l'information auprès des sourds sachant que l'œil de l'être humain est susceptible au moindre défaut.

Dans cette thèse, nous s'intéressons à la génération automatique de la transcription XML-Gloss à partir d'un simple texte en Anglais fournit en entrée. Plusieurs approches et algorithmes seront décrits dans les chapitres qui suivent. Avant tout, nous décrivons une interface de programmation écrite en Java abrégé en API (*Application Programming Interface*) pour la création des fichiers XML sans faire recours à l'édition manuelle.

7. API de création de XML-Gloss

7.1. Description JDOM

JDOM[49] est une API du langage Java développée indépendamment de Sun Microsystems. Elle permet de manipuler des données XML plus simplement qu'avec les

[49] API JDOM est disponible sur le site [http://www.jmdoudoux.fr/java/dej/chap-jdom.html]

API classiques. Le but de JDOM est la facilité de la manipulation au sens large de document XML : lecture d'un document, représentation sous forme d'arborescence, manipulation de cet arbre, définition d'un nouveau document, exportation vers plusieurs formats cible, etc. JDOM est donc un modèle de documents objets open source dédié à Java pour encapsuler un document XML. Il propose aussi une intégration de SAX, DOM, XSLT et XPath. JDOM n'est pas un parseur : il a d'ailleurs besoin d'un parseur externe de type SAX ou DOM pour analyser un document et créer la hiérarchie d'objets relative à un document XML.

- Un document XML est encapsulé dans un objet de type Document qui peut contenir des objets de type Comment, ProcessingInstruction et l'élément racine du document encapsulé dans un objet de type Element.
- Les éléments d'un document sont encapsulés dans des classes dédiées : Element, Attribute, Text, ProcessingInstruction, Namespace, Comment, DocType, EntityRef, CDATA.
- Un objet de type Element peut contenir des objets de type Comment, Text et d'autres objets de type Element.

7.2. API d'annotation de gloses

Pour créer notre API, il faut construire chaque élément de fichier XML puis les ajouter les uns aux autres de façon logique, selon leur apparence dans l'annotation en gloses. Un noeud est une instance de la classe "org.jdom.Element". Les méthodes de l'API sont décrites dans le Tableau 5. Les méthodes couvrent tous les points qu'on a déjà parlés dans les sections précédentes.

L'ordre de l'appel de chaque méthode doit suivre l'ordre d'apparition des signes. Tout d'abord, on crée la phrase et ses différents blocs. Ensuite, on crée les signe un par un. Puis, on rajoute les propriétés relatives à chaque signe (ou jeton) selon l'ordre. Enfin, on génère le fichier résultant après un test de validation selon le schéma de XML-Gloss. Nous commençons donc par décrire les différentes méthodes qui vont se charger de créer l'arborescence adéquate :

Tableau 5. Les principales méthodes pour la création d'un fichier XML-Gloss

Les méthodes	Fonction
`public void create_bloc ()`	Créer un bloc de phrases « élément racine » qui respecte les règles et les transformations du schéma XML et de la feuille de style.
`public void create_sentence (String,String,String)`	Créer un nouvel élément « sentence » et l'ajouter en tant qu'élément de racine.
`public void create_clause (String)`	Créer une clause pour spécifier la nature de la mimique faciale si elle existe, sinon une clause nulle.
`public void create_token (String)`	Créer un signe unique qui se fait seulement avec sa convention de signation.

`public void add_compound(String, String, String)`	Créer un signe qui donne dans sa traduction en anglais plus qu'un mot.
`public void add_fingerspelling (String)`	Créer une séquence de signes de caractères alphabétiques utilisés pour épeler un mot.
`public void add_numeral (String)`	Créer un signe numéral qui exprime l'âge.
`public void add_property (String, String)`	Créer un signe avec un changement de sens associés avec la morphologie grammaticale.
`public void add_id (String,String)`	Créer un signe tout en donnant un identifiant
`public void add_idref (String,String)`	Créer un signe tout en indiquant la référence associée.
`public void add_classifier (String, String)`	Créer un signe en indiquant le type de classificateur produit.
`public void add_description (String)`	Créer un signe de description pour un classificateur ou bien pour un « handshap ».
`public void add_duplicate (String, int)`	Créer un signe avec répétition. L'entier en argument sert à indiquer le nombre de répétition.
`public void add_one_h (String)`	Créer un signe qui est généralement produit avec deux mains, mais qui est fait avec une seule main.
`public void add_two_h (String)`	Créer un signe qui est généralement produit avec une seule main, mais qui est fait avec les deux mains.
`public void add_hs (String)`	Ajouter une configuration de la main spécifique à un jeton
`public void add_age (String)`	Créer un signe qui indique l'âge de quelque chose (tenir compte du préfixe {AGE$_0$}.
`public static void affiche()`	Afficher l'arborescence complète du fichier XML résultant.
`public static void enregistrer (String)`	Cette méthode permet d'enregistrer la sortie.

8. Evaluation

Après avoir présenté notre API de transcription de gloses, nous allons décrire, dans ce qui suit, quelques exemples extraits des travaux de Liddel [65]. Plus que 100 phrases ont été créées avec cette API avec un taux de précision de 97%. L'étape d'évaluation s'est déroulée comme suit :

- Préparer la liste des phrases en Anglais avec leurs transcriptions correspondantes.
- Pour chaque phrase, on suit une liste d'instructions pour générer le XML-Gloss en utilisant l'API développée.
- Afficher sur le navigateur la transcription résultante grâce à la feuille de style XSLT.
- Comparer la transcription avec la transcription générée. Si le rendu est identique, on valide la phrase et sa transcription.

Voici deux exemples décrit étape par étape.

8.1. Exemple 1

Soit la phrase en anglais suivante "*I don't like chocolate*". Sa transcription correspondante est :

neg
PRO-1 LIKE CHOCOLATE

La transcription en XML-Gloss manuelle est :

```
(15)  <sentence lang="asl" src_lang="en" src_sentence="i don't like chocolate.">
      <clause type_clause="neg">
            <token type_tk="pro1st"/>
            <token>LIKE</token>
            <token>CHOCOLATE</token>
      </clause>
      </sentence>
```

Les différentes instructions pour créer le fichier XML-Gloss sont :

```
(16)  g.create_sentence("en","asl"," i don't like chocolate");
      g.create_clause("neg");
      g.create_token("i");
      g.create_token("like");
      g.create_token("chocolate");
      affiche ();
      enregistre ("fichier1.xml");
```

Le fichier XML-Gloss généré est le suivant :

```
(17)  <?xml version="1.0" encoding="UTF-8"?>
      <?xml-stylesheet type='text/xsl' href='form.xsl'?>
      <bloc xmlns:noNamespaceSchemaLocation="schemaXML.xsd">
      <sentence src_lang="en" lang="asl" src_sentence=" i don’t like chocolate">
      <clause type_clause="neg">
            <token>I</token>
            <token>LIKE</token>
            <token>CHOCOLATE</token>
      </clause>
      </sentence>
```

Le rendu final est :

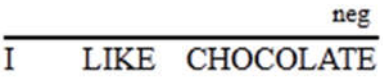


Figure 35. Transcription XML-Gloss de la phrase "I don't like chocolate"

8.2. Exemple 2

Soit la phrase en anglais suivante "*I have going to the conference for five years*". Sa transcription correspondante est :

PRO-1 GO [HABITUAL] CONFERENCE UP-TO-NOW {FIVE}{YEAR}

La transcription en XML-Gloss manuelle est :

(18)
```
<sentence src_lang="en" lang="asl" src_sentence="i have going to the
conference for five years.">
<clause type_clause="none">
      <token type_tk= "pro1st"/>
      <token property="HABITUAL">GO</token>
      <token>CONFERENCE</token>
      <token compound="-">
      <token>UP</token><token>TO</token><token>NOW</token>
      </token>
      <token numeral="yes">FIVE</token><token numeral="yes">YEAR</token>
</clause>
</sentence>
```

Les différentes instructions pour créer le fichier XML-Gloss sont :

(19)
```
g.create_sentence("en","asl","i have going to the conference for five
years.");
g.create_clause("none");
g.create_token("i");
g.add_property("go", "habitual");
g.create_token("conference");
g.add_compound("up", "to", "now", "-");
g.add_numeral("five");
g.add_numeral("year");
affiche ();
enregistre ("fichier2.xml");
```

Le fichier XML-Gloss généré est le suivant :

(20)
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type='text/xsl' href='form.xsl'?>
<bloc xmlns:xsi="http://www.w3.org/2001/XMLSchema_instance"
xmlns:noNamespaceSchemaLocation="schemaXML.xsd">
<sentence src_lang="en" lang="asl" src_sentence="i have going to the
conference for five years.">
      <clause type_clause="none">
```

```
                <token>I</token>
                <token property="HABITUAL">GO</token>
                <token>CONFERENCE</token>
                <token compound="-">
                <token>UP</token>
                <token>TO</token>
                <token>NOW</token>
                </token>
                <token numeral="yes">FIVE</token>
                <token numeral="yes">YEAR</token>
        </clause>
</sentence>
```

Le rendu final est :

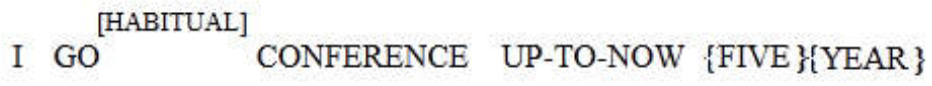


Figure 36. Transcription en XML-Gloss de la phrase "i have going to the conference for five years"

9. Conclusion

Dans la première partie de ce chapitre, nous avons d'abord présenté les différentes conventions et concepts utilisés pour décrire les gloses en ASL. Puis, l'architecture du système d'annotation à base des gloses en utilisant le langage XML permettant d'avoir une bonne représentation des signes à manipuler a été exposée dans la deuxième partie, tout en détaillant ses caractéristiques et son mode de construction. Enfin dans une dernière partie, nous avons décrit quelques exemples pour tester et valider notre API.

Chapitre 4 : Génération de discours en ASL à partir des règles de dépendances

1. Introduction

Nous avons vu dans les chapitres précédents, que les langues des signes sont très spécifiques, elles sont finalement très proches du point de vue structurel des langues vocales, au canal près de leur expression. Cela n'est pas étonnant dans la mesure où la langue, qu'elle soit vocale ou signée a, du point du vue psycholinguistique, la même vocation. Aussi, elle utilise les mêmes schémas cognitifs pour s'organiser [104].

La nécessité de passer par un niveau abstrait de représentation structurelle pour les langues est primordiale. C'est ce qu'on a fait tout au long du chapitre précédent. Au cours de ce chapitre, nous proposons une nouvelle approche de génération de la langue des signes Américaine (ASL) à partir de règles de dépendances grammaticales. A la fin du chapitre, nous présenterons l'évaluation du système de génération de discours en ASL proposé.

2. Motivation et contributions

Au cours de ce chapitre nous allons présenter une nouvelle approche de génération d'énoncés en ASL afin de traduire un texte Anglais et au même temps construire un corpus parallèles entre ces deux langues. L'approche proposée a été présenté dans le projet VisiCast [38] [105], mais elle s'est limitée au niveau syntaxique. Dans ce travail, nous intégrons plus que 52 relations de dépendances grammaticales lors de la génération. Ceci nous a permis de générer aussi les composantes non-manuelles.

Les recherches sur l'analyse lexicale, syntaxique, morphologique et sémantique d'un texte anglais est très avancées, et on trouve plusieurs outils qui donnent des résultats avec des taux de précision proche de 100% et un rappel (recall) proche de 90%.

Notre approche est décomposée en deux parties principales : la première est le traitement automatique du texte donné en entrée et le représenter sous forme d'un graphe sémantique. La seconde étape est la génération de la transcription XML-Gloss en utilisant l'API décrit dans le chapitre précédent.

3. Eléments de grammaire

Il est légitime de penser que les mêmes grandes catégories grammaticales sont partagées entre les langues signées et les langues vocales. Les éléments issus de l'observation des langues signées qui viennent remettre en cause les définitions et l'organisation de ces catégories (notamment la primauté du verbe comme constructeur de la phrase) ne font que renforcer les propositions des études sur l'oral et sur les langues des

signes. Par ailleurs, notre objectif étant la mise en œuvre d'un traducteur automatique, nous avons jugé opportun de ne pas chercher à créer de nouvelles catégories.

Nous avons préféré observer comment les énoncés de l'ASL que nous devions traiter pouvaient être décrits avec les outils traditionnels, utilisés pour les langues vocales. Ces outils étant également utilisés pour d'autres langues, ils sont donc a priori suffisamment malléables pour rendre compte d'éléments qu'on l'on pourra trouver dans l'ASL et pas dans les autres langues. Le choix de l'ASL vient du fait qu'elle est la seule langue des signes qu'on l'a défini une grammaire claire [65].

3.1. Noyau verbal

Le noyau verbal existe dans la mesure où des verbes sont clairement identifiés comme telles en langues vocales. Le verbe en ASL est l'entité principale pour la construction de l'énoncé en tenant compte des dépendances avec les autres signes du discours.

Partant de l'exemple illustré dans la Figure 37, le verbe "**ASK**" a deux signes différents. Pour le premier la configuration de la main est orientée vers la personne qui est en face du signeur. Pour le second, l'index de la main dominante est orienté vers le signeur.

Figure 37. Deux formes du verbe "ASK" en ASL. Celui à gauche correspond à la phrase "I ASK YOU" et celui à droite correspond à la phrase 'YOU ASK ME'.

Le phénomène illustré dans la Figure 37 a été observé dans plusieurs langues des signes outre que l'ASL [106], comme la langue des signes Australiennes [107], la langue des signes Brésilienne [108], la langue des signes Britanniques [109], la langue des signes Allemande [110], la langue des signes Japonaise [111] et autres.

Certaines recherches ont examiné les changements dans l'orientation et la direction du mouvement pour marquer l'accord du verbe, car la différence entre les deux formes correspond à une différence de sens qui est souvent marquée dans les langues parlées par accord de personne avec le sujet et l'objet. L'accord du verbe fait généralement référence à certains covariance systématique entre une propriété sémantique formelle ou d'un élément et une propriété formelle de l'autre, par exemple entre l'objet et le sujet.

Corbett [112] élargit cette définition en précisant qu'il ya quatre éléments principaux de la covariance systématique :

- Contrôleur (un élément qui détermine accord)
- Cible (un élément dont la forme est déterminée par accord)
- Domaine (l'environnement syntaxique dans lequel l'accord a lieu)
- Fonctions (les propriétés que le contrôleur est d'accord avec l'objet)

Une question fondamentale concernant le phénomène illustré à la Figure 37 est de savoir comment réaliser l'accord du verbe avec le sujet et l'objet, et si c'est le cas, quelles sont les caractéristiques pertinentes à la réalisation. Plusieurs approches ont été proposées dans les recherches linguistiques de l'ASL. On peut citer par exemple l'analyse R-locus proposé par Lillo Martin et Klima en 1990 [113].

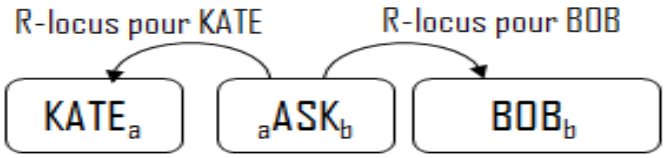


Figure 38. La relation R-locus dans la phrase "*Kate asked Bob a question*"

Dans l'exemple de la Figure 38, la relation R-locus permet de spécifier les accords avec le verbe 'ASK'. Selon cette analyse, la relation est expliquée comme un accord entre un syntagme nominal et un verbe dans le sens où ils partagent un indice référentiel, qui est réalisé ouvertement comme un R-locus.

3.2. Classificateurs

Supalla [114] considère que les modificateurs ou classificateurs, déjà décrit dans le chapitre précédent, sont classés en cinq catégories et dont certain possèdent des sous-catégories :

- Les classificateurs sémantiques : représentent les noms avec des caractéristiques appropriées aux référents lors d'un discours (humains, animale ou véhicule) ;
- Les spécificateurs de taille et de forme : qui désignent les noms selon les caractéristiques géométriques visuels de leurs référents :
  - statique : qui se composent d'une forme de la main (ou de la combinaison de deux mains) indiquant la taille / forme d'une entité ;
  - trajectoire : qui ont un mouvement de la main (s) qui décrit une entité de taille / forme, et dans lequel la forme de l'articulateur manuel désigne la dimension de cette entité.
- Les classificateurs instrumentaux, qui sont également classés de deux types :
  - classificateurs à main instrumentales, dans lequel la main représente une main qui tient et / ou manipule une autre entité ;
  - classificateurs à outils, dans lequel la main est un outil qui est en cours de manipulation.
- Classificateurs "Bodypart" : les parties du corps se représenter (par exemple, des mains, des yeux) ou membres (par exemple, mains, pieds) ;
- Un classificateur du corps : le corps du signataire représente une entité animée.

Dans la Figure 39, un exemple d'utilisation de classificateur. Dans la photo (a), le signe "BOOK" a été signé en utilisant la configuration de la main plate, par contre et pour le même signe, dans la photo (b) la configuration de la main est C (voir l'annexe A pour les configurations de mains en détails). Ici, le changement de configuration est causé par la sémantique de la phrase.

Dans le premier cas, l'action de mettre un livre dans la bibliothèque est similaire à l'action de placer un livre en réalité indépendant de la langue et de même pour la seconde action. L'information sémantique est extraite des prépositions grammaticales "in" et "from" de deux phrases comme l'indique la Figure 40.

Figure 39. Exemple de classificateur "flat" et "thick" du signe "BOOK"

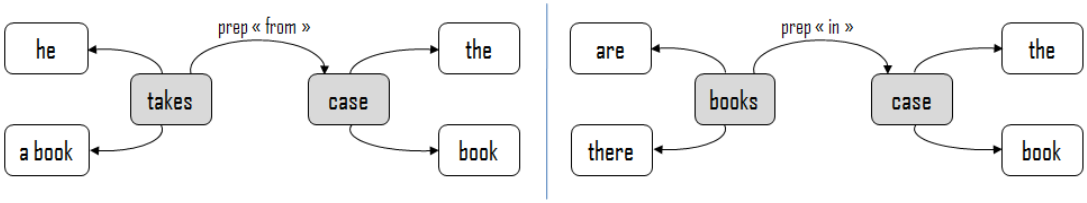


Figure 40. Les relations de dépendances "*IN*" et "*FROM*" dans deux phrases en anglais

Le phénomène des classificateurs peut-être reçu comme le rôle le plus important dans les langues des signes et que cela mérite. Le processus de formation du classificateur (ou modificateur) est considéré comme productif et évolutif selon contexte. Comme les orateurs, les signataires disposent de plusieurs moyens pour exprimer les relations spatiales entre les entités et les mouvements des entités à travers l'espace ; les classificateurs ne sont qu'un sous-ensemble de ceux-ci.

Les utilisateurs des langues des signes ont une gamme de classificateurs à leur disposition pour l'expression de l'existence, l'emplacement, le mouvement et le transfert, ainsi que la forme et l'orientation des entités. Ces classificateurs peuvent être combinés, mais les signataires peuvent également utiliser un seul de ces dispositifs, en se concentrant sur dé-focalisation ou un aspect particulier d'un événement.

3.3. Pronoms

Le terme "pronom" a été utilisé dans les langues parlées pour se référer non seulement aux pronoms personnels, mais aussi aux "proformes", y compris les pronoms démonstratifs, les pronoms indéfinis, les pronoms interrogatifs, les pronoms relatifs, etc.

Dans la langue des signes, les systèmes pronominaux ont été identifiés au moins aussi loin que le milieu des années 70 [115]. Depuis, le terme "pronom" a été largement utilisée pour désigner les signes dans différentes langues des signes qui ont la fonction des pronoms personnels indicatifs. C'est, en montrant des signes qui renvoient au signataire, au destinataire ou à des références non adressés.

Comme pour les langues parlées, le terme a été étendu pour désigner proformes comme indéfinis, interrogatifs, et les pronoms relatifs. Généralement en étude linguistique, les pronoms personnels jouent le rôle d'un sujet dans un discours en langue des signes. Pour les autres, il faut passer par une analyse sémantique en premier lieu.

Ceci est dû au fait que la modalité joue un rôle important. Le signataire place tous les objets et les acteurs, hors lui-même, dans son espace et se fait référence par son index ou par autres configurations de mains selon le nombre.

3.4. Le temps

À travers la langue des signes, la stratégie la plus courante pour localiser un événement au cours du temps est par l'intermédiaire d'adverbes temporels. Une phrase qui ne contient aucune référence de temps est soit interprétée dans le laps de temps établi précédemment dans le discours ou - par défaut - comme présent. Pourtant, les langues des signes ont généralement un signe lexical "**NOW**", qui peut être utilisé avec insistance ou de contraste pour indiquer présent de l'indicatif selon Friedman [115]. Par exemple, prenons la phrase en anglais suivante "*Last week the meeting started at ten and ended at a quarter to three.*" , sa transcription en ASL est la suivante: "**LAST-WEEK MEETING START TEN END QUARTER TO THREE**". Ici, la composante temporelle est définie par l'utilisation de deux mots "*last week*" qui forment un adverbe temporelle.

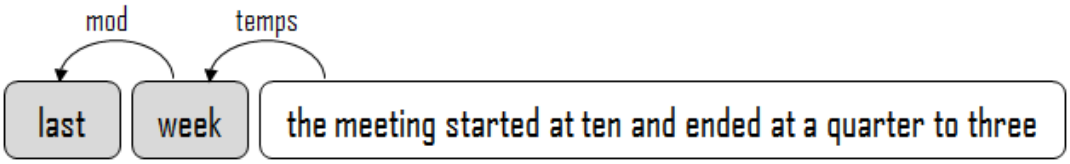


Figure 41. Relation de dépendance temporelle dans la phrase "*Last week, the meeting started at ten and ended at a quarter to three*"

Dans la Figure 41, la relation de dépendance temporelle pointe sur le mot "week" qui en relation avec le mot "last" aussi. Ici, pour déterminer le temps d'action dans la phrase, il faut extraire toute la préposition temporelle "last week".

3.5. Construction d'un énoncé

La construction d'un énoncé en SL fait recours à l'alignement des signes à partir d'une phrase source. Dans cette section, nous discuterons les problématiques relatives à l'ordre des signes quel que soit la modalité du langage. Il s'agit notamment de l'aspect fonctionnel de l'ordre, des problèmes articulatoires liés à la simultanéité dans les SL, et à la question de savoir si on peut identifier un ordre de signes de base. Bien que le terme "ordre constituant" est plus précis en cas de présence d'un sujet, un objet et un verbe. Mais, dans la plupart des cas, on a plus de termes à signer.

Dans notre étude, nous suivrons la démarche de Bouchard [116], qui a identifié trois aspects fondamentaux :

- un aspect fonctionnel, où l'ordre des éléments fournit des informations sur la combinaison de mots et qui, à son tour, fournit des règles sur la façon d'interpréter la phrase ;
- un aspect articulatoire qui suppose qu'on ne peut signer deux signes à la fois (comme pour les langues parlées) ;
- la présomption de l'existence d'un ordre des signes (mots) de base

A partir de l'identification des électeurs discrets, la recherche cross-linguistique et typologique a identifié une série d'associations entre les ordres spécifiques dans des langues et des fonctions particulières. Par exemple, les relations d'ordre ont été identifiés entre un verbe et ses arguments, qu'ils soient exprimés en affixes ou des phrases distinctes, qui permettent d'identifier la structure propositionnelle de la clause. Nous pouvons nous référer à une langue qui présente ce comportement comme argument de configuration. Ceci peut être réalisé indirectement par l'intermédiaire d'un système de relations grammaticales (sujet, objet, etc) ou directement via rôles sémantiques (l'agent, patient, etc.). Greenberg [117] a proposé un travail sur l'ordre typologique des mots dans un discours, qui caractérise les langages comme SVO, SOV, etc. et ceci suppose l'omniprésence de ce rôle de l'ordre dans la détermination de la signification propositionnelle. Cet ordre typologique a été respecté dans la construction d'énoncé en ASL en se basant sur l'aspect fonctionnel.

Pour l'aspect articulatoire, et par analogie aux langues parlées, on ne peut pas signer des mots à la fois puisque on ne peut pas prononcer deux mots à la fois aussi. Par contre, cet aspect soulève des questions sur la séquence chronologique et la discrétisation en se reliant directement à la modalité. En effet, les langues des signes peuvent exprimer du sens ou de l'information, outre que les signes, grâce à la simultanéité. La simultanéité est effectuée par le couplage entre les gestes manuels et les gestes non-manuels en même temps. Ceci permet à rajouter de l'information afin de signer les phrases interrogatives (*wh* ou *yes/no*), la négation, les structures de type "*topic*", etc. Et pour le troisième aspect, la présomption de l'ordre des signes n'est pas confirmée puisque on trouve jusqu'à aujourd'hui des travaux sur les langues des signes qui supposent que l'ordre des signes est libre [118] [119].

3.6. Types de phrases

Lorsque on étudie la construction de l'énoncé dans les langues des signes, on est face à définir les types de phrases de la SL. Par analogie, dans les langues parlées, les types de phrases sont établis. On trouve les phrases exclamatives, interrogatives, déclaratives, impératives, etc. Cependant, les langues des signes peuvent être utilisés pour exprimer une affirmation, poser une question, donner un ordre, donc, il n'est pas surprenant qu'il y en a des formes grammaticales associés à ces usages conversationnels. En effet, dans la littérature des langue des signes, on trouve un ensemble considérable de travaux sur les structures de phrases déclaratifs, négatifs et interrogatifs, par contre, les recherches sur les autres types de phrases sont extrêmement limitées. Pour l'ASL, on s'intéresse aux types suivants :

- les questions de type "*yes/no*"
- les questions de type "*wh*"
- les phrases négatives

3.7. Négation

L'expression de la négation propositionnelle en langues des signes est caractérisée par nombreuses propriétés morphologiques et syntaxiques attesté pour les langues

parlées, par analogie. Cependant, les gestes non-manuels de la négation, comme la secousse de la tête et les expressions faciales, jouent un rôle très important dans ces langues. Ils interagissent de diverses façons avec les gestes manuels suivant une structure syntaxique de clauses négatives et introduisant ainsi des caractéristiques spécifiques de modalité.

Par exemple, dans la Figure 42, la négation est exprimé par une animation faciale marqué par un trait en haut du signe "**BUY**". Ici, la négation est marqué au niveau du verbe seulement, le sujet et l'objet sont signés à part. Dans d'autres, la négation peut suivre toute la phrase lors d'une interprétation et ce qui est rare.

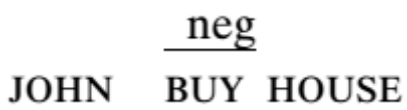


Figure 42. Transcription en glose de la phrase "*John doesn't buy the house*"

En ASL, et comme on adit, la négation est effectué par des composantes non-manuelles. Certes, et dans d'autres langues des signes, comme celle de la Jordanie, la négation peut être signé. Autrement dit, le signe de négation existe déjà dans le dictionnaire comme on le voit dans la Figure 43 [120].

Figure 43. Geste manuel pour exprimer la négation dans la langue des signes de la Jordanie

Indépendamment de la façon de signer la négation (manuel ou non-manuel), la négation est extraite de la relation de dépendance grammaticale entre le verbe d'action et les adverbes de négation. Pour nous, on s'intéresse seulement à la génération de cette relation de négation à partir d'un texte ou une phrase et ensuite construire une transcription en XML-Gloss selon la langue des signes cible souhaitée.

3.8. Coordination et subordination

Identifier la coordination et la subordination dans la langue des signes n'est pas facile parce que les relations morphologiques et syntaxiques, qui marquent les limites de la clause, comme conjonctions ou compléments, ne sont pas obligatoire, généralement. Les coordonnées sont introduites par les conjonctions de coordination "*but, so, comma, etc.*" et leurs équivalents fonctionnels.

A première vue, on peut penser à une décomposition des phrases subordonnés en deux parties ou plus et traiter chaque phrase à part. Mais, on rencontre un problème de coréférence qui consiste pour plusieurs expressions différentes contenues dans une phrase ou dans un document, à désigner le même objet. Par exemple une personne, un lieu, un événement, ou encore une date [121].

Prenons l'exemple suivante, étant donné une phrase en anglais "*Mary said she would help me*", le mot "*she*" et le mot "*Mary*" référent à la même personne (voir Figure 44). Donc, l'étude de coordination et de subordination est très importante afin de tenir compte des relations de dépendances entre les signes et surtout lors de la création de l'espace de signation.

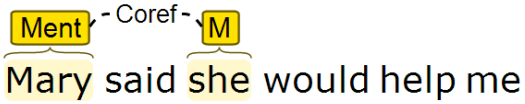


Figure 44. La relation de coréférence dans la phrase "*Mary sais she would help me*"

3.9. Les émotions

Les émotions, de façon générale, sont des états motivationnels. Elles sont constituées d'impulsions, de désirs ou d'aversions ou, plus généralement, elles comportent des changements de motivation. Elles poussent l'individu à modifier sa relation avec un objet, un état du monde, un état de soi, ou à maintenir une relation existante malgré des obstacles ou des interférences.

Dans les langues des signes, la modalité visuelle-gestuelle fournit un système à deux canaux dans lequel l'information linguistique distincte peut être exprimée en même temps par le visage et par les mains. Les sourds, comme les personnes non-sourdes, expriment les six émotions (joie, tristesse, colère, peur, dégoût et surprise), selon la classification d'Ekman [122], par des expressions faciales. Ces expressions faciales émotionnelles et linguistiques diffèrent dans leur portée et le temps de l'expression et les muscles du visage utilisés lors de l'interprétation [123]. Et elles interviennent dans le contexte général du discours.

Ici, il est nécessaire de déterminer à partir d'une phrase, si c'est possible, de reconnaître l'émotion afin de la transcrire et tenir compte de son état lors de la synthèse.

4. Discussions

L'étude des relations de dépendances grammaticales entre les mots dans une phrase en Anglais, nous montre qu'à partir de ces relations on peut générer des discours en langue des signes Américaine (ASL). Les structures de phrases et les types de construction d'énoncés proposés par les linguistiques, nous montrent que l'ASL a une grammaire dirigée par des règles.

Donc, étant donnée une phrase en entrée, peut-on construire un énoncé en ASL ? Pour répondre à cette question les sourds emploient leurs cognitions pour l'interprétation d'une manière rapide et implicite. Tout d'abord, le cerveau ordonne les signes (par exemple : "*temps → sujet → objet → verbe*").

Ensuite, le sourd créé son espace de signation en plaçant les différents sujets et objets en tenant compte des relations sémantiques entre les objets. Enfin, le sourd procède à l'interprétation de son discours. D'après les sections précédentes, l'analyse syntaxique, morphologique et sémantique est très intéressante et qui sera notre point de départ dans notre système de traduction automatique vers les langues des signes à partir des règles de dépendances entre les signes extraites des relations grammaticales.

Au cours du reste de ce chapitre, nous décrivons notre approche de génération des discours en ASL à partir d'un texte en anglais.

5. Description de l'approche proposée

Avant de commencer la description de l'approche, nous présentons un aperçu sur le domaine de la traduction automatique. La traduction automatique (TA) de textes écrits s'inscrit dans le domaine très large du traitement automatique du langage naturel.

Les approches employées pour la construction de systèmes de TA peuvent être divisées en deux grandes classes différentes : l'approche directe est encore appelée l'approche de première génération, et les approches de deuxième génération consistent essentiellement en deux approches : la méthode de transfert et la méthode interlingua qui utilise une représentation interlinguale.

L'approche directe est une stratégie qui enlève toute étape dans les traitements de traduction : le traitement du texte de la langue source en entrée conduit directement au texte de la langue cible désirée en sortie du système. Le trait le plus représentatif de cette approche est que toute analyse de la langue source est dirigée par les particularismes de la langue cible. Cette approche représente la première génération des traducteurs automatiques apparu dans les années soixante.

Les approches de deuxième génération représentent le besoin de rendre les analyses des langues sources indépendantes des langues cibles et les générations indépendantes des langues source. La méthode interlingua propose d'analyser le texte source dans une forme de représentation intermédiaire à partir de laquelle le texte ciblé est généré. La représentation intermédiaire inclut toute l'information nécessaire pour la génération du texte cible sans tenir compte du texte d'entrée.

La méthode est interlinguale dans le sens où elle est neutre entre deux ou plusieurs langues. La représentation intermédiaire est indépendante de la langue source et de la langue cible, le but des analyses en langue source est la création de la représentation interlinguale et la langue cible n'a aucun effet sur les processus d'analyses. La représentation interlinguale est une structure arborescente, une ontologie dans un domaine ou une structure désambiguïsée.

La traduction par transfert propose comme la méthode interlingua, un ensemble d'analyses indépendantes de la langue cible et de générations indépendantes de la langue source. Seulement, les systèmes basées sur la méthode de transfert construisent des représentations intermédiaires qui sont dépendantes de la langue cible. A l'issue des analyses de la langue source, la représentation intermédiaire est manipulée par des

composants du système dépendants de la langue cible. Le système est alors lié non seulement aux deux langues mais aussi au sens de la traduction.

L'analyse et la génération sont séparées en deux modules morphologique et syntaxique. Le transfert est découpé en deux phases, le transfert lexical et le transfert structurel. Après les phases d'analyses morphologique et syntaxique, la structure résultante est passée au transfert lexical pour acquérir les unités lexicales de la langue cible puis au transfert structurel pour donner la structure syntaxique intermédiaire cible avant la génération syntaxique dans la langue cible.

Nous utilisons la méthode de la traduction par transfert pour effectuer la traduction automatique de l'anglais vers la langue des signes Américaine, la langue cible. Nous ajoutons une analyse complémentaire, qui peut être définie comme une préparation au passage à la langue cible, en extrayant des textes source des données indispensables à la mise en place du transfert.

Une idée sous-jacente à la suite d'analyses dans la langue source est de résoudre à chaque fois que l'on monte d'un niveau d'analyse, l'ambiguïté due au niveau d'analyse inférieure. L'analyse syntaxique doit choisir alors entre les diverses formes celle qui convient. L'analyse sémantique doit pouvoir choisir l'acception de sens convenant au contexte de la phrase. La résolution automatique de l'ambiguïté lexicale sémantique a principalement engendré deux types d'approches provenant de domaines différents :

- La première approche provient du domaine de la recherche d'informations et de domaines connexes. Elle utilise une représentation informatique des sens des mots pour discriminer de manière automatique le sens en contexte. L'avantage est donc de remplacer le codage manuel énorme dans le second type d'approche, par une procédure automatique de choix de ces sens. Comme les traitements sont effectués sur de grands corpus de textes où doit être minimisé le temps d'exécution, des méthodes statistiques sont utilisés pour calculer les sens des mots dans le texte.
- La deuxième approche est celle généralement issue du génie linguistique. Elle repose sur le codage dans un dictionnaire des mots et des sens des mots en fonction de ses critères morphosyntaxiques et de relations lexicales entre les sens des mots. L'inconvénient majeur de cette méthode est l'effort important demandé par le codage manuel des sens des mots dans le dictionnaire. L'avantage majeur est l'étude linguistique des mots et de leurs acceptions de sens pour leur intégration dans le dictionnaire. Dans cette approche, les sens des mots sont désambiguïsés de manière manuelle dans le dictionnaire.

Notre approche consiste à utiliser une analyse syntaxique dans la lignée de la seconde méthode, couplée à une représentation des sens des mots et une comparaison des mots provenant de la première méthode. Toutes ces étapes sont illustrées dans le triangle de Vauquois de la Figure 45 [124].

La démarche retenue consiste à implémenter sous forme d'espace vectoriel les diverses notions d'un thesaurus d'idées de manière à pouvoir récupérer les idées des sens des mots donnés dans un lexique sémantique. Les idées de chaque mot, manipulées sous forme de vecteurs, sont remontées des feuilles de la structure syntaxique vers les points

supérieurs pour obtenir les idées représentatives des phrases et des textes en traitement. Les vecteurs associés au texte représentent alors le contexte du texte.

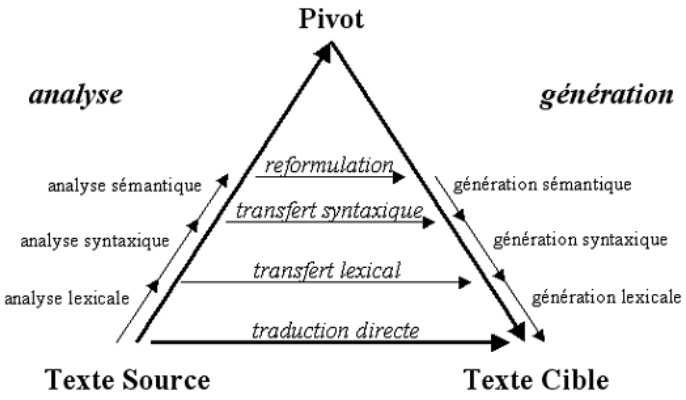


Figure 45. Principe de traduction automatique selon le triangle de Vauquois

5.1. Architecture du système proposé

L'organisation des niveaux de traitements linguistiques de notre système est à l'image de celles des niveaux du triangle de Vauquois. Ainsi, comme le montre la Figure 46, le système, pour l'analyse, est organisé suivant trois niveaux principaux : lexical, syntaxique et sémantique, dont les briques logicielles constituent une chaîne de traitements linguistiques.

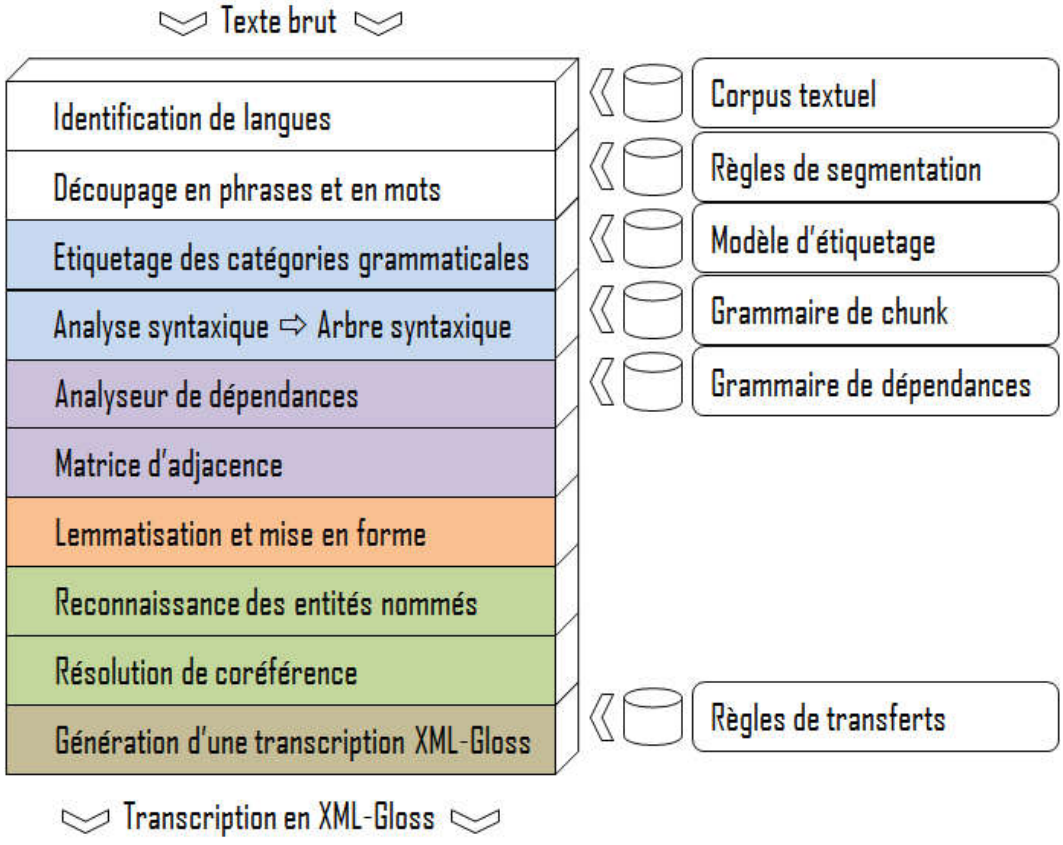


Figure 46. Architecture du système de traduction proposé

L'ensemble est architecturé autour de plusieurs modules supervisés par un module de contrôle. Ces modules contiennent les différentes données en fonction desquelles l'analyse et la génération de message vont être faites :

- Données pour la segmentation : découpage d'un texte en phrases, mots ou signes, numéros, signes de ponctuation, etc.
- Lexiques monolingues, avec des informations morphologiques pour l'association de chaque mot à ces différentes analyses hors contexte
- Traduction mot-à-mot à partir d'un dictionnaire
- Grammaire de chunks, pour la séparation du message en constituants de base non-imbriqués : groupe nominaux, groupes prépositionnels, groupes verbaux, etc.
- Grammaire de dépendances, pour les relations syntaxiques entre les mots.
- Traduction de traits, qui permet de mettre en relation des traits lexicaux et les étiquettes grammaticales
- Règles de construction sémantique, pour affiner la pertinence des relations syntaxiques et ne retenir que celles dégageant un sens
- Règles de transformation de graphe, pour la reformulation lorsque, pour un énoncé donné dans la langue source, l'analyse d'un énoncé attendu dans la langue

Au cours de processus d'implémentation, les lexiques, la ou les grammaire(s), la traduction de traits et les données de segmentation sont compilés avant d'être utilisés par le système. En ce qui concerne son fonctionnement, le système proposé prend en entrée un texte, le segmente en paragraphes, phrases et mots et génère actuellement plusieurs niveaux d'analyse :

- l'analyse minimale
- l'analyse syntaxique (chunking et/ou dépendances)
- l'analyse sémantique et la génération de formules ontologiques.

L'idéal est de coupler l'analyse syntaxique avec une analyse sémantique qui affine la désambiguïsation. Le résultat de l'analyse syntaxique peut être utilisé pour l'analyse de corpus, en entrée d'outils d'extraction d'informations ou d'indexation de textes et de traitement de requêtes.

Il est donc nécessaire que le mécanisme soit robuste, c'est -à-dire qu'il soit capable de donner toujours une solution, qu'elle soit correcte pour un maximum de phrases des textes traités, ou dégradée pour les autres.

5.2. Aperçu du fonctionnement du système

Revenons à la chaîne de traduction du système proposé en simulant le traitement d'un exemple. Rappelons les principales phases de l'analyse qui se fait de façon classique suivant le schéma présenté à la Figure 46 dont nous avons transformé la sortie qui, pour les autres langues, est l'énoncé obtenu dans la langue cible. Ainsi, après une étape de segmentation, les informations morphologiques, syntaxiques et sémantiques de chacun des mots de l'énoncé source sont recherchées dans un lexique. Une grammaire de dépendance est ensuite utilisée pour construire le(s) arbre(s) représentant la structure

syntaxique de l'énoncé. L'arbre obtenu pour l'analyse d'une phrase comme "*Kate gave chocolate for each boy, yesterday*" est donné par la Figure 47. En effet, la structure de l'énoncé cible en ASL est "*temps → sujet → verbe → objet → complément d'objet*". La première étape est un prétraitement appliqué directement sur la phrase en entrée suivi d'une analyse de dépendance entre les mots qui sera décrite dans les sections suivantes. Ensuite, on passe à analyser les structures sémantiques entre les mots.

Suite à cette analyse, on génère une structure syntaxique propre à l'ASL, qui à partir de cette structure, on procède à la linéarisation et la mise en forme selon le système de transcription XML-Gloss.

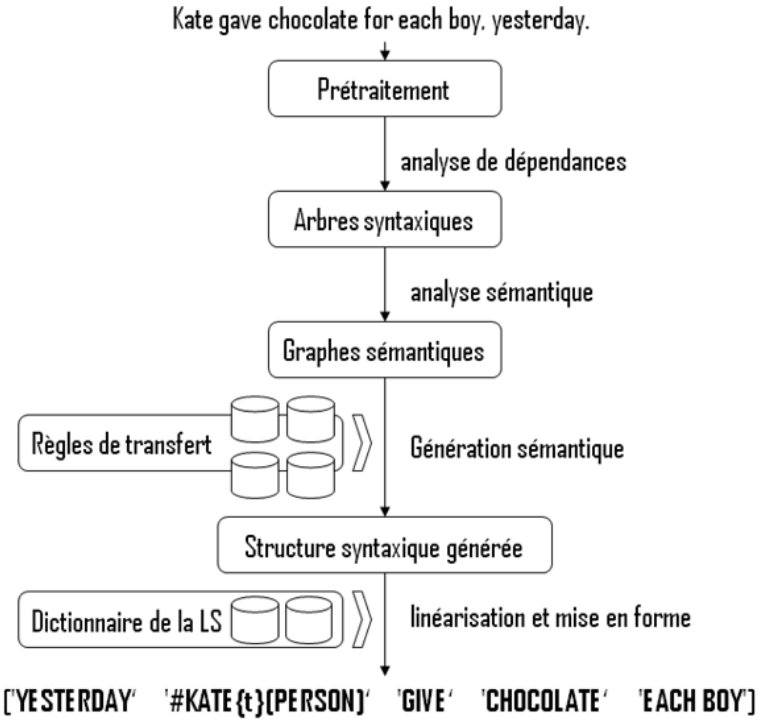


Figure 47. Aperçu du fonctionnement du système de traduction proposé

6. Implémentation

A partir de l'exemple précédent, nous décrivons la phase d'implémentation étape par étape. La phrase en entrée est "*Kate gave chocolate for each boy, yesterday*".

6.1. Prétraitement et analyse lexicale

Le prétraitement est une phase nécessaire et indispensable pour la conversion des données brutes en un format adéquat pour le traitement automatique dans notre système de génération d'énoncé en ASL. On commence toujours par segmenter le texte en phrases et pour chaque phrase on effectue le même prétraitement.

La première opération de prétraitement est découpage en mots (dit "*tokenisation*"), dont l'objectif est justement de transformer la chaîne de caractères fournie en entrée en jetons "*tokens*". Cette opération s'applique aux textes source et va prendre en compte les espaces pour séparer les mots, les nombres et la ponctuation. Dans notre exemple les jetons sont :

```
(1)  ["Kate" ; "gave" ; "chocolate" ; "for" ; "each" ; "boy" ; "," ; "yesterday"]
```

Ensuite, on transforme tous les caractères en minuscules. On aura :

```
(2)  ["kate" ; "gave" ; "chocolate" ; "for" ; "each" ; "boy" ; "," ; "yesterday"]
```

La "*tokenisation*" est une opération assez simple pour certaines langues (ex. français, Anglais) mais assez compliquée pour d'autres, surtout quand la segmentation des mots est nécessaire, comme c'est le cas pour le chinois, l'arabe et l'allemand.

6.2. Analyse grammaticale

A partir de l'analyse lexicale (l'ensemble des jetons), on procède à l'analyse grammaticale de chaque jeton afin de construire l'arbre syntaxique qui nous permet de passer à l'analyse sémantique. L'analyse syntaxique consiste à associer une catégorie grammaticale (nom, verbe, adjectif, adverbe, nom propre, etc.) pour chaque mot de la phrase en entrée. La liste complète des abréviations des catégories grammaticales (VBD, DT, NN, IN, etc.) est décrite dans le Tableau 6.

Pour notre phrase en entrée, on a :

```
(3)  "kate"        → NNP
     "gave"        → VBD
     "chocolate"   → NN
     "for"         → IN
     "each"        → DT
     "boy"         → NN
     ","           → SYM
     "yesterday"   → NN
     "."           → SYM
```

Pour l'étiquetage des catégories grammaticales, on a utilisé l'outil "*Stanford Log-linear Part-Of-Speech Tagger*" [125] qui est basé sur le modèle d'entropie maximale ("*Maximum Entropy Model*" en anglais). Le modèle assigne une probabilité pour chaque catégorie grammaticale t de l'ensemble des catégories grammaticales T (voir Tableau 6) pour un mot w donné dans son contexte h. Le modèle utilisé permet d'estimer la probabilité de la séquence d'étiquettes $(t_1, \dots, t_n)$ de la phrase $(w_1, \dots, w_n)$ à partir de :

$$P(t_1, \dots, t_n | w_1, \dots, w_n) = \prod_{i=1}^{n} P(t_i | t_1, \dots, t_{i-1} | w_1, \dots, w_n) \approx \prod_{i=1}^{n} P(t_i | h_i)$$

Le taux de précision globale de l'étiquetage est 96.72% pour des mots connus lors de l'apprentissage et 84% pour les mots inconnus. Par exemple, que le taux de précision de la catégorie grammaticale "IN" est 97.3%., pour le "VB" est 94% et le "NNPS" est 41.1%.

Tableau 6. Liste des catégories grammaticales en Anglais

POS	English Part-of-Speech	Example
CC	Coordinating Conjunction	and, but, for, nor, or, so, yet
CD	Cardinal Number	12, 20, 5, etc.
DT	Determiner	the, a, my, some, which, both, etc.
EX	Existential there	there
FW	Foreign word	
IN	Preposition	in, from, with, throughout, etc.
JJ	Adjective	yellow, fast, slow, etc.
JJR	Adjective, comparative	bigger, etc.
JJS	Adjective, superlative	wildest, fastest, etc.
LS	List item marker	1 2 3 4 , i ii iii, etc.
MD	Modal	can, may, must, will, shall
NN	Noun, singular or mass	car, bus, name, etc.
NNS	Noun, plural	dogs, cats, etc.
NNP	Proper noun, singular	Kate, John, etc.
NNPS	Proper noun, plural	Carolinas, etc.
PDT	Predeterminer	all, etc.
POS	Possessive ending	’s
PRP	Personal pronoun	I
PRP$	Possessive pronoun	your, etc.
RB	Adverb	never, now, soon, incredibly, etc.
RBR	Adverb, comparative	faster, etc.
RBS	Adverb, superlative	fastest, etc.
RP	Particle	off, etc.
SYM	Symbol	. , ! ? ; : - etc.
TO	To	to
VB	Verb, base form	eat, go, swim, etc.
VBD	Verb, past tense	ate
VBG	gerund or participle participle	eating
VBN	Verb, past participle	eaten
VBP	Non-3rd person singular present	eat
VBZ	3rd person singular present	eats
WDT	Wh-determiner	which
WP	Wh-pronoun	what
WP$	Possessive wh-pronoun	whose
WRB	Wh-adverb	where

6.3. Analyse syntaxique

Après avoir déterminé les catégories grammaticales de chaque mot de la phrase donné en entrée, on passe maintenant à l'analyse syntaxique afin de construire l'arbre syntaxique. L'arbre syntaxique [126] contient un nœud pour chaque mot. Le rôle du parseur est d'établir une connexion entre deux nœuds pour créer un nouveau nœud. La

tâche d'un analyseur de dépendances entre les nœuds est de prendre une série de mots et de lui imposer des liens. Dans ce qui suit, nous allons citer quelques hypothèses :

- **Unicité** : le résultat du processus d'analyse est un arbre unique (avec une unique racine) comprenant tous les mots dans la chaîne d'entrée.
- **Adjacence** (contiguïté) : si le mot A dépend mot B, alors tous les mots entre A et B sont également subordonnées à B.
- **1-mot** : l'analyseur examine un seul mot à la fois et le place dans l'arbre.
- **Gauche→Droite** : l'analyseur traite la phrase en entrée de gauche à droite sans faire recours à un retour arrière afin de minimiser les ambigüités.
- **Empressement** : pour chaque passage de gauche à droite, le parseur doit établir un lien entre deux mots.
- **Aucune ambiguïté** : on suppose qu'il n'y aura pas d'ambiguïté lors de l'établissement des liens entre les nœuds [127].
- **No-Null** : on suppose qu'il n'y a pas d'éléments nuls ou lien nulle.
- **Atomicité** : les mots sont des éléments non analysables et il n'y a pas d'opérations sur les fonctionnalités ou la structure interne des mots.

A partir de ces hypothèses, la stratégie d'analyse syntaxique est évidente. Il existe trois stratégies :

- **Stratégie 1** ("*Brute-force*"): pour chaque paire de mots, on examine le lien de dépendances si on une règle grammaticale. Si on a n mots dans la phrase en entrée, le nombre de paires est $n(n-1)$. La complexité de la construction de l'arbre est $O(n^2)$. Si on a un retour arrière ("*backtracking*" en anglais), la complexité de l'algorithme devient $O(n^3)$.
- **Stratégie 2** ("*Exhaustive left-to-right search*") : on examine les mots un par un et on essaye d'établir un lien reliant chaque mot en tant que racine (de l'arbre en totalité ou d'un sous-arbre) ou qui dépend des mots analysés précédemment. Le principe de la stratégie est présenté dans l'algorithme 1.
- **Stratégie 3** ("*Enforcing uniqueness*") : le principe de cette stratégie est que lorsque un mot est une racine (arbre ou sous-arbre) alors il ne peut pas avoir une autre racine. C.-à-d. lorsque on examine tous les mots, on suppose qu'il y'a un lien, sinon, le nœud est une racine et on arrête l'itération. L'algorithme de cette stratégie est décrit dans l'algorithme 2. La complexité de l'algorithme est $O(n^2)$.

Algorithme 1 : ESH (*Exhaustive left-to-right search, heads first*)

```
Entrée: A partir d'une phrase composé de n mots et une grammaire G
Sortie: liens entre les n mots de la phrase en entrée
1.    Début
2.    pour i de 1 n faire
3.          pour j de i-1 à 1 pas -1 faire
4.                si règle(mot[i],mot[j])∈G
5.                      mot[j] est le parent du mot[i]
6.                      mot[j] dépend du mot[i]
7.                finsi
8.          finpour
9.    finpour
10.   fin
```

Algorithme 2 : ESHU (*Exhaustive search, heads first, with uniqueness*)

```
Entrée: A partir d'une phrase composé de n mots et une grammaire G
Sortie: liens entre les n mots de la phrase en entrée
1.    Début
2.    pour i de 1 n faire
3.          pour j de i-1 à 1 pas -1 faire
4.                si le mot[i] n'a pas de lien alors
5.                      si règle(mot[i],mot[j])∈G
6.                            mot[j] est le parent du mot[i]
7.                            si le mot[j] n'a pas de lien alors
8.                                  mot[j] dépend du mot[i]
9.                            finsi
10.                     finsi
11.               finsi
12.         finpour
13.   finpour
14.   fin
```

A partir des trois algorithmes de recherche de dépendances, on établit les liens de dépendances entre les mots en se basant sur le principe de parent-fils. La condition "règle(mot[i],mot[j])∈G" est vérifiée à partir des règles appartenant à la grammaire donné à l'entrée. Cette grammaire est définie par des règles lexicales et des règles internes. Les règles lexicales sont extraites à partir de l'analyse grammaticale de l'étape précédente qui consiste à donner une étiquette grammaticale pour chaque mot de la phrase donnée en entrée. Pour les règles internes de la grammaire, on se base sur le modèle de Collins [128]. Ce modèle probabiliste est basé sur les statistiques extraites à partir d'un corpus où chaque phrase du corpus a un arbre syntaxique selon une grammaire lexicale donné par un linguiste [129].

Pour notre exemple, les règles lexicales sont :

```
(4)  NNP("kate",NNP)      → kate
     VBD("gave",VBD)      → gave
     NN("chocolate",NN)   → chocolate
     IN("for",IN)         → for
     DT("each",DT)        → each
     NN("boy",NN)         → boy
     SYM(",",SYM)         → ,
     NN("yesterday",NN)   → yesterday
     SYM(".",SYM)         → .
```

Et les règles internes sont:

```
(5)  S       → NP VP SYM
     NP      → NNP
     NP      → NN
     NP      → DT NN
     PP      → IN NP
     VP      → VBD NP PP SYM NP
```

A partir des règles lexicales et les règles internes de la grammaire G, on construit l'arbre syntaxique de la phrase "*Kate gave chocolate for each boy, yesterday*". L'arbre est illustré dans la Figure 48. L'arbre est construit à partir de l'algorithme 2, dont la complexité est $O(n^2)$, cité précédemment dont on suppose qu'il n'y aucune ambiguïté grâce à l'hypothèse de l'atomicité. A ce point, si on va tenir compte de l'ambiguïté et les accords entre les verbes et les autres mots, la construction de l'arbre syntaxique devient un problème NP_Complet [130].

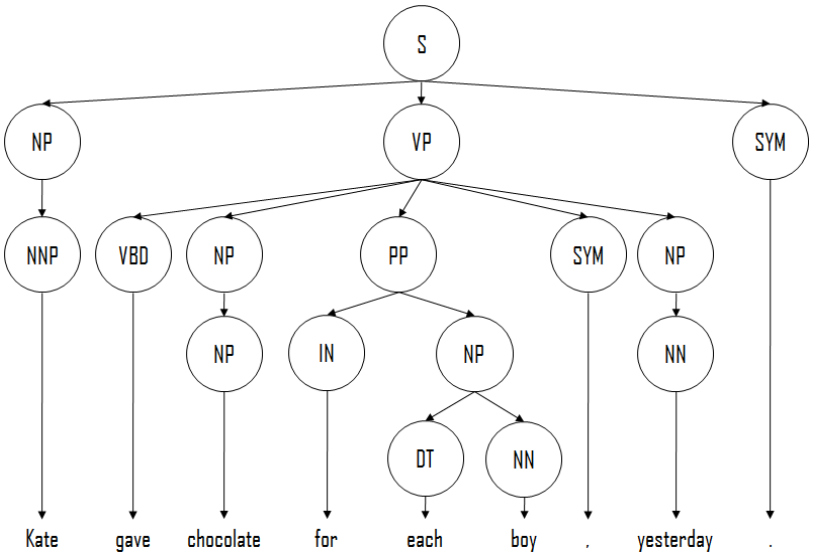


Figure 48. Arbre syntaxique de la phrase "*Kate gave chocolate for each boy, yesterday*"

6.4. Analyse de dépendances

L'analyse de dépendances permet d'étiqueter les relations grammaticales fournit par l'analyse syntaxique. La relation de dépendances entre deux mots est binaire : une partie dite "agent" et l'autre partie est le "dépendant". Pour extraire les relations, on se base sur le travail de Marneffe et al. [131]. Leur représentation contient 53 relations de dépendances. On cite seulement quelques relations dépendances qui sont les plus utilisés lors de la génération d'un discours en ASL (le reste des dépendances ainsi que la hiérarchie sont décrites dans l'annexe D et E) :

- **18. det** - determiner: il transforme le nom dont il est, en un syntagme nominal.
- **19. dobj** - objet direct : Le complément d'objet direct du syntagme VP.
- **28. nsubj** - sujet nominal : c'est un syntagme nominal qui présente le sujet syntaxique d'une clause.
- **40. prep** - modificateur propositionnel : Un modificateur propositionnel d'un verbe, adjectif ou d'un nom, est une préposition qui sert à modifier le sens du verbe, de l'adjectif, du nom, ou même d'une autre préposition.
- **49. root** - root: permet de définir la racine et le point de départ dans la phrase. C'est un élément fictif.
- **50. tmod** - modificateur temporel : il s'agit d'une expression qui sert à exprimer le temps dans une phrase.

Les indices (18,19, 28, 40, 49, 50, etc.) seront utiles pour des traitements ultérieurs. A partir des relations, on construit le graphe de dépendances commençant par la relation "*root*", sachant que les relations ne couvrent pas tous les mots, mais plutôt sur les mots principaux. Plus généralement, tous les mots s'accordent sur le sens de la relation de dépendance lorsqu'il existe une relation de subordination, c.à.d. lorsqu'il existe une relation actancielle (entre tête et son actant) ou une relation modificative (entre tête et modificateur). La Figure 49 illustre le graphe de dépendances de notre phrase exemple.

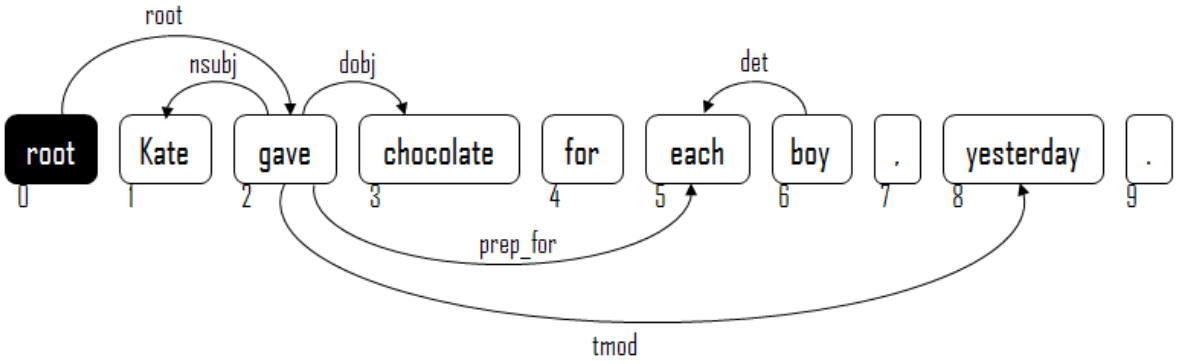


Figure 49. Les relations de dépendances de "*Kate gave chocolate for each boy, yesterday*"

6.5. Matrice d'adjacence

A partir des relations de dépendances, on définit un graphe fini G à n sommets, tel que n est le nombre de mots de la phrase en entrée. L'arête reliant deux sommets i et j est notée (V_i, V_j). L'ensemble des arêtes est V. Pour notre exemple, le graphe de dépendances est illustré dans la Figure 50.

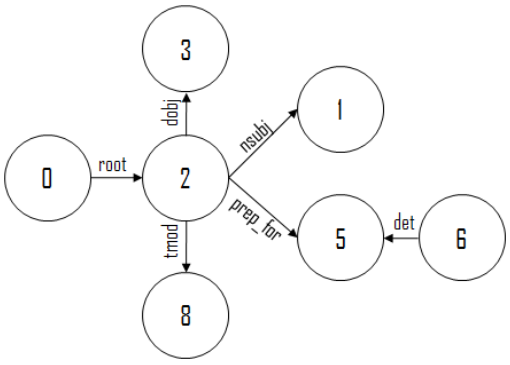


Figure 50. Graphe de dépendance de la phrase "*Kate gave chocolate for each boy, yesterday*"

La matrice d'adjacence A de taille $n \times n$ est déterminé à partir de G. Un élément de la matrice a_{ij} (une arête entre deux sommets i et j) est défini comme suit:

$$a_{ij} = \begin{cases} 0 \; si \; (V_i, V_j) \notin E \\ > 0 \; sinon \end{cases}$$

Si la valeur de a_{ij} est strictement supérieur à zéro, alors sa valeur correspond à l'index d'une relation de dépendance décrite dans l'annexe D. A partir de cette définition et de notre exemple, on construit la matrice d'adjacence de la Figure 51. La complexité de construction de cette matrice est $O(m)$ tel que m est le nombre des arêtes du graphe G.

	00	01	02	03	04	05	06	07	08	09
00	0	0	0	0	0	0	0	0	0	0
01	0	0	28	0	0	0	0	0	0	0
02	49	0	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0	0	0
04	0	0	0	0	0	0	0	0	0	0
05	0	0	0	0	0	0	18	0	0	0
06	0	0	40	0	0	0	0	0	0	0
07	0	0	0	0	0	0	0	0	0	0
08	0	0	50	0	0	0	0	0	0	0
09	0	0	0	0	0	0	0	0	0	0

Figure 51. Matrice d'adjacence du graphe de dépendance de la phrase "*Kate gave chocolate for each boy, yesterday*"

Pour générer l'énoncé en ASL, il faut définir la règle de la sortie, et c'est ce qu'on fait au début du chapitre. En effet, la structure de l'énoncé doit respecter la règle suivante :

(6) "*temps → sujet → verbe → objet → complément d'objet*".

A partir de la matrice d'adjacence, on extrait les composantes qui construisent l'énoncé par ordre. Pour chaque composante, on récupère tous les mots en relation à partir de sa colonne. Par exemple, l'extraction des composants se fait comme suit :

(7) "*tmod → nsubj → root → dobj → prep_for*"

La composante "*prep_for*" contient aussi une relation récursive qui est la relation "*det*". Donc, l'ordre d'extraction sera :

(8) "*tmod → nsubj → root → dobj → prep_for+det*"

Le résultat de l'algorithme d'extraction est :

(9) "*yesterday kate{t} gave chocolate each-boy*"

Le suffixe {t} est appliqué aux sujets de la phrase afin de mentionner la composante non-manuelle "*topic*".

Les règles de transfert vers la langue des signes Américaine à partir des relations de dépendance grammaticale sont données par les linguistes. Aussi, lors de la génération de l'énoncé, il faut spécifier à l'avance le type de l'énoncé (SVO, SOV, etc.). Ceci fonctionnalité, malgré quelle est manuelle, mais elle nous donne la possibilité de rendre la génération des discours en langues des signes plus généralisée.

6.6. Lemmatisation et mise en forme

La lemmatisation consiste à associer un lemme à chaque mot du texte. Si le mot ne peut pas être lemmatisé (nombre, mot étranger, mot inconnu ou sa fonction grammaticale est FW), ce mot sera épelé ("*fingerspelled*") en utilisant le symbole #. Aussi, tous les mots sera transformé en majuscules. Pour notre exemple, on aura :

```
(10) "kate"        → Lemme : kate          → KATE
     "gave"        → Lemme : give          → GIVE
     "chocolate"   → Lemme : chocolate     → CHOCOLATE
     "each"        → Lemme : each          → EACH
     "boy"         → Lemme : boy           → BOY
     "yesterday"   → Lemme : yesterday     → YESTERDAY
```

Le résultat est le suivant :

(11) **"YESTERDAY KATE{t} GIVE CHOCOLATE EACH-BOY"**

6.7. Reconnaissance des entités nommées

La tâche de reconnaissance d'entités nommées s'intéresse à un certain nombre d'unités lexicales particulières, que sont les noms de personnes, les noms d'organisation et les noms de lieux, ensemble auquel sont souvent ajoutés d'autres syntagmes comme les dates, les unités monétaires et les pourcentages. Son objectif est double : il s'agit, d'une part, d'identifier ces unités dans un texte, et, d'autre part, de les catégoriser en fonction de types sémantiques prédéfinis.

Le résultat de ces processus correspond à l'annotation des entités, laquelle se matérialise le plus souvent via des balises encadrant l'entité. Ceci, nous permet de déterminer par exemple les noms des personnes afin de les épeler lors de la transcription. Ou bien, identifier une date ou les quantités [132].

Dans notre exemple, la phrase "*Kate gave chocolate for each boy, yesterday*" contient une entité nommé de type "*personne*" qui est le mot "Kate". Dans ce cas, la transcription de ce signe devient "#KATE{t}", et l'énoncé devient :[50]

(12) **"YESTERDAY #KATE{t} GIVE CHOCOLATE EACH-BOY"**

6.8. Résolution des coréférences

La résolution de coréférence consiste à trouver toutes les expressions qui se réfèrent à la même entité dans un texte comme on a vue dans la section précédente. Pour déterminer les entités coréférentiels dans une phrase, on a utilisé l'outil "*Stanford Coreference Resolution*"[51] [133]. Voici un exemple :

[50] Site web du "Stanford Named Entity Recognition" [http://nlp.stanford.edu/ner/index.shtml]
[51] Site web du "Stanford Coreference Resolution" [http://nlp.stanford.edu/projects/coref.shtml]

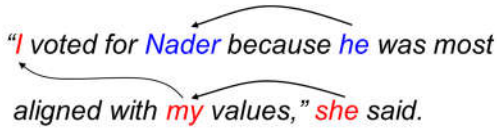


Figure 52. Exemple de coréférence

La résolution de coréférence permet de rajouter des indices entre l'entité "root" déjà extraite, le sujet et les objets. Dans cette exemple, l'entité "*Nader*" sera indexé par *x* et le verbe "vote" sera indexé lui aussi par *x* et par *y* qui fait référence à la première personne au singulier. Ensuite, l'entité "*he*" sera remplacé par une composante non-manuelle faisant référence à *x*. De même pour l'entité "*she*" qui sera référée par *y*. Le taux de précision de la résolution des coréférences est 83.8% et la taux du rappel (*recall*) est 73%.

6.9. Génération de la transcription XML-Gloss

La génération de la transcription XML-Gloss est une tâche simple. Il suffit de parcours les entités résultant des algorithmes de construction d'énoncé en ASL et faire un appel des méthodes de l'API. Prenons le même exemple, l'énoncé final est :

(13) "**YESTERDAY #KATE{t} GIVE CHOCOLATE EACH-BOY**"

Les différentes instructions pour créer le fichier XML-Gloss sont :

```
(21)  g.create_sentence("en","asl","kate gave chocolate for each boy,
      yesterday.");
      g.create_clause("none");
      g.create_token("yesterday");
      g.create_token("kate");
      g. add_property("kate","t");
      g.add_fingerspelling("kate");
      g.create_token("give");
      g.create_token("chocolate");
      g.add_compound("each", "boy", "-");
      enregistre ("sentence.xml");
```

Le fichier XML-Gloss généré est le suivant :

```
(22)  <?xml version="1.0" encoding="UTF-8"?>
      <?xml-stylesheet type='text/xsl' href='form.xsl'?>
      <bloc xmlns:xsi="http://www.w3.org/2001/XMLSchema_instance"
      xmlns:noNamespaceSchemaLocation="schemaXML.xsd">
      <sentence src_lang="en" lang="asl" src_sentence="kate gave chocolate for
            each boy, yesterday">
      <clause type_clause="none">
            <token>YESTERDAY</token>
            <token property="t" fs="yes">KATE</token>
```

```
            <token>GIVE</token>
            <token>CHOCOLATE</token>
            <token compound="-"><token>EACH</token><token>BOY</token></token>
      </clause></sentence>
```

Le rendu final est :

t
YESTERDAY #KATE GIVE CHOCOLATE EACH-BOY

Figure 53. Transcription en gloses de la phrase "*Kate gave chocolate for each boy, yesterday.*"

7. Evaluation

L'évaluation de l'approche proposée est indispensable pour assurer la qualité de la transcription générée par nos algorithmes. Pour évaluer notre système, nous avons commencé par comparer manuellement chaque phrase et sa transcription et la transcription générée. Mais ceci, nous a pris énormément de temps, sachant que le nombre de phrases dépasse $100k$.

L'idée est d'évaluer seulement les règles de transfert entre les deux langues (Anglais et ASL) et ceci nous a réduits énormément le temps et le coût de l'évaluation. Autrement dit, prenons une phrase e en anglais et sa matrice d'adjacence M. On définit une règle de transfert $R(i \Rightarrow j)$ par :

(14) "tmod + nsubj + root + dobj + prep_for-det" ⇒ "T + S + V + O + CO"

Au niveau évaluation, on vérifie seulement la règle de transfert pour toutes les phrases ayant une structure de type $R[i]$. Pour notre système, nous avons évalué 820 règles de transferts extraites des livres d'apprentissage des langues des signes Américaine avec un taux de précision est égale à 82% pour 6720 phrases calculé à partir de la formule suivante :

$$T(précision) = \frac{Nbre(phrases\ valides)}{Nbre(phrases)} \times 100$$

Dans l'annexe D, nous avons présenté les catégories de dépendances grammaticales avec des exemples de transcription générés automatiquement pour chaque relation de dépendance à partir de notre système.

8. Conclusion

L'approche de génération de discours en ASL à partir des règles de dépendances grammaticales présente une solution intéressante pour la traduction automatique d'un texte vers une transcription de la langue des signes. L'architecture globale de ce système de génération a été décrite avec les différentes briques constituant ce système.

Cette approche souffre de quelques limites. La première est que la fonction de traduction n'est pas bijective, autrement dit, seul la génération d'un texte en Anglais vers l'ASL est faisable. La deuxième est que l'évaluation était une tâche complexe puisqu'elle est manuelle et aucune approche automatique n'a été trouvée.

Chapitre 5 : Architecture du système de traduction automatique statistique ASL-Anglais

1. Introduction

Dans le chapitre précédent, nous avons présenté une approche de génération de discours en ASL à partir des règles de dépendances grammaticales, et on est arrivé au faite que cette fonction de traduction est non-bijective, c.à.d. un seul sens est valide.

Sachant que cette thèse vise à traduire un texte ou une transcription dans les deux sens. C'est pour cette raison, pour traduire une transcription en ASL vers un texte en anglais, nous nous s'intéressons aux approches de traduction automatique à partir des statistiques collectés du corpus construit automatiquement en utilisant l'approche décrite précédemment.

Les traducteurs automatiques statistiques se basent essentiellement sur un modèle de traduction. Au cours de ce chapitre, nous allons présenter la démarche de la génération d'une traduction possible en Anglais à partir d'un discours en ASL annoté en XML-Gloss en utilisant les traducteurs automatiques statistiques.

2. Principes de la traduction automatique statistique

Dans cette section, nous abordons les principes fondamentaux de la SMT (*Statistical Machine Translation* en anglais), avant d'introduire une approche basée sur l'alignement et la traduction de segments sous-phrastiques (à base de phrases).

Nous expliquons ensuite le fonctionnement d'un décodeur permettant de générer des hypothèses de traductions, suivi par la description d'une implémentation. Puis, nous présentons différentes méthodes d'évaluation automatiques des hypothèses de traduction produites par les systèmes.

2.1. Principes de la traduction automatique statistique

Brown et al. [134] proposent un modèle probabiliste selon lequel une phrase P dans la langue source a une traduction possible T dans la langue cible selon la probabilité $p(T|P)$.

Cette affirmation peut être interprétée comme la probabilité qu'un traducteur humain produise la phrase cible T, connaissant la phrase source P. Ainsi, ce modèle permet de chercher une phrase $\hat{T}$, qui soit une traduction possible de P en maximisant la probabilité $p(T|P)$, selon les observations effectuées dans un corpus parallèle. La SMT peut donc se définir selon l'équation :

$$\hat{T} = \max_{T} p(T|P) = \max_{T} p(P|T) \cdot P(T) \quad (1)$$

Ce formalisme permet d'isoler deux éléments :

- un modèle de traduction p(T|P), contenant les probabilités inter-langues,
- un modèle de langage p(T), permettant d'évaluer la probabilité d'une séquence de mots.

Cette théorie constitue les fondements de l'approche TAS, dont l'une des premières implémentations a été réalisée dans le cadre du projet Candide chez IBM [135]. Dans les travaux de Brown et al., [136] sont présentés les modèles statistiques les plus populaires, aujourd'hui connus sous le nom de modèles IBM.

Ces modèles permettent l'alignement des mots entre des énoncés source et des énoncés cible, au préalable alignés (voir Figure 54). Il aura fallu attendre une dizaine d'années pour que les premiers systèmes se basant sur une méthode d'alignement autre que celle uniquement basée sur les mots permettent d'atteindre des performances significatives. L'alignement et la traduction de segments sous-phrastiques (en anglais, *Phrase-Based Machine Translation* ou PBMT) [137] [138] est depuis devenue la référence des approches purement statistiques en traduction automatique.

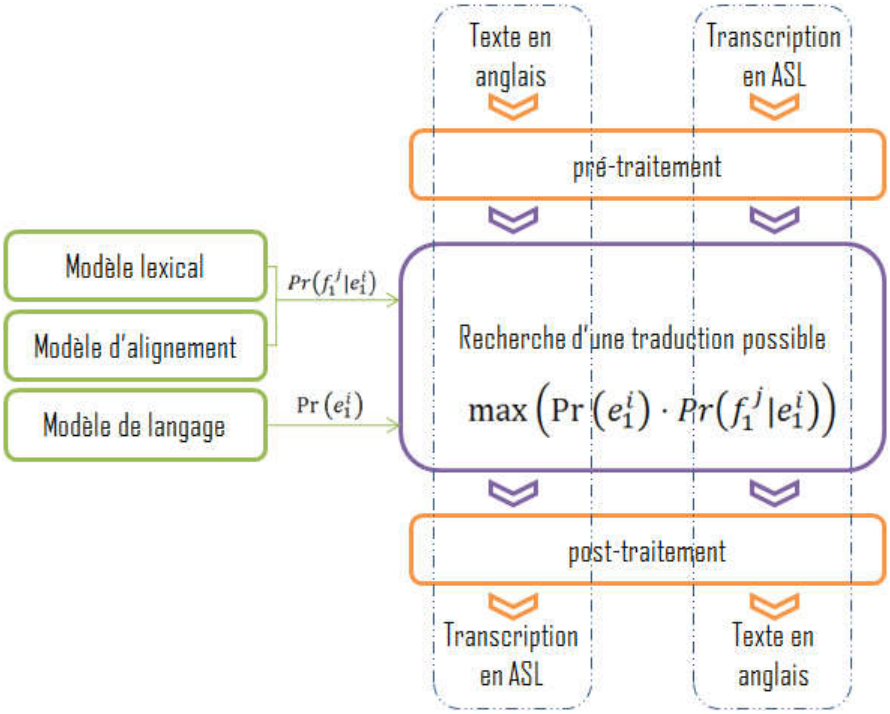


Figure 54. Modèle de traduction à base de statistiques

2.2. Traduction automatique sous-phrastique

L'approche PBMT est basée sur le principe qu'un mot ou un groupe de mots dans une langue peut être traduit par un mot ou un groupe de mots dans une autre langue. Cette première étape d'alignement nécessite l'utilisation d'un corpus bilingue parallèle, dont l'alignement au niveau des phrases est fourni.

Alignement des mots

En partant de l'équation (1), [136] introduisent l'alignement au niveau des mots pour l'estimation des probabilités d'un couple de traductions. La probabilité $p(T)$ étant délivrée par le modèle de langage de la langue cible, ils cherchent donc à estimer la probabilité d'observer une phrase P connaissant une phrase T selon l'équation :

$$p(P|T) = \sum_{a} p(P, a|T) = \sum_{a} \prod_{i=1}^{l} \left(P_i, a_i, l|P_1^{i-1}, a_1^{i-1}, T\right) \quad (2)$$

où l est la longueur de la phrase P, a_i est la position du mot cible aligné avec le mot source P_i. L'alignement des mots est donc obtenu, selon l'algorithme de Viterbi [139], en maximisant la probabilité :

$$a' = \max_{a} p(P, a|T) \quad (3)$$

De cette modélisation découlent les modèles IBM, numérotés de 1 à 5, permettant d'associer aux mots sources et cibles des probabilités d'alignement différentes. Ces estimations concernant le ré-ordonnancement des mots peuvent prendre en compte plusieurs paramètres, liés par exemple à la distribution des probabilités dans l'ordre des mots (équiprobables pour le modèle IBM 1), la longueur des phrases source et cible, la position d'un mot cible aligné avec un mot source (information prise en charge dans le modèle IBM 2), la fertilité relative à chaque mot (implémentée dans les modèles IBM 3, 4 et 5), etc. La fertilité d'un mot dans la langue source peut être calculée selon la moyenne des nombres de mots composant sa traduction dans la langue cible, suivant les observations faites dans un corpus parallèle.

Les travaux de Och et Ney [140] tendent à montrer que le modèle IBM4 permet d'obtenir les meilleurs résultats d'alignements.

Alignement de segments

De manière générale, l'alignement de mots entre deux phrases est bi-directionnel, de la langue source vers la langue cible. Ainsi, deux alignements sont obtenus, l'un issu de l'intersection entre les deux langues, l'autre provenant de l'union entre ces dernières. La Figure 55 permet d'illustrer ce principe, en représentant les mots de deux phrases sur les lignes et les colonnes d'une matrice d'alignement. Les cases noircies présentes les mots (ou ensembles de mots) alignés.

Une matrice est construite pour représenter les alignements du français vers l'anglais, et une autre pour l'anglais vers le français. La mise en commun des alignements présents dans ces deux matrices permet de générer une troisième matrice présentant les alignements issus de l'intersection (cases noircies) ou de l'union (cases grisées) des deux matrices initiales. Un exemple d'alignement de segments sous-phrastiques est présenté par la Figure 56.

Les travaux actuels utilisant l'approche PBMT se basent généralement sur cette méthode d'alignement, mais divergent parfois quant aux étapes permettant de délimiter

les segments dans une langue alignés avec des segments dans une autre langue [141]. Ces différences permettent toutefois d'accroître le nombre de paramètres associés à chaque alignement de segments [142]. Une seule exception est à souligner dans les approches d'alignements, les travaux de [143], où les auteurs tentent d'aligner directement les segments sous-phrastiques sans passer par un alignement au niveau des mots effectués par un modèle IBM.

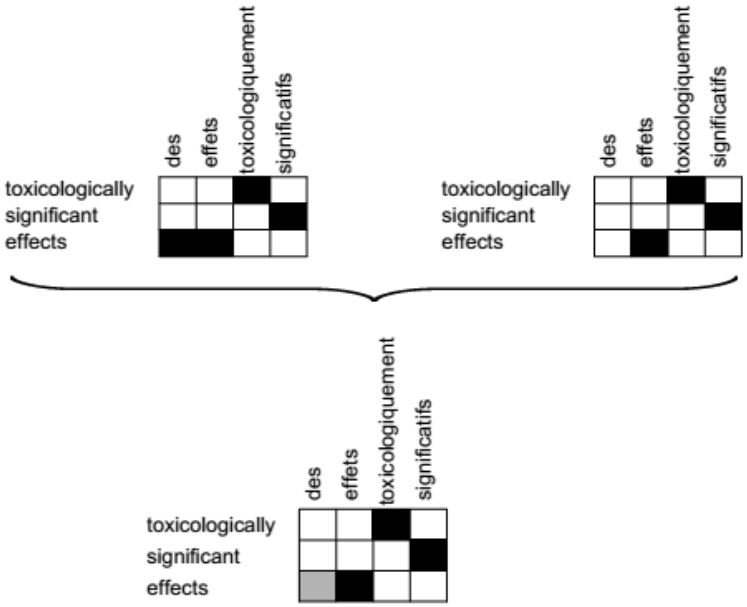


Figure 55. Alignement bi-directionnel au niveau des segments entre deux phrases

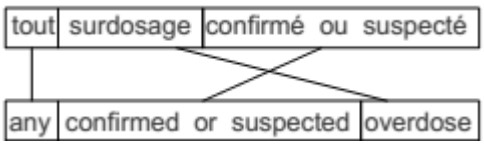


Figure 56. Deux phrases en relation de traduction dont les segments sont alignées

Ainsi, l'estimation des probabilités d'alignement issues de corpus parallèles donne lieu à la construction d'un modèle de traduction, constitué en partie d'une table de traduction. Cette table regroupe les couples de segments sous-phrastiques sources et cibles alignés, auxquels sont associés leurs scores résultant de l'estimation des probabilités d'observation apprises sur le corpus parallèle.

Un extrait d'une table de traduction construite à partir du corpus Europarl est présenté dans le Tableau 7. Les segments sous-phrastiques sont aussi appelés n-grammes, correspondant à des sous-séquences de n éléments construits à partir d'une séquence donnée (une phrase ou un segment de phrase).

Tableau 7. Extrait d'une table de traduction construite sur le corpus parallèle Europarl

Segment source	Segment cible	Score
de la sécurité industrielle et de	industrial safety and	(0,5 ; 1)
de la sécurité industrielle et	industrial safety and	(0,5 ; 1)
de la sécurité industrielle	industrial safety	(1 ; 1)
de la sécurité insuffisante	insufficient security	(0,5 ; 1)
de la sécurité insuffisante à nos	insufficient security at our	(1 ; 1)

Décodeur

La modélisation de l'ensemble des paramètres nécessaires à l'approche PBMT est donc généralement décomposée en plusieurs éléments :

- un modèle de traduction de segments sous-phrastiques, appelé table de traduction,
- un modèle de distorsion (ou de ré-ordonnancement) de segments, permettant de gérer l'alignement des segments,
- un modèle de langage, permettant de s'assurer de la grammaticalité des hypothèses de traduction produites.

L'architecture classique d'un décodeur PBMT est présentée par la figure 2.9.

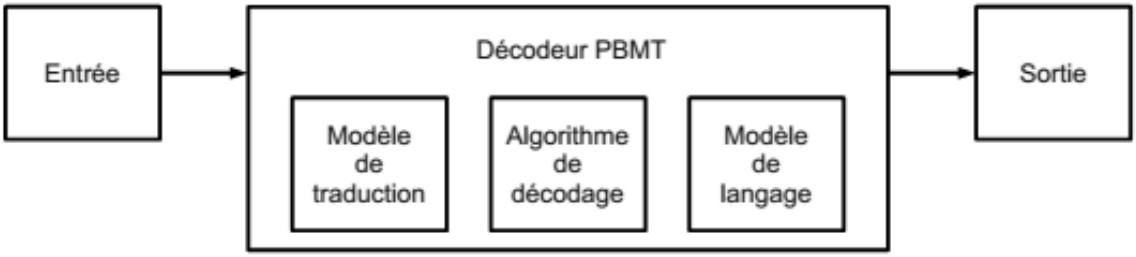


Figure 57. L'architecture d'un décodeur classique

Les scores associés à chaque modèle vont permettre au décodeur de sélectionner les segments cibles les plus probables lorsqu'une phrase source est à traduire. Chaque paramètre peut être défini par un ensemble de fonctions f_n ayant comme paramètre les phrases source P et cible T, ainsi que leurs alignements possibles.

En partant du modèle de traduction, le parcours des paires de segments sous-phrastiques (P_i, T_i) va permettre de générer des hypothèses de traduction. Puis, en s'appuyant sur un modèle de langage, les scores de ces hypothèses sont pondérés par la probabilité d'observer ces segments dans l'ordre proposé par le modèle de distorsion. Finalement, le décodeur peut sélectionner une traduction en suivant l'équation :

$$\begin{aligned} \hat{T} &= \max_{T} p(T|P) \\ &\propto \sum_{n} \lambda_n \log_{10}\left(f_n(a, P, T)\right) \end{aligned} \tag{4}$$

Les paramètres λ_n sont des poids associés à chaque fonction f_n et permettant de minimiser les erreurs de traduction, selon un corpus parallèle. Cette phase de réglage des poids des paramètres du modèle de traduction est appelée développement ou optimisation. Un corpus dédié à cette tâche est donc généralement utilisé et est appelé corpus de développement. Les poids sont optimisés de manière itérative selon les scores obtenus lors de la traduction du corpus parallèle de développement, en se basant sur une traduction de référence et une métrique automatique.

Cette phase d'optimisation est généralement effectuée selon une méthode proposée par [144] sous la dénomination MERT (*Minimum Error-Rate Training* en anglais). Cet

algorithme consiste à affecter à chaque fonction f_n le poids λ_n permettant d'obtenir les traductions les plus proches de la traduction de référence du corpus de développement. Concrètement, les λ_n sont sélectionnés chacun leur tour d'après les résultats de traduction du corpus, évalués par la métrique automatique BLEU. La progression du système est ainsi mesurée après chaque itération permettant de converger vers une solution sous-optimale, permettant d'améliorer le score général sur l'ensemble du corpus d'optimisation.

2.3. Implémentation

Après avoir présenté les principes fondamentaux de la traduction automatique statistique, nous décrivons dans ce qui suit notre architecture adapté au domaine des langues des signes en tenant compte des aspects linguistiques de l'ASL.

3. Architecture du traducteur automatique statistiques

L'architecture globale de notre système de traduction automatique d'un contenu en Anglais vers la langue des signes Américaine ASL ou inversement est décrit dans la Figure 58. Les différentes composantes du système sont emboîtées par des carrés gris. Avant de décrire les différentes composantes du système, nous présentons une légende des différents circuits :

→ Circuit de traduction d'un texte en anglais vers une transcription en ASL et une synthèse 3D.

→ Circuit de traduction d'une interprétation en ASL vers un texte en Anglais.

→ Circuit pour la construction d'un modèle de langage en Anglais et la génération par apprentissage artificiel d'une mémoire de traduction Anglais-ASL.

→ Circuit de raffinement des poids et des mesures générés à partir de la phase d'apprentissage.

→ Circuit d'évaluation manuelle et automatique de traductions générées automatiquement.

→ Circuit des mesures d'évaluation automatique à partir des outils existant.

L'architecture englobe toute les fonctionnalités de notre système de traduction. Les différentes composantes sont :

Entrée du système

- "**Voix**" : C'est le signal émis par l'être humain. Il est sauvegardé dans un fichier de format "*wav*" ou "*mp3*" pour des traitements futurs.
- Module "**Reconnaissance de la parole**" : à partir de la voix, ce module essaye de le transformer en texte. Au cours de cette, on ne s'intéresse pas à la reconnaissance automatique de la parole, et on va utiliser des outils existants comme CMUSphinx[52] qui transforme la parole en un fichier au format VoiceXML[53].

[52] CMUSphinx: outil open-source pour la reconnaissance automatique de la parole http://goo.gl/Cc4b05
[53] VoiceXML: (Voice eXtensible Markup Language ou langage de balisage extensible vocal) est un langage normalisé de programmation d'une application vocale. [http://www.w3.org/Voice/]

- "**Texte Anglais**" : C'est l'entrée de base pour le système de transcription vers la langue des signes Américaine. Dans certain cas, on fait recours à une étape de détection de la langue avant de commencer la phase de la traduction automatique.
- "**Acquisition vidéo**" : Le signal capturé par un WebCam ou tout périphérique vidéo qui contient une interprétation en langue des signes[54].
- Module "**Reconnaissance des gestes**" : Ce module permet d'extraire les gloses à partir des séquences d'images acquises par le périphérique d'acquisition vidéo 2D ou 3D.
- Module "**Génération XML-Gloss**" : A partir du module précédent, ce module transforme les gloses en une transcription XML-Gloss pour la traduction automatique.
- "**Texte ASL**" : C'est l'entrée principale du système de traduction ASL-Anglais.

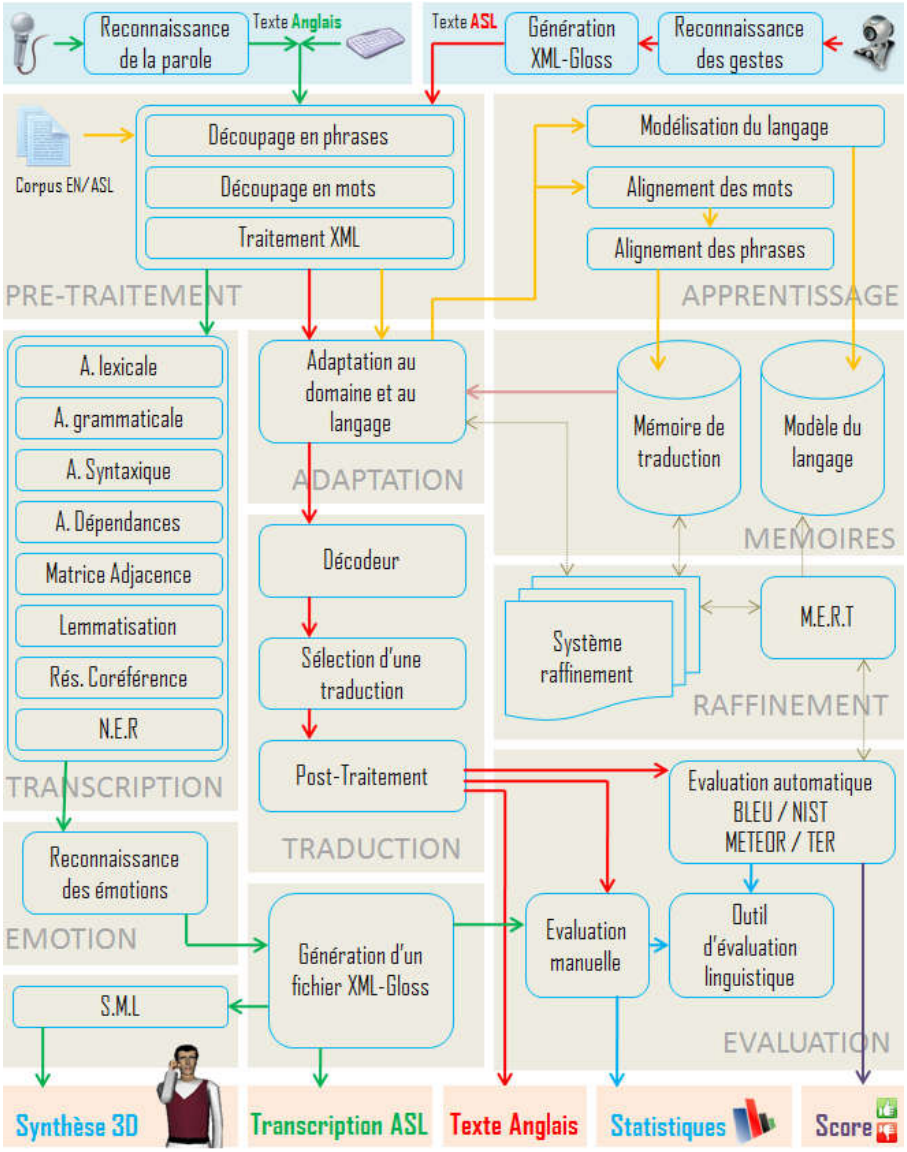


Figure 58. Architecture globale du système de traduction automatique en deux sens

[54] L'acquisition vidéo peut être capturée à partir des périphériques vidéo 2D ou 3D comme le Kinect qui capte aussi la profondeur à partir des capteurs infrarouges. La profondeur est utile surtout pour éliminer la superposition de la main devant le visage du signeur.

Prétraitement

- "**Corpus EN/ASL**" : C'est une collection de texte en Anglais et en ASL transcrit en XML-Gloss. Chaque langue est stockée dans un fichier à part.
- Module "**Découpage en phrases**" : C'est un algorithme qui découpe un texte volumineux en phrases. Cet algorithme est basé sur approche statistique et dont la base de connaissances est extraite à partir de corpus "*Wall Street Journal news*" [145] combiné avec "*the Brown Corpus*"[55]. Le taux d'erreur est 0.25% [146].
- Module "**Découpage en mots**" : C'est un algorithme qui permet d'extraire à partir d'une phrase l'ensemble des jetons qui la compose en utilisant les expressions régulières ou les espaces.
- Module "**Traitement XML**" : C'est une étape de nettoyage des balises XML et HTML qu'on les rencontres dans une phrase.

Apprentissage

- Module "**Alignement des phrases**" : C'est la première étape pour la création d'une mémoire de traduction, elle consiste à coupler deux phrases, une extraite à partir du corpus Anglais et l'autre à partir d'un corpus en ASL.
- Module "**Alignement des mots**" : A partir de deux phrases, l'algorithme d'alignement de mots permet de coupler pour chaque mot de la langue source sa traduction de la langue cible automatiquement.
- Module "**Modélisation du langage**":

Mémoires

- "**Modèle du langage**" : C'est la composante du système de traduction qui est en charge d'introduire les contraintes imposées par la syntaxe de la langue. Les modèles de langage utilisés habituellement sur de grands vocabulaires utilisent une approche statistique de type n-gram (2 ou 3-gram) permettant de prendre en compte des contraintes locales d'accord [147]. Ainsi, avec un modèle 3-gram, la probabilité d'un chemin W contenant n mots m est : $P(W) = \prod_{i=1}^{n} P(m_i|m_{i-2}, m_{i-1})$.
- "**Mémoire de traduction**" : l'estimation des probabilités d'alignement issues de corpus parallèles donne lieu à la construction d'un modèle de traduction, constitué en partie d'une table de traduction. Cette table regroupe les couples de segments sous-phrastiques sources et cibles alignés, auxquels sont associés leurs scores résultant de l'estimation des probabilités d'observation apprises sur le corpus parallèle.

Raffinement

- Module "**Système de raffinement**" : Le système permet d'améliorer la qualité de la traduction à partir d'un ensemble de phrases couplées avec leurs traductions manuelles. Il existe deux approches de raffinement : MERT et MIRA "*Margin Infused Relaxed Algorithm*" [148]. Dans notre système, on utilisera MERT.
- Module "**MERT**" : Le "*Minimum Error Rate Training*" [144] est un algorithme supervisé qui détermine les poids optimaux pour une traduction efficace.

[55] "The Brown Corpus" contient à peu près 1.1 million de mots de différentes catégories de la langue Anglaise. Tous les mots sont étiquetés par des étiquettes grammaticales dont le nombre est 192 [http://khnt.aksis.uib.no/icame/manuals/brown/].

Adaptation

- Module "**Adaptation au domaine et au langage**" : C'est une étape de normalisation des contenus à utiliser lors de la phase de traduction. Les phrases doivent être alignées ainsi que les mots.

Transcription

- Module "**Transcription**" : l'approche de transcription d'un texte en Anglais vers l'ASL a été décrite dans le chapitre précédent.

Traduction

- Module "**Décodeur**" : Le décodeur permet de trouver l'ensemble des traductions générées automatiquement à partir d'une phrase donnée en entrée. L'approche de décodage sera décrite dans le chapitre suivant.
- Module "**Sélection d'une traduction**" : A partir du module "décodeur", cet algorithme permet de choisir la meilleure traduction possible.
- Module "**Post-Traitement**" : La post-édition de traductions fait référence à des modifications a posteriori des hypothèses produites par un système. Une étape d'édition peut impliquer des modifications et des corrections d'erreurs. Ainsi, l'édition d'une traduction automatique T' revient à générer la phrase T'', comportant les modifications grammaticales et syntaxiques nécessaires à l'amélioration de la lisibilité de T', tout en gardant le sens de la phrase source S. Cette étape implique un coût pouvant dépasser celui induit par la traduction humaine. Savoir si la post-édition de traductions automatiques est moins coûteuse que la traduction humaine dépend de la qualité des hypothèses produites par les systèmes.

Génération XML-Gloss

- Module "**Génération d'un fichier XML-Gloss**" : l'approche de transcription d'un texte en Anglais vers l'ASL a été décrite dans le chapitre 3.

Emotion

- Module "**Reconnaissance des émotions**" : Cet algorithme permet de reconnaître automatique l'émotion (à partir d'un ensemble contenant six émotions : joie, tristesse, dégoût, colère, surprise et peur selon le modèle de Paul Ekman [122]) à partir du texte en anglais donné en entrée du système. Cet algorithme sera décrit dans un chapitre à part.

Génération SML

- Module "**Génération SML**" : C'est l'avant-dernière étape qui consiste à convertir le fichier XML-Gloss en un format SML "*Sign Modeling Language*" [103] afin d'animer l'avatar et synthétiser en trois dimensions la phrase en Anglais donnée en entrée du système.

Evaluation

- Module "**Evaluation automatique**": L'évaluation automatique (évaluation objective) a besoin d'une ou plusieurs traductions qui seront considérées comme des références pour la traduction d'une phrase source. Les mesures automatiques ont pour but de déterminer le degré de ressemblance entre la traduction proposée par le système et la (les) traduction(s) de référence.
- Module "**Evaluation manuelle**" : Lors d'une évaluation humaine de la traduction automatique, on demande à plusieurs participants d'évaluer chaque traduction en

fonction de critère précis. Les critères de qualité peuvent être multiples et inclure, par exemple, des critères de correction grammaticale et de fidélité au sens du texte. Ces critères de qualité constituent la vraie mesure de la qualité du système, mais requièrent une coûteuse intervention humaine. Par ailleurs, toute évaluation subjective souffre des problèmes de non reproductibilité et de variabilité inter-annotateur.

- Module "**Outil d'évaluation linguistique**" : A partir des évaluations manuelles par des participants, on raffine encore une fois notre de système de traduction. Ceci nous permet de modifier les poids des modèles de langage ainsi que la mémoire de traduction afin d'obtenir des résultats meilleurs.

Sortie du système

- "**Synthèse 3D**" : C'est la sortie du système, un animation interprété par un agent conversationnel du projet WebSign [103].
- "**Transcription ASL**" : C'est le fichier XML-Gloss de la pharse traduite en ASL.
- "**Texte Anglais**" : C'est la traduction de la phrase ASL donnée en entrée du système.
- "**Statistiques**" : C'est ensemble de chiffres concernant le nombre de traductions, la taille des corpus parallèles, etc.
- "**Score**" : Le score est une collection de mesures obtenues par des outils d'évaluation automatique, par exemple, le score BLEU [74] (*Bilingual Evaluation Understudy* en anglais) ou autres qui seront décrit dans les sections suivantes.

4. Discussions, contributions et démarche

Jusqu'à maintenant, nous avons détaillé l'architecture globale de notre système de transcription et de traduction automatique d'un texte en anglais vers la langue des signes Américaines ou inversement. L'architecture est composé d'un ensemble de composants dont certains a été déjà développé et qu'on l'utilisera directement. Et d'autres ont été décrit et implémenté dans les deux chapitres précédents (Système de transcription Anglais-ASL et le système d'annotation en XML-Gloss).

Dans ce qui suit dans cette thèse, on s'intéresse à mettre en place un traducteur automatique statistique qui prend en entrée une transcription en Anglais et génère sa traduction en ASL. Sachant que cette approche est bijective, alors le sens inverse est implémenté implicitement. L'approche statistique nécessite des ressources linguistiques importantes pour aboutir des bonnes résultats, aussi, dans la littérature, on ne trouve pas des implémentations réelles de ce type de traducteurs pour les langues des signes.

Notre travail se situe donc à la construction d'une ressource linguistique Anglais-ASL en premier lieu, et à décrire en détails l'implémentation de l'approche statistique pour la traduction automatique d'une transcription ASL vers l'anglais et inversement.

5. Génération d'un corpus parallèle Anglais-ASL

Après avoir cité les problèmes rencontrés pour la création des corpus en langues des signes, dans les sections précédentes, à cause de leurs coûts. La disponibilité des corpus parallèles pose donc des problèmes, que ce soit pour la couverture du vocabulaire ou pour les structures syntaxiques des phrases à traduire.

De plus, les textes de domaines spécialisés sont en général plus difficiles à traduire automatiquement, de par le manque de corpus et du nombre important de mots hors vocabulaire. Les systèmes de traduction automatique fournissent les meilleures traductions lorsqu'il existe des données d'apprentissage du domaine de taille suffisante. Une des méthodes permettant de pallier le manque de données spécialisées est l'édition a posteriori de traductions issues de systèmes automatiques. Habituellement effectuée par des humains, cette étape de révision permet de s'assurer de la qualité des traductions et de les adapter, si besoin, aux domaines de spécialités [149].

De ce fait, la qualité des traductions automatiques agit directement sur l'effort de post-édition. Les chercheurs travaillant en traduction automatique ont donc rapidement envisagé l'automatisation de la post-édition, en se basant sur des approches statistiques similaires à la traduction automatique statistique, et permettant de réécrire du texte dans un cadre monolingue [150].

Dans ce cadre, nous avons fait recours à une approche de génération d'un corpus artificiel parallèle Anglais-ASL. L'idée vient de générer à partir d'un ensemble de phrases en Anglais, leurs transcriptions respectives en langue des signes Américaine annotés en XML-Gloss [151] en utilisant l'approche citée dans le chapitre précédent. Ensuite, on procède à une étape de validation manuelle des règles de transferts par des utilisateurs du système qui sont soit des experts du domaine linguistique de la langue des signes, soit des connaisseurs de cette langue. Les phrases en Anglais ont été extraites à partir du corpus Gutenberg[56] qui contient plus que quarante-deux-milles livres électronique et plus que cent-milles livres électroniques collectés à partir d'autres sites partenaires. Toutes ces ressources sont libres en droit d'accès. L'extraction de ces ressources a été réalisée en utilisant un outil qui parcours pour chaque livre et qui sauvegarde dans une base de données les phrases et les mots.

Cette phase a été effectuée sur quatre étapes. Les statistiques sont illustrées dans le Tableau 8. On note que les données extraites à partir de cette ressource sont bruitées et peuvent contenir des phrases erronées ou des textes incompréhensibles, etc.

Tableau 8. Statistiques des données extraites à partir du corpus Gutenberg

	Nombre de mots	**Nombre de phrases**
Corpus Anglais	1 595 579 658	79 611 533

Après avoir sauvegardé toutes ces ressources dans une base de données, nous avons procédés à un nettoyage des phrases qui sont invalides, qui sont :

i. Phrases non-anglaises
ii. Phrases dont le nombre de mots est strictement inférieure à 2.
iii. Phrases dont le nombre de mots est strictement supérieure à 10.
iv. Phrases dont la structure syntaxique est invalide.
v. Phrases qui contiennent une ambigüité dans le sens.
vi. Phrases complexes (subordonnés).

[56] Site Web du projet "Gutenberg," 2012 [http://www.gutenberg.org/]

La nouvelle taille du corpus de la langue Anglais est devenue 119860 phrases. Pour les trois premières étapes (i, ii et iii), on a utilisé un algorithme écrit en C++ ou Perl pour faire le nettoyage automatique. Pour le reste des étapes (iv, v et vi), on construit l'arbre syntaxique de la phrase, ensuite, on génère automatiquement sa traduction lexicale en ASL à partir des règles de transfert à base des étiquettes grammaticales. Si la structure finale de la phrase en ASL contient un sujet, un verbe et un objet, alors, on la garde [152]. Le nombre de phrases et de mots du corpus est illustré dans le tableau suivant.

Tableau 9. Statistiques du corpus après la phase de nettoyage

	Anglais		**A.S.L**	
	# Phrases	# Jetons	# Phrases	# Jetons
Corpus pour apprentissage	79907	669045	79907	370030
Corpus pour raffinement	26636	223522	26636	123816
Corpus pour évaluation	13317	111658	13317	61773

Dans la Figure 59 à droite, on remarque la proportionnalité du nombre de mots générés en ASL par rapport au nombre de mots en Anglais. Ceci pourra utile comme hypothèse lors la traduction automatique. Le corpus Anglais-ASL [153] est disponible gratuitement sous licence libre.

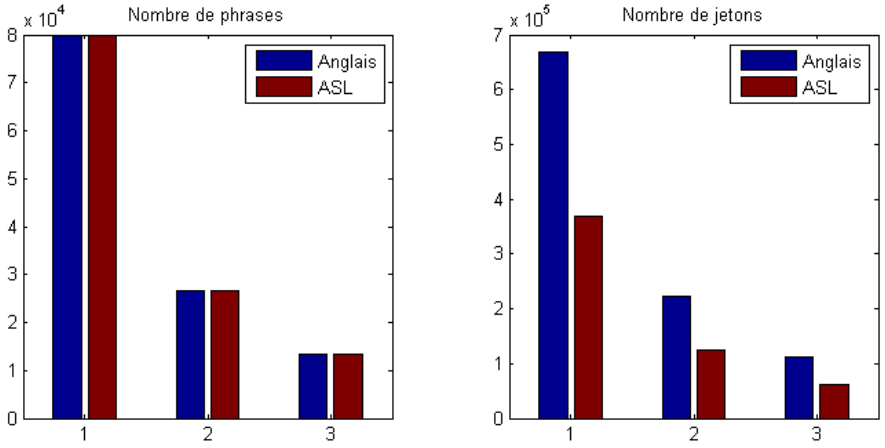


Figure 59. Aperçu de la taille du corpus Anglais-ASL pour apprentissage, raffinement et évaluation (gauche) Nombre de phrases (droite) Nombre de jetons

6. Implémentation et expérimentation

Après avoir construit le corpus parallèle Anglais-ASL, on procède à l'implémentation de notre traducteur automatique statistique basé sur une approche sous-phrastique qui sera décrite dans le chapitre 7. Cette approche statistique permet actuellement d'obtenir de bonnes performances et les mesures d'évaluations ont montré une prédominance de ce système. Aussi, nous avons choisi d'adopter ces méthodes dans nos travaux, proposées dans la boîte à outils de traduction automatique Moses[57] [154].

[57] Moses: Implémentation de l'approche statistique de la traduction automatique. C'est le successeur du traducteur Pharaoh [http://www.statmt.org/moses/?n=Moses.Overview].

Moses propose l'ensemble des outils nécessaires à la construction d'un modèle de traduction. Un décodeur permet aussi d'utiliser ces outils afin de produire des hypothèses de traduction d'un texte initial.

Plusieurs étapes sont nécessaires à la construction d'un système de traduction avec Moses et font appel aux différents outils mis à disposition. Tout d'abord, il est nécessaire d'effectuer un alignement au niveau des mots à partir du corpus parallèle. Puis, l'ensemble des alignements obtenus forment une table servant de base à la construction de la table ou mémoire de traduction [155] qui sera décrite dans le chapitre suivant. Cette dernière est élaborée selon les scores d'alignement obtenus par les segments sous-phrastiques présents dans le corpus parallèle. Un modèle de ré-ordonnancement est ensuite construit, contant les informations des positions dans les phrases des mots traduits par rapport aux mots traduits précédemment. Afin d'estimer les probabilités d'alignement de mots, Moses encapsule l'outil GIZA++[58] [140], qui implémente les algorithmes des modèles IBM 1-5 [156]. Nous utilisons aussi une version parallélisable de ce programme, MGIZA[59] [157] afin de réduire le temps lors de la phase d'alignement.

Les modèles de langage peuvent, quant à eux, être construits à l'aide d'autres outils, comme celui proposé par le SRI-LM[60] [158] , ou encore par l'IRST-LM[61] [159] .

7. Evaluation

Évaluer la qualité des traductions produites par un système automatique peut se faire manuellement, par des humains. On parle alors d'évaluation subjective. Lorsque l'on dispose de références de traduction, l'évaluation automatique (objective) est alors possible. L'évaluation humaine étant généralement plus coûteuse, elle permet cependant d'avoir une meilleure estimation de la qualité des traductions. En effet, elle permet d'intégrer plusieurs critères comme la lisibilité, la justesse grammaticale et syntaxique, etc. [160].

Récemment, les coûts liés à l'évaluation manuelle de traductions ont pu être réduits par l'utilisation de plates-formes en ligne regroupant des personnes pouvant effectuer des tâches parcellisées. Un grand nombre d'évaluateurs sont donc disponibles simultanément et ce qui permet une répartition de l'effort d'évaluation des traductions. Des sous-ensembles de données à évaluer sont formés parmi les hypothèses de traduction produites par un système, puis soumis aux participants à l'évaluation. L'adéquation entre les utilisateurs s'obtient par le croisement des résultats obtenus. Ce protocole d'évaluation est présenté de manière plus détaillé dans les travaux de Callison-Burch [161]. Nous pouvons cependant relever des limites à ces méthodes d'évaluation manuelle. En effet, les plateformes d'évaluation à bas coût risquent d'être vues comme une norme dans le processus de création de ressources linguistiques. De ce fait, faire intervenir des professionnels pour évaluer des traductions automatiques pourrait devenir trop onéreux

[58] GIZA++: Outil d'alignement basé sur les modèles d'IBM pour le traducteur automatique statistique Moses, écrit en C++ [https://code.google.com/p/giza-pp/]
[59] MGIZA: Version parallèle et multi-thread de l'outil d'alignement GIZA++ écrit en C++ [http://www.kyloo.net/software/doku.php/mgiza:overview]
[60] SRI-LM: générateur des modèles de langage pour SMT [www.speech.sri.com/projects/srilm/]
[61] IRST-LM: générateur des modèles de langage pour SMT [http://hlt.fbk.eu/en/irstlm]

si les budgets dédiés à cette tâche se calquent sur les coûts réduits des plateformes en ligne.

Pour les traducteurs automatiques statistiques, plusieurs métriques ont été développées. Nous décrivons quelque uns dans ce qui suit.

7.1. Score BLEU

BLEU (*BiLangual Evaluation Understudy* en anglais) est algorithme pour l'évaluation de la qualité d'un texte traduit par un traducteur automatique. La qualité est déterminée à partir de la traduction générée automatiquement et sa référence traduite par un être humain [74]. Ce score ne tient pas compte de l'aspect grammatical. Le score est un nombre compris entre 0 et 1 ou bien en pourcentage[62]. Lorsque le score est proche de 1, on dit que la traduction est plus similaire que la traduction humaine. Si le score est égal à 1, alors la traduction automatique est identique à celle de la traduction manuelle.

BLEU est basé sur une formule inspiré de la formule de précision et du rappel. En effet, la précision est définie comme suit :

$$Précision = \frac{Nbre\ de\ candidats\ corrects}{Nbre\ de\ candidats\ références} \quad (5)$$

Pour le score BLEU, la formule devient :

$$Précision = \frac{Nbre\ de\ mots\ candidates\ m}{Nbre\ de\ mots\ candidates\ de\ la\ référence\ W_t} \quad (6)$$

Ici, la précision tient compte seulement du nombre des mots, sans tenir compte de la relation entre les mots. Donc, les modèles sont intégrés dans le calcul de la précision, on a donc :

$$Précision = \frac{\sum_{C\in\{candidats\}} \sum_{n-gram\in C} Nombre(n-gram)}{\sum_{C'\in\{candidats\}} \sum_{n-gram'\in C'} Nombre(n-gram')} \quad (7)$$

Ceci, nous permet d'avoir une corrélation plus efficace entre la traduction automatique et la traduction manuelle référence [162].

7.2. Score METEOR

METEOR (*Metric for Evaluation of Translation with Explicit ORdering* en anglais) [163] est métrique pour l'évaluation des traductions issues d'un traducteur automatique statistique. Cette métrique est basée sur la moyenne harmonique[63] de la précision et le rappel des modèles uni-gram.

L'algorithme METEOR démarre à partir d'un ensemble de phrases comme le score BLEU. Au départ, l'algorithme créée un alignement entre la phrase traduite automatiquement t et sa référence traduite manuellement r. L'alignement résultant est

[62] Dans les travaux de recherche en traduction automatique statistique, on utilise toujours les scores BLEU en pourcentage. Par exemple, si le score BLEU est 0.325, alors on dit que le score BLEU est 32.5.

[63] Moyenne harmonique H: C'est l'inverse de la moyenne arithmétique de l'inverse des termes. Elle définit comme suit : $H = \frac{n}{\frac{1}{a_1}+\frac{1}{a_2}+\cdots+\frac{1}{a_n}}$ tel que $\{a_1, a_2, \cdots, a_n\} \in \mathbb{R}^*$.

un ensemble de mappages entre les unités uni-gram. A partir de ces deux alignements on détermine la précision uni-gram P et le rappel R comme suit :

$$P = \frac{m}{w_t} ; R = \frac{m}{w_r} \quad (8)$$

tel que m est le nombre des uni-gram de la phrase t trouvé dans la phrase r, w_t est le nombre des uni-gram de la phrase t et w_r est le nombre des uni-gram de la phrase r.

A partir des équations (8), et en utilisant la formule de la moyenne harmonique, on définit une nouvelle mesure F_{mean}:

$$F_{mean} = \frac{10 \cdot P \cdot R}{R + 9 \cdot P} \quad (9)$$

Afin d'assurer une corrélation plus efficace, l'algorithme METEOR définit une pénalité d'alignement p qui se détermine à partir des groupements grammaticaux bi-gram:

$$p = 0.5 \cdot \left(\frac{c}{U_m}\right)^3 \quad (10)$$

Tel que c est le nombre des groupements grammaticales (chunks en anglais) et U_m est le nombre des uni-gram qui n'ont pas été aligné dans la phrase r. On arrive maintenant à définir le score METEOR M comme suit :

$$M = F_{mean} \cdot (1 - p) \quad (11)$$

7.3. Score F-Mesure

F-mesure ou F_1-score est une mesure de précision calculé à partir de l'équation (8) de la section précédente. Le score F_1 est interprété comme étant le poids moyen de la précision et le rappel. Il est défini comme suit :

$$F_1 = 2 \cdot \frac{precision.rappel}{precision + rappel} \quad (12)$$

D'une manière générale, le score F-mesure est calculé comme suit :

$$F_\beta = (1 + \beta^2) \cdot \frac{precision.rappel}{(\beta^2 \cdot precision) + rappel} \; ; \; \beta \in \mathbb{R}^* \quad (13)$$

7.4. Score NIST

NIST (abréviation du *US National Institute of Standards and Technology*[64]) [164] est une méthode d'évaluation des traductions automatiques. Elle est basée sur le score BLEU avec quelques modifications. Sachant que le score BLEU tient compte seulement des unités n-gram, NIST rajoute une variable poids extraites à partir des modèles de langages n-gram. C.-à-d., la chaîne bi-gram "*on the*" a un poids plus important que la

[64] Depuis 2001, l'institut NIST publie des résultats d'évaluation officiels de différents traducteurs automatiques existent. Ces résultats sont utiles pour les recherches dans le domaine de traduction automatique pour faire des comparaisons. [http://www.itl.nist.gov/iad/mig/tests/mt/2009/ResultsRelease/]

chaîne "*interesting calculations*", puisque cette dernière on la trouve rarement dans les corpus.

8. Conclusion et discussions

Au cours de ce chapitre, nous avons présenté l'architecture globale de notre système de traduction et transcription vers et à partir de la langue des signes Américaines, premier lieu. Ensuite, nous avons décrit notre démarche pour la construction d'un corpus parallèle artificiel parallèle Anglais-ASL qui sera la mémoire de traduction pour notre traducteur automatique statistique. Puis, nous avons introduit l'outil d'expérimentation afin d'implémenter concrètement notre approche. Les différents étapes de génération de la mémoire de traduction et de traduction seront décrites et détaillés dans les deux chapitres qui suivent. Enfin, nous avons présenté les mesures d'évaluation de la qualité de traduction qu'on les utilisera lors de l'expérimentation.

Chapitre 6 : Construction d'une mémoire de traductions lexicales "Anglais-ASL"

1. Introduction

Les traducteurs automatiques statistiques se basent essentiellement sur un corpus parallèle (une langue source et langue cible). A partir de cette ressource linguistique, le système de traduction construit sa mémoire de traduction suite à une étape d'apprentissage. Autrement dit, il essaye d'aligner automatiquement chaque mot de la langue source avec sa traduction dans la langue cible à partir du corpus. Généralement, chaque mot peut avoir plusieurs traductions, donc, chaque traduction aura un poids qui sera déterminé lors des itérations.

Au cours de chapitre, nous présentons notre démarche pour la construction d'une mémoire de traduction Anglais-ASL à partir du corpus ASL-PC12 décrit dans le chapitre précédent [153] (voir Tableau 9) inspiré des modèles IBM 1-5 [135]. Cette mémoire sera utilisée dans la phase de décodage. Nous commençons par la génération d'une mémoire de traduction lexical (mot-à-mot). Ensuite, on passe à la construction d'une mémoire sous-phrastique. Et on finit par une évaluation globale de l'approche.

2. Génération d'une mémoire de traduction lexicale

2.1. Définition

D'une façon générale, le terme mémoire de traduction désigne un type de dispositif informatique à l'intention des traducteurs humains, visant à promouvoir la réutilisation des traductions existantes. D'après Macklovitch et Russell [165], deux définitions de ce terme sont admises, que nous appelons informellement définition pratique et définition théorique [166]. La définition pratique se lit comme suit : *"...une mémoire de traduction [...] est un type d'outil d'aide à la traduction qui maintient une base de données de paires de phrases en langues source et cible, et qui extrait automatiquement la traduction des phrases d'un nouveau texte qui apparaissent dans cette base de données.*".

La définition théorique est, quant à elle, beaucoup plus générale : "*...une archive de traductions existantes, conçue de façon à faciliter la réutilisation des traductions.*".

La définition théorique est certes plus attrayante que la première. Malgré tout, il faut comprendre que celle-ci (la définition pratique) découle directement d'une réalité indéniable : elle décrit en fait beaucoup plus adéquatement la nature des traducteurs automatiques statistiques.

A partir de ces définitions, nous entamons la partie de la génération d'une mémoire de traduction lexicale à base de mots, en premier lieu. En second lieu, la génération à partir des phrases sera détaillée.

2.2. Traduction lexicale

Si on prend un mot source en ASL '**REVIEW**', on trouve plusieurs traductions possibles en anglais : 'review', 'reviews', 'reviewed', etc. La plupart des mots possèdent plusieurs traductions possibles. La notion des traducteurs automatiques à base de statistiques implique l'utilisation des statistiques extraites à partir des mots et des textes. Quel type de statistiques on a besoin pour traduire le mot '**REVIEW**' en ASL ou vice versa ? Si on suppose qu'on a un grand corpus ou collection de données en langue des signes Américaine ASL couplés avec une collection de données en anglais, où pour chaque phrase en Anglais correspond à sa traduction en ASL, on peut compter combien de fois a été traduite le mot '**REVIEW**' en 'review', 'reviews', etc. Le Tableau 10[65] montre les résultats possibles de la traduction du mot '**REVIEW**' sachant que le nombre d'occurrence du mot initial en anglais est 5 fois dans le corpus ASL.

Tableau 10. Nombre hypothétique des différentes traductions possibles du mot 'REVIEW'

Traduction de **'REVIEW'**	Nombre d'occurrences	Probabilité d'alignement
reviewed	1	1.0000000
reviews	3	0.6666667
review	9	0.4000000
for	7216	0.0001840
Ø[66]	-	0.0001261
of	10854	0.0001064
the	32608	0.0000326

Ici, on remarque que le mot 'reviewed' a été utilisé qu'une seule fois et il a été aligné avec **'REVIEW'** ce qui donne une probabilité de traduction égale 1, autrement dit, lors d'une nouvelle requête de traduction de ce terme, il sera traduit en 'reviewed'. Pour les autres, lorsqu'il le nombre d'occurrences augmente, la probabilité d'alignement diminue.

2.3. Estimation de la distribution probabiliste

Si on veut estimer ou bien calculer la distribution probabiliste de la traduction lexicale à partir des statistiques collectés des corpus parallèles Anglais-ASL, on a besoin de répondre à la question suivante : Quel la traduction la plus probable du mot '**REVIEW**' en anglais ou inversement ? Ceci nous mènent à trouver la fonction suivante :

[65] Les valeurs sont calculées à partir d'une expérimentation réelle appliquée sur un corpus Anglais-ASL contenant 79907 phrases.
[66] Le caractère Ø désigne que le mot 'REVIEW' n'a pas été traduit et a été remplacé par un jeton null.

$$P_f: e \rightarrow P_f(e) \quad (1)$$

Etant donné un mot f (c'est "**REVIEW**' dans notre exemple), on aura la probabilité pour chaque traduction possible e. Cette fonction retourne une grande valeur si e est une traduction possible et courante, sinon une valeur petite où e est rarement utilisé. Dans le cas de zéro, on a pas de traduction possible pour l'entrée f. La fonction donnée dans l'équation précédente, doit prouver deux propriétés :

$$\sum_e P_f(e) = 1 \text{ et } \forall e: 0 \leq P_f(e) \leq 1 \quad (2)$$

Pour calculer la distribution probabiliste d'un mot f, dans notre exemple le mot '**REVIEW**', il suffit de déterminer le nombre d'occurrences de ce mot f dans le corpus en ASL et calculer l'occurrence de ces traductions possibles dans le corpus en anglais. Puis, on calcule le ratio pour chaque sortie e. On aura donc :

$$P_f(e) = \begin{cases} 1.0000000 \text{ si e='reviewed'} \\ 0.6666667 \text{ si e='reviews'} \\ 0.4000000 \text{ si e='review'} \\ 0.0001840 \text{ si e='for'} \\ 0.0001261 \text{ si e=}\emptyset \\ 0.0001064 \text{ si e='of'} \\ 0.0000326 \text{ si e='the'} \end{cases} \quad (3)$$

Dans ce cas, le choix de la meilleure traduction possible est celui qui a la probabilité la plus importante. Le type d'estimation employé est appelé le maximum de vraisemblance[67] (*'Maximum Likelihood Estimation'* en anglais) qui maximise la ressemblance entre les données.

2.4. Alignement

A partir des distributions probabilistes déjà calculés précédemment pour les traductions lexicales, on peut mettre en place le premier traducteur automatique statistique, qui utilise seulement les traductions lexicales. Dans le Tableau 11, on montre les distributions probabilistes de trois jetons en ASL et leurs alignements en anglais. On note la probabilité de traduire un mot f en un mot en ASL e avec une fonction de probabilité conditionnelle t(e|f). On appelle le tableau résultant par T-tables.

Tableau 11. Les traductions lexicales pour 3 jetons composant la phrase 'YOU BLUE CAR'

YOU		BLUE		CAR	
e	**t(e|f)**	**e**	**t(e|f)**	**e**	**t(e|f)**
you	0.7969932	blue	0.6966292	car	0.8256757
your	0.7936965	∅	0.0002522	cars	0.7872340
∅	0.192914	the	0.0000652	∅	0.0001261

[67] Maximum de vraisemblance : l'estimation du maximum de vraisemblance est une méthode statistique courante utilisée pour inférer les paramètres de la distribution de probabilité d'un échantillon donné.

Dans cet exemple, la traduction mot-à-mot de la phrase '**YOU BLUE CAR**' en anglais est la suivante : 'your blue car'[68]. L'alignement entre les mots sources et les mots cibles est illustré dans la Figure 60. L'alignement entre une phrase source f en une phrase cible e est formalisé par la fonction d'alignement a. Cette fonction permet de mapper les positions des mots de f avec les positions des mots de e. Dans l'exemple précédent, l'alignement a devient : $a: \{1 \rightarrow 1; 2 \rightarrow 2; 3 \rightarrow 3\}$.

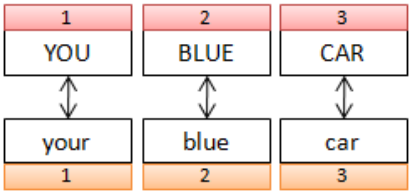


Figure 60. Diagramme d'alignement de la phrase 'YOU BLUE CAR' en ASL

Ceci est un simple alignement entre deux phrases ayant le même nombre de mots des deux sens et pour chaque mot de f correspond un mot de même position dans e. Généralement, les mots qui se correspondent n'ont pas les mêmes positions allant d'une phrase source vers une phrase cible comme cet exemple :

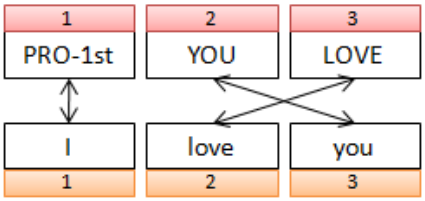


Figure 61. Diagramme d'alignement de la phrase 'i love you' et la phrase 'PRO-1ST YOU LOVE'

La fonction d'alignement a aura la valeur suivante : $a: \{1 \longrightarrow 1; 2 \longrightarrow 3; 3 \longrightarrow 2\}$. A part le changement de l'ordre des mots entre deux phrases source et cible, le nombre de mots varie aussi. Voici un exemple :

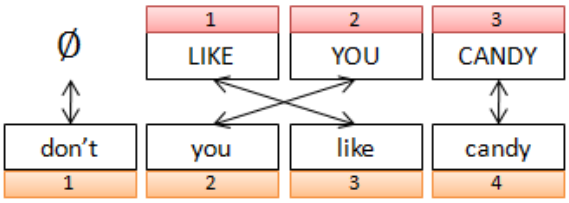


Figure 62. Diagramme d'alignement de la phrase 'D'ont you like candy ?'

Ici, la fonction d'alignement a devient : $a: \{2 \rightarrow 2; 3 \rightarrow 1; 4 \rightarrow 4\}$. On peut avoir aussi le cas de suppression d'un mot de la phrase source, autrement dit, il n'y a pas de correspondance entre le mot source et le mot cible. Dans le même exemple de la Figure 62, le mot source 'don't' a été éliminé dans la phrase cible. Ce cas est très fréquent dans l'alignement entre un texte en Anglais et un texte en ASL. Dans ce cas, le mot est remplacé par un jeton NULL (noté Ø).

[68] A partir des distributions probabilistes du tableau, la traduction de la phrase 'you blue car' est 'you blue car', mais, nous avons utilisé la traduction 'your blue car' puisqu'on va intégrer les modèles de langages dans ce qui suit. Autrement dit, la probabilité d'avoir le couple 'your blue' est plus importante que la probabilité du couple 'you blue'.

Dans cette partie, nous avons présenté les différentes possibilités de correspondance qu'on peut trouver lors de la traduction d'une phrase source f en une phrase cible e. Les différents modèles d'alignement possibles ont été formalisés par une fonction a qui tient compte de la suppression, ajout et changement d'ordre lors de la traduction. Dans ce qui suit, on définit la phrase source en Anglais par f et la phrase de langue cible par e.

Étant donné un corpus parallèle Anglais-ASL, comment peut déterminer les alignements entre une phrase source et une phrase cible afin de générer un dictionnaire ? Ceci peut se faire en implémentant les algorithmes d'apprentissage proposé par IBM [136]. Ces modèles d'alignements sont appelé 'IBM Model' qui se basent sur la probabilité des traductions lexicales en utilisant la fonction d'alignement a.

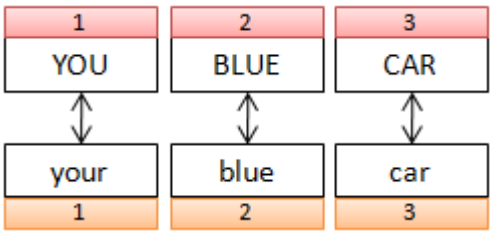
3. Algorithmes d'apprentissage et implémentation

3.1. Modèle d'alignement IBM 1

Équipé d'une fonction d'alignement et de distributions probabilistes lexicales, 'IBM Model 1' permet de calculer la probabilité de traduire une phrase f en une phrase e. Cette probabilité de traduction est noté P(e|f) –la probabilité d'avoir e si on a f. On va appliquer cette méthode pour notre modèle de traduction. On définit la probabilité de traduction d'une phrase source $f = (f_1, f_2, \dots, f_{l_f})$ de longueur l_f vers une phrase cible $e = (e_1, e_2, \dots, e_{l_e})$ de longueur l_e avec un alignement pour chaque mot en ASL (ou anglais) e_j vers un mot en Anglais (ou ASL) f_i selon la fonction a: j → i comme suit :

$$P(e,a|f) = \frac{\varepsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \qquad (4)$$

Dans l'équation précédente, la probabilité de la traduction est calculée à partir du produit de toutes les probabilités lexicales de chaque mot e_j pour j allant de 1 à l_e. Au niveau du quotient on a rajouté 1 pour tenir compte du jeton NULL. Donc, on aura $(l_f+1)^{l_e}$ alignements possible pour mapper les (l_f+1) mots de la phrase f avec tous mots de la phrase cible e. ε[69] est constante de normalisation de P(e, a|f) qui assure $\sum_{e,a} P(e,a|f) = 1$ [136]. Pour conclure cette partie, on applique l'équation précédente sur la phrase 'your blue car' qui est traduite en '**YOU BLUE CAR**'. On aura :



$$P(e,a|f) = \frac{\varepsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})$$
$$= \frac{\varepsilon}{3^4} \times \text{t(your,YOU)} \times \text{t(blue,BLUE)} \times \text{t(car,CAR)}$$
$$= \frac{\varepsilon}{4^3} \times 0.7936965 \times 0.6966292 \times 0.8256757$$
$$= 0.0071332\varepsilon$$

[69] La constante de normalisation ε est définie dans le travail de Brown et al..

La probabilité de traduire une phrase en ASL 'YOU BLUE CAR' vers l'anglais 'your blue car' (ou inversement[70]) est $0.0071332 \times \varepsilon$. On a donc notre premier traducteur automatique Anglais-ASL à base de statistiques.

Apprentissage des modèles de traductions lexicales

Nous avons introduit un modèle pour la traduction à partir des distributions probabilistes des traductions lexicales qui doivent être calculés à l'avance. Dans cette partie, on va présenter des méthodes pour l'apprentissage de ces distributions à partir d'un corpus parallèle (chaque phrase en langue source correspond à sa traduction en langue cibles). On utilisera dans ce qui suit l'Algorithme Espérance-Maximisation [167] (en anglais *Expectation Maximisation Algorithm*) abrégé en EM. Cet algorithme permet de trouver le maximum de vraisemblance des paramètres de modèles probabilistes lorsque le modèle dépend de variables latentes non observables et aussi qu'on a un problème de données incomplètes. L'objectif de cette section est d'estimer pour chaque mot en entrée f, et à partir d'une grande quantité de données, on calcule le nombre de traductions possibles. Nous sommes en face d'un problème de données incomplètes puisqu'on ne connait pas l'alignement de chaque mot e avec f. Le résultat de l'alignement sera donc une variable caché dans notre système d'apprentissage.

Algorithme Espérance-Maximisation (EM)

L'algorithme EM s'adresse aux problèmes de données incomplètes. C'est une méthode d'apprentissage itérative qui fonctionne comme suit :

i. Initialiser le modèle avec une distribution probabiliste uniforme.

ii. Appliquer le modèle sur les données existantes (Espérance).

iii. Apprendre le modèle à partir des données (Maximisation).

iv. Itérer les étapes 2 et 3 jusqu'à la convergence (généralement vers 1).

D'abord, on initialise notre modèle par une distribution probabiliste uniforme qui présentera un bon démarrage. La traduction lexicale initiale, pour le traducteur Anglais-ASL pour chaque mot f, est égale à la probabilité de traduire f pour chaque e du corpus. Ensuite, dans l'étape de l'espérance, on détermine les alignements manquant dans notre corpus pour chaque mot f avec le mot e. Puis, on apprend notre modèle à partir des valeurs résultantes du de l'étape (ii). On appliquera dans cette étape l'estimation du maximum de vraisemblance qui est une méthode statistique utilisée pour inférer les paramètres de la distribution de probabilité. On finit par itérer les deux étapes précédentes afin de converger les fonctions d'alignements pour chaque mot f vers 1.

Application de l'algorithme EM sur le modèle IBM 1

En partant de l'exemple précédent où on a calculé la probabilité de traduction en se basant sur les probabilités de traductions lexicales et de la Théorème de dérivation des fonctions composées on a :

[70] La fonction d'alignement a est une fonction bidirectionnelle.

$$P(a|e,f) = \frac{p(e,a|f)}{p(e|f)} \text{ et } P(e|f) = \sum_a P(e,a|f) \tag{5}$$

On obtient :

$$\begin{aligned} P(e|f) &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)}^{l_f} P(e,a|f) = \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)}^{l_f} \frac{\varepsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \\ &= \frac{\varepsilon}{(l_f+1)^{l_e}} \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)}^{l_f} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) = \frac{\varepsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i) \end{aligned} \tag{6}$$

A partir des équations 5 et 6, on aura :

$$P(a|e,f) = \frac{p(e,a|f)}{p(e|f)} = \frac{\frac{\varepsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})}{\frac{\varepsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)} = \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)} \tag{7}$$

Pour le quotient $\frac{\varepsilon}{(l_f+1)^{l_e}}$, on suppose qu'il égale à 1 si $l_f \sim l_e$ afin de rendre la complexité de l'algorithme d'apprentissage linéaire. L'équation 9 représente la deuxième étape de l'apprentissage –calcul de l'espérance. On passe maintenant à l'étape de maximisation où on calcule le nombre de traductions possibles à partir des probabilités d'alignements déjà calculées. Pour ce contexte on définit une fonction c appliqué sur une paire de phrases (e, f) comme suit :

$$c(e|f;e,f) = \sum_a P(a|e,f) \sum_{j=1}^{l_e} \delta(e,e_j)\delta(f,f_{a(j)}) \tag{8}$$

La fonction δ(x, y) dite 'Kronecker delta' est égale à 1 si x = y, sinon zéro. On rajouté cette fonction afin de simplifier l'équation 8. On a donc :

$$c(e|f;e,f) = \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)} \sum_{j=1}^{l_e} \delta(e,e_j) \sum_{i=0}^{l_f} \delta(f,f_{a(i)}) \tag{9}$$

La nouvelle distribution probabiliste de traduction sera :

$$t(e|f;e,f) = \frac{\sum_{(e,f)} c(e|f;e,f)}{\sum_e \sum_{(e,f)} c(e|f;e,f)} \tag{10}$$

L'algorithme 3 est une implémentation de la phase d'apprentissage EM appliqué sur le modèle IBM 1. L'algorithme assure les quatre étapes mentionnées précédemment. Si on prend par exemple un petit corpus composé de trois phrases en Anglais et leurs traductions en ASL chacun est composé de quatre mots :

i. *your car* | YOU CAR
ii. *your name* | YOU NAME
iii. *my name* | PRO-1st NAME

Les mots en Anglais sont (your, car, name, my) et les mots en ASL sont (YOU, CAR, NAME, PRO-1st). On initialise les distributions probabilistes t. La valeur t est uniforme qui égale à $\frac{1}{4} = 0.25$.

A partir de ce modèle, on exécute la première itération de l'algorithme EM (voir algorithme 3). Tous les alignements ont des valeurs semblables. Pour chaque itération, l'alignement se raffine et tend vers. Le Tableau 12 contient les différentes valeurs d'alignement pour chaque itération.

Algorithme 3 : Algorithme d'apprentissage EM appliqué sur le modèle IBM1

```
Entrée: set of sentence of pairs (e,f)
Sortie: translation probability t(e|f)
1.   Début
2.   initialize t(e|f) uniformly
3.   while not converged do
4.     count(e|f) = 0 for all e,f
5.     total(f) = 0 for all f
6.     for all sentence pairs (e,f) do
7.       for all words e in e do
8.         s-total(e) = 0
9.         for all words f in f do
10.          s-total(e) += t(e|f)
11.        end for
12.      end for
13.      for all words e in e do
14.        for all words f in f do
15.          count(e|f) += t(e|f)/s-total(e)
16.          total(f) += t(e|f)/s-total(e)
17.        end for
18.      end for
19.    end for
20.    for all english words f do
21.      for all ASL words e do
22.        t(e|f) = count(e|f)/total(f)
23.      end for
24.    end for
25.  end while
26.  fin
```

Le tableau suivant montre la convergence de certains alignements vers 1 (les lignes avec un fond bleu). Ceci indique que l'algorithme apprenne automatiquement à partir du corpus pour chaque mots (ou jetons) quelle est ou quelle sont leur traductions possibles avec une distribution probabiliste différente. Les courbes de Figure 64 montrent comment la traduction la plus probable converge vers 1 et les autres vers zéro pour un mot donné.

Aussi, on remarque que la convergence n'est pas efficace dès la première itération bien que le corpus parallèle contient seulement trois phrases en langue source et en langue cible. Il faut itérer le même algorithme plusieurs fois afin d'aboutir à une convergence vers zéro, ou vers 1. Par exemple, dans la Figure 63, la distribution probabiliste de la traduction lexicale $t(YOU|you)$ converge vers 1 après 200 itérations appliquées à un corpus parallèle contenant 740 phrases (Anglais/ASL).

La complexité de l'algorithme d'apprentissage est linéaire selon le nombre des phrases et quadratique en taille des phrases.

Tableau 12. Application de l'algorithme EM sur le modèle IBM1

		0	1	2	3	4	5	6	7	10	15	20
YOU	car	0,2500	0,5000	0,4286	0,3466	0,2755	0,2183	0,1741	0,1408	0,0828	0,0460	0,0315
YOU	name	0,2500	0,2500	0,1818	0,1208	0,0752	0,0444	0,0252	0,0139	0,0021	0,0001	0,0000
YOU	my	0,2500	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
YOU	your	0,2500	0,5000	0,6364	0,7479	0,8344	0,8961	0,9371	0,9630	0,9933	0,9997	1,0000
NAME	your	0,2500	0,2500	0,1818	0,1208	0,0752	0,0444	0,0252	0,0139	0,0021	0,0001	0,0000
NAME	car	0,2500	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
NAME	my	0,2500	0,5000	0,4286	0,3466	0,2755	0,2183	0,1741	0,1408	0,0828	0,0460	0,0315
NAME	name	0,2500	0,5000	0,6364	0,7479	0,8344	0,8961	0,9371	0,9630	0,9933	0,9997	1,0000
PRO-1st	name	0,2500	0,2500	0,1818	0,1313	0,0904	0,0596	0,0378	0,0232	0,0046	0,0002	0,0000
PRO-1st	car	0,2500	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
PRO-1st	your	0,2500	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
PRO-1st	my	0,2500	0,5000	0,5714	0,6534	0,7245	0,7817	0,8259	0,8592	0,9172	0,9540	0,9685
CAR	my	0,2500	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CAR	your	0,2500	0,2500	0,1818	0,1313	0,0904	0,0596	0,0378	0,0232	0,0046	0,0002	0,0000
CAR	name	0,2500	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CAR	car	0,2500	0,5000	0,5714	0,6534	0,7245	0,7817	0,8259	0,8592	0,9172	0,9540	0,9685

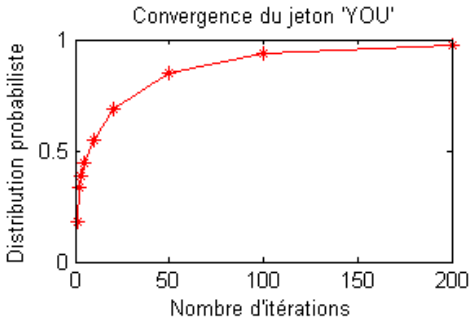


Figure 63. Evolution de la distribution probabiliste du jeton 'YOU' durant 100 itérations

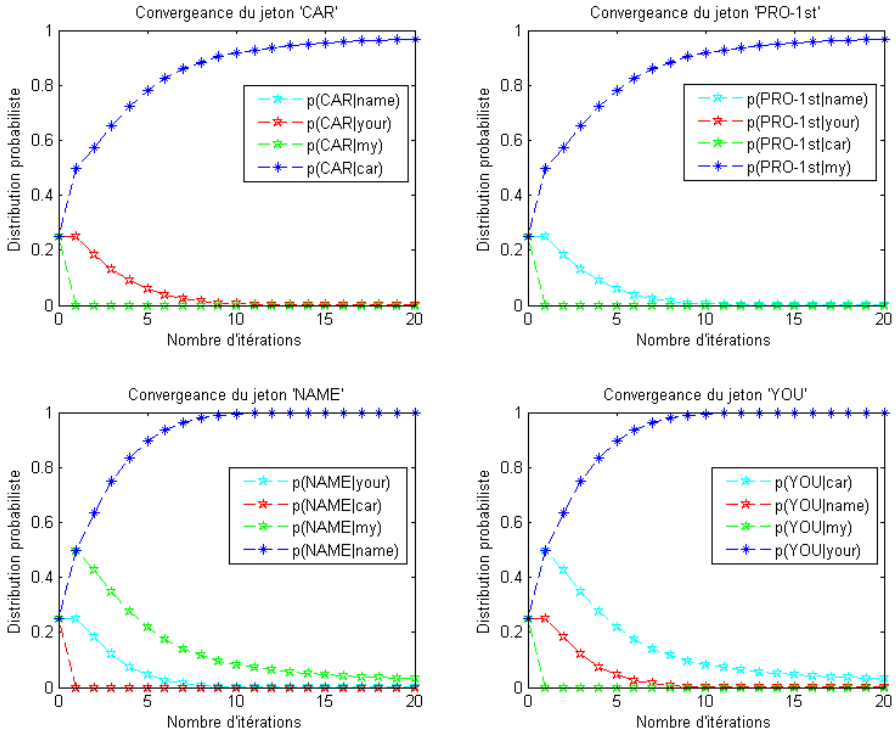


Figure 64. Convergence des traductions lexicales les plus probables vers 1 calculée à partir du modèle d'alignement IBM1

Optimisation à base des chaînes de caractères similaires

Afin de réduire le nombre d'itérations et accélérer la phase d'apprentissage pour la construction d'une mémoire de traduction lexicale Anglais-ASL, nous avons fait recours à l'intégration de la notion de similarité des chaînes de caractères. Puisque, la majorité des mots en anglais sont similaires à leurs traductions ASL et inversement. Pour cela, on a utilisé la distance de Jaro-Winkler [168] [169]. La distance de Jaro-Winkler mesure la similarité entre deux chaînes de caractères. La distance est une valeur comprise entre 0 et 1. La distance Jaro-Winkler d_w de deux chaînes S_1 et S_2 est défini comme suit :

$$d_w = d_j + \left(\ell p\left(1 - d_j\right)\right) \tag{11}$$

Où ℓ est la longueur du préfixe commun (maximum 4 caractères), p est un coefficient qui permet de favoriser les chaînes avec un préfixe commun[71] et d_j est la distance de Jaro entre S_1 et S_2 défini comme suit :

$$d_j = \frac{1}{3}\left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-t}{m}\right) \tag{12}$$

Où $|S_1|$ est la longueur de la chaîne de caractères S_1, $|S_2|$ est la longueur de la chaîne de caractères S_2, m est le nombre de caractères correspondants et t est le nombre de transpositions[72]. Cette distance est intégrée lors de l'initialisation des distributions probabilistes avant la phase de l'apprentissage. Au départ, la distribution $p(e|f)$ est initialisé à $\frac{1}{l_e}$ tel que l_e est le nombre de mots en langue cible e. Dans notre cas, la distribution probabiliste d'une traduction lexicale $p(e|f)$ sera définie comme suit :

$$t(e|f) = \alpha \cdot \frac{1}{l_e} + \beta \cdot d_w(e,f) \tag{13}$$

Où α et β deux coefficients vérifiant les conditions suivantes: $\alpha, \beta \in \mathbb{R}$ et $\alpha + \beta = 1$. Ces deux coefficients nous permettent de choisir le poids de l'utilisation des distances de Jaro-Winkler lors de l'initialisation. Pour notre expérimentation, on initialise α à 0.2 et β à 0.8 qui favorise l'utilisation de la similarité des chaînes de caractères.

Les expérimentations ont été effectué sur le même corpus qu'on l'a utilisé précédemment contenant trois phrases (voir Tableau 13). On remarque que les distributions des traductions les plus probable converge rapidement vers 1 [170].

La Figure 65 montre l'évolution de la convergence vers des traductions lexicales de quatre mots (CAR, PRO-1st, NAME, YOU) appris à partir du corpus contenant trois phrases. On remarque bien que la distribution converge plus rapidement par rapport à l'exemple précédent. Ce qui montre l'efficacité de notre approche.

Reprenons le même exemple précédent du couple p(YOU|you), la Figure 66 (la courbe en bleue) montre la distribution probabiliste en utilisant la distance de Jaro-Winkler converge vers 1 à partir de 20 itérations.

---

[71] Dans les travaux de Winkler, la valeur de p est 0.1.

[72] Dans l'annexe F, nous décrivons par des exemples comment calculer la distance Jaro-Winkler entre deux chaînes de caractères S_1 et S_2.

Tableau 13. Application de l'algorithme EM sur le modèle IBM1 en utilisant la distance de Jaro-Winkler

		init	1	2	3	4	5	6	7	10	15	20
YOU	car	0,0500	0,0643	0,0667	0,0638	0,0591	0,0540	0,0494	0,0452	0,0358	0,0264	0,0209
YOU	name	0,0500	0,0239	0,0125	0,0066	0,0035	0,0018	0,0010	0,0005	0,0001	0,0000	0,0000
YOU	my	0,0500	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
YOU	your	0,8033	0,9028	0,9486	0,9726	0,9854	0,9922	0,9958	0,9978	0,9997	1,0000	1,0000
NAME	your	0,0500	0,0266	0,0166	0,0092	0,0049	0,0026	0,0013	0,0007	0,0001	0,0000	0,0000
NAME	car	0,4722	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
NAME	my	0,0500	0,1000	0,1234	0,1175	0,1043	0,0908	0,0791	0,0694	0,0497	0,0333	0,0250
NAME	name	0,8500	0,7718	0,8978	0,9462	0,9703	0,9834	0,9908	0,9949	0,9992	1,0000	1,0000
PRO-1st	name	0,0500	0,2043	0,0897	0,0472	0,0262	0,0147	0,0083	0,0046	0,0008	0,0000	0,0000
PRO-1st	car	0,4437	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
PRO-1st	your	0,4881	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
PRO-1st	my	0,0500	0,9000	0,8766	0,8825	0,8957	0,9092	0,9209	0,9306	0,9503	0,9667	0,9750
CAR	my	0,0500	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CAR	your	0,0500	0,0705	0,0348	0,0182	0,0097	0,0052	0,0028	0,0015	0,0002	0,0000	0,0000
CAR	name	0,4722	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CAR	car	0,2900	0,9357	0,9333	0,9362	0,9409	0,9460	0,9506	0,9548	0,9642	0,9736	0,9791

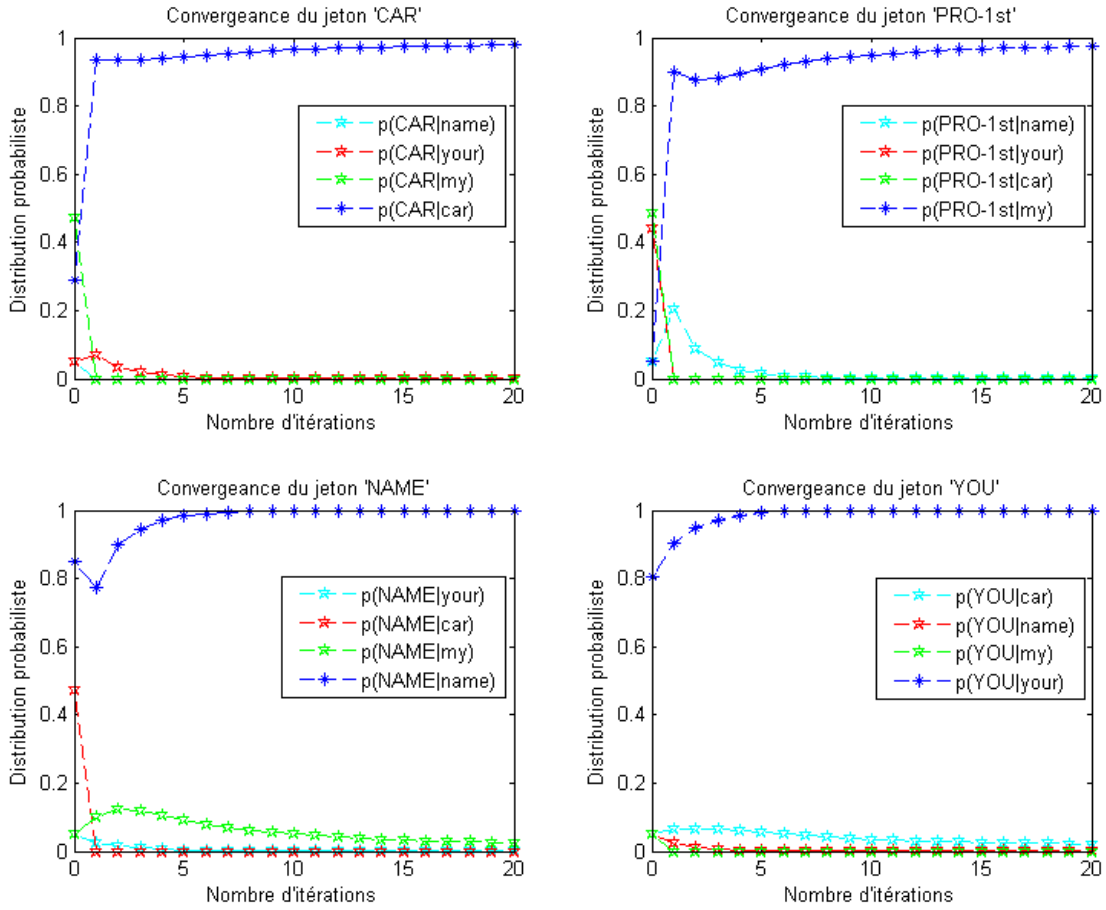


Figure 65. Convergence des traductions lexicales les plus probables vers 1 calculée à partir du modèle d'alignement IBM1 en utilisant la distance de Jaro-Winkler

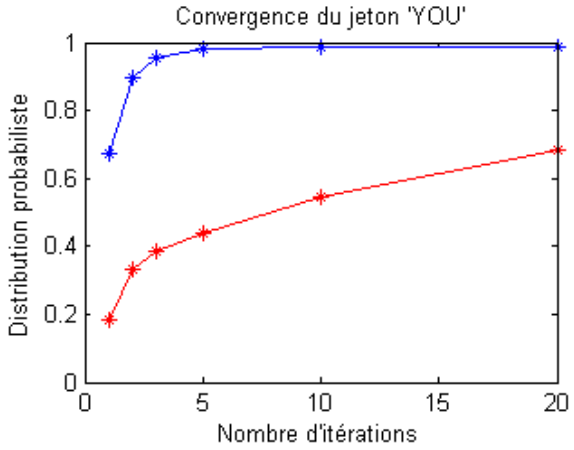


Figure 66. Evolution de la distribution probabiliste du jeton 'YOU' durant 20 itérations (en rouge) sans distance de Jaro-Winkler (en bleu) avec distance de Jaro-Winkler

Jusqu'à maintenant, nous avons présenté une implémentation du modèle d'apprentissage IBM1 sans et avec l'utilisation des distances de Jaro-Winkler. Ceci, nous a permis de construire automatiquement des dictionnaires de traductions lexicales de chaque mot en langue source vers une langue cible à partir d'un corpus parallèle. On a montré qu'à chaque itération, la distribution probabiliste est raffinée jusqu'à la convergence vers la valeur 1. Aussi, on a optimisé le nombre d'itérations, qui réduit le temps d'apprentissage, en utilisant la notion de similarité des chaînes de caractères.

3.2. Détermination de l'incertitude

L'objectif de cette partie est de déterminer la meilleure traduction à partir de données existantes qui manquent une certaine information qui est l'alignement et dont on ne connait pas certains paramètres (probabilité de traduction). Notre modèle est un modèle de traduction qui vise à traduire un texte en Anglais vers un texte en langue des signes Américaine. A ce point, nous avons appris notre modèle en utilisant l'algorithme EM et on peut mesurer maintenant la traduction. Nous avons le modèle défini pour P(e|f). Étant donnée en entrée les informations résultants de la phase d'apprentissage et prenant la phrase '**YOU CAR**' de l'exemple précédent, quelle est la probabilité pour traduire la phrase Anglais en 'your car' ? On démarre de l'équation[73] suivante :

$$P(e|f) = \frac{\varepsilon}{{l_f}^{l_e}} \sum_{j=1}^{l_e} \sum_{i=1}^{l_f} t(e_j|f_i) \tag{14}$$

Au départ, la distribution uniforme est 0.25 pour toutes les traductions. On a:

$$P(\text{your car}|\text{YOU CAR}) = \frac{\varepsilon}{2^2}(0.25 + 0.25) \times (0.25 + 0.25) = 0.0625\varepsilon \tag{15}$$

Après la première itération, la probabilité t(e|f) devient :

[73] Dans cette équation, on ignore le jeton NULL.

$$t(e|f) = \begin{cases} 0.50 \text{ si e=your et f=YOU} \\ 0.25 \text{ si e=car et f=YOU} \\ 0.25 \text{ si e=your et f=CAR} \\ 0.50 \text{ si e=car et f=CAR} \end{cases} \quad (16)$$

Par conséquent :

$$P(\text{your car}|\text{YOU CAR}) = \frac{\varepsilon}{2^2}(0.5 + 0.25) \times (0.25 + 0.5) = 0.1406\varepsilon \quad (17)$$

Et on converge vers :

$$t(e|f) = \begin{cases} 1.00 \text{ si e=your et f=YOU} \\ 0.00 \text{ si e=car et f=YOU} \\ 0.00 \text{ si e=your et f=CAR} \\ 1.00 \text{ si e=car et f=CAR} \end{cases} \quad (18)$$

Et par conséquent :

$$P(\text{your car}|\text{YOU CAR}) = \frac{\varepsilon}{2^2}(1 + 0) \times (0 + 1) = 0.25\varepsilon \quad (19)$$

De la même façon, on continue à calculer les autres distributions probabilistes. Le tableau suivant nous donne une idée sur les probabilités résultantes à partir du reste des phrases. On remarque l'amélioration de la vraisemblance.

On mesure la progression par une incertitude du modèle. Dans un traducteur automatique à base de statistiques, l'incertitude *PP* est définie comme suit:

$$\log_2 \text{PP} = -\sum_S \log_2 \text{P}(\text{e}_S|\text{f}_S) \quad (20)$$

Cette incertitude nous permet de converger vers un minimum local. Dans le Tableau 14, l'incertitude, au cours de la phase d'apprentissage, diminue de 4096 vers 113.8.

Tableau 14. Evolution de la distribution probabiliste de P(e,f) et de l'incertitude au cours de l'apprentissage en utilisant le modèle IBM 1

	initial	1ère ité.	2ème ité.	3ème ité.	…	Résultat
$P(\text{your car}\|\text{YOU CAR})$	0.0625	0.1875	0.1905	0.1913	…	0.1875
$P(\text{your car}\|\text{YOU CAR})$	0.0625	0.1406	0.1790	0.2075	…	0.2500
$P(\text{your car}\|\text{YOU CAR})$	0.0625	0.1875	0.1907	0.1913	…	0.1875
Incertitude ($\varepsilon = 1$)	4095	202.3	153.6	131.6	…	113.8

3.3. Application de l'algorithme EM sur le modèle IBM 2

Nous avons présenté dans la partie précédente le modèle IBM 1 pour un traducteur automatique vers et à partir de la langue des signes Américaine. Cependant, il existe d'autres modèles :

- Modèle IBM 2: il supporte la traduction lexicale du modèle IBM 1 et rajoute un nouveau paramètre dans l'équation de l'alignement $a(i|j, l_f, l_e)$.
- Modèle IBM 3: il supporte toute les caractéristiques du modèle IBM 2 et rajoute un modèle de richesse $n(\emptyset, f)$.

- Modèle IBM 4: En plus du modèle IBM 3, ce modèle supporte le modèle probabiliste de la distorsion lors de la traduction. Il formalise l'ajout d'un NULL.
- Modèle IBM 5: Ce modèle est le plus avancé, il permet de calculer la défaillance de la traduction lors de la phase de l'apprentissage.

Le modèle IBM 2 se caractérise par une fonction d'alignement qui supporte le changement d'ordre d'un mot de la phrase source vers un mot de la phrase cible, comme on le remarque dans la Figure 67.

La traduction d'un mot en Anglais de position i vers un mot en ASL de position j est modélisé par une distribution probabiliste d'alignement comme suit : $a(i|j, l_f, l_e)$ tel que l_f est la longueur de la phrase en Anglais et l_e est la longueur de la phrase en ASL. Le processus de traduction dans ce modèle est composé de deux phases : traduction lexicale $t(e|f)$ (celle du modèle déjà présenté précédemment) et alignement a.

La première étape est modélisée par une probabilité de traduction t(e|f). La deuxième étape permet d'aligner les mots après la traduction lexicale (voir Figure 67). On note que la fonction d'alignement a mappe pour chaque mot en ASL j avec un mot en Anglais i = a(j) et la distribution probabiliste d'alignement fait la même chose dans le sens inverse.

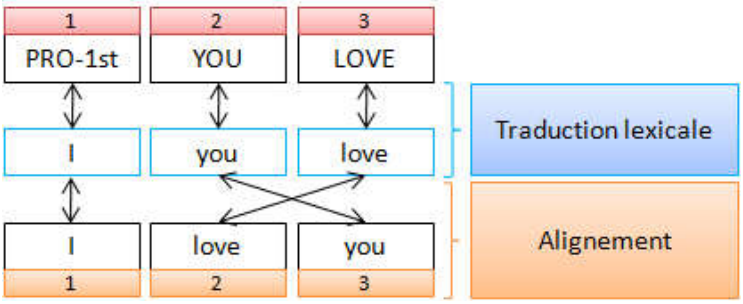


Figure 67. Exemple de traduction en utilisant le modèle d'IBM 2

Les deux étapes sont combinées mathématiquement par la formule suivante :

$$P(e,a|f) = \varepsilon \prod_{j=1}^{l_e} t\left(e_j|f_{a(j)}\right) a\left(a(j)|j, l_e, l_f\right) \tag{21}$$

En partant de P(e|f) déjà défini précédemment, on a :

$$P(e|f) = \varepsilon \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_e} t\left(e_j|f_{a(j)}\right) a\left(a(j)|j, l_e, l_f\right) \tag{22}$$

$$= \varepsilon \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t\left(e_j|f_{a(j)}\right) a\left(a(j)|j, l_e, l_f\right) \tag{23}$$

A partir de (19), on détermine P(a|e, f) à partir du nombre de traductions lexicales c:

$$c(e|f; e, f) = \sum_{j=1}^{l_e} \sum_{i=0}^{l_f} \frac{t(e|f) a\left(a(j)|j, l_e, l_f\right) \delta\left(e, e_j\right) \delta(f, f_i)}{\sum_{i'=0}^{l_f} t(e|f_{i'}) a\left(i'|j, l_e, l_f\right)} \tag{24}$$

Et le nombre de calcul pour les alignements possibles sera :

$$c(i|j,l_e,l_f;e,f) = \frac{t(e_j|f_i)a(a(j)|j,l_e,l_f)}{\sum_{i'=0}^{l_f} t(e_j|f_{i'})a(i'|j,l_e,l_f)} \quad (25)$$

L'algorithme d'apprentissage du modèle 2 est similaire à celui d'IBM1. En plus l'initialisation au départ par une distribution uniforme, on les initialise par les valeurs résultantes de la première phase d'apprentissage. Ensuite, on itère autant de fois pour calculer t(e|f) et $a(i|j,l_f,l_e)$ jusqu'à converger.

Pour ce modèle aussi, on applique la notion de similarité des chaînes de caractères lors de l'initialisation comme on a fait pour le modèle IBM1 (voir équation 13). L'algorithme 4 est une implémentation de l'algorithme d'apprentissage du modèle IBM2 qui est similaire à l'algorithme IBM1. L'initialisation de t(e|f) et $a(i|j,l_f,l_e)$ n'est pas uniforme au départ, au contraire du modèle précédent, on initialise ces deux paramètres par des estimations extraites de quelques itérations du modèle IBM1 (voir algorithme 3).

Algorithme 4 : Algorithme d'apprentissage EM appliqué sur le modèle IBM2

```
Entrée: set of sentence pairs (e,f)
Sortie: probability distributions t(lexical translation) and a(alignment)
1.   carry over t(e|f) from Model 1
2.   initialize a(i|j,le,lf) = 1/(lf+1)for alli,j,le,lf
3.   while not convergeddo
4.    //initialize
5.    count(e|f)=0 for all e,f
6.    total(f)=0 for all f
7.    counta(i|j,le,lf)=0 for all i,j,le,lf
8.    totala(j,le,lf)=0 for all j,le,lf
9.    for all sentence pairs (e,f) do
10.         le = length(e), lf = length(f)
11.         //compute normalization
12.         for j=1..le do //all word positions in e
13.               s-total(ej)=0
14.               for i =0..lf do //all word positions inf
15.                     s-total(ej)+=t(ej|fi) * a(i|j,le,lf)
16.               end for
17.         end for
18.         //collect counts
19.         for j=1..le do// all word positions in e
20.               for i =0..lf do //all word positions in f
21.                     c = t(ej|fi) * a(i|j,le,lf) / s-total(ej)
22.                     count(ej|fi)+=c
23.                     total(fi)+=c
24.                     counta(i|j,le,lf)+=c
25.                     totala(j,le,lf)+=c
26.               end for
27.         end for
28.   end for
29.   //estimate probabilities
30.   t(e|f)=0 for all e,f
31.   a(i|j,le,lf)=0 for all i,j,le,lf
32.   for all e,fdo
33.         t(e|f) = count(e|f) / total(f)
34.   end for
35.   for all i,j,le,lf do
36.         a(i|j,le,lf) = counta(i|j,le,lf) / totala(j,le,lf)
37.   end for
38.  end while
```

L'estimation de la distribution probabiliste de la traduction lexicale du modèle IBM1, ce n'est un cas particulier du modèle IBM2 si on a: $a(i|j, l_f, l_e) = \frac{1}{l_f+1}$.

L'implémentation de ce modèle d'apprentissage est décrite dans l'algorithme 4.

De même que l'algorithme 3, la complexité de l'algorithme IBM2 est quadratique lors de l'initialisation (puisque on fait recours à IBM1 lors de l'initialisation) et quadratique lors de l'apprentissage.

3.4. Application de l'algorithme EM sur le modèle IBM 3

Jusqu'à moment, on n'a pas modélisé le nombre de mots résultants après la traduction pour chaque mot en entrée. Généralement, chaque mot en Anglais (ou ASL) est traduit en un seul mot en ASL (ou en anglais), mais on peut trouver des cas où un mot en ASL est traduit en deux mots à la fois comme le signe '**THANK-YOU**' qui peut être traduit en '*Thank you*' en anglais. Et même le cas de la suppression d'un mot dans la phrase source. Nous voulons maintenant modéliser un modèle de richesse des mots en entrée directement avec les distributions probabiliste $n(\emptyset|f)$. Pour chaque mot en Anglais f, la distribution probabiliste indique en combien de mots sera traduit $\emptyset = 1,2, \ldots$. Le modèle de richesse doit assurer aussi la suppression d'un mot en entrée lorsque $\emptyset = 0$.

En plus, on peut ajouter un jeton dit NULL qui représentera un mot en ASL (ou anglais) n'ayant pas une correspondance dans la phrase en Anglais (ou ASL). Le modèle de richesse du jeton NULL est déterminé de la même façon $n(\emptyset|\text{NULL})$.

L'apprentissage du modèle IBM s'illustre en quatre étapes comme l'indique cet exemple :

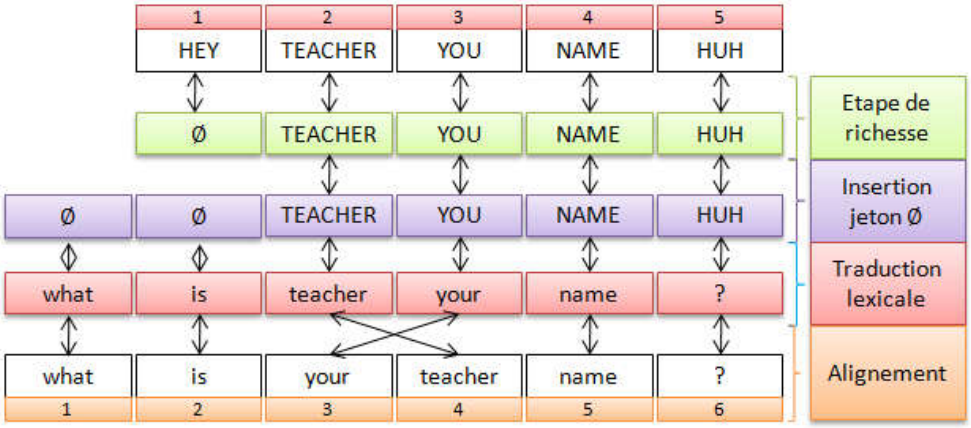


Figure 68. Les différentes étapes de traduction du modèle IBM 3

Plus formellement, le modèle IBM 3 peut être dérivé de l'équation 21 qui pour chaque mot en entrée f_i on génère plusieurs mots $\emptyset_i$ par le calcul de la distribution probabiliste $n(\emptyset_i|f_i)$. Le nombre d'insertion du jeton NULL est $\emptyset_0$. On a donc $\sum_{i=1}^{l_f} \emptyset_i = l_e - \emptyset_0$ mots générés à partir des mots en entrée et au maximum $l_e - \emptyset_0$ jetons NULL. La probabilité de générer un jeton NULL $\emptyset_0$ est :

$$P(\emptyset_0) = \binom{l_e - \emptyset_0}{\emptyset_0} P_1^{\emptyset_0} P_0^{l_e - 2\emptyset_0} \quad (26)$$

P_0 est la probabilité qu'aucun jeton NULL sera inséré et P_1 indique qu'un seul jeton NULL sera généré. Le terme combinatoire $\binom{l_e - \emptyset_0}{\emptyset_0}$ indique le nombre d'insertion d'un jeton au cours de la phase d'apprentissage. Si on combine le modèle de richesse avec la probabilité d'insérer un jeton NULL, on a :

$$\binom{l_e - \emptyset_0}{\emptyset_0} P_1^{\emptyset_0} P_0^{l_e - 2\emptyset_0} \prod_{i=1}^{l_f} \emptyset_i!\, n(\emptyset_i | e_i) \tag{27}$$

Pour chaque mot en Anglais, on factorise la probabilité de richesse $n(\emptyset_i|e_i)$. Le factoriel $\emptyset_i!$ résulte pour chaque alignement. Pour modéliser le modèle IBM 3 appliqué à la langue des signes, on combine tous les modèles proposés (alignement, richesse et distribution probabiliste de la traduction lexicale), on a :

$$P(e|f) = \sum_{a(1)=0}^{l_f} \cdots \sum_{a(l_e)}^{l_f} \prod_{j=1}^{l_e} \binom{l_e - \emptyset_0}{\emptyset_0} P_1^{\emptyset_0} P_0^{l_e - 2\emptyset_0} \prod_{i=1}^{l_f} \emptyset_i! n(\emptyset_i|e_i) \times \prod_{j=1}^{l_e} t\left(e_j | f_{a(j)}\right) d(j|a(j), l_e, l_f) \tag{28}$$

La Figure 69 montre un exemple de convergence vers 1 de la traduction lexicale t(what|∅). L'estimation de la distribution probabiliste est calculée à partir d'un petit corpus parallèle ASL-Anglais contenant quatre phrases seulement :

i. *your car* | YOU CAR

ii. *your name* | YOU NAME

iii. *my name* | PRO-1st NAME

iv. *what is your name ?* | NAME YOU[74]

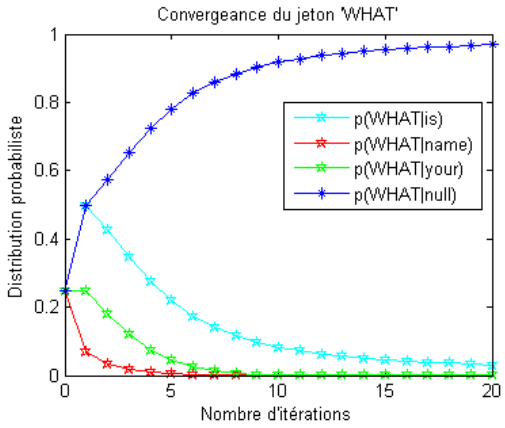


Figure 69. Evolution de la distribution probabiliste de la traduction lexicale (WHAT,∅) et sa convergence vers 1

La complexité de l'algorithme d'apprentissage des estimations de la distribution probabiliste selon le modèle IBM3 est exponentielle. Pour réduire le temps d'apprentissage, on fait recours à des heuristiques comme le hill climbing [171] qui réduit l'espace des alignements possible.

[74] Dans certaines références, la traduction de la phrase 'What is your name ?' en ASL est 'NAME YOU WHAT' et dans d'autres références est 'NAME YOU' et le signe 'WHAT' est transformé en une expression faciale. Dans notre cas, nous utilisons la deuxième traduction en vue de montrer la convergence de la traduction lexicale.

3.5. Les modèles IBM 4 et 5

Le modèle IBM 3 avec raffinement par la fonction de distance de Jaro-Winkler est un modèle d'un traducteur automatique par excellence à base de statistiques. Ce modèle IBM 4 rajoute une amélioration au niveau de l'alignement pour les phrases très longues (dépassant les 20 mots) où l'alignement d'un mot en position 20 dépend du mot de position 19. On parle d'un alignement relatif à une position donnée. Pour le modèle IBM 5, il rajoute un modèle où on a plusieurs traductions pour un seul mot en entrée dans la même position ce qui n'est pas le cas pour la paire de langues Anglais-ASL. Pour cette section, on s'est intéressé seulement aux trois premiers modèles où on réduit la complexité d'apprentissage et l'appliquer sur une langue des signes. Aussi, les modèles d'IBM ne font plus un état de l'art pour les traductions, mais ils sont la source d'intuition pour développer de nouvelles approches pour les traducteurs automatiques statistiques.

4. Alignement

Après avoir présenté les modèles d'apprentissage pour les estimations des traductions lexicales à partir d'un corpus parallèle, nous décrivons dans ce paragraphe la tâche finale pour la construction d'une mémoire de traductions lexicales qui est l'alignement.

La seule façon de visualiser l'alignement entre une phrase source et sa traduction est d'utiliser une matrice comme le montre la

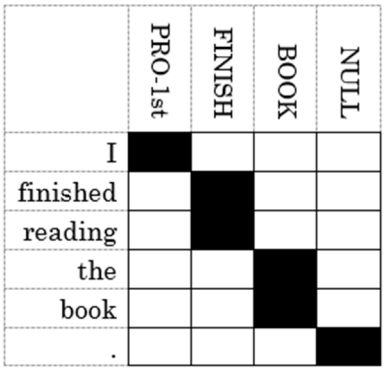


Figure 70. L'alignement comme le montre la matrice n'est pas mot-à-mot et aussi on peut y avoir le cas de ne pas traduire un mot (jeton null). Les cases noires indiquent que l'élément est aligné entre le mot en ligne et sa traduction en colonne, sinon la case est coloré en blanc.

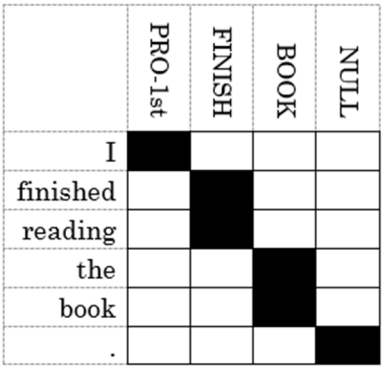


Figure 70. Matrice d'alignement de la phrase 'I finished reading the book.' (les lignes) et sa traduction 'PRO-1st FINISH BOOK' (les colonnes)

On remarque aussi, que l'alignement n'est pas identique pour une phrase donnée en Anglais et sa traduction en ASL ou inversement comme le montre la Figure 71.

L'alignement est déterminé à partir des estimations des distributions probabilistes des traductions lexicales $t(e|f)$. Donc, chaque alignement aura un score à la fin de phase de détermination.

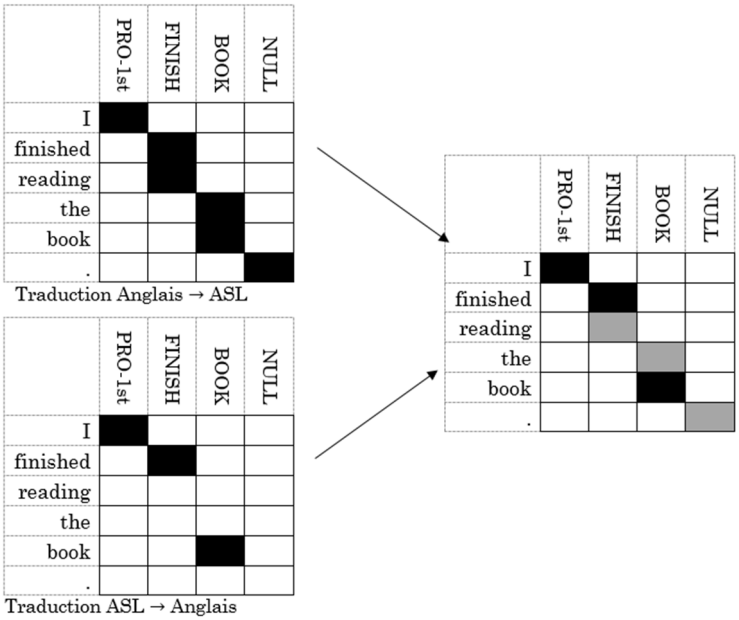


Figure 71. Deux matrices d'alignement de la phrase 'I finished reading the book.' et sa traduction 'PRO-1st FINISH BOOK' (en haut) de l'anglais vers l'ASL (en base) de l'ASL vers l'anglais (à droite) couplage entre les deux alignements

5. Evaluation

Depuis longtemps, les chercheurs essayent de développer des méthodes d'évaluation de l'alignement et même de la traduction. La qualité de ces méthodes est déterminée à partir d'une collection de test avec un ratio d'or validé par des experts humains. Pour avoir un standard d'évaluation, les annotateurs doivent respecter des directives strictes qui résolvent les ambigüités. Une métrique commune pour évaluer l'alignement des mots après la phase d'apprentissage appelé Taux d'Erreur de l'Alignement 'Alignment Error Rate AER)', défini comme suit :

$$AER(S,P;A) = \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{29}$$

Tel que A est l'alignement, A est l'ensemble des alignements sûres et P l'ensemble des alignements les plus probable. La qualité d'alignement A est déterminé à partir d'un alignement sûr S et un alignement possible P. Le taux d'erreur parfait est 0%.

La mesure de précision est le chevauchement entre l'ensemble des liens hypothétiques et l'ensemble des liens annotés manuellement (l'alignement parfait ou '*gold*'). Cette mesure de précision est définie comme suit :

$$Precision(A, P) = \frac{|A \cap P|}{|A|} \quad (30)$$

Le rappel mesure le pourcentage des liens qui se trouvent dans l'alignement hypothétique. Il est définit comme suit :

$$Rappel(A, S) = \frac{|A \cap S|}{|S|} \quad (31)$$

6. Conclusion

Au cours de ce chapitre, nous avons construit notre mémoire de traductions lexicales Anglais-ASL à partir d'un corpus parallèle. L'étape de construction a nécessité une phase d'apprentissage dont on a fait recours à des algorithmes d'estimation [155]. En plus, afin de réduire le nombre d'itérations, nous avons intégré l'algorithme de similarité des chaînes de caractères à base des distances de Jaro-Winkler, qui a donné des résultats plus rapides [170]. Cette mémoire de traduction est l'entrée du système de traduction automatique statistique qui sera décrit dans les chapitres qui suivent.

Chapitre 7 : Traduction automatique statistique Anglais-ASL

1. Introduction

Dans le chapitre précédent, nous avons décrit une méthode d'alignement à base de mots ou jetons. Mais, cette entité élémentaire n'aboutit pas à une efficacité meilleure. Puisque, la plupart des mots en ASL (ou en anglais) sont traduites en plusieurs mots dit segment[75] et vice-versa. Dans ce cas, les segments sont les unités élémentaires à traduire. Ceci nous permet de conserver la sémantique et le contexte de la phrase lors de la traduction. La mémoire à base de segments, qui sera utilisée par le traducteur automatique à base de segments (en anglais *phrase-based machine translation*) a été introduite par Koehn [172]. Ce modèle de traduction est maintenant utilisé dans la plupart des traducteurs automatiques statistiques[76]. Pour nous, on s'intéresse à son implémentation pour les langues des signes spécialement pour la langue des signes Américaine ASL. Au cours de chapitre, nous décrivons l'approche de construction d'une mémoire de traduction à base de segments pour le couple de langues Anglais-ASL. Ensuite, nous décrivons l'approche à base de statistiques pour génération des traductions à partir d'un texte par l'utilisateur et à partir de la mémoire de traductions à base de segments. Enfin, nous entamons la phase d'évaluation de notre système ainsi que le calcul du score BLEU.

2. Mémoire de traduction à partir des segments

2.1. Modèle Standard

Pour mieux comprendre le principe du modèle à base de segments, nous commençons à étudier un exemple de près. Commençons par étudier deux exemples :

- la phrase en anglais '*Did you see the new clothing store they put in ?*' et sa traduction en ASL '**WHAT-DO YOU SEE NEW CLOTHES STORE THEY PUT**'.
- la phrase en anglais '*It's nice. It's next to the grocery store.*' et sa traduction en ASL '**IT NICE. FOOD STORE NEXT TO**'.

Les figures suivantes montrent un alignement possible de chaque phrase et sa traduction. On remarque que les entités alignées sont des groupements de mots.

[75] Segment: ensemble de mots unitaire.
[76] Google Translate est un traducteur automatique statistique à base de segments [http://translate.google.com/]

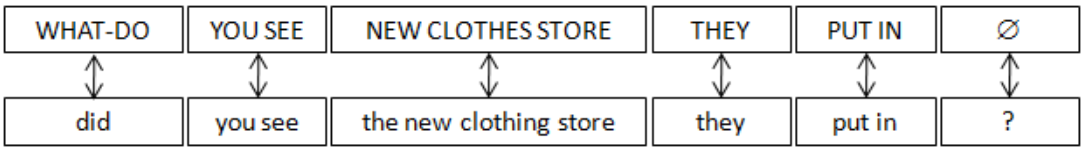


Figure 72. Alignement à base de segments de la phrase en anglais 'Did you see the new clothing store they put in ?' et sa traduction en ASL 'WHAT-DO YOU SEE NEW CLOTHES STORE THEY PUT'

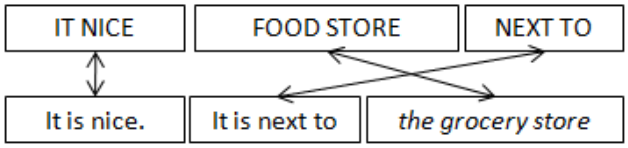


Figure 73. Alignement à base de segments (avec croisement) de la phrase en anglais 'It is nice. It is next to the grocery store' et sa traduction en ASL 'IT NICE FOOD STORE NEXT TO'

D'après les figures, un groupement de mots dit segment est une ensemble de jetons. Chaque segment de la langue source se correspond à un segment de la langue cible à partir d'une table de traductions à base de segments.

Par exemple, le Tableau 15 montre l'estimation des distributions probabilistes des traductions possibles du segment 'NEXT TO'[77] :

Tableau 15. Les probabilités d'alignement possibles du segment 'NEXT TO'

Traduction de **'NEXT TO'**	Probabilité d'alignement
next to	0.8236514
it is next to	0.4532145
it is next to ,	0.1000450
, it is next to	0.8745230
, it is next to ,	0.0232540
. it is next to	0.0221520
to be next to	0.0006548
Ø[78]	0.0000325

L'alignement à base de segments est plus efficace de point de vue contexte par rapport à l'alignement mot-à-mot déjà décrit dans le chapitre précédent. Ceci est un avantage qui nous rend la traduction généré automatiquement plus fluide et proche de la traduction ou l'interprétation humaine. Aussi, la traduction à partir des segments nous permet de résoudre les problèmes d'ambigüités.

2.2. Formulation mathématique

A partir du modèle mathématique déjà utilisé dans les deux chapitres précédents, on a :

[77] La traduction de la phrase 'NEXT TO' est prise du site [www.start-american-sign-language.com]
[78] De même que la traduction lexicale, le caractère Ø désigne que le segment 'NEXT TO' n'a pas été traduit et a été remplacé par un jeton null.

$$e = argmax_e p(e|f) \quad (1)$$

En utilisant la règle de Bayes, l'équation (1) devient :

$$e = argmax_e p(f|e) P_{LM}(e) \quad (2)$$

Tel que:

- e est la phrase cible (la traduction).
- f est la langue source.
- $P_{LM}(e)$ est le modèle du langage de la langue cible.

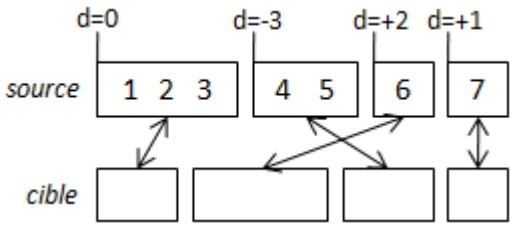
Cette formulation est identique à celle du modèle IBM (alignement mot-à-mot), pour l'alignement à segments, on décompose $p(f|e)$ en :

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \emptyset(\bar{f}_i|\bar{e}_i) d(start_i - end_i - 1) \quad (3)$$

Tel que :

- la phrase f est décomposé en I segments noté $\bar{f}_i$ et de même pour la langue cible.
- Le segment $\bar{f}_i$ est traduit en $\bar{e}_i$.
- L'estimation de traduire le segment $\bar{f}_i$ en $\bar{e}_i$ est défini par l'équation $\emptyset(\bar{f}_i|\bar{e}_i)$.
- $d(start_i - end_i - 1)$ est un modèle de réorganisation des segments à partir des distances sachant que:
  - $start_i$ est la position du premier jeton dans le segment source qui est traduit en un jeton à la position i du segment cible.
  - end_i est la position du dernier jeton du segment source.
  - La distance de réorganisation est défini comme suit: $start_i - end_i - 1$.

La distance de réorganisation (*reordering distance* en anglais) est le nombre de mots sautés (en avant ou en arrière) comme le montre la Figure 74.



Cible	Segments	Mouvements	d
1	1-3	début	0
2	6	saut 4-5	+2
3	4-5	recul 4-6	-3
4	7	saut 6	+1

Figure 74. Calcul de la distance de réorganisation d'une phrase source vers une phrase cible

Pour déterminer la probabilité de la distance d, on utilise une fonction exponentielle décroissante. La fonction est définit comme suit :

$$d(x) = \propto^{|x|} \quad (4)$$

Tel que $\propto$ est un paramètre qui varie entre 0 et 1 ($\propto \in [01]$). Cette fonction sera utilisée lors de la construction du mémoire de traduction à base de segments.

Le modèle mathématique de l'équation (3) définit un le principe classique d'un traducteur automatique statistique à base de segments [141].

2.3. Approche

L'apprentissage de la mémoire de traduction à base de segments commence après la phase d'alignement de traductions lexicales discutées dans le chapitre précédent. Les paires de segments sont extraits à partir de la matrice d'alignement symétrique (alignement ASL-Anglais et alignement Anglais-ASL) comme le montre la figure :

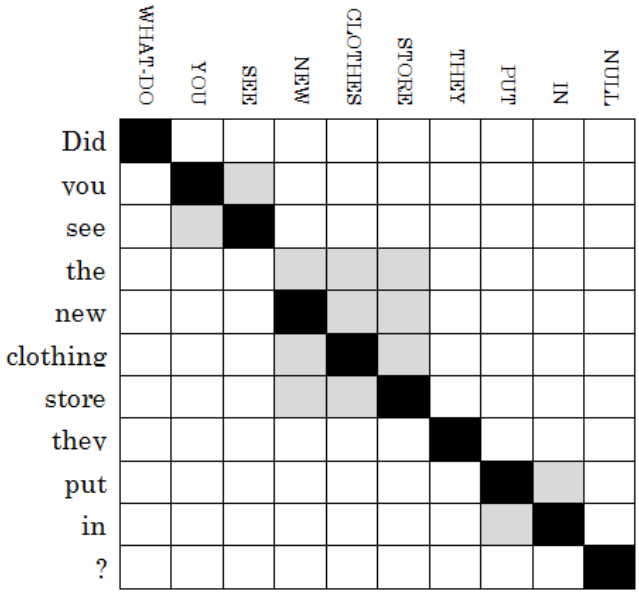


Figure 75. Matrice d'alignement de la phrase en anglais 'Did you see the new clothing store they put in ?' et sa traduction en ASL 'WHAT-DO YOU SEE NEW CLOTHES STORE THEY PUT IN'

Dans cet exemple, deux paires de phrases sont extraites :

- YOU SEE | you see
- the new clothing store | NEW CLOTHES STORE
- put in | PUT IN

Les segments extraites à partir de la matrice d'alignement symétrique sont généralement les phrases les plus utilisés en langue source ou cible et qui sont généralement des phrases courtes.

A partir de la matrice d'alignement, on essaye d'extraire un alignement de segments. Donc, on détermine pour chaque paire de segments $(\bar{f}, \bar{e})$ un alignement possible A. La fonction d'alignement doit satisfaire la condition suivante : pour tous les jetons $f_1, f_2, \dots, f_n$ du segment $\bar{f}$, on a un alignement A pour les jetons $e_1, e_2, \dots, e_n$ du segment $\bar{e}$ et inversement tel que:

$$\begin{aligned} \forall e_i \in \bar{e}: (e_i, f_j) \in A \Longrightarrow\ & f_j \in \bar{f}\ \wedge\ \forall f_j \in \bar{f}: (e_i, f_j) \in A \\ \Longrightarrow\ & e_i \in \bar{e} \wedge\ \exists e_i \in \bar{e}, f_j \in \bar{f}: (e_i, f_j) \in A \end{aligned} \tag{5}$$

Cette condition nous s'assure qu'aucun jeton ne trouve dans deux segments à la fois. A partir du quelle, nous définissons notre algorithme de génération d'une mémoire de traduction à base de segments.

2.4. Algorithme d'extraction

Le principe d'extraction des segments à partir des alignements déjà déterminés à partir des modèles d'alignement IBM est le suivant: parcourir tous les segments possibles de la phrase cible et trouver le segment de longueur minimal[79] de la langue source. L'algorithme s'arrête quand on identifie tous les alignements de chaque segments de longueur minimal de la langue cible et son correspondant de longueur minimal aussi de la langue source à partir de la matrice d'alignement. L'implémentation de cette fonction d'extraction est décrite dans l'algorithme 5.

L'algorithme 5 détermine pour chaque segment $e_{start}, \dots, e_{end}$ son segment correspondant de longueur minimal $f_{start}, \dots, f_{end}$. Si la condition est satisfaite, on ajouter la paire de segments à la mémoire de traduction.

Après avoir extrait les paires de segments, on calcule l'estimation de la distribution probabiliste de chaque paire. Celle-ci est déterminé à partir de nombre de paires noté $count(\bar{e}, \bar{f})$. La fonction d'estimation est noté $\emptyset(\bar{f}, \bar{e})$.

Algorithme 5 : Algorithme d'extraction des segments

```
Entrée: word alignment A for sentence pair (e,f)
Sortie: set of phrase pairsBP
1.   for estart = 1 ... length(e) do
2.    for eend = estart ... length(e) do
3.          //find the minimally matching foreign phrase
4.          (fstart,fend) = ( length(f), 0 )
5.          for all(e,f) ∈ A do
6.                if estart ≤ e ≤ eend then
7.                      fstart = min( f, fstart ) ; fend = max( f, fend )
8.                end if
9.          end for
10.         add extract(fstart,fend,estart,eend) to set BP
11.   end for
12.  end for

function extract(fstart,fend,estart,eend)

1.   return {} if fend == 0//check if at least one alignment point
2.   // check if alignment points violate consistency
3.   for all (e,f) ∈ A do
4.    return {} if e < estart or e > eend
5.   end for
6.   // add pharse pairs (incl. additional unaligned f)
7.   E={} ; fs = fstart
8.   repeat
9.    fe = fend
10.   repeat
11.         add phrase pair (estart .. eend, fs .. fe) to set E
12.         fe++
13.   until fe aligned
14.   fs--
15.  until fs aligned
16.  return E
```

[79] Un segment de longueur minimal est un segment dont le nombre de jetons est le plus petit.

L'estimation $\emptyset(\bar{f},\bar{e})$ est définit par :

$$\emptyset(\bar{f},\bar{e}) = \frac{count(\bar{e},\bar{f})}{\sum_{\bar{f}_i} count(\bar{e},\bar{f}_i)} \quad (6)$$

Cette fonction détermine tous les pairs de segments possibles. Imaginons, si on a un corpus parallèle de grande taille (en giga-octets), la génération de la mémoire de traductions à base de segments nécessite une mémoire machine très importante puisque le nombre des paires est exponentiel. Pour réduire le temps d'exécution et diminuer la taille de la mémoire de traductions, on fait recours à des optimisations et des extensions du modèle d'alignement comme on fait pour le modèle d'alignement mot-à-mot [173].

La complexité de l'algorithme d'extraction est $O(n^2)$ si on a une phrase de la langue source de taille n et $O(m^2)$ si la taille de la phrase de la langue cible est m. Si m=n, la complexité est $O(n^4)$.

2.5. Phase d'apprentissage

Après avoir déterminer les paires de segments possibles, on passe à l'étape de l'estimation des traductions probabilistes de chaque paire $p(\bar{e},\bar{f})$. Puisqu'on a réduit l'espace de paires possibles dans le paragraphe précédent, on est face maintenant à un problème de données incomplètes. Sachant que l'algorithme EM s'adresse aux problèmes de données incomplètes, comme on l'a vue dans le chapitre précédent, on utilise cette méthode pour la détermination de la mémoire de traduction à partir des segments:

i. Initialiser le modèle avec une distribution probabiliste uniforme, $p(\bar{e},\bar{f})$ est identique pour toutes les paires.

ii. Phase de calcul de l'espérance: on calcule $count(\bar{e},\bar{f})$

iii. Phase de maximisation: apprendre le modèle à partir des données (Maximisation) et on itère jusqu'à la convergence.

Lors de l'apprentissage, on peut rajouter des heuristiques d'optimisation du temps de calcul afin d'accélérer la phase d'apprentissage [174].

3. Modélisation du langage

Dans un système de traduction automatique statistique, le rôle du modèle de langage est de restituer des contraintes linguistiques en guidant le décodage [175]. Il permet donc d'éliminer des hypothèses les suites de mots ayant une probabilité très faible et tente de garantir le fait qu'une suite de mots choisie sera aussi grammaticalement correcte que possible. Pour modéliser les contraintes d'une langue, les modèles de langage statistiques les plus souvent utilisés attribuent une probabilité à chaque séquence de mots W de longueur k, exprimée par :

$$P(W) = P(W_1)\prod_{i=1}^{k} P(W_i|W_1,\dots,W_{i-1}) = P(W_1)\prod_{i=1}^{k} P(W_i|h_i) \quad (7)$$

où h_i représente l'historique du mot W_i. Par conséquent, $h_i = W_1, \dots, W_{i-1}$. La représentation de la probabilité $P(W)$ du modèle de langage utilisée ici est celle de la traduction. Pour la traduction automatique statistique, cette probabilité serait représentée par $P(e)$.

Les modèles de langage de type N-Gramme sont les modèles statistiques les plus utilisés lorsqu'il s'agit de construire un système traduction. Dans ce genre de modèle, l'historique d'un mot est composé des $n-1$ mots qui le précèdent, n étant l'ordre du modèle, généralement d'une valeur de trois ou quatre, bien que des modèles d'ordre cinq voire sept soient parfois utilisés pour des tâches spécifiques. Cette limitation à un ordre de trois ou quatre s'explique principalement par les contraintes techniques et un manque de données d'apprentissage, empêchant de prendre en compte un large historique. Pour des ordres allant de un à quatre, on parlera alors de modèle uni-gramme ($n = 1$), bi-gramme ($n = 2$), trigramme ($n - 3$) et quadri-gramme ($n - 4$). Pour un ordre supérieur à 1, l'équation (7) peut s'écrire :

$$P(W) \simeq P(W_1) \prod_{i=1}^{k} P(W_i | W_{i-n+1}, \dots, W_{i-1}) \tag{8}$$

Ce type de modèle permet de couvrir aisément la totalité des phrases exprimables dans un langage donné. Lors de la construction d'un tel modèle, l'apprentissage consiste à estimer un ensemble de probabilités à partir d'un corpus de données textuelles. Plusieurs techniques existent afin de réaliser cet apprentissage [176], la plus courante étant l'estimation par maximum de vraisemblance : la distribution des probabilités du modèle est celle qui maximise la vraisemblance du corpus d'apprentissage. L'équation suivante exprime cette estimation :

$$P_{MV}(W_i | W_{i-n+1}, \dots, W_{i-1}) = \frac{c(W_{i-n+1}, \dots, W_{i-1})}{c(W_{i-n+1}, \dots, W_i)} \tag{9}$$

Où $c(W_{i-n+1}, \dots, W_{i-1})$ représente la fréquence d'apparition du N-Gramme $W_{i-n+1}, \dots, W_{i-1}$ dans le corpus d'apprentissage.

Comme nous venons de l'exposer, la qualité d'un modèle de langage N-Gramme dépend fortement de la quantité de données textuelles à notre disposition. L'insuffisance de données reste toujours un des problèmes principaux rencontrés, même avec de très grands corpus, de fait certains N-Grammes n'apparaîtront jamais dans le modèle. Il est même possible que certains mots du lexique qui sera utilisé pour le décodage soient absents du corpus d'apprentissage et par conséquent du modèle généré. Pour ces raisons, il existe diverses techniques de lissage [177] qui tentent de compenser ce manque. Elles permettent de généraliser le modèle en attribuant une probabilité non-nulle à un événement inobservé dans le corpus d'apprentissage.

Le lissage par repli (en anglais *back-off*) est un de ces mécanismes permettant de compenser le manque de données pour certains N-Grammes [178]. Dans cette technique, on utilisera une probabilité issue d'un N-Gramme d'ordre inférieur $n-1, n-2, \dots$ lorsque le N-Gramme d'ordre n pour un mot et un historique donné n'a pas été observé.

4. Décodage

4.1. Approche

Le décodage en SMT comprend la recherche des hypothèses t ayant les plus grandes probabilités pour être considérées comme étant les traductions de la phrase source en question, suivant le modèle de traduction $P(e|f)$ [179]. Le modèle $P(e|f)$ est une combinaison log-linéaire de quatre principaux composants : un ou plusieurs modèles trigramme de la langue cible, un ou plusieurs modèles de traduction basés sur les segments, un modèle de distorsion qui rend compte des différences dans l'ordre des mots en langues source et cible, et un modèle de longueur qui rend compte des différences de longueur entre les deux langues. A partir des deux équations (2) et (3) cités au début du chapitre, on définit la meilleur traduction e_{best} par :

$$e_{best} = argmax_e \prod_{i=1}^{I} \emptyset(\overline{f_i}|\overline{e_i}) d(start_i - end_i - 1) P_{LM}(e) \quad (10)$$

Plusieurs composantes contribuent dans le calcul du score e_{best}:

- L'estimation de la distribution probabiliste à partir des segments $\emptyset$: Lorsqu'on a une phrase en entrée $\overline{f_i}$ à traduire en $\overline{e_i}$ pour l'ensemble des segments $i = 1 \dots I$, on récupère la valeur de l'estimation de la distribution probabiliste à partir de la mémoire de traduction à base de segments déjà calculée lors de la phase d'apprentissage pour chaque paire de segments $\emptyset(\overline{f_i}|\overline{e_i})$.
- Le modèle de réorganisation d: pour cette composante, on essaye de calculer la valeur end_i du dernier jeton du segment pour chaque segment à partir d'une jeton $start_i$.
- Le modèle du langage P_{LM}: comme on déjà parlé dans la section précédente, cette composante assure la qualité de la traduction vers la langue cible. Le score est raffiné par un score extrait du modèle n-gramme. Par exemple, pour une modèle bi-gramme (2-gram), la probabilité d'une phrase commençant par le jeton '*he*' est $P_{LM}(he|\langle S \rangle)$. La probabilité pour le jeton '*he*' est suivi par le jeton '*does*' est $P_{LM}(does|he)$.

4.2. Exemple

Prenons un exemple, soit la phrase f : '**NAME YOU WHAT**' (en ASL) à traduire en e (en anglais). Tout d'abord, le décodeur essaye d'extraire les différentes segments possible de la phrase source $\overline{f_i}$. Ensuite, pour chaque segment, on cherche sa traduction à partir de la mémoire de traduction et affecter un score pour ce couple. Puis, pour tous les couples extraits, on calcule le score global de la traduction possible. Enfin, après avoir calculé tous les scores des différentes hypothèses possibles, on détermine la meilleure traduction ayant le score le plus important. Dans la Figure 76, on remarque que la traduction de la phrase 'NAME YOU WHAT' s'effectue sur trois étapes. La première étant de traduire le segment 'WHAT' en '*what is*'. La deuxième étape consiste à traduire lexicalement le jeton 'YOU' en '*your*'. Enfin, le segment 'NAME' est traduit en '*name ?*'.

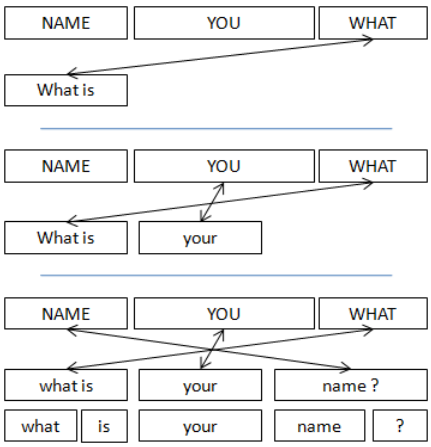


Figure 76. Les différentes étapes de traductions de la phrase 'NAME YOU WHAT'

Pour cet exemple et en utilisant le corpus déjà cité précédemment, le nombre d'hypothèses pour cette traduction est 5862 pour les différents segments.

La Figure 77 montre une sélection de quelques hypothèses de traductions possibles de chaque segment de la phrase source. Un score est affecté pour chaque hypothèse qui nous permet ensuite de calculer le score globale en tenant compte du modèle de réorganisation[80] et du modèle du langage[81]. Le nombre d'hypothèse augment en exponentiel lorsqu'on augmente la taille du corpus.

Le problème du décodage est classé comme étant comme étant un problème NP-Complet. En littérature, on trouve plusieurs techniques de décodage afin de minimiser le nombre d'hypothèses ou de minimiser le temps de détermination des meilleures traductions possibles [180] [181].

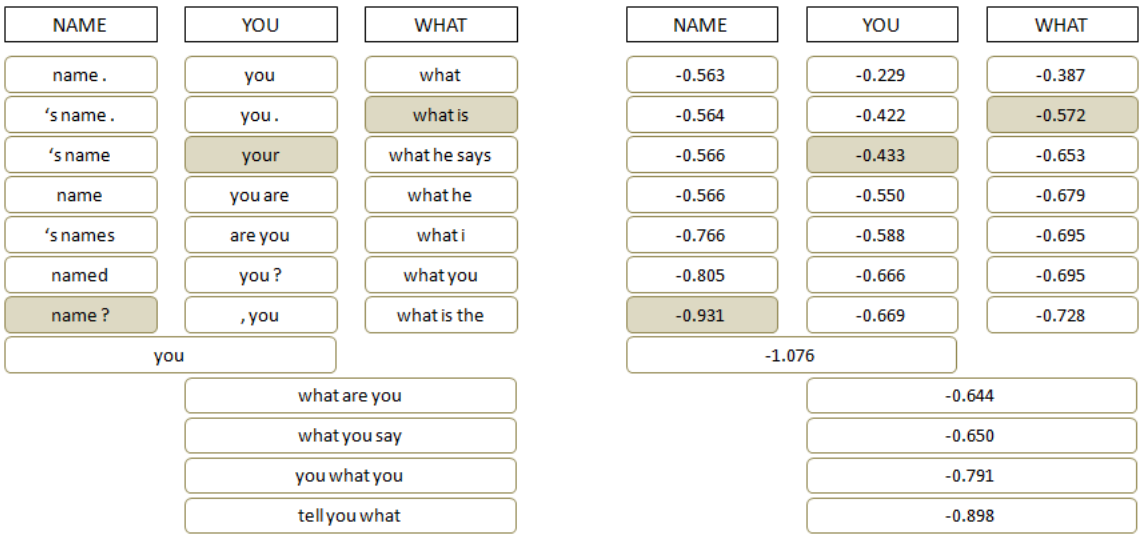


Figure 77. Grille des différentes traductions possibles de la phrase 'NAME YOU WHAT' (à droite) les traductions possibles (à gauche) score en logarithmique

[80] Dans cette étape, on calcule tous les alignements possibles pour chaque.
[81] A partir du modèle du langage généré par l'outil SRI-LM ou IRST-LM, on récupère les valeurs des modèles n-gramme (2 et 3) pour chaque segment de la langue cible.

5. Evaluation

Il est à noter qu'un bon décodeur, c'est-à-dire celui qui propose rapidement la traduction optimale au sens des modèles, n'est appréciable à l'utilisateur que dans la limite où les modèles sont bons. Nous étudions dans ce chapitre le comportement des décodeurs que nous avons implémentés.

L'évaluation humaine est une méthode pour déterminer la performance d'un système de traduction. Les évaluations humaines de la traduction automatique s'intéressent à plusieurs aspects de la traduction, comme l'adéquation, la fidélité, et la maîtrise de la traduction. L'évaluation humaine est très discutée, en témoigne le nombre des travaux sur le sujet [182]. Le problème majeur de l'évaluation humaine est le temps qu'elle nécessite. Elle correspond donc d'avantage à une situation où l'on souhaite évaluer un système stable.

Dans un premier temps, nous détaillons les résultats de la traduction d'un texte en langue des signes Américaine vers un texte en anglais. La Figure 78 présente l'évolution des performances du système au fur et à mesure de l'augmentation de la taille du corpus d'évaluation et d'apprentissage. Les données sont présentées dans le Tableau 16. La métrique utilisée est le score BLEU [74].

Tableau 16. Variation du score BLEU selon la taille du corpus d'évaluation

#evaluation	BLEU	precision				BP	ratio length	hyp_len	ref_len
		1-gram	2-gram	3-gram	4-gram				
1000	31,99	72,1	43,1	28,6	19,6	0,881	0,887	7320	8250
2000	33,62	74,9	45,7	30,7	21,2	0,871	0,878	14293	16272
3000	35,06	75,5	46,9	32,2	22,5	0,875	0,882	21654	24539
4000	36,60	76,1	48,1	33,1	23,2	0,881	0,887	29348	33075
5000	35,97	76,2	47,9	33,0	23,1	0,881	0,887	37112	41823
6000	36,23	76,3	47,9	33,1	23,3	0,884	0,890	44356	49813
7000	35,76	76	47,3	32,4	22,7	0,887	0,893	51841	58054
8000	35,54	75,8	47	32,1	22,4	0,888	0,894	59237	66274
9000	35,61	75,6	46,9	32,2	22,4	0,89	0,896	66675	74444
10000	35,38	75,5	46,7	31,9	22,2	0,89	0,895	74437	83125

Dans une seconde étape, nous avons refait la même opération en utilisant la technique des chaînes similaires basés sur la distance de Jaro-Winkler. Cette technique a augmenté le score BLEU qui reflète la qualité de la traduction comme le montre la Figure 78. Le score BLEU moyen est 35.17 et 46.35 en utilisant la distance Jaro-Winkler.

Tableau 17. Variation du score BLEU selon la taille du corpus d'évaluation en utilisant l'approche des chaînes similaires

#evaluation	BLEU	precision				BP	ratio length	hyp_len	ref_len
		1-gram	2-gram	3-gram	4-gram				
1000	41,19	81,3	52,3	37,8	28,8	0,881	0,887	7320	8250
2000	43,82	85,1	55,9	40,9	31,4	0,871	0,878	14293	16272
3000	45,16	85,6	57	42,3	32,6	0,875	0,882	21654	24539
4000	46,20	85,7	57,7	42,7	32,8	0,881	0,887	29348	33075
5000	47,57	87,8	59,5	44,6	34,7	0,881	0,887	37112	41823
6000	49,73	89,8	61,4	46,6	36,8	0,884	0,890	44356	49813
7000	48,66	88,9	60,2	45,3	35,6	0,887	0,893	51841	58054
8000	48,04	88,3	59,5	44,6	34,9	0,888	0,894	59237	66274
9000	46,61	86,6	57,9	43,2	33,4	0,89	0,896	66675	74444
10000	46,58	86,7	57,9	43,1	33,4	0,89	0,895	74437	83125

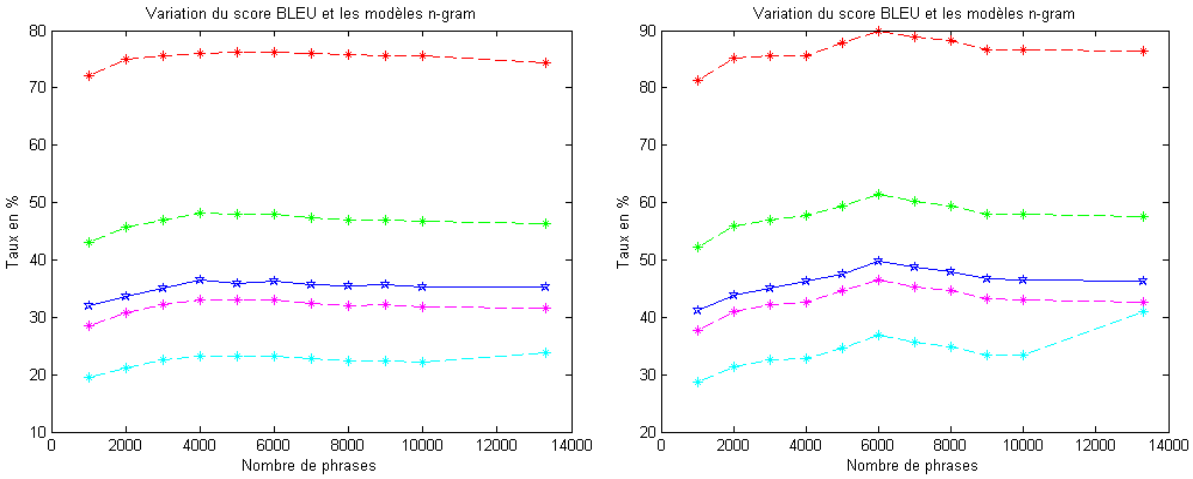


Figure 78. Variation des résultats d'évaluation (bleu) score BLEU (cyan) Modèle 4-gram (magenta) Modèle 3-gram (vert) Modèle 2-gram (rouge) Modèle 1-gram (à gauche) sans utiliser la distance Jaro-Winkler (à droite) en utilisant la distance Jaro-Winkler

6. Conclusion

Au cours de ce chapitre, nous avons implémenté un décodeur pour la traduction automatique statistique alimenté d'une mémoire de traduction à partir des segments extraits d'un corpus Anglais-ASL. Les expériences montrent que le score BLEU de la traduction automatique d'une transcription en ASL vers un texte en anglais (ou inversement) est bon en utilisant la technique des distances de Jaro-Winkler.

Conclusion Générale

Les progrès de la recherche depuis un demi-siècle ont permis à la traduction automatique d'entrer dans nos vies quotidiennes. La traduction automatique rend accessibles tous les jours des millions de pages web aux internautes du monde entier. La traduction des langues de signes est un thème de recherche plus récent, car elle combine ces deux problèmes scientifiques complexes : le premier étant la traduction et le second étant le traitement automatique des langues des signes de point de vue transcription. Mais on imagine sans mal les applications qu'elle rend possibles : systèmes d'interprétations en ligne, accès à l'information, indexation cross-lingue de contenus multimédias, assistant pour l'échange d'informations, l'éducation et l'apprentissage pour les enfants ou personnes souhaitant comprendre les langues des signes, etc.

Les études sur les langues des signes sont récentes et innovantes et au même temps sont nombreux, allant des études sur l'aspect linguistique, cognitif et grammaticales jusqu'à la création des corpus, la traduction automatique et la synthèse en temps réel. Aussi, on peut classifier ses études par pays, puisque les langues des signes ne sont pas universelles et ne partagent pas les mêmes structures syntaxiques, phonologiques, lexicales, morphologiques et sémantiques.

Ceci est un avantage pour nous, pour faire des recherches sur domaine. Cette thèse vise à développer des ressources linguistiques en langue des signes afin de les traiter pour la traduction automatique, soit d'un texte en Anglais, par exemple, vers la langue des signes Américaine, soit de la langue des signes Américaine vers le texte.

Malgré l'existence d'outils de notation et d'annotation, chacun présente des inconvénients. Cependant, pour l'annotation textuelle en glose, ayant plus d'avantage pour un traitement automatique, il n'existe pas une représentation XML détaillé de ce système pour le stockage et l'animation via des avatars signants.

Cette mission n'est pas assez simple puisque les problématiques citées précédemment sont nombreux ou bien plusieurs travaux sont arrivés à des résultats satisfaisants. En effet, pour traitement des SL, on doit toujours passer par une forme écrite pour la représentation, la modélisation et la traduction.

Cette thèse a porté sur la traduction automatique et plus particulièrement sur la traduction de la langue des signes Américaine (ASL) automatiquement. Conformément à l'état de l'art, les systèmes de traduction développés au cours de cette thèse reposent sur des modèles statistiques. Ces modèles sont constitués d'un grand nombre de paramètres de l'ordre de plusieurs millions. Leur entraînement nécessite des textes parallèles : des milliers ou de préférence des millions de phrases traductions l'une de l'autre, grâce auxquelles les paramètres des modèles sont estimés. Ces textes ont été générés automatiquement en utilisant des approches guidés par les graphes de dépendances grammaticales.

Nos recherches ont débuté grâce aux modèles d'alignement IBM appliqué sur les transcriptions en ASL, un modèle à base de mots performant. Ensuite, un décodeur a été

entièrement développé au cours de cette thèse en s'inspirant de la littérature à ce sujet et son fonctionnement a été décrit en détail. L'avènement du décodeur libre Moses au milieu de l'année 2006 eut un impact décisif sur cette thèse. Moses est un décodeur par groupes de mots à l'état de l'art. Il a pu être intégré au cadre expérimental mis en place pour le décodeur précédent et a immédiatement obtenu de très bonnes performances. Cela nous a donné l'opportunité de poursuivre nos recherches avec un autre modèle de traduction.

Nos expériences ont montré que le modèle d'alignement appris en utilisant les distances de Jaro-Winkler améliorait les performances obtenues : entre 8 et 11 point Bleu suivant le sens de traduction.

Pour le rendu final, nous avons proposé un nouveau système d'annotation basé sur la glose, en rapport avec nos besoins spécifiques pour la traduction automatique vers et à partir des langues des signes. Ainsi, dans le cadre de notre travail, nous n'avons pas eu besoin de transcrire l'ASL autrement qu'au moyen des gloses, puisque nous ne nous sommes pas intéressés par la description des gestes (configuration et emplacement) des mains, mais plutôt sur le changement de forme et de configuration dans le contexte.

En effet, nous partons d'une représentation textuelle de la langue des signes Américaine, qui sera notre langue d'étude tout au long de cette thèse, vers une représentation au moyen des gloses, à base du langage "XML". Ce travail a présenté deux contributions essentielles, la première étant la modélisation d'un système de transcription pour l'annotation de la langue des signes, et en deuxième partie la traduction automatique selon deux approches avec une étude comparative avec d'autres travaux.

Comme perspective à cette thèse, l'extension du modèle proposé à d'autres paires de langues et spécialement la langue Arabe qui présente beaucoup de défis allant de la structure grammaticales jusqu'à la sémantique. Et on plus, l'intégration de nouvelles approches d'apprentissage et de génération automatique pour la modélisation des modèles de langues assez efficaces.

Bibliographie

[1] A. Guylhem, «Informatisation d'une forme écrite de la langue des signes française,» Thèse de Doctorat, 2008.

[2] S. C. William, «Sign language structure: an outline of the visual communication system of the American deaf,» *Journal of Deaf Studies and Deaf Education,* pp. 3-37, 2005.

[3] C. S. Bailey, K. Dolby et C. Letourneau, The Canadian Dictionary of Asl, T. U. o. A. Press, Éd., 2002.

[4] M. Filhol, «Modèle descriptif des signes pour un traitement automatique des langues des signes,» 2008.

[5] L. Kervajan, «Contribution à la traduction automatique Français / Langue des Signes Française (LSF) au moyen de personnages virtuels,» Université de Provence, 2011.

[6] A. Risler, «Construction/Déconstruction de l'espace de signation,» chez *12ème conférence annuelle sur le Traitement Automatique des Langues Naturelles*, Dourdan, 2005.

[7] C. Cuxac, «La Langue des Signes Française (LSF) : les voies de l'iconicité,» *L'information grammaticale,* vol. 98, pp. 56-57, 2003.

[8] C. Cuxac et E. A. Pizzuto, «Émergence, norme et variation dans les langues des signes : vers une redéfinition notionnelle,» *Langage et société,* n° %1131, pp. 37-53, 2010.

[9] M. Filhol et A. Braffort, «Description lexicale des signes, Intérêts linguistiques d'un modèle géométrique,» *Traitement Automatique des Langues,* vol. 48, n° %13, pp. 151-177, 2007.

[10] C. Valli et C. Lucas, Linguistics of American Sign Language: An Introduction, Gallaudet University Press, 2000.

[11] H.-Y. Su et C.-H. Wu, «Improving Structural Statistical Machine Translation for Sign Language With Small Corpus Using Thematic Role Templates as Translation Memory,» *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 17, n° %17, pp. 1305-1315, 2009 .

[12] C. Neidle, «SignStream Annotation: Addendum to Conventions used for the American Sign Language Linguistic Research Projec,» 2007.

[13] C. Neidle, D. MacLaughlin, B. Bahan, R. G. Lee et J. Kegl, «The SignStream™ Project,» 1997.

[14] T. J. Castelli, M. Betke et C. Neidle, «Facial Feature Tracking and Occlusion Recovery in American Sign Language,» 2005.

[15] C. Neidle, A. Thangali et S. Sclaroff, «Challenges in Development of the American Sign Language Lexicon Video Dataset (ASLLVD) Corpus,» chez *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon*, 2012.

[16] C. Neidle et C. Vogler, «A new web interface to facilitate access to corpora:,» chez *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon*, 2012.

[17] H. Ebbinghau et J. Heßmann, «Signs and Words. Accounting for Spoken Language Elements in German Sign Language,» *International Review of Sign Linguistics,* vol. 1, pp. 23-56.

[18] T. Hanke, «HamNoSys - representing sign language data in language resources and language processing contexts,» chez *LREC 2004, Workshop proceedings : Representation and processing of sign languages*, 2004.

[19] D. Blanck, T. Hanke, I. Hofmann, S.-E. Hong, O. Jeziorski, T. Kleyboldt, L. König, S. König, R. Konrad, G. Langer, R. Nishio, C. Rathmann, S. Vorwerk et S. Wagner, «The DGS Corpus Project, Development of a Corpus Based Electronic Dictionary German Sign Language-German,» chez *Conference of Theoretical Issues in Sign Language Research*, 2010.

[20] T. Hanke et J. Storz, «iLex – A Database Tool for Integrating Sign Language Corpus Linguistics and Sign Language Lexicography,» chez *The sixth international conference on Language Resources and Evaluation*, 2008.

[21] D. Stein, J. Forster, U. Zelle, P. Dreuw et H. Ney, «RWTH-Phoenix: Analysis of the German Sign Language Corpus,» chez *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010.

[22] E. Auer, A. Russel, H. Sloetjes, P. Wittenburg, O. Schreer, S. Masnieri, D. Schneider et S. Tschöpel, «ELAN as Flexible Annotation Framework for Sound and Image Processing,» chez *Proceedings of the Seventh International Conference on Language Resources and Evaluation LREC'10*, 2010.

[23] M. Abdel-Fattah, «Arabic Sign Language: A Perspective,» *Journal of Deaf Studies and Deaf Education,* vol. 10, n° %12, pp. 212-221, 2005.

[24] A. Schembri, «The British Sign Language Corpus Project,» chez *Fourth UK Council on Deafness Annual Deafness Conference*, 2007.

[25] D. Broeder, F. Offenga, D. Willems et P. Wittenburg, «The IMDI Metadata set, its Tools and accessible Linguistic databases,» chez *Proceedings of the IRCS Workshop on Linguistic Databases*, 2001.

[26] S.-E. Hong, «Agreement verbs in Korean Sign Language (KSL),» chez *Theoretical Issues in Sign Language Research Conference*, 2006.

[27] C. Padden, «The relation between space and grammar in ASL verb morphology,» chez *International Conference on Theoretical Issues in Sign Language Research*, 1990.

[28] W. Janis, «A crosslinguistic perspective on ASL verb agreement,» chez *International Conference on Theoretical Issues in Sign Language Research*, 1995.

[29] C. Rathmann et G. Mathur, «Is verb agreement the same cross-modally ?,» chez *Modality and Structure in Signed and Spoken Languages*, Cambridge University Press, 2002, pp. 370-404.

[30] I. Miyamoto, «Research on Korean Sign Language (part 1),» *Sign Language Communication Studies,* vol. 33, n° %19, pp. 24-27, 1999.

[31] I. Miyamoto, «Research on Korean Sign Language (part 2),» *Sign Language Communication Studies,* vol. 34, n° %19, pp. 49-56, 1999.

[32] Y.-H. Chiu, C.-H. Wu, H.-Y. Su et C.-J. Cheng, «Joint optimization of word alignment and epenthesis generation for Chinese to Taiwanese sign synthesis,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, n° %11, pp. 28-39, 2007.

[33] R. San-Segundo, J. M. Pardo, J. Ferreiros, V. Sama, R. Barra-Chicote, J. M. Lucas, D. Sánchez et A. García, «Spoken Spanish generation from sign language,» *Journal Interacting with Computers,* vol. 22, n° %12, pp. 123-139, 2010.

[34] A. Braffort, A. Choisier, C. Collet, C. Cuxac, P. Dalle, I. Fusellier, R. Gherbi, G. Jausions, G. Jirou, F. Lejeune, B. Lenseigne, N. Monteillard, A. Risler et M.-A. Sallandre, «Projet LS-COLIN. Quel outil de notation pour quelle analyse de la LS ?,» chez *Journées Recherches sur la langue des signes.*, 2001.

[35] C. Cuxac, «Arguments pour une éducation bilingue de l'enfant sourd,» chez *Les rencontres de Saint-Jacques, "La LSF, son enseignement"*, 2000.

[36] J. Bungeroth, D. Stein, P. Dreuw, N. Hermann, S. Morrissey, A. Way et L. v. Zijl, «The ATIS Sign Language Corpus,» chez *International Conference on Language Resources and Evaluation*, 2008.

[37] A. Bernard, F. Encrevé, F. Jeggli et S. Gonzalez, L'interprétation en langue des signes : Français/Langue des signes française, Presses Universitaires de France - PUF, 2007.

[38] A. J. Bangham, S. J. Cox, R. Elliott, J. Glauert, I. W. Marshall, S. Rankov et M. Wells, «Virtual Signing : Capture, Animation, Storage and Transmission - An Overview of the VisiCast Projet,» chez *IEEE seminair on Speech and Language Processing for Disabled and Elderly People*, 2000.

[39] D. Grinberg, J. Lafferty et D. Sleator, «A robust parsing algorithm for link grammars,» chez *Proceedings of the Fourth International Workshop on Parsing Technologies*, 1995.

[40] I. Marshall et E. Safar, «Extraction of Semantic Representations from Syntactic CMU Link Grammar linkages,» chez *Recent Advances in Natural Language Processing (RANLP)*, 2001.

[41] T. Hanke, H. Popescu et C. Schmaling, «eSIGN - HPSG-assisted Sign Language Composition,» chez *The Gesture Workshop 2003*, 2003.

[42] R. Elliott, J. R. W. Glauert, V. Jennings et R. Kennaway, «SiGML Notation and SiGMLSigning Software System,» chez *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, 2004.

[43] R. Kennaway, «Experience and Requirements for a Gesture Description Language for Synthetic Animation,» chez *5th International Workshop on Gesture and Sign Language based Human-Computer Interaction*.

[44] S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, M. Tutt et S. Abbott, «TESSA, a system to aid communication with deaf people,» chez *Fifth International ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2002)*, Edinburgh, Scotland, 2002.

[45] I. Marshall et E. Safar, «The Architecture of an English-Text-to-Sign-Languages Translation System,» chez *Recent Advances in Natural Language Processing (RANLP)*, 2001.

[46] S. Cox, M. Lincoln, M. Nakisa, M. Wells, M. Tutt et S. Abbott, «The Development and Evaluation of a Speech to Sign Translation System to Assist Transactions,» *International Journal of Human Computer Interaction,* vol. 16, n° %12, pp. 141-161, 2003.

[47] M. Huenerfauth, L. Zhao, E. Gu et J. Allbeck, «Evaluating American Sign Language Generation Through the Participation of Nativ ASL Signers,» chez *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, 2007.

[48] M. Huenerfauth, «American Sign Language natural language generation and machine translation,» *ACM SIGACCESS Accessibility and Computing,* vol. 81, pp. 12-15, 2005.

[49] M. Huenerfauth, «A Multi-Path Architecture for Machine Translation of English Text into American Sign Language Animation,» chez *HLT-SRWS '04 Proceedings of the Student Research Workshop at HLT-NAACL 2004*, 2004.

[50] M. Huenerfauth, «Generating American Sign Language Classifier Predicates for English-to-ASL Machine Translation,» 2006.

[51] P. Lu et M. Huenerfauth, «CUNY American Sign Language Motion-Capture Corpus: First Release,» chez *Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, The 8th International Conference on Language Resources and Evaluation (LREC 2012)*, 2012.

[52] L. C. DeWitt, T. T. Lam, D. S. Silverglate, E. Mitford et C. J. Wideman, «Method for animating 3-D computer generated characters». United States Brevet 6535215, 8 Mars 2003.

[53] Y. Chabeb, O. El Ghoul et M. Jemni, «WebSign: A Web-Based Interpreter of Sign Language,» chez *International Conference on Informatics and its Applications*, 2006.

[54] M. Jemni et O. El Ghoul, «A system to make signs using collaborative approach,» chez *Computers Helping People with Special Needs*, 2008.

[55] A. Othman, O. El Ghoul et M. Jemni, «SportSign: A Service to Make Sports News Accessible to Deaf Persons in Sign Languages,» chez *12th International Conference on Computers Helping People with Special Needs*, 2010.

[56] M. Jemni, O. El Ghoul, N. Ben Yahya et M. Boulares, «Sign language MMS to make cell phones accessible to the deaf and hard-of-hearing community,» chez *Conference and Workshop on Assistive Technology for People with Vision and Hearing Impairments (CVHI 2007)*, 2007.

[57] B. Harris, «La traductologie, la traduction naturelle, la traduction automatique et la sémantique,» *Cahier de linguistique,* vol. 2, pp. 133-146, 1973.

[58] A. B. Grieve-Smith, «English to American Sign Language Machine Translation of Weather Reports,» chez *Proceedings of the Second High Desert Student Conference in Linguistics (HDSL2)*, 1999.

[59] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler et M. Palmer, «A machine translation system from English to American Sign Language,» chez *Proceedings of the 4th Conference of the Association for Machine Translation in the Americas on Envisioning Machine Translation in the Information Future*, 2000.

[60] S. M. Shieber, «Restricting the Weak-Generative Capacity of Synchronous Tree-Adjoining Grammars,» *Computational Intelligence,* vol. 10, pp. 371-385, 1994.

[61] L. Zhao, «Synthesis and acquisition of laban movement analysis qualitative parameters for communicative gestures,» 2001.

[62] J. v. Eijck, Discourse Representation Theory, Stichting Mathematisch Centrum, 2005.

[63] R. D. Levine et D. Meurers, «Head-Driven Phrase Structure Grammar: Linguistic Approach, Formal Foundations, and Computational Realization",» chez *Encyclopedia of Language and Linguistics, Second Edition*, Oxford: Elsevier, 2006.

[64] A. L. Speers, «Representation of American Sign Language for Machine Translation,» 2001.

[65] S. K. Liddell, Grammar, gesture, and meaning in American Sign Language, Cambridge University Press, 2003.

[66] B. Joan, Lexical Functional Syntax, Blackwell, 2001.

[67] L. v. Zijl et G. Olivrin, «South African sign language assistive translation,» chez *Proceedings of the IASTED International Conference on Telehealth/Assistive Technologies*, 2008.

[68] C. Fellbaum, «WordNet and wordnets, Second Edition,» chez *Encyclopedia of Language and Linguistics*, Oxford: Elsevier, 2005, pp. 665-670.

[69] R. San Segundo, J. M. Montero, J. M. Guarasa, R. d. Córdoba, J. Ferreiros et J. M. Pardo, «Proposing a Speech to Gesture Translation Architecture for Spanish Deaf People,» *Journal of Visual Languages and Computing,* vol. 19, n° %15, pp. 523-538, 2008.

[70] W. Wayne, «Understanding spontaneous speech: The phoenix system,» chez *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1991.

[71] W. A. Woods, «Transition Network Grammars for Natural Language Analysis,» *Communications of the ACM,* vol. 13, n° %110, pp. 591-606, 1970.

[72] S. Morrissey et A. Way, «Manual labour: tackling machine translation for sign languages,» *Machine Translation,* vol. 27, n° %11, pp. 25-64, 2013.

[73] S. Morrissey, A. Way, D. Stein, J. Bungeroth et H. Ney, «Combining Data-Driven MT Systems for Improved Sign Language,» chez *Machine Translation Summit XI*, 2007.

[74] K. Papineni, S. Roukos, T. Ward et W.-J. Zhu, «BLEU: a method for automatic evaluation of machine translation,» chez *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002.

[75] W. C. Stokoe, «Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf (reprint),» *Journal of Deaf Studies and Deaf Education,* vol. 10, n° %11, pp. 3-37, 2005 (1960).

[76] W. C. Stokoe, Sign Language Structure: The First Linguistic Analysis of Amercian Sign Language, Linstok Press; Revised edition, 1978.

[77] P. Dreuw, C. Neidle, V. Athitsos, S. Sclaroff et H. Ney, «Benchmark Databases for Video-Based Automatic Sign Language Recognition,» chez *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, 2008.

[78] K. Jaballah et M. Jemni, «Toward Automatic Sign Language Recognition from Web3D Based Scenes,» chez *12th International Conference on Computers Helping People with Special Needs*, 2010.

[79] E. Chételat-Pelé, «Les Gestes Non Manuels en Langue des Signes Française ; Annotation, analyse et formalisation : application aux mouvements des sourcils et aux clignements des yeux,» 2010.

[80] D. C. Baynton, Forbidden Signs: American Culture and the Campaign against Sign Language, University Of Chicago Press, 1998.

[81] K. Emmorey, Language, Cognition, and the Brain: Insights from Sign Language Research, Routledge, 2002.

[82] C. J. Neidle, The Syntax of American Sign Language: Functional Categories and Hierarchical Structure, MIT Press, 2000 .

[83] S. K. Liddell et R. E. Johnson, «American Sign Language: The Phonological Base,» *Sign Language Studies,* vol. 64, pp. 195-278, 1989.

[84] W. Sandler, Sign Language and Linguistic Universals, Cambridge University Press, 2006.

[85] P. Jouison, «Analysis and linear transcription of sign language discourse,» chez *Current trends in European sign language research. Proceedings of the 3rd European Congress on Sign Language Research*, 1990.

[86] R. v. Laban et L. Roderyk, Laban's Principles of Dance and Movement Notation, Macdonald & Evans, 1975 .

[87] B. Farnell, «Movement notation systems,» chez *The World's Writing Systems*, 1996, pp. 855-879.

[88] D. Newkirk, Signfont Handbook, Edmark Corp, 1987.

[89] T. Stuart, «A grammar of SignWriting,» 2011.

[90] T. Johnston, «From archive to corpus: transcription and annotation in the creation of signed language corpora Department of Linguistics,» *International journal of corpus linguistics,* vol. 15, n° %11, pp. 106-131, 2010.

[91] E. S. Klima, The Signs of Language, MA: Harvard University Press, 1979.

[92] O. Crasborn, H. Sloetjes, E. Auer et P. Wittenburg, «Combining video and numeric data in the analysis of sign languages within the ELAN annotation software,» chez *Proceedings of the 2nd Workshop on the Representation and Processing of Sign languages: Lexicographic matters and didactic scenarios*, 2006.

[93] A. B. Grieve-Smith, «SignSynth: A Sign Language synthesis application using Web3D and Perl,» chez *Gesture and Sign Language in Human-Computer Interaction*, 2002.

[94] C. Steinback, «Perception of Sign Language and Its Application to Visual Communications for Deaf People,» *Journal of Deaf studies and Deaf education,* vol. 10, n° %14, pp. 390-401, 2005.

[95] C. Dubuisso et L. Lacerte, «La glose dans la transcription et l'analyse des langues signée,» *Spécificités de la recherche linguistique sur les langues signées, Montréal: Les cahiers scientifiques de l'Acfas,* 1996.

[96] A. Lonjon et J.-J. Thomasson, Modélisation XML, Édition Eyrolles, 2006 .

[97] M. Richard, «Person deixis in American Sign Language,» chez *Theoretical Issues in Sign Language Research*, University of Chicago Press, 1990, pp. 175-190.

[98] S. Fischer, «Two processes of reduplication in American Sign Language,» *Foundations of Language,* vol. 9, n° %14, p. 469–480, 1973..

[99] S. K. Liddell, «Unrealized-inceptive aspect in American Sign Language: Feature insertion in syllabic frames,» *Papers from the 20th Regional Meeting of the Chicago Linguistic Society,* p. 257–270, 1984.

[100] S. K. Liddel, «An investigation into the syntactic structure of American Sign Language,» 1977.

[101] T. Supalla, «Morphology of verbs of motion and location in American Sign Language,» chez *Proceedings of the Second National Symposium of Sign Language Research and Teaching*, 1978.

[102] M. Huenerfauth, «Spatial Representation of Classifier Predicates for Machine Translation into American Sign Language,» chez *Workshop on Representation and Processing of Sign Language, 4th Internationnal Conference on Language Ressources and Evaluation*, 2004.

[103] M. Jemni et O. EL Ghoul, «An avatar based approach for automatic interpretation of text to Sign language,» chez *9th European Conference for the Advancement of the Assistive Technologies in Europe*, 2007.

[104] U. Bellugi, P. Howard et S. K. Edward, «Brain organization for language: Clues from sign aphasia,» *Human Neurobiology,* vol. 2, pp. 155-170, 1983.

[105] E. Safar et I. Marshall, «The Architecture of an English-Text-to-Sign-Languages Translation System,» chez *Recent advances in natural language processing (RANLP)*, 2001.

[106] C. Padden, «Interaction of Morphology and Syntax in American Sign Language,» 1983.

[107] T. Johnston et A. Schembri, Australian Sign Language: An Introduction to Sign Language Linguistics, Cambridge University Press, 2007.

[108] R. d. Quadros et J. Quer, «Back to Back(wards) and Moving on: On Agreement, Auxiliaries and Verb Classes,» chez *The 9th Theoretical Issues in Sign Language Research Conference*, 2008.

[109] R. Sutton-Spence et B. Woll, The Linguistics of British Sign Language: An Introduction, Cambridge University Press, 1999.

[110] C. Rathmann, «The Optionality of Agreement Phrase: Evidence from Signed Languages,» 2000.

[111] S. Fischer, «The Role of Agreement and Auxiliaries in Sign Languages,» *Sign Linguistics Phonetics, Phonology and Morpho-syntax,* vol. 98, n° %13, pp. 103-119, 1996.

[112] G. G. Corbett, Agreement, Cambridge University Press, 2006.

[113] D. Lillo-Martin et E. Klima, «Pointing out Differences: ASL Pronouns in Syntactic Theory.,» *Theoretical Issues in Sign Language Research,* vol. 1, pp. 191-210, 1990.

[114] T. Supalla, «The Classifier System in American Sign Language,» *C. Craig (Ed.), Noun Classification and Categorization. J. Benjamins,* pp. 18-214, 1986.

[115] L. A. Friedman, «Space, Time, and Person Reference in American Sign Language,» *Language,* vol. 51, n° %14, pp. 940-961, 1975.

[116] D. Bouchard, «Sign Languages and Language Universals: The Status of Order and Position in Grammar,» *Sign Language Studies,* vol. 91, pp. 101-160, 1997.

[117] J. H. Greenberg, Universals of Language, Second Edition, Cambridge, MA: MIT Press, 1966.

[118] K. Hale, «Warlpiri and the Grammar of Non-configurational Languages,» *Natural Language and Linguistic Theory,* vol. 1, pp. 5-47, 1983.

[119] R. Pensalfini, A Grammar of Jingulu: An Aboriginal Language of the Northern Territory, Pacific linguistics, Research School of Pacific and Asian Studies, Australian National University, 2003.

[120] B. Hendriks, «Negation in Jordanian Sign Language: A Cross-linguistic Perspective,» *Trends in Linguistics Studies and Monographs,* vol. 188, pp. 103-128, 2007.

[121] L. Yifan, M. Petr, R. Marek et W.-S. Loren, «Identification of Pleonastic It Using the Web,» *Journal of Artificial Intelligence Research,* vol. 34, pp. 339-389, 2009.

[122] P. Ekman, «Handbook of Cognition and Emotion,» chez *Handbook of Cognition and Emotion*, 1999, pp. 45-60.

[123] A. Bhatara, E.-M. Quintin, B. Levy, U. Bellugi, E. Fombonne et D. J. Levitin, «Perception of emotion in musical performance in adolescents with Autism spectrum disorder,» *Autism Research,* vol. 3, pp. 214-225, 2010.

[124] B. Vauquois et C. Boitet, «Automated translation at Grenoble University,» *Journal Computational Linguistics - Special issues on machine translation,* vol. 11, n° %11, pp. 28-36, 1985.

[125] K. Toutanova et C. D. Manning, «Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger,» chez *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, 2000.

[126] M. A. Covington, «A Fundamental Algorithm for Dependency Parsing,» chez *Proceedings of the 39th annual ACM southeast conference*, 2001.

[127] S. P. Abney, «A computational model of human parsing,» *Journal of Psycholinguistic Research,* vol. 18, p. 129–144, 1989.

[128] M. Collins, «Head-Driven Statistical Models for Natural Language Parsing,» *Computational linguistics,* vol. 29, n° %14, pp. 589-637, 2003.

[129] A. K. Joshi et Y. Schabes, «Tree-Adjoining Grammars and Lexicalized Grammars,» Department of Computer & Information Science, 1991.

[130] G. E. Barton, R. C. Berwick et E. S. Ristad, Computational Complexity and Natural Language, MIT Press Cambridge, 1987.

[131] M.-C. d. Marneffe, B. MacCartney et C. D. Manning, «Generating Typed Dependency Parses from Phrase Structure Parses,» chez *Proceeding of The fifth international conference on Language Resources and Evaluation, LREC*, 2006.

[132] D. Nadeau et S. Sekine, «A survey of named entity recognition and classification,» *Lingvisticae Investigationes,* vol. 30, n° %11, pp. 3-26, 2007.

[133] K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky et C. Manning, «A Multi-Pass Sieve for Coreference Resolution,» chez *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010.

[134] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer et P. S. Roossin, «A statistical approach to machine translation,» *Computational Linguistics,* vol. 16, n° %12, pp. 79-85 , 1990.

[135] A. L. Berger, P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, J. R. Gillett, J. D. Lafferty, R. L. Mercer, H. Printz et L. Ureš, «The Candide system for machine translation,» chez *Proceedings of the workshop on Human Language Technology HLT'94*, 1994.

[136] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra et R. L. Mercer, «The mathematics of statistical machine translation: parameter estimation,» *Computational Linguistics - Special issue on using large corpora: II,* vol. 19, n° %12, pp. 263-311, 1993 .

[137] F. J. Och, «Statistical Machine Translation: From Single-Word Models to Alignment Templates,» 2002.

[138] R. Zens, F. J. Och et H. Ney, «Phrase-Based Statistical Machine Translation,» chez *Annual German Conference on AI : Advances in Artificial*, 2002.

[139] A. Viterbi, «Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,» *IEEE Transactions on Information Theory,* vol. 13, n° %12, p. 260–269, 1967.

[140] F. J. Och et H. Ney, «A systematic comparison of various statistical alignment models,» *Computational linguistics,* vol. 29, n° %11, pp. 19-51, 2003.

[141] P. Koehn, F. J. Och et D. Marcu, «Statistical phrase-based translation,» chez *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 2003.

[142] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin et D. Radev, «A smorgasbord of features for statistical machine translation,» chez *Proceedings of the 2004 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 2004.

[143] D. Marcu et W. Wong, «A phrase-based, joint probability model for statistical machine translation,» chez *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 2002.

[144] F. J. Och, «Minimum error rate training in statistical machine translation,» chez *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, 2003.

[145] M. P. Marcus, M. A. Marcinkiewicz et B. Santorini, «Building a large annotated corpus of English: The Penn Treebank,» *Computational linguistics,* vol. 19, n° %12, pp. 313-330, 1993.

[146] D. Gillick, «Sentence boundary detection and the problem with the US.,» chez *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

[147] F. Béchet, A. Nasr, T. Spriet et R. d. Mori, «Modèles de langage à portée variable : Application au traitement des homophones,» chez *Actes de la 6ème conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, 1999.

[148] E. Hasler, B. Haddow et P. Koehn, «Margin infused relaxed algorithm for moses,» *The Prague Bulletin of Mathematical Linguistics,* vol. 96, n° %11, pp. 69-78, 2011.

[149] H. P. Krings, Repairing texts: empirical investigations of machine translation post-editing processes, Vol. 5 éd., Kent State University Press, 2001.

[150] J. Allen et C. Hogan, «Toward the development of a post editing module for raw machine translation output : A controlled language perspective,» chez *Third International Controlled Language Applications Workshop (CLAW-00)*, 2000.

[151] A. Othman, Z. Tmar et M. Jemni, «Toward developing a very big Sign Language Parallel Corpus,» chez *The 13th International Conference on Computers Helping People with Special Needs 'ICCHP', Lecture Note in Computer Science LNCS Springer*, 2012.

[152] Z. Tmar, A. Othman et M. Jemni, «A Rule-Based Approach for Building an Artificial English-ASL Corpus,» chez *International Conference on Electrical Engineering and Software Applications ICEESA*, 2013.

[153] A. Othman et M. Jemni, «English-ASL Gloss Parallel Corpus 2012: ASLG-PC12,» chez *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC'12*, 2012.

[154] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin et E. Herbst, «Moses: Open Source Toolkit for Statistical Machine Translation,» chez *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007.

[155] A. Othman et M. Jemni, «A probabilistic model for Sign Language Translation Memory,» chez *Proceedings of the International Symposium on Intelligent Informatics ISI'12*, 2012.

[156] F. J. Och et H. Ney, «Improved statistical alignment models,» chez *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 2000.

[157] Q. Gao et S. Vogel, «Parallel implementations of word alignment tool,» chez *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, 2008.

[158] A. Stolcke, «Srilm-an extensible language modeling toolkit,» chez *Seventh International Conference on Spoken Language Processing*, 2002.

[159] M. Federico, N. Bertoldi et M. Cettolo, «Irstlm : an open source toolkit for handling large scale language models,» chez *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[160] L. Specia, «Exploiting objective annotations for measuring translation post-editing effort,» chez *Proceedings of the 15th Conference of the European Association for Machine Translation*, 2011.

[161] C. Callison-Burch, «ast, cheap, and creative : Evaluating translation quality using amazon's mechanical turk,» chez *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009.

[162] C. Callison-Burch, M. Osborne et P. Koehn, «Re-evaluating the Role of BLEU in Machine Translation Research,» chez *11th Conference of the European Chapter of the Association for Computational Linguistics EACL*, 2006.

[163] S. Banerjee et A. Lavie, «METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments,» chez *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.

[164] G. Doddington, «Automatic evaluation of machine translation quality using n-gram co-occurrence statistics,» chez *Proceedings of the second international conference on Human Language Technology Research*, 2002.

[165] E. Macklovitch et R. Graham, «What's been forgotten in translation memory,» chez *Envisioning Machine Translation in the Information Future*, 2000.

[166] M. Simard, «Mémoires de traduction sous-phrastiques,» 2003.

[167] A. P. Dempster, M. L. Nan et D. B. Rubin, «Maximum Likelihood from Incomplete Data via the EM Algorithm,» *Journal of the Royal Statistical Society. Series B (Methodological),* vol. 39, n° %11, pp. 1-38, 1977.

[168] M. A. Jaro, «Advances in record linking methodology as applied to the 1985 census of Tampa Florida,» *Journal of the American Statistical Society,* vol. 84, n° %1406, pp. 414-420, 1989.

[169] W. E. Winkler, «The state of record linkage and current research problems,» 1994.

[170] A. Othman et M. Jemni, «Statistical Sign Language Machine Translation: from English written text to American Sign Language Gloss,» *International Journal of Computer Science Issues,* vol. 8, n° %15, pp. 65-73, 2011.

[171] S. Russell et P. Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2009.

[172] P. Koehn, Statistical Machine Translation, Cambridge University Press, 2010.

[173] R. Zens et H. Ney, «Improvements in Phrase-Based Statistical Machine Translation,» chez *Proceedings of the Human Language Technology Conference HLT-NAACL*, 2004.

[174] P. Spirtes, R. Thomas et C. Meek, «Heuristic greedy search algorithms for latent variable models,» chez *Proceedings of AI & STAT'97*, 1997.

[175] A. Rousseau, «La traduction automatique de la parole,» 2012.

[176] M. Federico et D. M. Renato, «Language modelling,» *Spoken Dialogues with Computers,* pp. 199-230, 1998.

[177] S. F. Chen et G. Joshua, «An empirical study of smoothing techniques for language modeling,» chez *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, 1996.

[178] S. Katz, «Estimation of probabilities from sparse data for the language model component of a speech recognizer,» *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 35, n° %13, pp. 400 - 401, 1987.

[179] H. Hoang et P. Koehn, «Design of the Moses Decoder for Statistical Machine Translation,» chez *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, 2008.

[180] A. Arun, B. Haddow et P. Koehn, «A Unified Approach to Minimum Risk Training and Decoding,» chez *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, 2010.

[181] W. Xu et P. Koehn, «Extending Hiero Decoding in Moses with Cube Growing,» *The Prague Bulletin of Mathematical Linguistics,* vol. 98, n° %11, pp. 133-142, 2012.

[182] «The ARPA MT evaluation methodologies: evolution, lessons, and future approaches,» chez *Proceedings of the 1994 Conference, Association for Machine Translation in the Americas*, 1994.

Annexe A : Configuration des mains en ASL

Les configurations des mains de langue des signes sont spécifiées par les attributs distincts de l'avant-bras, les doigts et le pouce[82]. Ils sont utilisés pour la distinction entre les signes de n'importe quelle langue visio-gestuelles. Le tableau suivant contient toutes les configurations des mains avec une description pour la langue des signes Américaine.

Configuration	Image
1	
4	
6	
Open8	
A	
BentB	
C	
D	
I	
K	
N	
FlatO	
S	
BentV	
Y	

Configuration	Image
Bent3	
Claw 5	
7	
9	
OpenA	
OpenB	
FlatC	
E	
H	
OpenM	
OpenN	
SmallO	
T	
X	
ILY	

Configuration	Image
3	
5	
8	
Flat9	
B	
FlatB	
Small-C	
G	
L	
M	
O	
R	
V	
OpenX	
Corna	

[82] Liste des configurations des mains en langue des signes Américaines extraites du site web : [http://en.wiktionary.org/wiki/Appendix:Sign_language_handshapes]

Annexe B : Schéma XSD du modèle d'annotation en gloses

Une version disponible pour téléchargement sur le site :
[http://www.achrafothman.net/aslsmt/schemaXML.xsd]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns="C:\Users\java\Desktop\ASL"
attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="C:\Users\java\Desktop\ASL" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!-- Déclarations de types simples -->
<xsd:simpleType name="type_sen">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="neg"/>
<xsd:enumeration value="wh_q"/>
<xsd:enumeration value="q"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="type-clause">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="cond"/>
<xsd:enumeration value="direction"/>
<xsd:enumeration value="dur"/>
<xsd:enumeration value="t"/>
<xsd:enumeration value="hn"/>
<xsd:enumeration value="rh"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="type-tk">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="pro1st"/>
<xsd:enumeration value="poss1st"/>
<xsd:enumeration value="pro1st-pl"/>
<xsd:enumeration value="poss1st-pl"/>
<xsd:enumeration value="pro2nd"/>
<xsd:enumeration value="poss2nd"/>
<xsd:enumeration value="pro3rd-he"/>
<xsd:enumeration value="pro3rd-she"/>
<xsd:enumeration value="pro3rd-it"/>
<xsd:enumeration value="pro3rd-they"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:annotation>
<xsd:documentation>Schéma de définition de l'ASL</xsd:documentation>
</xsd:annotation>
<xsd:element name="bloc">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="sentence" minOccurs="1" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="sentence">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="token" type="tokens" minOccurs="1" maxOccurs="unbounded"/>
<xsd:element name="clause" type="clause" minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="lang" type="xsd:string" use="required" default="asl"/>
<xsd:attribute name="src_lang" type="xsd:language" use="required" default="en"/>
<xsd:attribute name="src_sentence" type="xsd:string" use="required"/>
<xsd:attribute ref="type-sen" type="xsd:string" use="optional"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="tokens">
```

```
<xsd:complexType>
<xsd:sequence>
<xsd:element name="token" type="xsd:string" minOccurs="0"maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="id" type="xsd:string" use="optional"/>
<xsd:attribute ref="type-tk" use="optional"/>
<xsd:attribute name="fs" type="xsd:boolean" default="0" use="optional"/>
<xsd:attribute name="idref" type="xsd:string" use="optional"/>
</xsd:complexType>
</xsd:element>
 <xsd:element name="clause">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="tokens" minOccurs="1" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute ref="type-cl" use="required"/>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Annexe C : Feuille de styles XSLT du système de transcription en gloses

Une version disponible pour téléchargement sur le site :
[http://www.achrafothman.net/aslsmt/formexsl.xsl]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output indent="yes" method="html"/>
<xsl:template match="/">
<html><head>
<title>ASL</title>
<style type="text/css">
.blackText {font-family:arial;color:#000000; }
.largeYellowText {font-family:arial;font-size:14pt;color:#ffff00;}
.largeBlackText {font-family:arial;font-size:14pt;color:#000000;}
.borders {border-left:1px solid #000;border-right:1px solid #000;border-top:1px solid
#000;border-bottom:1px solid #000;}
</style>
</head>
<body><span><b>American Sign Language </b></span><p/>
<xsl:apply-templates/></body></html>
</xsl:template><xsl:template match="bloc">
<xsl:for-each select="sentence"><br/>
<div style="padding:20px; border:1px solid #CCCCCC;">
<div><xsl:value-of select="@src_sentence" /> (emotion: <xsl:value-of select="@emotion"
/>)(sentiment: <xsl:value-of select="@sentiment" />)</div>
<div style="margin:20px; padding:20px;">
<xsl:for-each select="clause">
<span style="border-top:2px solid #000000;">
<xsl:for-each select="token[@compound]">
<xsl:choose>
<xsl:when test="../@type_clause='none'">
<xsl:variable name="i" select="number(1)"/>
<span style="border-top:3px solid #FFFFFF;">
<xsl:value-of select=" (child::token)[$i]"/>
<xsl:value-of select="./@compound"/>
</span>
<xsl:if test="child::token [$i+1]!=child::token[last()]">
<span style="border-top:3px solid #FFFFFF;">
<xsl:value-of select=" (child::token)[$i+1]"/>
<xsl:value-of select="./@compound"/></span>
</xsl:if>
<xsl:if test="child::token [$i+2]!=child::token[last()]">
<span style="border-top:3px solid #FFFFFF;">
<xsl:value-of select=" (child::token)[$i+2]"/><xsl:value-of select="./@compound"/></span>
</xsl:if>
<span style="border-top:3px solid #FFFFFF;">
<xsl:value-of select="( child::token)[last()]"/></span>
</xsl:when>
 <xsl:when test="../@type_clause!='none'">
<span style="padding-left:15px;">
<xsl:variable name="i" select="number(1)"/>
<xsl:value-of select=" (child::token)[$i]"/><xsl:value-of select="./@compound"/>
<xsl:if test="child::token [$i+1]!=child::token[last()]">
<xsl:value-of select=" (child::token)[$i+1]"/><xsl:value-of select="./@compound"/>
</xsl:if>
<xsl:if test="child::token [$i+2]!=child::token[last()]">
<xsl:value-of select=" (child::token)[$i+2]"/><xsl:value-of select="./@compound"/>
</xsl:if>
<xsl:value-of select="(child::token)[last()]"/></span>
</xsl:when></xsl:choose></xsl:for-each>
<xsl:for-each select="token[not(@compound)]">
```

```
<xsl:if test="./@fs">
<xsl:choose>
<xsl:when test="./@fs='yes' and ../@type_clause='none'">
 <span style="border-top:3px solid #FFFFFF; margin:-13px; padding-left:10px; font-
weight:bold;">#</span>
</xsl:when>
<xsl:when test="./@fs='yes'">
<span style="position:relative; left:12px; font-weight:bold;">#</span>
</xsl:when></xsl:choose></xsl:if>
<xsl:if test="./@type_tk='AGE0'">
<xsl:choose>
<xsl:when test="./@type_tk='AGE0' and ../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF; margin:-7px; font-weight:bold;">{AGE0}-
</span></xsl:when>
<xsl:when test="./@type_tk='AGE0'">
<span style="position:relative; left:12px; font-weight:bold;">{AGE0}-</span>
</xsl:when></xsl:choose></xsl:if>
<xsl:if test="./@numeral">
<xsl:choose>
<xsl:when test="./@numeral='yes' and ../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF; margin:-5px; padding-left:10px; font-
weight:bold;">{</span>
</xsl:when>
<xsl:when test="./@numeral='yes'">
<span style="position:relative; left:12px; font-weight:bold;">{</span>
</xsl:when></xsl:choose></xsl:if>
<xsl:if test="./@classifier">
<xsl:choose>
<xsl:when test="./@classifier and ../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF; margin:-5px; padding-left:10px; font-
weight:bold;">CL:<xsl:value-of select="@classifier" /></span>
</xsl:when>
<xsl:when test="./@classifier">
<span style="position:relative; left:12px; font-weight:bold;">CL:<xsl:value-of
select="@classifier" /></span>
</xsl:when></xsl:choose></xsl:if>
<xsl:if test="./@two_h">
<xsl:choose>
<xsl:when test="./@two_h and ../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF; margin:-5px; padding-left:10px; font-
weight:bold;">2h:</span>
</xsl:when><xsl:when test="./@two_h">
<span style="position:relative; left:12px; font-weight:bold;">2h:</span>
</xsl:when></xsl:choose></xsl:if>
<xsl:if test="./@one_h">
<xsl:choose>
<xsl:when test="./@one_h and ../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF; margin:-5px; padding-left:10px; font-
weight:bold;">1h:</span>
</xsl:when>
<xsl:when test="./@one_h">
<span style="position:relative; left:12px; font-weight:bold;">1h:</span>
</xsl:when></xsl:choose></xsl:if>
<xsl:if test="./@hs">
<xsl:choose><xsl:when test="./@hs and ../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF; margin:-5px; padding-left:10px; font-
weight:bold;">HS:<xsl:value-of select="@hs" /></span>
</xsl:when><xsl:when test="./@hs">
<span style="position:relative; left:12px; font-weight:bold;">HS:<xsl:value-of select="@hs"
/></span>
</xsl:when></xsl:choose></xsl:if>
<xsl:if test="./@description">
<xsl:choose>
<xsl:when test="./@description and ../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF; margin:-5px; padding-left:10px; font-
weight:bold;">'</span>
```

```
</xsl:when>
<xsl:when test="./@description">
<span style="position:relative; left:12px; font-weight:bold;">'</span>
</xsl:when></xsl:choose></xsl:if>
<xsl:if test="./@type_tk and ../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF;">
<xsl:choose>
<xsl:when test="./@type_tk='pro1st'">I </xsl:when>
<xsl:when test="./@type_tk='poss1st'">MY</xsl:when>
<xsl:when test="./@type_tk='pro1st_pl' ">WE</xsl:when>
<xsl:when test="./@type_tk='poss1st_pl' ">OUR</xsl:when>
<xsl:when test="./@type_tk='pro2nd' ">YOU</xsl:when>
<xsl:when test="./@type_tk='poss2nd' ">YOUR</xsl:when>
<xsl:when test="./@type_tk='pro3rd_he' ">HE</xsl:when>
<xsl:when test="./@type_tk='pro3rd_she' ">SHE</xsl:when>
<xsl:when test="./@type_tk='pro3rd_it' ">IT</xsl:when>
<xsl:when test="./@type_tk='pro3rd_they' ">THEY</xsl:when>
<xsl:otherwise>
</xsl:otherwise></xsl:choose></span></xsl:if>
<xsl:if test="./@type_tk and ../@type_clause!='none'">
<span>
<xsl:choose>
<xsl:when test="./@type_tk='pro1st'">I</xsl:when>
<xsl:when test="./@type_tk='poss1st'">MY</xsl:when>
<xsl:when test="./@type_tk='pro1st_pl' ">WE</xsl:when>
<xsl:when test="./@type_tk='poss1st_pl' ">OUR</xsl:when>
<xsl:when test="./@type_tk='pro2nd' ">YOU</xsl:when>
<xsl:when test="./@type_tk='poss2nd' ">YOUR</xsl:when>
<xsl:when test="./@type_tk='pro3rd_he' ">HE</xsl:when>
<xsl:when test="./@type_tk='pro3rd_she' ">SHE</xsl:when>
<xsl:when test="./@type_tk='pro3rd_it' ">IT</xsl:when>
<xsl:when test="./@type_tk='pro3rd_they' ">THEY</xsl:when>
<xsl:otherwise></xsl:otherwise>
</xsl:choose></span></xsl:if>
<xsl:if test="../@type_clause">
<xsl:choose>
<xsl:when test="../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF;padding-left:15px;"><xsl:value-of select="." />
<xsl:if test="./@id">
<sup style="padding:5px; position:relative; top:15px;right:3px;"><xsl:value-of select="@id"
/></sup></xsl:if>
<xsl:if test="./@idref">
<sup style="padding:5px; position:relative; bottom:1px;right:3px;"><img src="right.png"
/><xsl:value-of select="@idref" /></sup>
</xsl:if></span></xsl:when>
<xsl:otherwise>
<span style="padding-left:15px;"><xsl:value-of select="." />
<xsl:if test="./@id">
<xsl:choose><xsl:otherwise>
<sup style="padding:5px; position:relative; top:15px;right:3px;"><xsl:value-of select="@id"
/></sup>
</xsl:otherwise></xsl:choose></xsl:if>
<xsl:if test="./@idref"><xsl:choose><xsl:otherwise>
<sup style="padding:5px; position:relative; bottom:1px;right:3px;"><img src="right.png"
/><xsl:value-of select="@idref" /></sup>
</xsl:otherwise></xsl:choose></xsl:if></span></xsl:otherwise></xsl:choose></xsl:if>
<xsl:if test="./@numeral"><xsl:choose>
<xsl:when test="../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF; margin:-7px; padding-left:10px; font-
weight:bold;">}</span></xsl:when>
<xsl:otherwise>
<span style="position:relative; left:12px; font-weight:bold;">}</span>
</xsl:otherwise></xsl:choose></xsl:if>
<xsl:if test="./@duplication">
<xsl:choose><xsl:when test="../@type_clause='none'">
```

```
<span style="border-top:3px solid #FFFFFF; margin:-7px; padding-left:3px; font-
weight:bold;">+</span></xsl:when>
<xsl:otherwise>
<span style="position:relative; left:3px; font-weight:bold;">+</span>
</xsl:otherwise></xsl:choose></xsl:if>
<xsl:if test="./@property">
<xsl:choose><xsl:when test="../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF;padding-left:15px;">
<sup style="padding:5px; position:relative; bottom:8px;right:3px;">[<xsl:value-of
select="@property" />]</sup>
</span></xsl:when><xsl:otherwise><sup style="padding:5px; position:relative;
top:15px;right:3px;">[<xsl:value-of select="@property" />]</sup>
</xsl:otherwise></xsl:choose></xsl:if>
<xsl:if test="./@description">
<xsl:choose><xsl:when test="../@type_clause='none'">
<span style="border-top:3px solid #FFFFFF; margin:-5px; padding-left:10px; font-
weight:bold;">'</span></xsl:when>
<xsl:otherwise>
<span style="position:relative; left:3px; font-weight:bold;">'</span>
</xsl:otherwise></xsl:choose></xsl:if></xsl:for-each>
<xsl:if test="./@type_clause">
<xsl:choose><xsl:when test="./@type_clause='t'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">t</sup>
</xsl:when><xsl:when test="./@type_clause='cs'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">cs</sup>
</xsl:when><xsl:when test="./@type_clause='cond'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">cond</sup>
</xsl:when><xsl:when test="./@type_clause='rh'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">rh</sup>
</xsl:when><xsl:when test="./@type_clause='hn'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">hn</sup>
</xsl:when><xsl:when test="./@type_clause='mm'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">mm</sup>
</xsl:when><xsl:when test="./@type_clause='puff'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">puff</sup>
</xsl:when><xsl:when test="./@type_clause='neg'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">neg</sup>
</xsl:when><xsl:when test="./@type_clause='sta'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">sta</sup>
</xsl:when><xsl:when test="./@type_clause='int'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">int</sup>
</xsl:when><xsl:when test="./@type_clause='ps'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">ps</sup>
</xsl:when><xsl:when test="./@type_clause='th'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">th</sup>
</xsl:when><xsl:when test="./@type_clause='cha'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">cha</sup>
</xsl:when><xsl:when test="./@type_clause='pah'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">pah</sup>
</xsl:when><xsl:when test="./@type_clause='pow'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">pow</sup>
</xsl:when><xsl:when test="./@type_clause='wh_q'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">wh_q</sup>
</xsl:when><xsl:when test="./@type_clause='q'">
<sup style="padding:5px; position:relative; bottom:15px;right:3px;">q</sup>
</xsl:when></xsl:choose></xsl:if></span></xsl:for-each></div></div></xsl:for-
each></xsl:template></xsl:stylesheet>
```

Annexe D : Liste des relations de dépendances grammaticales

- **01. abbrev** - abréviation: Un modificateur d'abréviation d'un NP est une NP avec parenthèses qui sert à abréger le NP.
  "*The Australian Broadcasting Corporation (ABC)*" → **abbrev**(*Corporation*, *ABC*)
- **02. acomp** - complément adjectival: Un complément adjectival du verbe est une phrase adjectivale qui fonctionne comme le complément.
  "*She looks very beautiful*" → **acomp**(*looks*, *beautiful*)
- **03. advcl** - clause adverbiale: Une clause adverbiale d'un VP ou d'un S est une clause modifiant le verbe (clause temporelle, conséquence, clause conditionnelle, etc.).
  "*The accident happened as the night was falling*" → **advcl**(*happened*, *falling*)
- **04. advmod** - modificateur adverbial: il sert à modifier le sens du mot.
  "*Genetically modified food*" → **advmod**(*modified*, *genetically*)
- **05. agent** - agent : Un agent est le complément d'un verbe passif qui est introduit par la préposition "by".
  "*The man has been killed by the police*" → agent (killed, police)
- **06. amod** - modificateur d'adjectif: Un modificateur d'adjectif d'un NP est une phrase adjectivale qui sert à modifier le sens de la NP.
  "*Sam eats red meat*" → amod(*meat*, *red*)
- **07. appos** - modificateur appositionnel : Un modificateur d'apposition d'un NP est une NP immédiatement à la droite de la première NP qui sert à définir ou modifier cette NP.
  "*Sam, my brother*" → appos(*Sam*, *brother*)
- **08. aux** - auxiliaire : est un verbe juxtaposé à un verbe principal constituant ainsi avec lui une forme composée par exemple le verbe "*be*" et le verbe "*have*".
  "*Reagan has died*" → aux(*died*, *has*)
- **09. auxpass** - auxiliaire passif : de même que les auxiliaires sauf que la forme de la phrase est passive.
  "*Kennedy has been killed*" → auxpass(*killed*, *been*) et aux(*killed*,*has*)
- **10. cc** - coordination: c'est le rapport entre un élément de liaison et le mot de coordination de la conjonction.
  "*Bill is big and honest*" → cc(*big*, *and*)
- **11. ccomp** - clausal complement: le complement de clause d'un verbe ou adjective est une clause de dépendance avec un sujet interne dont la fonctionnalité est un objet de verbe ou adjective.
  "He says that you like to swim" → ccomp(says, like)
- **12. complm** - complementizer: est une conjunction de subordination utilisé par "that" ou "whether".
  "He says that you like to swim" → complm(like, that)
- **13. conj** - conjunct: la conjuction est une relation entre deux éléments connectés par une coordination comme “and”, “or”, etc.
  "Bill is big and honest" → conj(big, honest)
- **14. cop** - copula: “A copula” est une relation entre le complement d’un verbe copulaire et le verbe copulaire.
  "Bill is big" → cop(big, is)

- **15. csubj** - clausal subject: Un sujet clausal est un sujet syntaxique clausal d'une clause, par exemple, le sujet est lui-même une clause.
  "*What she said makes sense*" → csubj(*makes*, *said*)
- **16. csubjpass** - clausal passive subject: Un sujet passif clausal est un sujet syntaxique clausal d'une clause passive.
  "That she lied was suspected by everyone" → csubjpass(suspected, lied)
- **17. dep** - dependent: Une dépendance est appelée dep lorsque le système est incapable de déterminer une relation de dépendance plus précise entre deux mots.
  "Then, as if to show that he could, . . . " → dep(show, if)

- **18. det** - determiner: il transforme le nom dont il est, en un syntagme nominal.
  "*The man is here*" → det(*man*, *the*)
- **19. dobj** - objet direct: Le complément d'objet direct du syntagme VP.
  "*She gave me a raise*" → dobj(*gave*, *raise*)
- **20. expl** - expletive: Cette relation capture un existentiel «there». Le verbe principal de la clause est le gouverneur.
  "There is a ghost in the room" → expl(is, There)

- **21. infmod** - infinitival modifier: Le midificateur '….' qui permet de changer le sens de la phrase.
  "Points to establish are . . . " → infmod(points, establish)
- **22. iobj** - indirect object: L'objet indirect d'un VP est l'expression nominale qui est l'objet (datif) du verbe.
  "She gave me a raise" → iobj(gave, me)
- **23. mark** - marker: Un marqueur d'un complément de clause adverbial (advcl) est le mot qui l'introduit.
  "Forces engaged in fighting after insurgents attacked" → mark(attacked, after)
- **24. mwe** - multi-word expression: La relation d'expression multi-mots (modificateur) est utilisée pour certains idiomes multi-mots qui se comportent comme un mot fonctionnel unique.
  "I like dogs as well as cats" → mwe(well, as) et mwe(well, as)
- **25. neg** - negation modifier: Le modificateur de négation est la relation entre un mot de négation et le mot qu'il modifie.
  "Bill is not a scientist" → neg(scientist, not)
  "Bill doesn't drive" → neg(drive, n't)

- **26. nn** - noun compound modifier: Un modificateur de nom compose d'une NP est n'importe quelle nom qui modifie le sens.
  "Oil price futures" → nn(futures, oil) **nn**(futures, price)
- **27. npadvmod** - noun phrase as adverbial modifier : Cette relation capture différents endroits où quelque chose syntatiquement un syntagme nominal (NP) est utilisé comme modificateur adverbial dans une phrase.
  "The director is 65 years old" → npadvmod(old, years)
  "6 feet long" → npadvmod(long, feet)
  "Shares eased a fraction" → npadvmod(eased, fraction)
  "IBM earned $ 5 a share" → npadvmod($, share)
  "The silence is itself significant" → npadvmod(significant, itself)
  "90% of Australians like him, the most of any country" npadvmod(like, most)

- **28. nsubj** - sujet nominal: c'est un syntagme nominal qui présente le sujet syntaxique d'une clause.
  "*Clinton defeated Dole*" → nsubj(*defeated*, *Clinton*)
  "*The baby is cute*" → nsubj(*cute*, *baby*)
- **29. nsubjpass** - passive nominal subject: Un sujet nominal passif est un syntagme nominal qui est le sujet syntaxique d'une clause passive.
  "Dole was defeated by Clinton" → nsubjpass(defeated, Dole)
- **30. num** - numeric modifier: Un modificateur numérique d'un nom est n'importe quelle expression numérique qui sert à modifier la signification du nom.
  "Sam eats 3 sheep" → num(sheep, 3)
- **31. number** - element of compound number: Un élément de numéro composé est une partie d'une phrase nombre ou montant de devise.
  "I lost $ 3.2 billion" → number($, billion)
- **32. parataxis** - parataxis: La relation parataxis est une relation entre le verbe principal d'une clause et d'autres éléments.
  "The guy, John said, left early in the morning" → parataxis(left, said)
- **33. partmod** - participial modifier: Un modificateur participative est un participle de verbe qui sert à modifier le sens d'une NP.
  "Bill tried to shoot demonstrating his incompetence" → partmod(shoot, demonstrating)
- **34. pcomp** - prepositional complement: Ceci est utilisé lorsque le complément d'une préposition est une clause ou une phrase prépositionnelle.
  "We have no information on whether users are at risk" → pcomp(on, are)
- **35. pobj** - object of a preposition: L'objet d'une préposition est la tête d'un syntagme nominal suivant la préposition.
  "I sat on the chair" → pobj(on, chair)
- **36. poss** - possession modifier: La relation modi fi cateur de possession entre la tête d'un NP et son déterminant possessif, ou le complément d'un génitif.
  "their offices" → poss(offices, their)
- **37. possessive** - possessive modifier: The possessive modifier relation appears between the head of an NP and the genitive 's.
  "Bill's clothes" → possessive(John, 's)
- **38. preconj** - preconjunct: A preconjunct is the relation between the head of an NP and a word that appears at the beginning bracketing a conjunction (and puts emphasis on it), such as "either", "both", "neither").
  "Both the boys and the girls are here" → preconj(boys, both)
- **39. predet** - predeterminer: A predeterminer is the relation between the head of an NP and a word that precedes and modifies the meaning of the NP determiner.
  "All the boys are here" predet(boys, all)
- **40. prep** - modificateur propositionnel: Un modificateur propositionnel d'un verbe, adjectif ou d'un nom, est une préposition qui sert à modifier le sens du verbe, de l'adjectif, du nom, ou même d'une autre préposition.
  "*I saw a cat in a hat*" → prep(*cat*, *in*)
- **41. prepc** - prepositional clausal modifier: In the collapsed representation (see section 4), a prepositional clausal modifier of a verb, adjective, or noun is a clause

introduced by a preposition which serves to modify the meaning of the verb, adjective, or noun.
"He purchased it without paying a premium" → prepc without(purchased, paying)

- **42. prt** - phrasal verb particle: The phrasal verb particle relation identifies a phrasal verb, and holds between the verb and its particle.
"They shut down the station" → prt(shut, down)
- **43. punct** - punctuation: This is used for any piece of punctuation in a clause, if punctuation is being retained in the typed dependencies. By default, punctuation is not retained in the output.
"Go home! " → punct(Go, !)
- **44. purpcl** - purpose clause modifier: A purpose clause modifier of a VP is a clause headed by "(in order) to" specifying a purpose. At present the system only recognizes ones that have "in order to" as otherwise the system is unable to distinguish from the surface representations between these and open clausal complements (xcomp). It can also recognize fronted "to" purpose clauses in sentences.
"He talked to him in order to secure the account" → purpcl(talked, secure)
- **45. quantmod** - quantifier phrase modifier: A quantifier modifier is an element modifying the head of a QP constituent. (These are modifiers in complex numeric quantifiers, not other types of "quantification". Quantifiers like "all" become det.)
"About 200 people came to the party" → quantmod(200, About)
- **46. rcmod** - relative clause modifier: A relative clause modifier of an NP is a relative clause modifying the NP. The relation points from the head noun of the NP to the head of the relative clause, normally a verb.
"I saw the man you love" → rcmod(man, love)
- **47. ref** - referent: A referent of the head of an NP is the relative word introducing the relative clause modifying the NP.
"I saw the book which you bought" → ref (book, which)
- **48. rel** - relative: A relative of a relative clause is the head word of the WH-phrase introducing it.
"I saw the man whose wife you love" → rel(love, wife)
- **49. root** - root: permet de définir la racine et le point de départ dans la phrase. C'est un élément fictif.
"*I love French fries*." → root(*ROOT*, *love*)
"*Bill is an honest man*" → root(*ROOT*, *man*)
- **50. tmod** - modificateur temporel: il s'agit d'une expression qui sert à exprimer le temps dans une phrase.
"*Last night, I swam in the pool*" → tmod(*swam*, *night*)
- **51. xcomp** - open clausal complement: An open clausal complement (xcomp) of a VP or an ADJP is a clausal complement without its own subject, whose reference is determined by an external subject. These complements are always non-finite. The name xcomp is borrowed from Lexical-Functional Grammar.
"*He says that you like to swim*" → xcomp(*like*, *swim*)
- **52. xsubj** - controlling subject: A controlling subject is the relation between the head of a open clausal complement (xcomp) and the external subject of that clause.
"*Tom likes to eat fish*" → xsubj(*eat*, *Tom*)

Annexe E : Hiérarchie des relations de dépendances grammaticales

The grammatical relations defined in the above section stand in a hierarchy. The most generic grammatical relation, dependent (dep), will be used when a more precise relation in the hierarchy does not exist or cannot be retrieved by the system.

- root - root
- dep - dependent
  - aux - auxiliary
    - auxpass - passive auxiliary
    - cop - copula
  - arg - argument
    - agent - agent
    - comp - complement
      - acomp - adjectival complement
      - attr - attributive
      - ccomp - clausal complement with internal subject
      - xcomp - clausal complement with external subject
      - complm - complementizer
        - obj - object
        - dobj - direct object
        - iobj - indirect object
      - pobj - object of preposition
      - mark - marker (word introducing an advcl)
      - rel - relative (word introducing a rcmod)
  - subj - subject
    - nsubj - nominal subject
      - nsubjpass - passive nominal subject
    - csubj - clausal subject
      - csubjpass - passive clausal subject
  - cc - coordination
  - conj - conjunct
  - expl - expletive (expletive “there”)
  - mod - modifier
    - abbrev - abbreviation modifier
    - amod - adjectival modifier
    - appos - appositional modifier
    - advcl - adverbial clause modifier
    - purpcl - purpose clause modifier
    - det - determiner
    - predet - predeterminer
    - preconj - preconjunct
    - infmod - infinitival modifier
    - mwe - multi-word expression modifier
    - partmod - participial modifier
    - advmod - adverbial modifier

        - neg - negation modifier
    - rcmod - relative clause modifier
    - quantmod - quantifier modifier
    - nn - noun compound modifier
    - npadvmod - noun phrase adverbial modifier
        - tmod - temporal modifier
    - num - numeric modifier
    - number - element of compound number
    - prep - prepositional modifier
    - poss - possession modifier
    - possessive - possessive modifier (’s)
    - prt - phrasal verb particle
- parataxis - parataxis
- punct - punctuation
- ref - referent
- sdep - semantic dependent
    - xsubj - controlling subject

Annexe F : Calcul de la distance de Jaro-Winkler

La distance de Jaro-Winkler mesure la similarité entre deux chaînes de caractères. Plus la distance entre deux chaînes est élevée, plus elles sont similaires. Cette mesure est particulièrement adaptée au traitement de chaînes courtes comme des noms ou des mots de passe. Le résultat est normalisé de façon à avoir une mesure entre 0 et 1, le zéro représentant l'absence de similarité[83].

Distance de Jaro

La distance de Jaro entre chaînes S_1 et S_2 est définie par : $d_j = \frac{1}{3}\left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-t}{m}\right)$ où:

- $|S_i|$ est la longueur de la chaîne de caractères S_i ;
- m est le nombre de caractères correspondants (voir ci-dessous);
- t est le nombre de transpositions (voir ci-dessous).

Deux caractères identiques de S_1 et S_2 sont considérés comme correspondants si leur éloignement ne dépasse pas : $\left\lfloor \frac{\max(|S_1|,|S_2|)}{2} \right\rfloor - 1$.

Le nombre de transpositions est obtenu en comparant le $i^{\text{ème}}$ caractère correspondant de S_1 avec le $i^{\text{ème}}$ caractère correspondant de S_2. Le nombre de fois où ces caractères sont différents, divisé par deux, donne le nombre de transpositions.

Distance de Jaro-Winkler

La méthode introduite par Winkler utilise un coefficient de préfixe p qui favorise les chaînes commençant par un préfixe de longueur ℓ (avec $\ell \leq 4$). En considérant deux chaînes S_1 et S_2, leur distance de Jaro-Winkler d_w est : $d_w = d_j + \left(\ell p(1 - d_j)\right)$ où :

- d_j est la distance de Jaro entre S_1 et S_2
- ℓ est la longueur du préfixe commun (maximum 4 caractères)
- p[84] est un coefficient qui permet de favoriser les chaînes avec un préfixe commun.

Exemples

Soit deux chaînes S_1= MARTHA et S_2= MARHTA. La table de correspondance est :

	M	A	R	T	H	A
M	1	0	0	0	0	0
A	0	1	0	0	0	0
R	0	0	1	0	0	0
H	0	0	0	0	1	0
T	0	0	0	1	0	0
A	0	0	0	0	0	1

$m = 6$ (nombre de 1 dans la matrice)

$|S_1| = 6$ (nombre de caractères de la chaîne S_1)

$|S_2| = 6$ (nombre de caractères de la chaîne S_2)

[83] La définition est prise du site http://fr.wikipedia.org/wiki/Distance_de_Jaro-Winkler

[84] Winkler propose pour valeur de p égale à 0.1

Les caractères correspondants sont {M,A,R,T,H,A} pour S_1 et {M,A,R,H,T,A} pour S_2. En considérant ces ensembles ordonnés, on a donc 2 couples (T/H et H/T) de caractères correspondants différents, soit deux demi-transpositions. D'où $t = \frac{2}{2} = 1$. La distance de Jaro est : $d_j = \frac{1}{3}\left(\frac{6}{6} + \frac{6}{6} + \frac{6-1}{6}\right) = 0.944$

La distance de Jaro-Winkler avec $p = 0.1$ avec un préfixe de longueur $\ell = 3$ devient $d_w = 0.944 + \left(3 * 0.1(1 - 0.944)\right) = 0.961$.

Avec les chaînes S_1= DWAYNE et S_2= DUANE on trouve :

	D	U	A	N	E
D	1	0	0	0	0
W	0	0	0	0	0
A	0	0	1	0	0
Y	0	0	0	0	0
N	0	0	0	1	0
E	0	0	0	0	1

$m = 4$ (nombre de 1 dans la matrice)

$|S_1| = 6$ (nombre de caractères de la chaîne S_1)

$|S_2| = 5$ (nombre de caractères de la chaîne S_2)

$t = 0$ (nombre de transpositions).

La distance de Jaro est : $d_j = \frac{1}{3}\left(\frac{4}{6} + \frac{4}{5} + \frac{4-0}{4}\right) = 0.822$.

Celle de Jaro-Winkler avec $\ell = 1$: $d_w = 0.822 + \left(1 * 0.1(1 - 0.822)\right) = 0.840$.

Tout au long de cette thèse, nous avons utilisé la bibliothèque fuzzycomp[85] en python pour calculer la distance de Jaro-Winkler.

[85] fuzzycomp : est une bibliothèque de fonctions pour calculer la similarité entre deux chaînes de caractères. [https://code.google.com/p/fuzzycomp/]

Résumé

Dans cette thèse, nous traitons la traduction automatique vers les langues des signes. Nous avons commencé par l'analyse des problématiques et des systèmes existants afin de proposer une modélisation d'un traducteur automatique à base de statistiques d'un texte Anglais vers la langue des signes Américaine (Anglais/ASL). En effet, nous avons proposé une nouvelle approche pour la génération d'un corpus artificiel en utilisant des règles de dépendances grammaticales. Ce corpus parallèle est l'entrée de notre système de traduction afin de construire la mémoire de traduction statistique en se basant sur les algorithmes d'alignement IBM. Ces algorithmes ont été améliorés et optimisés par l'intégration des distances Jaro-Winkler. Ensuite, et à partir de la mémoire de traduction construite, nous avons modélisé et implémenté un décodeur permettant de traduire un texte en Anglais en entrée vers la langue des signes Américaine en utilisant un nouveau système de transcription basé sur l'annotation en gloses. Les résultats obtenus ont été évalués par la métrique d'évaluation BLEU utilisée dans le domaine de la traduction automatique.

Mots clés

Traitement automatique des langues des signes ; Traduction automatique ; Machine d'apprentissage

Abstract

In this thesis, we deal with machine translation to sign language. We start with studying existing systems and issues in order to propose a new model for statistical machine translation from written english text to American Sign Language (English/ASL). Indeed, we proposed a new approach aiming to build artificial corpus using grammatical dependencies rules. The parallel corpus was the input of our machine translation that has been used to create the statistical memory translation based on the IBM alignment algorithms. These algorithms have been improved and optimized by integrating Jaro-Winkler distances. Then, based on the constructed translation memory, we have modeled and implemented a decoder to translate an English text to the American sign language using a new transcription system based on gloss annotation. The obtained results were evaluated by the BLEU evaluation metric used in the field of machine translation.

Keywords

Sign Language Processing ; Machine Translation ; Machine Learning