

UNIVERSITÉ LIBRE DE BRUXELLES

Faculté de Lettres, Traduction et Communication

Configuration et évaluation
d'un système de prédiction de mots
au sein d'un logiciel de Communication
Améliorée et Alternative (CAA)
pour personnes handicapées
Étude de Presage et ACAT

Magali NORRÉ

Mémoire présenté sous la direction de Max
DE WILDE en vue de l'obtention du titre de
Master en Sciences et Technologies de l'In-
formation et de la Communication

Année académique 2017–2018



Magali Norré

**Configuration et évaluation d'un système de prédiction de mots
au sein d'un logiciel de Communication Améliorée et Alternative (CAA)
pour personnes handicapées**

Étude de Presage et ACAT

Résumé

Ce mémoire présente la configuration et l'évaluation d'un système de prédiction de mots ainsi que d'un logiciel de Communication Améliorée et Alternative (CAA) pour les personnes handicapées. En plus d'avoir une mobilité réduite, ces utilisateurs ont un usage de la parole plus ou moins altéré qui doit être compensé par une aide technologique proposant des modalités de saisie adaptées à leurs capacités. Afin d'accélérer leur vitesse de communication, différentes techniques de prédiction et de modélisation du langage sont utilisées. Ces approches sont étudiées par le biais de divers systèmes existants. Le paramétrage de prédicteurs statistiques est analysé, leur configuration en français étant évaluée par un simulateur et testée par une personne handicapée. Les résultats montrent qu'un modèle de langage lissé trigramme construit à partir d'un gros corpus littéraire comme Google Books Ngram permet d'économiser plus d'une saisie sur deux, les performances de ces systèmes variant en fonction de divers paramètres tels que les corpus, les prédicteurs (le nombre de n-grammes, le lissage, etc.), le nombre de prédictions lexicales à afficher et l'interface utilisateur.

Mots-clés : prédiction de mots, communication améliorée et alternative, technologie d'assistance, interaction homme-machine, traitement automatique du langage, handicap, accessibilité

Abstract

This thesis investigates the configuration and evaluation of a word prediction system as well as an Augmentative and Alternative Communication (AAC) device for people with disabilities. In addition to having a reduced mobility, these users have a more or less altered use of speech that must be compensated by a technological aid offering input methods adapted to their capabilities. In order to improve their communication speed, different prediction and language modeling techniques are used. A large diversity of AAC systems are nowadays available, which use specifically one or several of these techniques. The parameterization of statistical predictors is investigated, their configuration in French being evaluated by a simulator and tested by a disabled person. The results show that a smoothed trigram language model built from a large literary corpus like Google Books Ngram saves more than one keystroke out of two, the performance of these systems varying according to various parameters such as corpus, prediction model (number of n-grams, smoothing, etc.), number of displayed lexical predictions and user interface.

Keywords: word prediction, augmentative and alternative communication, assistive technology, human-computer interaction, natural language processing, disability, accessibility

But it must be recognized that the notion of "probability of a sentence" is an entirely useless one, under any known interpretation of this term.

Noam Chomsky (1969)

I believe that disabled people should concentrate on things that their handicap doesn't prevent them from doing and not regret those they can't do.

Stephen Hawking (2013)

Sommaire

Liste des figures	iv
Liste des tables	v
Liste des abréviations	vi
Introduction	1
I État de l’art	4
1 Communication améliorée et alternative	4
2 Systèmes de prédiction	12
3 Limites linguistiques, cognitives et ergonomiques	33
II Étude de cas	46
1 Objectif	46
2 Outils	47
3 Méthodologie	54
4 Résultats	68
5 Discussion	74
Conclusion	80
A Accessibilité du Web	84
B Code source	86
C Corpus Blog	98
D Questionnaire	100
Table des matières	104
Bibliographie	105

Résumé

Ce mémoire présente la configuration et l'évaluation d'un système de prédiction de mots ainsi que d'un logiciel de Communication Améliorée et Alternative (CAA) pour les personnes handicapées. En plus d'avoir une mobilité réduite, ces utilisateurs ont un usage de la parole plus ou moins altéré qui doit être compensé par une aide technologique proposant des modalités de saisie adaptées à leurs capacités. Afin d'accélérer leur vitesse de communication, différentes techniques de prédiction et de modélisation du langage sont utilisées. Ces approches sont étudiées par le biais de divers systèmes existants. Le paramétrage de prédicteurs statistiques est analysé, leur configuration en français étant évaluée par un simulateur et testée par une personne handicapée. Les résultats montrent qu'un modèle de langage lissé trigramme construit à partir d'un gros corpus littéraire comme Google Books Ngram permet d'économiser plus d'une saisie sur deux, les performances de ces systèmes variant en fonction de divers paramètres tels que les corpus, les prédicteurs (le nombre de n-grammes, le lissage, etc.), le nombre de prédictions lexicales à afficher et l'interface utilisateur.

Mots-clés : prédiction de mots, communication améliorée et alternative, technologie d'assistance, interaction homme-machine, traitement automatique du langage, handicap, accessibilité

Remerciements

Je voudrais remercier les personnes qui m'ont donné de nombreux conseils tout au long de la rédaction de ce mémoire.

Tout d'abord, mon directeur, M. De Wilde, pour l'encadrement, ses relectures et ses commentaires toujours pertinents. M. Hengchen et M. Jabon, pour avoir répondu à toutes mes questions concernant la partie liée au Web scraping et le développement du simulateur en Python. M. Antoine de l'Université de Tours, pour ces échanges si enrichissants dans un domaine qui lui tient à cœur et dont il est un grand spécialiste.

Mes remerciements vont aussi aux personnes qui m'ont aidé à constituer mes corpus et celles rencontrées grâce à l'association Comalso qui m'ont montré l'importance de la communication alternative qu'elle soit alphabétique voire pictographique. Je tiens également à mettre en avant une initiative, celle de M. Guitton, Mme Sauzéon et l'équipe d'Inria Learning Lab pour leur MOOC sur l'accessibilité numérique qui traite des systèmes à destination de toutes les personnes en situation de handicap.

Je dédie finalement ce travail à ma maman, dont on a diagnostiqué une autre maladie neurodégénérative cette année et qui a continué à me soutenir à sa façon.

Liste des figures

I.1	Complétion et liste de prédictions	10
I.1a	Soothsayer	10
I.1b	Presage : qprompter	10
I.2	Dictionnaire ou lexique	12
I.2a	Presage : base de données	12
I.2b	Structure en arbre	12
I.3	SibyLettre : exemple du mot « compter »	15
I.4	KeyGlass : exemple de la séquence « se... »	15
I.5	Sibylle : version K	17
I.6	VITIPI : transducteur	20
I.7	WordTree	21
I.7a	Arbre lexicographique : exemple des mots « a, ami, bon, bonne... »	21
I.7b	Interface utilisateur : exemple de la séquence « bo... »	21
I.8	CFG : exemple de la phrase « the cat eats the mice »	23
I.9	Interface utilisateur : exemple du mot « communication »	26
I.9a	PCA : mode verbal	26
I.9b	Sibylle : version 3.5	26
I.10	Fréquences et grammaire	28
I.10a	Windmill : interface utilisateur	28
I.10b	Chart parsing probabiliste	28
I.11	CFG : exemple de la phrase « je mange un canard en pyjama »	34
I.11a	Interprétation 1	34
I.11b	Interprétation 2	34
II.1	ACAT : tableau de bord	51
II.2	Claviers : mode Talk AZERTY et ABC ou App QWERTY et ABC	51
II.3	Sous-claviers : pavé numérique et ponctuation	52
II.4	Autres sous-claviers : navigation, touches de fonction et souris	53
II.5	Presage : fonctionnement	62
II.6	ACAT et Presage : test utilisateur	67
II.7	Résultats : corpus, prédicteurs, prédictions	72

Liste des tables

I.1	Systèmes de prédiction et leur KSR	32
II.1	Prédicteurs de Presage	48
II.2	Corpus français d'entraînement	55
II.3	Nombre de n-grammes utilisés pour le prédicteur par défaut et ARPA . . .	59
II.4	Formats de données des corpus	60
II.5	Exemple d'abréviations	61
II.6	Exemple de saisie de la phrase « on a sonné à la porte »	64
II.7	Extrait du corpus de test A	66
II.8	Résultats : corpus d'entraînement et de test A	68
II.9	Résultats : corpus d'entraînement et de test B	68
II.10	Résultats : prédicteur par défaut et nombre de n-grammes	69
II.11	Résultats : prédicteur par défaut et paramètres de lissage	69
II.12	Résultats : prédicteur par défaut et modèle de langage de l'utilisateur . . .	70
II.13	Résultats : prédicteur ARPA	71
II.14	Résultats : prédicteur par défaut et nombre de prédictions	71
A.1	AnySurfer et WCAG 2.0	85
C.1	Liste des articles utilisés	99
D.1	SUS : questionnaire de satisfaction	102

Liste des abréviations

AAATE Association for the Advancement of Assistive Technology in Europe

AAC Augmentative and Alternative Communication

ACAT Assistive Context-Aware Toolkit

ASL Analyse Sémantique Latente

CAA Communication Améliorée et Alternative

FALC Facile A Lire et à Comprendre

FST Fondation Suisse pour les Téléthèses

IHM Interaction Homme-Machine

IMC Infirmité Motrice Cérébrale

ISAAC International Society for Augmentative and Alternative Communication

KSPC Keystroke Per Character

KSR Keystroke Saving Rate

LDC Linguistic Data Consortium

LIS Locked-In Syndrome

NLTK Natural Language Toolkit

OOV Out Of Vocabulary

POS Part Of Speech

SLA Sclérose Latérale Amyotrophique

SUS System Usability Scale

TAL Traitement Automatique des Langues

UAAG User Agent Accessibility Guidelines

W3C World Wide Web Consortium

WAI Web Accessibility Initiative

WCAG Web Content Accessibility Guidelines

Introduction

*Before my operation my speech had been getting more slurred,
so only people who knew me well could understand me.*

But at least I could communicate.

Stephen Hawking (2013)

« Que feriez-vous si vous étiez un jour privé de la parole? » est la première question que pose l’ergothérapeute Cataix-Nègre (2017) dans l’introduction de son livre qui traite des communications alternatives. Dans une société où la circulation de l’information et de la communication est devenue un réel enjeu, l’écart avec les personnes en situation de handicap qui connaissent à la fois de graves difficultés pour se déplacer et communiquer pourrait s’avérer davantage problématique.

De nombreuses recherches sont néanmoins menées depuis des années dans le domaine du handicap et des aides logicielles. Certaines d’entre elles aboutissent à des prototypes opérationnels qui sont parfois commercialisés. Les personnes paralysées dont la communication est plus ou moins altérée ont désormais la possibilité de se procurer ces Communications Améliorées et Alternatives (CAA). Disponibles sur ordinateur ou tablette, la plupart intègrent des technologies comme la prédiction de textes, les claviers virtuels, la synthèse vocale ou la commande oculaire.

Ce mémoire vise à comprendre le fonctionnement des moteurs de prédiction de textes, ces systèmes utilisés dans les téléphones portables qui suggèrent des mots en fonction des lettres déjà entrées. Cette technique bien connue du grand public permet à l’utilisateur d’économiser les saisies requises pour composer un mot et accélérer la vitesse de saisie. Étant donné que certains utilisateurs handicapés ne peuvent se servir d’un clavier ou d’une souris, le recours à ces systèmes est alors nécessaire et se traduit par leur omniprésence dans de nombreuses communications alternatives modernes.

L’état de l’art permettra de passer en revue leurs caractéristiques et l’étude de cas comportera plusieurs points non abordés dans la littérature scientifique. D’une part, nous proposons de tester l’efficacité de corpus disponibles sur le Web comme la version française de Google Books Ngram qui n’est ni un corpus journalistique, ni un corpus oral et qui n’a jamais été utilisé dans le cadre d’un système de prédiction de mots. D’autre part, aucune étude sur le logiciel ACAT n’a encore été réalisée. Leur configuration en français a donc nécessité la création et le traitement de diverses ressources.

En focalisant notre propos sur le moteur de prédiction Presage et l'interface utilisateur ACAT, nous tenterons de déterminer s'il est possible de comparer Presage à d'autres systèmes, une fois que nous l'aurons configuré grâce à des connaissances en linguistique informatique. Le but sera d'exploiter au mieux les différents algorithmes et paramètres du prédicteur afin d'évaluer l'efficacité de logiciels libres. Pour ce faire, nous formulerons des questions concernant quatre éléments de notre configuration qui serviront de base à la structure de notre évaluation quantitative et qualitative.

La première question ciblera l'impact du choix du corpus. Nous en expérimenterons plusieurs et nous comparerons les résultats afin de voir si les performances du prédicteur en dépendent. Nous testerons également la généricité de ces différents corpus. Est-ce que des corpus spécifiques seront plus efficaces que des corpus génériques? Par exemple, est-ce qu'un corpus caractérisant l'idiolecte de l'utilisateur sera plus adapté qu'un corpus littéraire? La taille des corpus étant un facteur souvent étudié, nous vérifierons si un petit corpus de quelques milliers de mots sera effectivement moins performant qu'un plus gros en contenant des millions.

La deuxième question concernera les algorithmes de prédiction. Nous nous intéresserons aux modèles statistiques implémentés au sein du prédicteur. Est-ce qu'un modèle de langage lissé trigramme obtiendra de meilleurs résultats qu'un modèle unigramme ou bigramme? Quel gain apportent les différents paramètres de lissage de l'interpolation linéaire? La combinaison d'un modèle de langage général avec un modèle utilisateur est-elle efficace? Et quels seront les résultats d'un prédicteur utilisant un modèle de langage lissé et encodé en format ARPA? Pour répondre à ces questions, nous procéderons à des estimations théoriques afin d'évaluer ces prédicteurs.

La troisième question testera le nombre de prédictions lexicales à afficher. Est-ce que nous pouvons supposer que plus leur nombre augmente, plus il y a de chances que le mot prédit soit le bon et donc celui recherché par l'utilisateur? Dans la discussion, nous verrons que beaucoup d'études ont abordé ce sujet même si nous pouvons nous demander ce qu'il en est de l'usage réel qu'en font les personnes handicapées. Cette question est liée à la suivante étant donné que des facteurs cognitifs, mais aussi ergonomiques doivent être pris en compte.

La quatrième question étudiera la relation entre le prédicteur et son interface utilisateur. Aujourd'hui, des études apparaissent afin de trouver des solutions ergonomiques pour optimiser la présentation des prédictions et proposer de plus en plus de fonctionnalités de paramétrage. Après avoir répondu aux trois questions précédentes au moyen d'un simulateur, nous réaliserons un test utilisateur avec une personne handicapée afin d'évaluer son degré de satisfaction et si la configuration théorique lui convient en observant sa façon d'interagir avec l'interface.

Le mémoire a été divisé en deux chapitres : l'état de l'art et l'étude de cas.

Le chapitre I est essentiellement théorique puisqu'il exposera un état de l'art des communications alternatives ainsi que des recherches antérieures sur les systèmes de prédiction. Il présentera différentes sortes de CAA, dont celle proposant une prédiction de mots afin d'accélérer la saisie de phrases susceptibles d'être synthétisées pour permettre une meilleure communication. De nombreux systèmes existent déjà sur le marché en plus des divers projets et prototypes de recherche. Leur fonctionnement sera détaillé ainsi que leurs avantages et inconvénients. Nous présenterons la prédiction statistique avec les modèles de langage comme les n-grammaires, la prédiction symbolique basée sur des règles (morpho)syntaxiques et des contraintes sémantiques, mais également les versions hybrides qui combinent ces deux approches. Il existe beaucoup de formalismes. Nous verrons que des méthodes d'apprentissage automatique ainsi que diverses techniques plus avancées sont utilisées afin de permettre au système de s'adapter progressivement au style de langage de l'utilisateur et d'affiner les prédictions en apprenant de nouveaux mots.

Le chapitre II sera centré sur l'étude de cas de façon à répondre aux questions ci-dessus. Il décrira la configuration et l'évaluation de deux logiciels *open source* qui ont été conçus de manière totalement indépendante : Presage et ACAT. L'objectif de ce mémoire sera de trouver le meilleur paramétrage français qui minimise le nombre de saisies au clavier (calculable via différentes métriques) tout en essayant de ne pas alourdir davantage la charge cognitive et visuelle de l'utilisateur en évitant, par exemple, d'allonger inutilement la liste des propositions. Le code source étant public et mis à la disposition de tous, le créateur de Presage suggère non pas d'améliorer directement les performances de son prédicteur (car il faut toujours tenir compte des besoins individuels de chaque utilisateur), mais de tester sa propre configuration en fonction de leurs besoins. C'est pourquoi nous étudierons plusieurs paramètres en plus de l'interface utilisateur ACAT qui sera testée par une personne en situation de handicap. Un questionnaire de satisfaction lui sera également proposé. La configuration de ces deux systèmes fera donc l'objet d'une évaluation objective et subjective.

Enfin, nous conclurons en insistant sur certains points relatifs à la configuration et l'évaluation de ces systèmes. Nous aborderons également le TAL, les logiciels d'aide à la communication pour les personnes handicapées ainsi que la prédiction d'un point de vue éthique afin de mettre en avant certains risques et leurs implications.

Chapitre I

État de l’art

For a time, the only way I could communicate was to spell out words letter by letter by raising my eyebrows when somebody pointed to the right letter on a spelling card.

Stephen Hawking (2013)

Dans ce chapitre, nous aborderons la Communication Améliorée et Alternative (Section 1) avant de présenter le fonctionnement de différents systèmes de prédiction (Section 2) ainsi que les limites linguistiques, cognitives et ergonomiques relatives à la configuration et à l’évaluation de ces technologies (Section 3).

1 Communication améliorée et alternative

Selon l’association internationale ISAAC, la Communication Améliorée et Alternative (CAA) désigne l’ensemble des « moyens humains et matériels permettant de communiquer autrement ou mieux qu’avec les modes habituels et naturels, si ces derniers sont altérés ou absents ». ¹ Ce terme générique couvre donc à la fois un aspect supplétif et substitutif. Comme il s’agit de la traduction d’*Augmentative and Alternative Communication* (AAC), il existe différentes appellations en français (améliorée, augmentative ou augmentée, assistée, etc.), nous nous limiterons à la première. ²

1.1 Représentation du langage

La communication étant multimodale, il est possible de communiquer autrement que par le langage verbal. En fonction du handicap, Cataix-Nègre (2017) explique que les moyens de communication sont « multiples » et « variés » citant les « gestes, signes, images, objets, dessins, écrits... ». La transmission d’information passe donc par différents niveaux de représentation et peut s’appuyer sur un support simple ou

1. <http://www.isaac-fr.org>

2. Cet état de l’art sur les communications alternatives ne se veut pas exhaustif. Le lecteur pourra trouver plus d’informations dans le livre de Cataix-Nègre (2017) ou le chapitre de Pino (2014).

plus avancé comme une aide technologique. Un moyen qui est souvent recommandé aux patients atteints de troubles de la parole ou du langage – ne sachant parfois ni lire, ni écrire – consiste à recourir aux pictogrammes. Ces représentations graphiques expriment des concepts et sont regroupées en différents codes de communication.

Nous pouvons en citer plusieurs. Le Bliss est une langue idéographique de 3 000 symboles qu'il est possible de combiner à l'aide de règles de composition afin de créer de nouveaux mots. Les images issues du Makaton sont associées à un vocabulaire gestuel et des expressions faciales. Le système de pictogrammes polysémiques Minspeak® a été conçu par le linguiste Baker. Il nécessite que l'utilisateur ait à la fois un certain niveau mental ainsi qu'une très bonne mémoire visuelle selon l'association belge Comalso.³ Pour former le sens de « eat » ou « ate », l'utilisateur doit combiner le pictogramme représentant une « pomme » avec celui du « verbe » ou du « verbe +ed » (Tenny, 2016). Le pictogramme lexical « pomme » peut avoir différents sens (« fruit », « rouge », « faim », etc.)⁴ en fonction du contexte (Trost *et al.*, 2005), c'est-à-dire du pictogramme avec lequel il est combiné (*Semantic Compaction*).

Merlo (2015) a présenté une application sur tablette et smartphone qui permet la génération de phrases à l'aide de pictogrammes et de concepts qui seront ensuite synthétisées. Afin d'accélérer la composition de phrases, son moteur de prédiction sémantique s'appuie sur une grammaire par frames inspirée de la théorie du linguiste Fillmore (*Frame Semantics*). Chaque frame correspond à une « situation prototypique évoquée par des lexèmes ». Ainsi, après « manger », les pictogrammes proposés correspondront à des concepts comestibles comme « pomme ». En fonction des types sémantiques des objets (aliment, instrument, temps, etc.), plusieurs phrases peuvent être générées (« je mange une pomme », « je mange avec une fourchette », « je mange à 14 h »). Sevens *et al.* (2015) ont quant à eux proposé un système qui utilise un modèle de langage trigramme (cf. point 2.1.1) avec des *synsets* WordNet (point 2.2.3) pour générer, prédire et traduire des pictogrammes vers le néerlandais (ou inversement).

Ces différentes recherches contribuent au droit d'accès à l'information qui est écrit dans les lois et la Convention des Nations Unies pour les droits des personnes handicapées. En 2009, le projet européen Pathways a donné naissance au Facile A Lire et à Comprendre (FALC), une série de règles destinées à rendre l'information plus accessible aux personnes handicapées intellectuelles.⁵ Le guide recommande d'utiliser des mots faciles à comprendre, de faire des phrases simples et courtes, d'éviter les pronoms et de privilégier le présent. Il est néanmoins difficile d'obliger les personnes,

3. Bien qu'il ne s'agisse pas d'une référence académique, le site de Comalso propose un aperçu de ce que sont les communications alternatives (<http://comalso.be/aac/quest-ce-que-les-aac/>).

4. <http://www.minspeak.com/>

5. « L'information pour tous : Guide des règles européennes de mise en accessibilité des informations écrites pour les personnes handicapées mentales » édité par l'Unapei est disponible en ligne (<http://www.unapei.org/IMG/pdf/GuidePathways.pdf>).

quelles qu'elles soient, à simplifier leur écriture et à s'exprimer selon des « règles » même si elles peuvent s'avérer être utiles à d'autres personnes (*design for all*).

1.2 Aide au handicap

La Communication Améliorée et Alternative désigne toutes les aides à la communication destinées aux personnes handicapées. Il convient de noter que toutes ces technologies ne concernent pas les mêmes handicaps. Les systèmes que nous proposons d'étudier sont davantage adaptés aux personnes paralysées ayant un trouble sévère de la parole, mais ne présentant pas une importante déficience intellectuelle.

Parmi ce type de handicaps, nous pouvons citer les exemples repris par Ghe-dira (2009), à savoir : la Sclérose Latérale Amyotrophique (SLA) ou maladie de Charcot, l'Infirmité Motrice Cérébrale (IMC) bien que celle-ci soit souvent associée à des troubles cognitifs plus ou moins importants et le Locked-In Syndrome (LIS) aussi appelé syndrome d'enfermement.⁶ D'une part, il existe des variations au sein même de ces déficiences en fonction de divers facteurs comme le stade plus ou moins avancé de la pathologie si celle-ci est neurodégénérative. D'autre part, certains handicaps moteurs ou langagiers peuvent survenir à l'âge adulte après un Accident Vasculaire Cérébral (AVC), une trachéotomie, etc. ce qui implique que la personne maîtrisait déjà la communication écrite et aura tendance à privilégier ce médium plutôt que d'apprendre un système de représentation symbolique (Copestake, 1997).

Par conséquent, les besoins et les attentes varient énormément d'une personne à l'autre d'où l'impact de l'ergonomie et de la flexibilité dans l'adoption d'une technologie par le patient sur le long terme (cf. Section 3). Du point de vue de l'utilisateur, aucun système n'est parfait ; chacun possède des avantages et inconvénients comme le montrent à la fois Guerrier (2015) dans son état de l'art ou Meeùs dans son témoignage qui retrace son expérience des différents outils de communication qu'elle a testés.⁷ Les concepteurs de systèmes de communication alternative doivent donc tenter de prendre en compte la diversité des handicaps en laissant à l'utilisateur un vaste choix de paramètres et fonctionnalités d'accessibilité qu'il – ou son ergothérapeute – pourra configurer selon la spécificité de ses besoins. Il est néanmoins important de toujours documenter ces options et de les mettre suffisamment en évidence.

Les systèmes de CAA permettent de dépasser certaines limitations dues au handicap. Ils sont aujourd'hui disponibles sur différents supports (ordinateurs, tablettes, appareils spécifiques) et systèmes d'exploitation (Windows, Linux, MacOS ou iOS).⁸

6. Le journaliste français Jean-Dominique Bauby était atteint du LIS. Totale-ment paralysé après un accident vasculaire, il a écrit son livre *Le scaphandre et le papillon* avec sa paupière gauche.

7. <https://www.alis-asso.fr/meeus-mes-outils-de-communication-1> (Association du LIS).

8. Citons le logiciel Dasher aussi accessible sur Linux (<http://www.inference.org.uk/dasher/>) ou les applications d'AssistiveWare sur MacOS et iOS (Proloquo, Keeble, Keedogo, KeyStrokes, etc.).

La compatibilité avec d'autres types de logiciels et d'applications (messagerie électronique, navigation Internet, etc.) s'est progressivement développée. Certaines communications alternatives sont maintenant directement intégrées au sein de plateformes multitâches alors que d'autres incluent différentes fonctionnalités à leur interface utilisateur (Ghedira, 2009; Guerrier, 2015). Ces systèmes d'aide à la communication ont initialement été conçus pour permettre aux utilisateurs d'entrer leur message caractère par caractère dans un éditeur de texte par l'intermédiaire d'un clavier virtuel souvent associé à une liste de prédictions, d'une synthèse vocale et d'un dispositif physique d'entrée adapté à leur mobilité (Antoine et Maurel, 2007).

En fonction de ses capacités motrices, la personne a la possibilité de se servir d'un contacteur, d'un joystick, d'un détecteur de mouvements ou d'une commande oculaire qui peut contrôler un clavier à défilement automatique pointant chaque caractère. Le balayage du clavier se faisant linéairement (lettre par lettre) ou en mode ligne/colonne, le nombre d'appuis requis est réduit bien que le temps d'accès aux différents items soit plus long. Certains systèmes proposent donc parfois plusieurs modalités de saisie faisant office de clavier, souris ou défilement; c'est le cas de la plateforme multimodale de contrôle réalisée par Bianco *et al.* (2007). Pour communiquer, l'utilisateur d'une CAA peut utiliser une synthèse de parole qui oralise le message composé ou simplement l'afficher et le faire lire à l'interlocuteur. Les appareils portables de la gamme Lightwriter de Toby Churchill Ltd disposent, par exemple, d'un clavier physique avec prédiction de mots, d'une synthèse vocale, d'un grand haut-parleur ainsi que de deux écrans, l'un étant destiné à l'utilisateur et l'autre à l'interlocuteur.

Aujourd'hui, les CAA sont devenues un domaine de recherche prolifique, dont les progrès technologiques sont suivis et encouragés par plusieurs organisations professionnelles. Fondée en 1983, l'International Society for Augmentative and Alternative Communication (ISAAC) vise à améliorer la qualité de vie de toutes personnes utilisant une communication alternative. L'association est chargée de promouvoir le développement de la recherche. Elle organise des conférences biennales et publie la revue scientifique *AAC Augmentative and Alternative Communication*.⁹ Répartie en sections régionales ou nationales, nous pouvons mentionner l'ISAAC Francophone.¹⁰ L'Association for the Advancement of Assistive Technology in Europe (AAATE) organise également des conférences et gère plusieurs publications.¹¹ La Commission européenne finance aussi certains projets, ce fut le cas pour les prototypes d'aide à la communication KOMBE (Guenthner *et al.*, 1994) et FASTY (Trost *et al.*, 2005).¹²

9. <https://www.isaac-online.org>

10. <http://www.isaac-fr.org/isaac-fr/>

11. <http://aaate.net>

12. L'European Commission (2017, p. 16) continue de promouvoir « the concept of design for all and universal design through policies and research and innovation projects [https://ec.europa.eu/research/fp7/index_en.cfm], to develop interfaces and assistive technologies ».

1.3 Problématique et positionnement

Après avoir introduit les différents types de handicaps concernés et l'architecture générale des divers systèmes de communication alternative, nous allons aborder les technologies (point 1.3.1) ainsi que les mécanismes d'optimisation de la vitesse de saisie comme la complétion ou l'affichage d'une liste de prédictions (point 1.3.2).

1.3.1 Technologies et communication

L'Initiative pour l'Accessibilité au Web ou Web Accessibility Initiative (WAI) a été créée au sein du World Wide Web Consortium (W3C) en 1997. Elle a pour but d'améliorer l'accès des personnes handicapées aux protocoles Web en proposant diverses recommandations. Ainsi, sont apparues les *User Agent Accessibility Guidelines (UAAG) 2.0* qui favorisent l'accès via le navigateur Web et les technologies d'assistance, dont font partie les claviers de remplacement ainsi que les dispositifs de pointage alternatifs. Les directives garantissent que l'agent utilisateur est à la fois perceptible, utilisable, compréhensible et robuste. Ces quatre principes sont repris dans les *Web Content Accessibility Guidelines (WCAG) 2.0* (Caldwell *et al.*, 2008) qui veillent au contenu Web.

Parmi les diverses initiatives proposées, le site « Visa pour le Net » a par exemple été conçu pour être à la fois facile à lire et à utiliser pour toutes personnes ayant des difficultés de compréhension ; il a été entièrement rédigé en FALC.¹³ Dans son rapport du moniteur de l'accessibilité de 2017, AnySurfer révélait que 82 % des sites belges n'étaient pas accessibles aux personnes handicapées.¹⁴ Les critères « le focus est-il visible lors de la tabulation ? » et « le site est-il entièrement utilisable au clavier ? » qui concernent les déficients moteurs ont été rempli par 40 et 35 % des 173 sites évalués. Une liste reprenant les directives AnySurfer et les WCAG est disponible à l'Annexe A.¹⁵

En plus de l'accès problématique au Web et à toutes ses ressources, les handicapés moteurs sont parfois confrontés à de véritables problèmes de communication. C'est pourquoi nous nous intéresserons uniquement aux systèmes d'aide à la saisie permettant de l'accélérer et de remplacer la parole – alors déficiente – en la synthétisant via des mots transcrits à l'aide d'un clavier virtuel affiché à l'écran. Les études montrent toutefois une vitesse de saisie relativement faible si nous la comparons avec le débit de la communication orale estimé généralement à 150–200 mots par minute (Wood, 1996; Copestake, 1997; Schadle, 2003), les statistiques pour les CAA n'excédant pas 10 à 15 mots par minute. En plus de l'extrême lenteur de saisie qui ralentit la conversation et provoque parfois des situations embarrassantes¹⁶ (Garay-Vitoria et

13. <http://www.visapourlenet.be/>

14. <http://www.moniteurdelaaccessibilite.be/2017.html>

15. En France, le projet FACIL'iti (<http://www.facil-iti.fr/>) propose d'adapter l'affichage de certains sites pour les rendre plus accessibles en fonction de divers besoins (moteurs, cognitifs, visuels).

16. « The slow speed of typing creates a communication divide which can cause communication partners to lose interest or attempt to dominate the conversation » (Trnka et McCoy, 2007, p. 195).

Abascal, 2006), il faut également citer la fatigue motrice engendrée par ce type de systèmes chez les personnes handicapées (Niemeijer, 2005).

Bérard et Niemeijer (2004) relevaient déjà l'importance de l'ordinateur dans le contrôle de l'environnement de tous les jours et pour la communication avec le monde extérieur, le clavier étant le moyen le plus couramment utilisé pour la saisie de données quelles qu'elles soient. Destinés à remplacer les claviers physiques, les premiers claviers virtuels sont apparus dans les années 1980, en même temps que d'autres technologies dédiées à la CAA (Cataix-Nègre, 2017). En 1984, par exemple, l'appareil Muta-vox est doté d'un module linguistique fonctionnant sur le principe des pictogrammes qui permet de recomposer grammaticalement une phrase. Ainsi, « je / demain / avoir / beau / poupée » devient « demain, je vais avoir une belle poupée » (Bianco *et al.*, 2007).

Dans le domaine des synthèses vocales, le projet HECTOR de la Fondation Suisse pour les Téléthèses (FST) est lancé la même année.¹⁷ Bien qu'insistant sur la complémentarité entre les systèmes technologiques et non technologiques, Cataix-Nègre (2017, pp. 29–30) reconnaît que ces technologies ont un impact énorme sur l'efficacité de l'expression des personnes améliorant ainsi leur indépendance et autonomie. Elle note que ces logiciels dotés d'une synthèse vocale ont beaucoup évolué, passant par l'ordinateur fixe, portable, de poche avant de finalement être disponibles sur tablettes. L'utilisation de la synthèse vocale compense alors la modalité défaillante et est souvent associée à une aide à la saisie qui s'est, elle aussi, beaucoup développée.

De récentes études (Boissière *et al.*, 2015; Pouplin, 2016) ont ainsi montré que le Traitement Automatique des Langues (TAL) n'était plus le seul domaine concerné par l'optimisation de la vitesse de saisie de textes. L'Interaction Homme-Machine (IHM) et l'ergonomie de l'interface doivent être prises en compte lorsqu'on veut concevoir un système d'aide qui soit à la fois performant et adapté aux besoins réels des différents utilisateurs. La réduction de la fatigue et de l'effort relative à la charge cognitive et visuelle de l'utilisateur fait aujourd'hui l'objet de nombreuses recherches. Nous présenterons certaines fonctionnalités de paramétrage des prédicteurs et interfaces utilisateurs lorsque nous détaillerons le fonctionnement de ces systèmes (cf. Section 2).

1.3.2 Complétion et liste de prédictions

Afin d'accélérer la vitesse de saisie de textes et diminuer la fatigue des utilisateurs, il existe diverses techniques. Les logiciels de complétion et de prédiction qui offrent des suggestions de mots ou expressions en font partie (Pino, 2014), il est important de les distinguer.¹⁸ Initialement conçus pour les personnes handicapées, ces

17. Fin 2016, la FST a fusionné avec une entreprise commerciale (<http://www.fst.ch>).

18. Dans sa thèse, Badr (2011) propose une classification des claviers logiciels selon la présentation des résultats des systèmes de prédiction. Après avoir fait la distinction entre la prédiction de caractères et de mots, il classe dans la seconde catégorie la complétion, les listes et la désambiguïsation.

systèmes d'optimisation de la saisie se sont démocratisés et sont aujourd'hui utilisés par le grand public sur les interfaces limitées comme les dispositifs mobiles (Antoine, 2011). Les claviers étant relativement réduits, différentes méthodes (telles que la désabréviation et la désambiguïsation à la volée avec le T8 ou T9) ont fait leur apparition afin de composer des messages plus rapidement tout en limitant les erreurs de saisie.

Les systèmes de complétion permettent de compléter un mot en ne proposant qu'une suggestion commençant par les caractères saisis, le choix s'affinant naturellement à chaque saisie. Lorsque l'utilisateur décide de retenir le mot proposé en le validant, celui-ci est directement intégré au texte permettant ainsi « d'économiser les dernières saisies qui auraient été nécessaires à la composition du mot » (Pouplin, 2016, p. 42). Contrairement à la complétion, les logiciels de prédiction affichent une liste de mots présentée au-dessus du clavier ou à côté; horizontalement, verticalement voire les deux à la fois comme c'est le cas de l'interface utilisateur ACAT. Nous verrons que la présentation des prédictions lexicales peut être optimisée afin de réduire autant que possible le conflit cognitif entre écriture (du message) et lecture (des prédictions).

Boissière et Dours (2001) ont proposé le système de complétion VITIPI qui affiche les fins de mots lorsqu'il n'y a plus d'ambiguïté possible. Ils font remarquer que vu qu'il n'existe pas de liste de prédictions, la question de la surcharge cognitive ne se poserait plus (Boissière *et al.*, 2012, 2015). Les logiciels de complétion et de prédiction peuvent être intégrés à un clavier virtuel. L'utilisateur accepte les mots prédits en cliquant dessus à l'aide d'un dispositif de pointage combiné au système de défilement. La sélection de mots se fait via un seul clic ou avec des raccourcis clavier. Le prototype de complétion néerlandais Soothsayer¹⁹ (Figure I.1a) de Stoop et van den Bosch (2014) fonctionne avec la touche espace, tandis que l'éditeur de texte (qprompter) intégré au moteur de prédiction Presage (Figure I.1b) comprend les touches de fonction F1 à F6 pour sélectionner les six propositions – bien qu'il soit possible d'utiliser la souris.²⁰

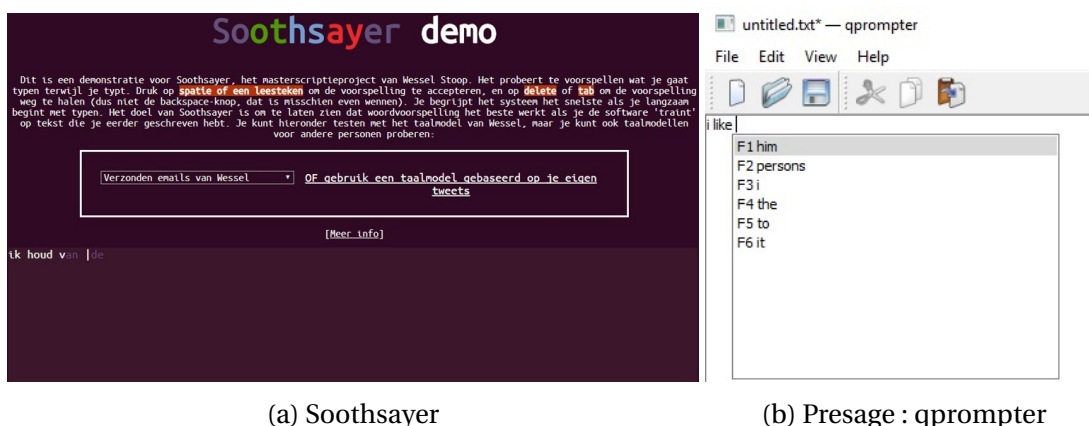


FIGURE I.1 – Complétion et liste de prédictions

19. Une démo de ce système de complétion est disponible en ligne (<http://soothsayer.cls.ru.nl/>).

20. À noter qu'aucun de ces deux prototypes de recherche ne propose de clavier virtuel.

Ainsi, supposons que l'utilisateur veuille écrire « i like him » ou « i like persons who ». Après avoir entré « i like », le système peut afficher les six propositions ci-dessus, c'est-à-dire celles que l'utilisateur pourrait vouloir écrire. Dans l'exemple, l'utilisateur a la possibilité d'appuyer sur F1 ou F2 pour écrire les mots « him » ou « persons ». Le système entrera le mot avec une espace. Dans le cas d'un système sans défilement et si aucune lettre n'a encore été entrée, l'utilisateur aura besoin d'effectuer une saisie à la place de quatre ou huit (en comptant l'espace). Plus le mot désiré est long, plus le taux d'économie (cf. KSR, Sous-section 3.3) sera rentable. Dans le cas où le mot recherché aurait été le déterminant « a » et que celui-ci serait dans la liste des propositions, un appui aurait malgré tout été économisé grâce à l'espace automatique.²¹

Le système proposé par Bérard et Niemeijer (2004) comprend trois niveaux de prédiction : la complétion, la prédiction du mot suivant et la prédiction de groupe de mots. Ils font remarquer que plus ce niveau est haut, plus la charge cognitive augmente. Selon eux, la prédiction du mot suivant et de groupes de mots offre un meilleur gain que la complétion ; surtout lorsque le mot prédit a déjà été appris par le système (cf. *infra*). Ils ont également montré que plusieurs facteurs affectaient la réduction de l'effort de ce type de systèmes : « l'algorithme de prédiction, le nombre de suggestions affichées, la façon dont elles sont présentées (verticalement ou horizontalement ; alphabétiquement ou par fréquence), l'apprentissage du nouveau vocabulaire et le niveau de prédiction » (Niemeijer, 2005).

Beaucoup de logiciels d'aide à la communication disponibles dans le commerce proposent des claviers virtuels avec une prédiction de mots en français. Pour le verbal, Blache et Rauzy (2007, p. 48) avaient cité : WiViK, qui comprenait déjà une option de défilement permettant de contrôler le système d'exploitation ; Eurovocs Suite, un système dont le module de prédiction Skippy ferait, aujourd'hui, économiser jusqu'à 50 % des frappes, qui évoluerait en fonction de l'usage langagier de l'utilisateur et serait disponible dans plusieurs langues. Nous pouvons aussi mentionner le logiciel d'aide à la lecture et à l'écriture SprintPlus qui, en plus de prédire une liste de mots, propose des alternatives phonétiques. Le système récent Grid 3 sur ordinateur et tablette Windows est doté du moteur d'intelligence artificielle SwiftKey pour la prédiction de mots.

Pouplin (2016, pp. 43–44) avait également cité les logiciels Penfriend et WordQ. D'après lui, ils s'améliorent automatiquement et présentent de nombreux paramètres comme changer le nombre, l'affichage ou le positionnement de la liste de mots, mais aussi la possibilité d'activer le module d'apprentissage et d'adaptation. WordQ permettrait par exemple de « proposer le mot bateau si le mot "pato" a été saisi ». Dans la section suivante, nous expliquerons le fonctionnement de ces algorithmes.

21. En supposant qu'il s'agisse de la fin d'une phrase et que l'utilisateur n'aurait, pour une raison ou une autre, pas voulu de l'espace, il perdrait une saisie pour l'effacer. Néanmoins, dans certains systèmes comme Presage, il existe une option qui, après une espace, permet de coller le dernier mot au point précédemment entré. Dans ce cas, l'utilisateur n'aurait ni gagné, ni perdu une saisie.

2 Systèmes de prédiction

Les nombreux systèmes de prédiction de textes proposent une solution d'optimisation de la vitesse de saisie en temps réel qui vise à réduire le nombre d'appuis nécessaires en se basant sur certaines informations contextuelles : statistiques ou linguistiques. Afin de modéliser le langage et l'adapter à l'utilisateur, il existe différents types d'approches qui peuvent néanmoins être complémentaires et concerner à la fois la prédiction de lettres ainsi que la complétion ou prédiction de mots. Les formalismes que nous décrirons (comme les n-grammaires) peuvent donc être utilisés pour les caractères, mais aussi dans d'autres types de prédicteurs de textes que ceux des CAA.

Nous verrons que tous ces algorithmes prédictifs, intégrés à diverses interfaces utilisateurs, reposent à la fois sur des paramètres, mesures et fonctionnalités hétérogènes rendant la comparaison ainsi que l'interprétation des résultats extrêmement difficile comme le font remarquer Garay-Vitoria et Abascal (2006) dans leur état de l'art. Les auteurs évoquent également la possibilité de prendre en compte d'autres unités linguistiques comme les syllabes et les morphèmes afin de simuler le comportement de l'utilisateur en prédisant le texte qu'il a l'intention de communiquer (écrire ou dire).

Ils mentionnent que les informations statistiques et linguistiques sont stockées dans des tables appelées dictionnaires ou lexiques. Le nombre ainsi que l'organisation de ces structures de données sous la forme de liste d'items ordonnés séquentiellement, d'arbre, etc. varient d'un système de prédiction à l'autre (Figures I.2a et I.2b). Leur contenu peut être soit fixé préalablement par le concepteur et donc identique pour tous les utilisateurs, soit adapté automatiquement en fonction de ceux-ci.

DB Browser for SQLite - C:\Program Files\presage\share\presage\database_en.db

File Edit View Help

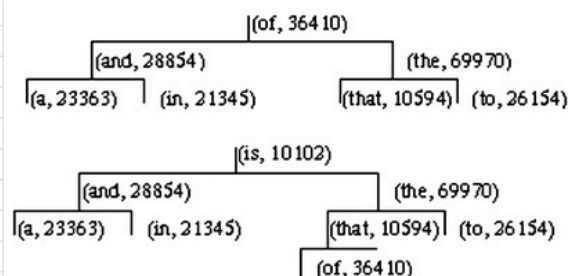
New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Table: _3_gram

	word_2	word_1	word	count
	Filter	Filter	Filter	Filter
30250	is	and	for	1
30251	is	and	my	1
30252	is	and	what	1
30253	is	annoyed	with	1
30254	is	arrested	if	1
30255	is	art	s	1
30256	is	as	bad	1
30257	is	as	good	1
30258	is	asked	lord	1
30259	is	at	once	1
30260	is	at	your	1

(a) Presage : base de données composée de n-grammes de mots et de leur fréquence



(b) Structure en arbre composée de mots et de leur fréquence, reproduite et adaptée de Garay-Vitoria et Abascal (2006)

FIGURE I.2 – Dictionnaire ou lexique

2.1 Approche statistique

Aussi connue sous le nom d'approche stochastique ou probabiliste, la méthode statistique est utilisée dans beaucoup de systèmes de prédiction (Leshner *et al.*, 1999). Elle modélise le langage à l'aide de fréquences et de modèles markoviens de langage (point 2.1.1) basés sur les n-grammes (bigramme, trigramme, etc.), qui sont parfois associés à des techniques stochastiques plus avancées (point 2.1.2) comme le lissage ou certaines interpolations. Il est possible d'adapter ce type de systèmes d'une langue à l'autre, mais également à un vocabulaire ou à un registre de langage plus spécifique.

De nombreux claviers virtuels gratuits ou payants associés à une prédiction de mots existent. Beaucoup de ces prédicteurs utilisent une approche statistique et un clavier statique sans dispositif de défilement. C'est le cas du moteur PolyPredixTM commercialisé par AssistiveWare²² (Bérard et Niemeijer, 2004; Niemeijer, 2005). Selon eux, l'approche stochastique supporte beaucoup de langues différentes et permet plus de liberté à l'utilisateur en matière de style d'écriture bien qu'elle ne garantisse pas que les phrases soient grammaticalement correctes contrairement aux systèmes de type syntaxique. Leur modèle statistique est utilisé pour concevoir les dictionnaires de base des différentes langues proposées (anglais, français et allemand).

Leur système apprend aussi de nouveaux mots et les combinaisons de mots que l'utilisateur saisit. Il s'agit d'une méthode d'auto-apprentissage souvent utilisée dans les CAA selon Hunnicutt et Carlberger (2001). Les fréquences sont réajustées en fonction de l'usage des utilisateurs. Le dictionnaire de PolyPredixTM contiendrait l'information sur les fréquences des mots et sur le moment où l'utilisateur les a entrés, ce qui permettrait de prendre en compte la récence. La faculté d'apprentissage du système implique que celui-ci apprend aussi les erreurs de saisie et les mots mal orthographiés. Selon un manuel, le prédicteur comprend une option qui, si on n'entre pas toujours la même erreur, la supprimera automatiquement après un certain temps.²³

La probabilité d'occurrence des lettres et des mots diffère d'une langue à l'autre. Les tout premiers systèmes – d'abord développés pour l'anglais (Trost *et al.*, 2005) – ne reposaient que sur ces différentes valeurs fréquentielles (Garay-Vitoria et Abascal, 2006). Les éléments proposés étaient ainsi triés sur la base de cet unique critère sans prendre en compte la moindre information, c'est-à-dire indépendamment de leur contexte. Il s'agit d'une méthode très limitée, mais qui nécessite, néanmoins, peu d'espace mémoire et qui permet de réduire le temps d'accès moyen aux touches lors de l'utilisation du clavier à défilement automatique du curseur. D'après Wood (1996), l'approche par fréquence ne s'adapte pas assez à l'usage des utilisateurs; les fréquences n'étant pas basées sur la syntaxe, mais sur des généralités du langage.

22. <http://www.assistiveware.com>

23. Cette fonctionnalité limitant l'apprentissage aux mots correctement orthographiés est reprise dans le manuel 3.5 de Proloquo4Text (<http://assistiveware.com/product/proloquo4text/resources>).

2.1.1 Modèle markovien de langage

L'ordre d'apparition des éléments dans une phrase est souvent décrit par des modèles n -grammes, issus de la théorie de l'information. Aussi utilisés dans d'autres domaines du TAL (comme la reconnaissance de la parole), ils permettent de calculer la probabilité d'occurrence de séquences de n mots, lettres ou caractères à partir d'un corpus d'entraînement, c'est-à-dire $P(m_1...m_n)$ décomposable en une chaîne de probabilités (I.1). Un n -gramme correspond à un modèle génératif de Markov d'ordre n . Dans le cas d'un bigramme ($n = 2$), il s'agit ici de calculer la probabilité d'occurrence du mot en cours de composition à partir du précédent (déjà entré par l'utilisateur). Pour un modèle trigramme ($n = 3$), il faut donc prendre en compte les deux derniers mots. Par conséquent, plus le n est grand, plus la quantité de données nécessaires sera importante entraînant un risque de surapprentissage du modèle qui ne généralise pas. Les n -grammaires sont donc toujours limitées puisqu'elles ne peuvent modéliser les dépendances sur de longues distances ($n > 3$) en se basant sur un contexte suffisant.²⁴

$$\begin{aligned} P(m_1...m_n) &= P(m_1) \cdot P(m_2|m_1) \cdot P(m_3|m_1, m_2) \cdot \dots \cdot P(m_n|m_1...m_{n-1}) \\ &= \prod_{k=1}^n P(m_k|m_1...m_{n-1}) \end{aligned} \quad (\text{I.1})$$

Comme d'autres modèles statistiques, les n -grammes sont dépendants du corpus sur lequel ils sont entraînés (Jurafsky et Martin, 2009, p. 126). Leur constitution représente un coût élevé et soulève toujours beaucoup de questions.²⁵ Par exemple, Guenther *et al.* (1994) critiquaient l'approche statistique en raison de la taille conséquente du corpus dont il est nécessaire de disposer et le fait qu'il n'en existe pas de spécifique pour le domaine de la CAA. Bien qu'ils aient entraîné et testé leur système VITIPI sur un corpus spécialisé lié à la météo, Boissière *et al.* (2012, 2015) affirment qu'il faudrait « un corpus parfaitement adapté au registre de l'utilisateur ». D'après eux, il serait inutile de mettre dans le lexique des mots qui ne seront jamais utilisés par celui-ci. Ils critiquent les résultats de Trnka et McCoy (2007) qui avaient démontré que « pour construire un modèle n -gram performant, il fallait disposer d'une *grande* quantité de textes généraux, d'une quantité *raisonnable* de textes du même style et sur le même sujet, et d'une *petite* quantité de textes similaires à ce que va écrire l'utilisateur ».

Selon Jurafsky et Martin (2009, p. 121), le Web ne serait pas non plus suffisant pour donner de bonnes estimations en raison, notamment, de la créativité du langage. Chaque jour, de nouvelles phrases sont créées; il n'est dès lors pas possible de tous les prendre en compte. La langue évolue en permanence comme en témoignent les plu-

24. Les premiers systèmes prédictifs reposaient sur un modèle unigramme ($n = 1$). Trnka *et al.* (2006) l'ont utilisé pour prédire le premier mot de chaque phrase vu qu'elles ont tendance à commencer par un petit ensemble de mots restreints. Ils citent les déterminants, les marqueurs discursifs, etc.

25. Le choix du corpus étant un paramètre important quand on étudie un système de prédiction statistique, ce dernier fera encore l'objet de discussion dans le Chapitre II.

riels irréguliers, les « langages » issus des réseaux sociaux et les nombreux néologismes qui soulèvent aussi le problème des mots hors vocabulaire (cf. point 2.1.2). Il est, par conséquent, difficile d'anticiper toutes ces évolutions et nous verrons qu'il est nécessaire de pouvoir faire abstraction de certaines richesses du langage. Nous expliquerons comment Antoine (2011) prend en compte la représentativité du corpus pour le calcul des probabilités et traite le problème de l'éparpillement des données.

Les travaux de son laboratoire de recherche, issus de l'Université François Rabelais de Tours (Schadle, 2003; Wandmacher, 2008) concernent aussi bien la prédiction de lettres que de mots; elles sont représentées respectivement par un modèle pentagramme ($n = 5$) et quadrigramme ($n = 4$). SibyLettre tient compte des quatre derniers caractères saisis, tandis que Sibylle se base sur les trois derniers mots. Le clavier dynamique repose sur le réagencement en temps réel des lettres au fur et à mesure que l'utilisateur saisit des caractères. Les lettres les plus probables sont à chaque fois présentées en haut du clavier, réduisant le temps d'accès moyen aux caractères grâce au défilement linéaire. Pour le mot « compter » (Figure I.3), la 2^e lettre s'obtient directement alors que la 3^e ne nécessite qu'un déplacement supplémentaire du curseur.



FIGURE I.3 – SibyLettre : exemple du mot « compter », reproduite d'Antoine (2011)

Toujours afin d'optimiser l'accès aux lettres, Raynal (2007) a présenté un système de prédiction appelé KeyGlass utilisant un modèle bigramme qui consiste à proposer à l'utilisateur des lettres supplémentaires en fonction de celle qu'il vient de saisir. Ces caractères sont affichés à l'aide de nouvelles touches semi-transparentes, les KeyGlasses. Celles-ci viennent à chaque fois se positionner de manière stratégique et récursive sur la lettre sélectionnée (Figure I.4). Comme il le mentionne, ce sont systématiquement les mêmes caractères prédits qui reviennent pour chacune des 26 lettres de l'alphabet, ce qui favorise l'apprentissage chez un utilisateur régulier qui pourrait également gagner en vitesse en anticipant les mouvements relatifs au pointage. Le système ayant évolué, nous verrons qu'il utilise aussi une approche symbolique et hybride.

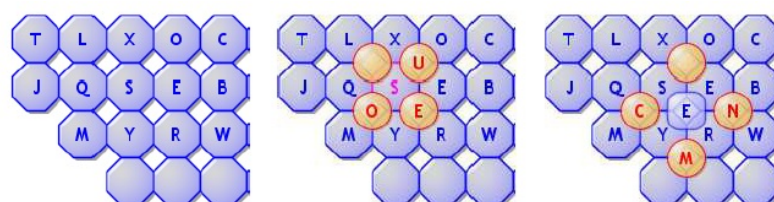


FIGURE I.4 – KeyGlass : exemple de la séquence « se... », reproduite de Raynal (2007)

Les modèles n-grammes utilisés dans les systèmes de prédiction de textes ont souvent tendance à évoluer au cours des étapes de conception de ces prototypes. Les différentes évaluations permettent de complexifier les modèles à l'aide de diverses méthodes statistiques et ainsi tenter de dépasser leurs limites relatives aux corpus. D'après Trnka et McCoy (2007), ces prédicteurs devraient être entraînés et testés sur une langue similaire aux besoins de réels utilisateurs. Cependant, ils ajoutent que vu qu'il n'existe pas de tels corpus, d'autres alternatives doivent être trouvées (cf. *infra*). Ces systèmes étant conçus pour synthétiser des conversations spontanées et variées, ils font néanmoins remarquer que les utilisateurs de CAA pourraient, par exemple, préférer faire des phrases plus courtes sans certaines disfluences. En outre, il convient de rappeler que ces systèmes sont également utilisés pour l'écrit et destinés à différents usages quotidiens, que ce soit pour la rédaction de mails, de documents, d'articles, etc.

2.1.2 Techniques stochastiques avancées

Il existe plusieurs techniques d'adaptation de modèles de langage généraux qui permettent au système de prendre en compte les mots inconnus ou hors vocabulaire ne se trouvant pas dans le dictionnaire, appelés *Out Of Vocabulary (OOV)* en anglais. Selon Trnka (2010), les noms propres en font partie et sont problématiques dans un système de prédiction de textes. La probabilité de retrouver les noms propres que l'utilisateur a l'habitude de saisir dans le corpus d'entraînement est relativement faible. Il avance à cela deux raisons : le fait que les données ayant servi à concevoir le modèle proviennent de différents domaines spécifiques ainsi que le possible décalage entre les événements référencés dans le corpus d'entraînement et ceux de l'utilisateur.

Ces considérations sont importantes. Dans le cas où un mot ne se trouve pas dans le dictionnaire du système, le modèle lui assignera une probabilité nulle et l'utilisateur devra saisir toutes les lettres. Afin de tenir compte des OOVs, le repliement est une technique qui consiste à ôter une part des probabilités pour les réaffecter sur tous les n-grammes. Les n-grammaires ont donc pour inconvénient d'être particulièrement dépendantes du corpus d'entraînement. Pour réduire l'écart entre celui-ci et la diversité des tâches pour lesquelles ces systèmes sont destinés, le lissage par interpolation permet de combiner des sources d'informations hétérogènes en associant, par exemple, un modèle général appris sur un large corpus générique avec un plus petit entraîné « sur les productions de l'utilisateur » (Wandmacher et Antoine, 2007b, p. 79).

$$P_{\text{global}}(m_n) = \lambda_i \cdot P_{\text{utilisateur}}(m_n) + (1 - \lambda_i) \cdot P_{\text{général}}(m_n) \quad (\text{I.2})$$

où $0 < \lambda_i \leq 1$ et $\sum_i \lambda_i = 1$ (λ_i étant un paramètre de lissage). Durant l'évolution du système Sibylle (Figure I.5),²⁶ Antoine (2011) a testé différentes techniques avan-

26. Le logiciel français Sibylle a également été développé en anglais et en allemand. La figure est reproduite du site du projet qui est toujours en cours et accessible librement (<http://k-lab.fr/sibylle/>).

cées afin d'adapter le modèle de langage – interpolé linéairement ici – à l'utilisateur (I.2), mais aussi au registre et au thème du discours. Une de leurs évaluations avait effectivement montré une dégradation du KSR (cf. Sous-section 3.3) pouvant atteindre 13,6 % si le registre du corpus d'entraînement est différent de celui utilisé pour le test (Wandmacher et Antoine, 2007b). Wandmacher (2008) a donc testé leur modèle quadrigramme avec le paramètre λ_i estimé par l'algorithme EM, l'auto-adaptation du dictionnaire, un modèle utilisateur dynamique, l'utilisation d'un modèle cache ou l'analyse sémantique latente sur des corpus de test de registre différent.²⁷

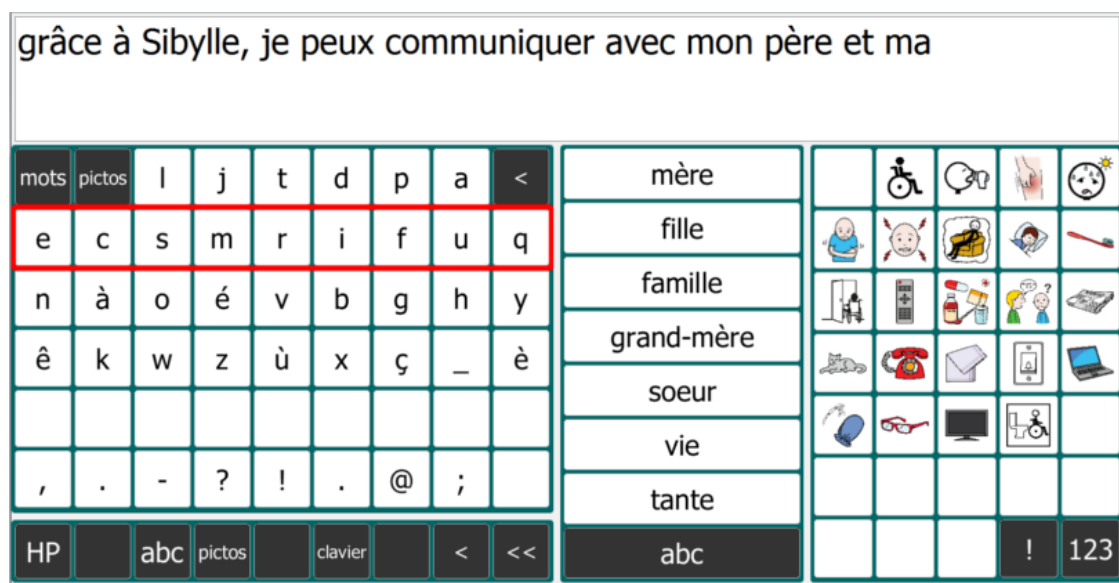


FIGURE I.5 – Sibylle : version K (2014)

Le modèle cache est basé sur le principe de récence, c'est-à-dire qu'il estime que les n derniers mots saisis par l'utilisateur doivent avoir une probabilité d'occurrence plus élevée. Leur valeur fréquentielle est donc augmentée automatiquement suivant une constante prédéfinie et associée à une « fonction de décroissance temporelle » (Wandmacher et Antoine, 2007b, p. 85).²⁸ D'après Hunnicutt et Carlberger (2001), ce mécanisme consistant à « promouvoir » la récence d'un mot (et donc à le faire apparaître plus tôt dans la liste des propositions) avait déjà été exploité par de nombreux prédictors. Ils remarquent que le choix quant à l'ordre de ces mots et au sens du terme « récence » relève davantage de considérations « heuristiques » qu'« algorithmiques ».²⁹ Ils ont également constaté que bien que ce principe soit abondamment évoqué dans la littérature sur les systèmes de prédiction, il existait peu d'études expliquant la façon dont elles implémentaient réellement l'information de récence.

27. Nous reviendrons sur les résultats de ces différentes techniques (cf. Sous-section 2.4, Table I.1).

28. À noter qu'ils ont montré que l'efficacité d'un modèle cache était limitée; c'est la raison pour laquelle nous ne le testerons pas dans Presage bien qu'un algorithme similaire soit déjà implémenté.

29. Autrement dit, ce choix n'est basé que sur la logique et l'expérience. En plus de ce mécanisme de récence, ils abordent d'autres méthodes heuristiques : l'apprentissage de nouveaux mots, la majusculation automatique de la première lettre d'une phrase et l'affichage des formes fléchies des mots.

Nous avons cité l'Analyse Sémantique Latente (ASL). Il s'agit d'un modèle vectoriel souvent employé dans le cadre de l'indexation automatique ou la recherche d'informations. Il permet d'établir automatiquement des relations de similarité entre un contexte et les termes dont ils sont issus. Ainsi, Wandmacher et Antoine (2007a,b) expliquent qu'après « mon père était professeur en mathématiques et je pense que... », le système proposera des mots relatifs à cette thématique comme « professeur » (dont la probabilité est de 0,0117), « mathématiques » (0,0109), « enseigné » (0,0083), « enseignait » (0,0053), « mathématicien » (0,0049), etc. Cette approche de type *bag of words* est connue pour considérer les phrases comme un « sac de mots » qui ne prend pas en compte l'ordre des mots (syntaxe), mais uniquement leur sens (sémantique).

Les auteurs avaient d'abord testé l'analyse sémantique latente sans interpolation (I.3) où un mot m_n est représenté comme un vecteur, h désigne l'historique (m_1, \dots, m_n) qui peut être représenté par la somme des vecteurs, γ est un facteur de température et le dénominateur sert à vérifier que la somme des probabilités est égale à 1.

$$P_{asl}(m_n|h) = \frac{(\cos(\vec{m}_n, \vec{h}) - \cos_{\min}(h))^\gamma}{\sum_k (\cos(\vec{m}_k, \vec{h}) - \cos_{\min}(h))^\gamma} \quad (I.3)$$

Ils l'ont ensuite testée avec un modèle n-gramme interpolé géométriquement (I.4) où n représente le nombre de termes. Ils ont expérimenté d'autres interpolations (linéaire avec ou sans pondération du coefficient par une mesure de confiance) afin de combiner différemment les informations issues de l'analyse sémantique latente.

$$P_{\text{global}}(m_n) = \frac{P_{\text{base}}(m_n)^{\lambda_i} \cdot P_{asl}(m_n)^{1-\lambda_i}}{\sum_{j=1}^n P_{\text{base}}(m_n)^{\lambda_j} \cdot P_{asl}(m_n)^{1-\lambda_j}} \quad (I.4)$$

Trnka *et al.* (2006) avaient également évalué leur modèle n-gramme associé à des techniques de *topic modeling* pour l'anglais, mais en utilisant le corpus de conversations téléphoniques *Switchboard* afin d'adapter les prédictions en déterminant le sujet du message.³⁰ Avant d'appliquer leur modèle à tous les mots, ils ont d'abord cherché à enrichir ce qui est appelé le *Fringe Vocabulary* par opposition au *Core Vocabulary* (c'est-à-dire les mots les plus couramment utilisés qui font souvent partie des classes de mots fermées même si les autres ne sont pas exclus).³¹ Tout comme l'analyse sémantique latente, le *topic modeling* consiste à augmenter la probabilité des mots liés par une relation de similarité et diminuer ceux qui ne le sont pas. Les auteurs ont testé deux méthodes différentes pour intégrer l'information sémantique aux modèles de langage lissés (bigramme et trigramme), mais que nous ne détaillerons pas.

30. Chaque conversation téléphonique (représentant alors un document) est reliée à un des 70 sujets (*topics*). Le corpus est disponible en ligne (<https://catalog.ldc.upenn.edu/LDC97S62>).

31. Ces termes sont souvent employés en AAC. Pino (2014) donne l'exemple suivant : « I want that cookie » où tous les mots proviennent du *core vocabulary*, excepté « cookie » qui n'est ni un déterminant, ni un modal dont il serait possible d'améliorer la vitesse de saisie via des interfaces ou des méthodes spécifiques telles que la technique de *semantic compaction* ou de désabréviation (Trnka *et al.*, 2006).

2.2 Approche symbolique

Nous avons vu que les systèmes statistiques cherchaient à maximiser la probabilité d'occurrence de séquences d'éléments (caractères, lettres ou mots), l'approche symbolique vise plutôt à vérifier leur validité (Schadle, 2003). Afin d'afficher certaines prédictions et améliorer leur pertinence, elle s'appuie sur l'analyse de diverses catégories de connaissances linguistiques *a priori* : morpho-lexicales (point 2.2.1), syntaxiques (point 2.2.2) ou sémantiques (point 2.2.3), reposant sur des formalismes différents ainsi que des règles et transformations qu'il faut parfois encoder manuellement.

L'obtention de ces informations requiert généralement plus d'efforts en termes de temps. Ces systèmes sont souvent compliqués à concevoir et à adapter puisqu'une reprogrammation des règles est nécessaire en cas de changement de langue (Boissière *et al.*, 2015). Cette approche plus précise est néanmoins restreinte à un ou plusieurs domaine(s) particulier(s). Nous verrons qu'elle peut également être utilisée en parallèle avec la correction automatique de fautes de frappe ou d'orthographe « *en ligne* » au moyen de procédures d'inférence (Boissière et Dours, 2001).³²

Certains systèmes ont proposé de combiner différents types de connaissances. Par exemple, le prototype KOMBE (Guenthner *et al.*, 1994; Pasero *et al.*, 1994) prend en compte le lexique, la syntaxe, la sémantique ainsi que des concepts. Il est toutefois limité puisqu'il s'apparente à un dialogue docteur-patient, destiné à guider les personnes souffrant de SLA dans la composition de leurs phrases. Comme le disent Antoine et Maurel (2007, p. 28), il s'agit d'une « communication finalisée, orientée vers un but précis et où le libre arbitre de l'utilisateur est le plus souvent restreint ».

2.2.1 Aspects morpho-lexicaux

Les études et états de l'art consultés relatifs à la prédiction ne distinguent généralement que l'approche statistique et syntaxique, les aspects morphologiques (c'est-à-dire la structure interne des mots) et lexicaux (les mots) étant intrinsèquement liés à cette dernière. Nous avons néanmoins décidé d'aborder ici les systèmes qui sont basés sur des transducteurs (non probabilistes) ainsi que des arbres lexicographiques. Nous introduirons ensuite quelques notions de *Part Of Speech (POS) tagging*, appelé étiquetage morphosyntaxique en français.

Dans le système de complétion VITIPI (Version Interprétant un Texte Imparfaitement écrit pour les Personnes Inexpérimentées), Boissière et Dours (2001) avaient d'abord modélisé la base lexicale (c'est-à-dire le lexique reprenant les mots) par des transducteurs (dé)factorisés non probabilistes avant de tester un modèle trigramme pour élargir le contexte (Boissière *et al.*, 2012). Développé à l'Institut de Recherche en

32. L'inférence désigne la capacité d'un système à déduire des informations (ou tirer des conclusions) à partir de connaissances qui n'ont pas été encodées explicitement.

Informatique de Toulouse (IRIT), le projet propose de se baser sur les morphèmes, autrement dit, les constituants d'un mot, soit le plus petit élément linguistique qui possède une fonction grammaticale ou sémantique.

Ce système présente une façon de traiter le problème des flexions très fréquentes en français et qui concernent aussi bien les noms que les verbes. Lorsque l'utilisateur commence, par exemple, à entrer un nom comme « égalisateur » ou « directeur » (Figure I.6), il est possible de deviner les lettres des radicaux « égalisat » et « direct » en ne tenant pas compte des différents suffixes flexionnels possibles comme « eur », « rice » ou « ion ». Le système complète d'abord le radical qui est une partie jugée relativement non ambiguë. L'utilisateur peut alors saisir la première lettre du morphème flexionnel « e », « r » ou « i » et les caractères restants s'afficheront automatiquement.

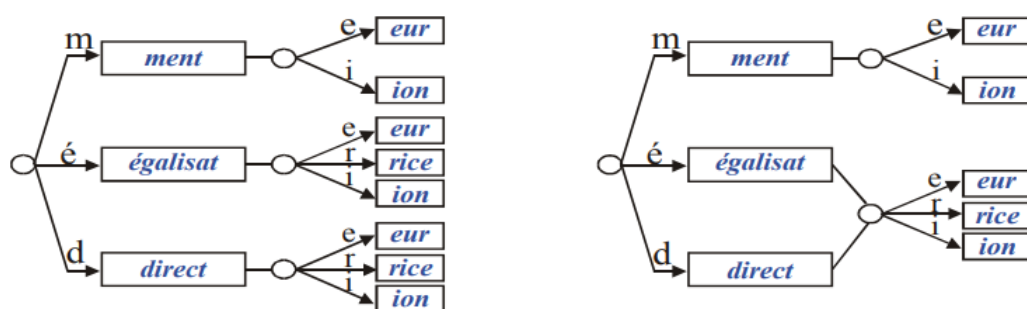


FIGURE I.6 – VITIPI : transducteur défactorisé et transducteur factorisé, reproduite de Boissière et Dours (2001)

Un transducteur est un automate composé de transitions qui comprennent des états, c'est-à-dire une entrée (ici la lettre saisie par l'utilisateur) et une sortie (les lettres affichées par le système), mais dont l'action est l'affichage des chaînes de caractères. Les auteurs ont conçu une technique de factorisation afin de regrouper les propriétés morpho-lexicales (ou fins de mots) communes « sans inférer de mots nouveaux dans la base » (Boissière et Dours, 2001). Afin de traiter le problème des OOVs et différentes altérations possibles (fautes de frappe ou d'orthographe entraînant une substitution, suppression ou élision de caractères), ils ont développé des procédures d'inférence.

Certaines de ces inférences servent à prédire de nouveaux mots par analogie d'états. Le système peut être construit à partir d'un corpus de mots standard ou ceux de l'utilisateur. Ils ont constaté que plus le vocabulaire de leur système était grand, plus celui-ci serait confronté à des ambiguïtés. Autrement dit, l'utilisateur devra alors saisir plus de lettres pour les lever, ce qui implique une baisse des performances. Or, une réduction de la taille du vocabulaire restreindrait également les possibilités de l'utilisateur. Ils ont, par conséquent, décidé de prendre en compte les mots précédents afin de limiter le choix des mots susceptibles d'être choisis à un instant t .

Également développés à l'IRIT, les deux systèmes de prédiction KeyGlass (Raynal, 2007) et WordTree (Badr et Raynal, 2009; Badr, 2011) fonctionnent avec un arbre

lexicographique constitué à partir d'un corpus de plus de 130 000 mots. Il s'agit d'une autre façon de représenter le langage. Dans WordTree (Figure I.7), elle permet à l'utilisateur de ne sélectionner qu'une partie des mots. Afin de rentabiliser le temps que l'utilisateur perd en consultant la liste de prédictions (parfois inutilement si le mot désiré ne s'y trouve pas), les auteurs proposent de considérer les lettres comme des mots que l'utilisateur sélectionnerait dans une liste classique.

Contrairement aux automates et transducteurs qui reposent sur des transitions et des états, l'arbre lexicographique (Figure I.7a) est basé sur des nœuds et des feuilles. Une lettre est ici représentée par un nœud, tandis qu'un mot est modélisé par le chemin qui relie la racine aux feuilles.³³ Supposons que l'utilisateur entre « bo » et que le mot désiré « bouleversant » ne s'affiche pas contrairement à « bouleverse », il peut toujours cliquer sur le « s » pour obtenir « boulevers ». Il économise ainsi directement les lettres qui précèdent le caractère sélectionné. Leur module de prédiction a par exemple été testé avec un clavier logiciel AZERTY et une liste verticale (Figure I.7b).

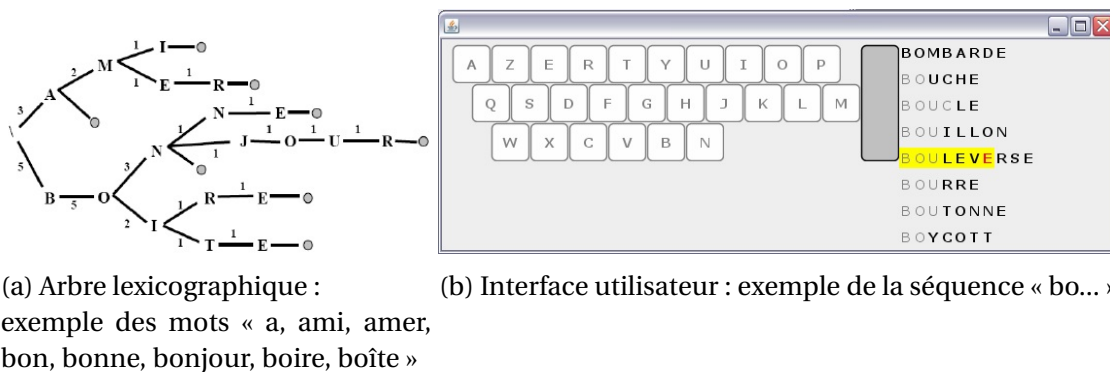


FIGURE I.7 – WordTree, reproduite de Badr et Raynal (2009)

Des systèmes de prédiction ont exploité les catégories lexicales des mots. Appelée *Part Of Speech (POS)* en anglais, les parties du discours permettent de distinguer les termes entre eux. Les catégories les plus courantes sont le nom, le verbe, l'adjectif, l'adverbe, le déterminant et la préposition. Elles sont regroupées dans des *tagsets* dont la taille varie fortement, par exemple, le *French Treebank* en compte 13 et le *Penn Treebank* 36 (ou 45 avec la ponctuation). Nous verrons lorsque nous présenterons l'approche hybride que le corpus *Le Monde* du *French Treebank* a par ailleurs été utilisé dans la PCA (Blache et Rauzy, 2007), mais avec 24 catégories morphosyntaxiques.³⁴

Les systèmes de prédiction syntaxique reposent sur des dictionnaires ou corpus annotés dont chaque mot a été associé à des informations grammaticales à l'aide d'étiquetteurs linguistiques, statistiques ou hybrides. L'analyseur Cordial a servi à concevoir l'une des versions de Sibylle (Schadle, 2003). À noter qu'un mot peut appartenir à plu-

33. À noter que le passage d'un nœud à l'autre est ici soumis à un poids afin de prendre en compte l'utilisation régulière des préfixes, ceux-ci sont incrémentés de 1 suite à l'ajout d'un nouveau mot.

34. Ils citent, par exemple, le $Noun_c$ (commun), le $Noun_p$ (propre), le V_{ms} (participe passé), le V_{mc} (conjugué), etc. dont des catégories plus sensibles aux traits de genre, nombre et personne.

sieurs catégories. Le système HandiAS (Maurel *et al.*, 1998) utilise une base de connaissances contenant les mots et les parties du discours. Elle stocke par exemple « danse » en tant que nom (féminin singulier) et verbe (1^{re} du singulier). La phrase en cours de saisie est analysée avec une grammaire formalisée par des automates stochastiques.

2.2.2 Aspects syntaxiques

Les propriétés syntaxiques des langues naturelles ont également été intégrées au sein de systèmes prédictifs. Elles sont encodées dans des grammaires reprenant un ensemble de règles plus ou moins complexes. Il est nécessaire de les reprogrammer si on veut adapter le système à d'autres langues ou à un registre de langage spécifique, certaines structures syntaxiques n'étant pas présentes dans tous (Boissière *et al.*, 2015).³⁵ En TAL, l'analyse grammaticale ou *parsing* a pour but de reconnaître une phrase (ici celle de l'utilisateur) et de lui assigner une structure syntaxique. Dans le cadre de la prédiction, elle permet de restreindre le choix des mots qui seront proposés. Les règles s'appuient sur la catégorie morphosyntaxique des mots (cf. point 2.2.1), leurs informations morphologiques et syntaxiques étant stockées dans un dictionnaire.

Wood (1996) justifie le recours à la prédiction syntaxique en soulignant que beaucoup de systèmes prédictifs proposent des mots qui rendent la phrase grammaticalement incorrecte. Dans ce cas, l'utilisateur perd du temps à consulter la liste de prédictions et lire les propositions, ce qui engendrerait une augmentation de la charge cognitive et de la frustration inutile vu que le système n'apporterait alors aucun bénéfice. D'après Garay-Vitoria et Abascal (1997), un système de prédiction n'est néanmoins pas un correcteur orthographique destiné à corriger les fautes que produirait l'utilisateur. Afin d'afficher de bonnes prédictions lexicales et améliorer la vitesse de la communication, il ne serait donc pas nécessaire d'encoder une grammaire complète couvrant tous les cas possibles et suffisant de fournir les règles les plus probables.

Dans sa thèse, Wood (1996) a proposé d'utiliser une grammaire générative de type CFG (*Context-Free Grammar*) dans son prototype Windmill pour la prédiction des mots anglais (cf. Figure I.10a). Pour la conception de son lexique, il a adapté le *British National Corpus Basic C5 Tagset*. La notation employée pour formaliser les règles structurant les phrases est appelée une *Grammar Description Language* (ou *GDL*). Pour représenter la phrase « the cat eats the mice » (Figure I.8), le système nécessite l'encodage d'au moins quatre règles grammaticales : $SENT \rightarrow NP + VP$, $VP \rightarrow V$, $VP \rightarrow TV + NP$ et $NP \rightarrow ART + N$ où *SENT* désigne la phrase, *VP* le syntagme verbal, *NP* le syntagme nominal, *V* le verbe, *TV* le verbe transitif, *N* le nom et *ART* l'article.³⁶

35. La langue varie en fonction de divers facteurs comme l'âge ou le registre (dialectal, populaire, technique, etc.). La linguistique variationniste étudie par exemple la variation diastatique (relative aux différences sociales des locuteurs), diaphasique (stylistique) ou diamésique (liée au canal utilisé).

36. Afin de dépasser les limites des grammaires CFG qui rencontrent des problèmes d'accord, il a utilisé des suffixes dans ses règles. Exemple : $VP(singular) \rightarrow VVT(third\ person\ singular)NP(plural)$.

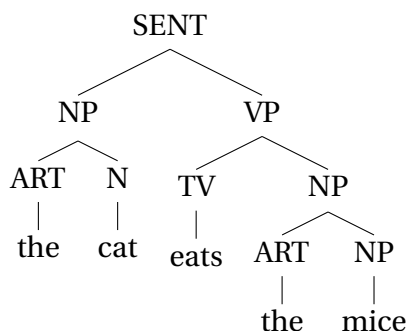


FIGURE I.8 – CFG : exemple de la ph. « the cat eats the mice », adaptée de Wood (1996)

Ses résultats ont montré que quel que soit le corpus de test utilisé (articles du *Telegraph*, littérature scientifique ou transcriptions de conversation), la prédiction syntaxique était inférieure à la méthode statistique en termes de taux d'économie de saisie (cf. KSR, Sous-section 3.3), à savoir : 37,29 % contre 50,31 % en moyenne. La combinaison des deux approches (*infra*) a toutefois obtenu le meilleur pourcentage (55,10 %). La prédiction syntaxique est souvent utilisée avec des informations statistiques.

Nous verrons que c'est également le cas du système testé par Garay-Vitoria et Abascal (1997) ainsi que dans SibyMot (Schadle, 2003). Le premier est basé sur une méthode de *chart parsing* (cf. Figure I.10b), tandis que le deuxième repose sur un *shallow parsing*, deux techniques permettant de gérer les ambiguïtés différemment. Le système de Garay-Vitoria et Abascal (1997) qui sera expliqué dans l'approche hybride utilise une méthode de type *bottom-up*. Elle consiste à parcourir l'arbre syntaxique en partant des mots de la phrase. Garay-Vitoria et Abascal (2006) ont constaté que la technique inverse (*top-down*) avait aussi été appliquée dans le cadre de la prédiction.

2.2.3 Aspects sémantiques

Certains projets ont tenté d'encoder des connaissances sémantiques afin de formaliser le sens des mots et ainsi limiter la liste de prédictions comme le ferait l'analyse sémantique latente. Garay-Vitoria et Abascal (1997, p. 242) relevaient déjà la « difficulty in associating semantic content to every word, and to process it in a reasonably short time ». Asnani *et al.* (2015) confirment que cette approche reste la plus difficile à implémenter car la spécification de catégories sémantiques pour chaque mot est toujours une tâche compliquée ; ils notent aussi que les méthodes d'allocation automatique (*semantic clustering, classification, etc.*) ne permettent pas de le faire en temps réel.

Des prototypes de prédiction ont néanmoins représenté la sémantique des mots ainsi que leur contexte. Selon Schadle (2003), le système Compansion aurait été doté d'un analyseur sémantique permettant de reformuler une phrase grâce à la grammaire des cas de Fillmore. Le prototype KOMBE (Guenthner *et al.*, 1994) que nous avons déjà cité exploite les autres niveaux de représentation du langage à l'aide de formalismes écrits en Prolog, un langage de programmation basé sur la logique des prédicats du

premier ordre (*First-Order Logic*).³⁷ Ils sont encodés dans divers fichiers (GrammaireDoc, AccesLexDoc, LexiqueDoc, etc.). Ainsi, le SemanticDoc contient des règles de composition sémantique alors que le ConceptuelDoc détaille les contraintes conceptuelles reliées aux différents domaines, un extrait des fichiers français a été repris :

```
/* SemanticDoc (p. 45) */
groupe_verbal =
< x, et( [ [verbe_1 , x] , y] , [ groupe_nominal , y] ) >; % regles;

/* ConceptuelDoc (pp. 46–47) */
être_vivant.humain.non_humain; % domaines;
humain.ou(homme.femme, adulte.enfant);
je(humain); % contraintes;
est_fruit(aliment);
manger(humain);
manger(humain, aliment);
```

Les fichiers reprenant ces nombreuses règles ne sont donc pas identiques pour le prototype allemand. Développée dans le cadre du programme européen TIDE,³⁸ l'application KOMBE vise à aider le patient à composer ses phrases dans un ensemble de domaines restreint (santé, hygiène, nourriture, etc.). Contrairement aux autres systèmes de prédiction, celui-ci analyse la partie gauche de la phrase afin de proposer à l'utilisateur une liste contenant tous les mots possibles du dictionnaire qui rendent la phrase correcte d'un point de vue lexical, syntaxique et sémantique. Après « je peux manger / avaler des », seuls les noms communs au pluriel qui désignent des objets pouvant être mangés / avalés par un humain (aliment) seront proposés, éliminant les mots peu probables et réduisant ainsi le nombre de saisies effectuées par l'utilisateur.

La logique du premier ordre étant à la fois coûteuse en termes de ressources à développer et sujette à l'ambiguïté des langues naturelles vu que les mots sont ambigus (surtout si plusieurs domaines sont pris en compte), des auteurs comme Copestake (1997) ou Vescovi (2004) ont évoqué la possibilité d'utiliser la base lexicale WordNet.³⁹ Elle comprend des ontologies et des ensembles de mots liés par des relations sémantiques (*synsets*). Gelšvartas *et al.* (2016) les utilisent pour extraire les synonymes, les hyperonymes ou les hyponymes des mots entrés par l'utilisateur. Ces informations sont intégrées à un modèle de langage n-gramme destiné à prédire les OOVs quand aucun mot ne serait disponible dans le dictionnaire du prédicteur pour la saisie en cours.

37. La logique du premier ordre est un langage de représentation d'une phrase qui lie un ou plusieurs argument(s) (terme(s)) par un prédicat (relation) de la forme : prédicat(x,y).

38. Le projet *Technology Initiative for Disabled and Elderly People* s'appuie sur la première version du logiciel ILLICO qui a été conçu au Laboratoire d'Informatique de Marseille (Pasero *et al.*, 1994).

39. Développée initialement en anglais et accessible en ligne, WordNet (<http://wordnet.princeton.edu/>) dispose aujourd'hui de versions multilingues (<http://compling.hss.ntu.edu.sg/omw/>), dont le Wordnet Libre du Français (<http://alpage.inria.fr/~sagot/wolf.html>).

2.3 Approche hybride

L'approche hybride ou mixte tente de combiner les avantages des deux types de connaissances que nous venons de voir, c'est-à-dire qu'elle tient compte des informations issues d'études statistiques et symboliques tout en tentant de minimiser leurs inconvénients respectifs (Boissière *et al.*, 2015). L'apprentissage des modèles de langage stochastiques se fait toujours automatiquement à partir de corpus plus ou moins représentatifs, mais des connaissances linguistiques explicites (comme des attributs (morpho)syntaxiques ou sémantiques) sont ajoutées. Leur pondération détermine l'ordre des mots auxquels ces différentes ressources sont rattachées. Badr (2011, p. 21) précise néanmoins que « ces propriétés ne sont pas spécifiées directement à la conception », il ajoute que cette approche consistant à fusionner plusieurs méthodes prédictives a été employée « par divers systèmes et s'est montrée assez performante du point de vue de la qualité des lettres et des mots prédits ».

Copestake (1997) avait par exemple testé un modèle de langage bigramme comprenant les probabilités de transition par mots associé aux POS extraites du *Penn Treebank*, le corpus annoté anglais déjà cité. Elle mentionne avoir constaté une dégradation des performances en raison des données d'entraînement collectées qui, selon elle, comportaient beaucoup de questions, de phrases à l'impératif et d'interjections. Elle a ensuite étendu le tagset à 80 tags et a confirmé qu'elle obtenait de meilleurs résultats, la spécification de POS permet de mieux prédire certains accords.⁴⁰ Elle a aussi travaillé sur la cogénération, une technique similaire au fonctionnement du système KOMBE (Guenthner *et al.*, 1994). Elle permet d'aider l'utilisateur à composer ses phrases puisque ce dernier n'a qu'à compléter des schémas prédéfinis. L'approche proposée ici nécessiterait l'utilisation de connaissances statistiques sur les collocations de mots et symboliques via une grammaire syntaxique et des ontologies sémantiques.

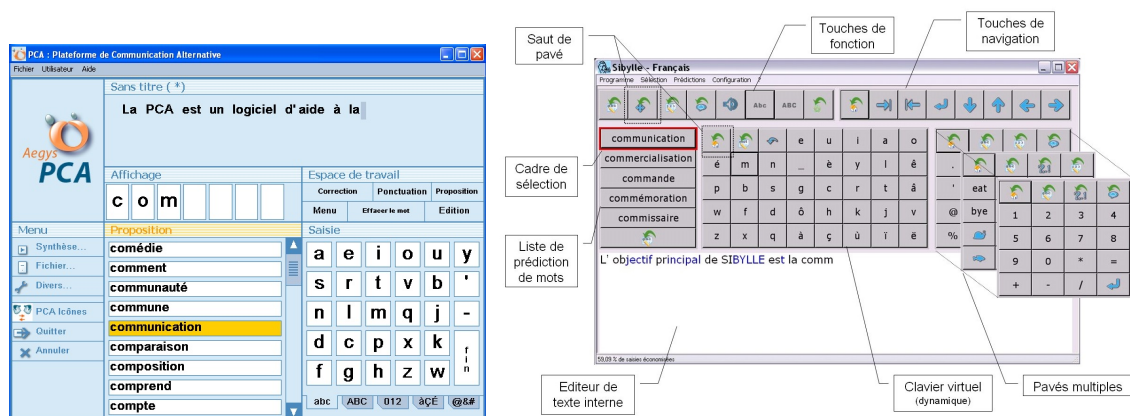
Le système français HandiAS (Maurel *et al.*, 1998; Maurel et Le Pévédic, 2001) a été développé à l'Institut de Recherche en Informatique de Nantes (IRIN). Il propose une prédiction lexicale et syntaxique sous la forme d'une liste de mots. La partie statistique est constituée d'un dictionnaire fréquentiel contenant le nombre d'occurrences des mots fléchis et ce qu'ils appellent un Numéro de Coefficient d'Usage (NCU) des lemmes. La partie symbolique s'appuie sur un schéma de phrases et la notion d'acceptabilité. Il s'agit d'automates finis (ou *Finite-State Automata* en anglais) probabilistes qui sont évolutifs et dont les transitions renvoient vers des chaînes nominales, verbales ou des ruptures comme les conjonctions ou la ponctuation. Ces automates reprennent donc les différentes catégories qui seront associées aux fréquences. Ils peuvent être étendus à de nouvelles règles grammaticales. Le corpus d'apprentissage syntaxique

40. Dans sa thèse, Trnka (2010) présente des études montrant que les modèles n-POS offrent de moins bonnes prédictions que de simples modèles n-grammes étant donné le manque de spécificité du *Penn Treebank* qui ne permettrait pas de représenter tous les aspects syntaxiques (et sémantiques).

qui a aussi servi de corpus de test était la *Charte de l'environnement Malgache*.

Le prototype a été doté d'un module d'adaptation à l'utilisateur qui permet une introduction manuelle ou automatique de nouveaux mots. Si les mots sont connus, leur nombre d'occurrences est incrémenté automatiquement. S'il s'agit, au contraire, de OOVs, le système génère des hypothèses en fonction de l'analyse de surface de certains traits (genre, nombre, temps) liés aux catégories. Il attribue alors un NCU provisoire peu élevé aux mots pour la session courante. L'utilisateur confirme leur intégration dans le dictionnaire s'il compte les ré-utiliser et le NCU est recalculé. La partie morphosyntaxique analyse uniquement le contexte gauche de la phrase. Les automates ne sont pas déterministes car ce sont les fréquences associées à leur transition qui déterminent le choix du chemin à suivre. Le modèle de langage est simple vu qu'il ne fait qu'une analyse de co-occurrences de mots. Ce système d'aide à la communication est resté à la phase de prototype et n'a pas pu être testé en situation réelle.

Développée au Laboratoire Parole et Langage de l'Université d'Aix-Marseille, la Plateforme de Communication Alternative (PCA) de Blache et Rauzy (2007) est un logiciel d'aide à la communication (Figure I.9a).⁴¹ Tout comme Sibylle (Figure I.9b),⁴² l'interface utilisateur de la PCA intègre une liste de prédictions verticale, un éditeur de texte interne, un clavier virtuel (ici statique), mais dont l'organisation des lettres a été modifiée et qui comprend des sous-claviers ainsi que plusieurs touches de fonction. Le moteur de prédiction utilise un lexique français général de 320 000 mots contenant les fréquences d'usage et les traits morphosyntaxiques. La grammaire probabiliste est construite à partir d'un corpus arboré, le *French Treebank*. Le système vérifie d'abord les fréquences de mots de façon à proposer ceux qui sont les plus fréquents avant que la prédiction morphosyntaxique ne modifie « les fréquences lexicales en fonction du contexte syntaxique du mot en cours de saisie » (Blache et Rauzy, 2007, p. 53).



(a) PCA : mode verbal, reproduite de Blache et Rauzy (2007)

(b) Sibylle : version 3.5

FIGURE I.9 – Interface utilisateur : exemple du mot « communication »

41. Il est téléchargeable sur le site du projet (<http://www.lpl-aix.fr/~pca/PCAWeb/fr/accueil.html>).

42. La figure est reproduite du site (http://www.info.univ-tours.fr/~antoine/SIBYLLE/Sibylle_fr/).

Le moteur de prédiction se base sur le modèle des patrons, une sous-classe des modèles de Markov cachés (ou *Hidden Markov Model* en anglais). Contrairement aux n-grammes, les états du modèle sont des séquences de catégories de longueur variable, par exemple : $P(\text{Det}, \text{Noun})$ ou $P(\text{Det}, \text{Adj}, \text{Noun})$. Comme HandiAS, la PCA intègre un algorithme pour gérer les OOVs, c'est-à-dire un modèle basé sur les fréquences d'usage des mots de l'utilisateur qui sont stockées dans son lexique personnel grâce à un module d'apprentissage.⁴³ En raison de la bonne couverture lexicale, ces OOVs peuvent, par défaut, être classifiés en tant que nom propre. Ils font l'objet d'une validation auprès de l'utilisateur avant d'être ajoutés et fusionnés au sein du lexique général. Afin de prendre en compte différents types de handicaps, le système propose deux sortes de communication : verbale (orthographique) ou non verbale (par pictogrammes avec reformulation iconique). Le logiciel a été distribué par la société Aegys de 2003 à 2011.

Dans le projet européen FASTY (FASter TYping for disabled persons, IST-2000-25420), Trost *et al.* (2005) ont testé plusieurs méthodes dont un modèle markovien de langage pour la prédiction. Il ne s'agit pas d'un modèle unigramme comme la PCA, mais d'un bigramme de mots $P(m_n|m_{n-1})$ interpolé linéairement avec le coefficient λ_i estimé par l'algorithme EM et associé à un modèle triclasse de POS $P(c_n|c_{n-1}c_{n-2})$ comme formulé en (I.5) où h désigne l'historique. Le système comprend une grammaire modélisée par des transducteurs à états finis et deux dictionnaires (général ainsi que celui de l'utilisateur). Tout comme KOMBE (Guenthner *et al.*, 1994), le projet est multilingue. Il a été développé en français, italien, allemand, néerlandais et suédois. Le but de FASTY était de concevoir un système de prédiction qui soit adaptable dans plusieurs langues européennes. Le choix d'intégrer des informations morphosyntaxiques a été facilité par l'existence d'étiqueteurs de POS disponibles pour ces langues.

$$P(m_n|h) \approx \lambda_i \cdot P(m_n|m_{n-1}) + (1 - \lambda_i) \cdot P(m_n|c_n) \cdot P(c_n|c_{n-1}c_{n-2}) \quad (\text{I.5})$$

Nous avons évoqué le système KeyGlass (cf. Figure I.4) qui visait à réduire la distance nécessaire entre les lettres grâce à l'ajout dynamique de touches (Raynal, 2007). L'auteur a évalué les performances théoriques en termes de taux de caractères correctement prédits des deux prédictors étudiés. En faisant varier leur pondération respective, il a ainsi pu déterminer laquelle permet d'obtenir les meilleurs résultats lorsque quatre KeyGlasses sont proposées. En combinant les prédictions du bigramme de lettres à 10 % et celles de l'arbre lexicographique (Figure I.7a) à 90 %, il a obtenu un taux de bonnes prédictions de 74,86 %, c'est-à-dire plus que ce qu'il avait obtenu pour les systèmes individuels (61,15 % et 62,40 %). À noter que les simulations de saisie ont été réalisées sur les mots de la version 3 de l'Officiel du Scrabble.⁴⁴

43. Ils ont constaté que l'apprentissage de leur modèle utilisateur était rapide, autrement dit, une amélioration du taux d'économie de saisie (cf. KSR, Sous-section 3.3) de 2 % après 2 000 mots appris.

44. <https://www.fisf.net/officiel-du-scrabble/presentation.html>

La version du système Windmill qui a montré les meilleures performances en combinant les deux approches présente chaque entrée de son lexique ainsi : « VVT,2; NN0,1 = view : 435 , 57 » (Wood, 1996, p. 74). Dans cet exemple, le verbe transitif à la 2^e personne (VVT,2) et le nom singulier (NN0,1) « view » ont tous les deux un coefficient de récence de 435 et une fréquence d'usage dynamique de 57. La grammaire CFG fonctionne grâce aux connaissances morphosyntaxiques et les données statistiques permettent d'ordonner les mots dans la liste de prédictions. Supposons que l'utilisateur entre « I », le système proposera les verbes qui correspondent aux structures dans la *Sequence List* sur la base de la pondération du coefficient de récence et de la fréquence d'usage affichée à côté des prédictions (Figure I.10a). Garay-Vitoria et Abascal (1997) ont quant à eux utilisé une grammaire où chaque règle est associée à un poids qui est adapté en fonction de la fréquence d'usage des utilisateurs (Figure I.10b).

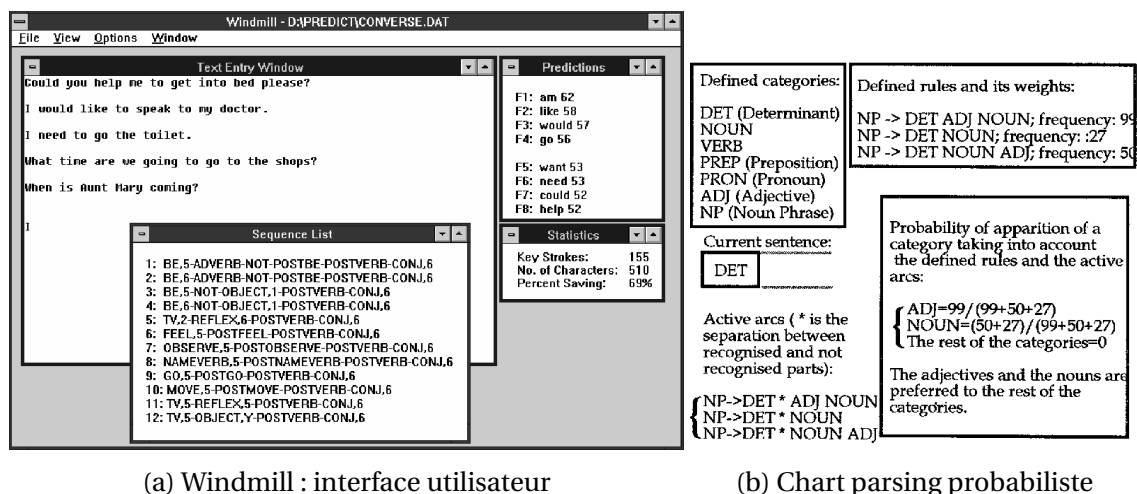


FIGURE I.10 – Fréquences et grammaire, reproduite de Wood (1996) ainsi que de Garay-Vitoria et Abascal (1997)

Comme évoqué au point concernant les aspects syntaxiques, les auteurs ont utilisé une méthode de programmation dynamique, le *chart parsing* qui permet d'éviter de devoir répéter une analyse en conservant des morceaux d'arbres en mémoire. Le système vérifie si la catégorie syntaxique du mot saisi et celles du contexte correspondent aux constituants d'une règle devenant alors « active ». Ils ont également associé des informations statistiques à leurs différentes règles afin de restreindre le choix des propositions en prenant en compte les mots et les catégories les plus probables. Malgré la plus haute complexité du système, leurs résultats ont montré que quel que soit le nombre de propositions affichées, l'ajout de l'information syntaxique n'apportait qu'un gain de 1 à 3 % en termes de taux d'économie de saisie (cf. KSR, Sous-section 3.3). Comme l'ont fait remarquer Trnka *et al.* (2006), Garay-Vitoria et Abascal (1997) ont testé leur système avec une technique appelée le rejet implicite.⁴⁵

45. Elle suppose que l'utilisateur sélectionnera toujours le mot désiré dès qu'il apparaît dans la liste. S'il ne le choisit pas et entre des lettres, cette proposition sera alors exclue de toute autre prédiction.

La première version de Sibylle a permis d'évaluer une modélisation stochastique trigramme intégrant des connaissances de type syntaxique (Schadle, 2003). SibyMot est doté d'un modèle structurel de langage à têtes de *chunks* (tri-*chunks*). Basés sur une analyse syntaxique de surface ou *shallow parsing*, il s'agit de prendre en compte les segments minimaux non récursifs présentés, par exemple, sous la forme de : [le chat*] [est sorti*] [,] [le téléphone*] [a encore sonné*] où * désigne les têtes des *chunks*. Ce système a pour but d'étendre la taille du contexte de quelques mots à l'ensemble de la phrase tout en évitant d'avoir un nombre plus élevé de paramètres au sein du modèle. La représentation de la phrase passe par plusieurs étapes, dont le POS tagging suivi de la segmentation de l'énoncé en *chunks*. L'analyseur identifie des informations au niveau du mot et du *chunk* comme le lemme, la flexion, le rôle ou l'étiquette (parmi un ensemble de 99 tags pour les mots et 61 pour leur grammaire des *chunks*).

2.4 Synthèse des systèmes

Les systèmes décrits dans cette section ont donc permis de montrer que diverses interfaces utilisateurs et approches avaient été envisagées. Bien que robustes, les modèles de langage statistiques sont dépendants de la représentativité des larges corpus avec lesquels ils ont été construits et testés. Le choix de ceux-ci reste une question dominante dans beaucoup d'applications, y compris dans le domaine de la prédiction ; c'est la raison pour laquelle nous avons choisi de l'approfondir dans notre étude de cas. Wandmacher et Antoine (2007b, p. 79) avaient par ailleurs constaté qu'en comparaison avec les « travaux classiques sur l'adaptation de modèles, la spécificité de la communication assistée réside dans le fait que les données utilisateurs resteront toujours limitées, du fait de l'extrême lenteur de la saisie sur les systèmes AAC ».

En plus de la problématique du corpus et des OOVs qui peuvent aussi être liés à des erreurs de saisie (fautes de frappe ou d'orthographe que l'utilisateur oublierait de corriger), nous avons vu que les modèles markoviens de langage étaient limités à cause du risque de surapprentissage qui se manifeste par l'éparpillement des données entraînant une dégradation des performances prédictives. Les *n*-grammes ne permettent de modéliser qu'un contexte restreint à quelques mots. Afin de réduire ces inconvénients, des techniques stochastiques avancées comme le lissage et les interpolations sont testées. L'analyse sémantique latente est aussi une méthode statistique plus complexe ne se limitant pas à quelques mots, mais qu'il est nécessaire de combiner à un modèle *n*-gramme autrement il ne pourrait pas prédire certains mots comme les *stop words*.

Les prédictions des systèmes symboliques sont relativement plus précises, mais ces systèmes restent limités à un domaine spécifique et sont plus coûteux à concevoir que ceux basés sur une approche statistique car les règles sont encodées par des linguistes. Il faut également plus de connaissances différentes et il est difficile d'adap-

ter le système d'une langue à l'autre. À notre connaissance, il n'existe aujourd'hui pas de logiciels prédictifs purement symboliques. Bien que l'approche statistique soit plus générale, nous avons évoqué le fait que les systèmes commerciaux se basaient souvent sur les fréquences sans prendre en compte le contexte avec une simple liste de mots constituant le lexique. Plusieurs prototypes de recherche ont testé des modèles n-grammes associés à d'autres types de ressources (n-POS, n-classe, n-*chunk*, etc.).

Ces différentes grammaires probabilistes ont pour but d'étendre et de préciser le contexte de prédiction vu que les modèles n-grammes prennent souvent en compte les trois derniers mots et que certaines contraintes grammaticales ne pourraient pas être respectées. Si nous prenons, par exemple, la phrase de Schadle (2003, p. 71) : « les pommes de l'archiduchesse sont cuites » ; il est difficile d'accorder le verbe « cuites » avec le mot dont il dépend, à savoir : « pommes ». Pour faire fonctionner ces systèmes, il est nécessaire de disposer d'une grande quantité d'informations accessibles sous la forme de corpus. Leur annotation pour les langues européennes peut aujourd'hui se faire automatiquement à l'aide d'étiqueteurs de POS très précis (Trost *et al.*, 2005).

Enfin dans le cadre spécifique de la prédiction de mots, nous pouvons nous demander si les deux approches remplissent le même objectif. Nous avons vu qu'un système symbolique imposait le plus souvent de respecter la syntaxe et l'orthographe de la langue, or cela peut s'avérer très difficile pour certains patients cérébrolésés. Nous pourrions donc considérer que les systèmes symboliques sont plus utiles en apprentissage ou rééducation de la langue pour aider les utilisateurs à apprendre la grammaire. Dans ce cas, ils pourraient servir les logopèdes ou les enseignants. Alors qu'un système statistique, bien plus souple, sera adapté avant tout à la communication sans finalité autre qu'échanger. Cette approche serait privilégiée par les ergothérapeutes. Ainsi, ce paragraphe s'inspire directement des réflexions d'Antoine et Maurel (2007, p. 29) :

“ Le problème qui se pose ici n'est donc pas celui de la robustesse, mais de sa gestion du point de vue de l'utilisateur. Lorsqu'un début d'énoncé est agrammatical, le système ne peut plus effectuer aucune prédiction [...] Cette question est très sensible dans le cas de l'aide au handicap où les erreurs de saisie sont fréquentes et où la maîtrise de la langue est parfois imparfaite (enfants en cours d'apprentissage, troubles langagiers associés...). Ces approches ne sont donc pas utilisables en pratique pour l'aide à la communication et aucun système commercial ne les emploie. Elles retrouvent en revanche une utilité dans la perspective d'une aide à l'apprentissage ou à la rééducation [de la langue]. Dans ce cas, il est très important que le système n'accepte que des énoncés syntaxiquement corrects.

”

Nous avons vu que ces systèmes disposaient souvent de fonctionnalités additionnelles qui permettent d'éviter certaines saisies comme l'ajout automatique d'espace après un mot, la majuscule automatique en début de phrase, mais également la prise en compte de nouvelles entités lexicales grâce à des modèles ou des dictionnaires propres à l'utilisateur. Comme nous venons de l'évoquer, les systèmes de prédiction symbolique peuvent éviter de faire certaines fautes (Boissière *et al.*, 2015). Renaud *et al.* (2010) ont étudié la grammaticalité et la qualité de la prédiction de mots. Ils ont conclu que les problèmes liés à la mauvaise interprétation du contenu (et donc de l'intention de l'utilisateur) seraient en réalité une question de perspective en fonction que l'on évalue la tâche de complétion en tant que scripteur ou lecteur.

Cet état de l'art concernant les systèmes de prédiction est loin d'être exhaustif.⁴⁶ Plutôt que de recenser toute la littérature qui présente, par exemple, des prototypes basés sur l'approche statistique majoritairement utilisée aujourd'hui, nous avons préféré donner un aperçu des diverses méthodes qui ont été testées principalement dans des projets français ou européens (et donc multilingues). À notre connaissance, aucune étude sur un prototype d'aide à la communication proposant une prédiction n'a été réalisée en Belgique. Les principaux centres de recherche dans ce domaine pour la langue française sont ceux de l'Université François Rabelais de Tours (Sibylle), de l'IRIT (WordTree, KeyGlass, VITIPI), de l'IRIN (HandiAS) et de l'Université d'Aix-Marseille (PCA, ILLICO). À noter que certains systèmes sont parfois plus détaillés que d'autres.

Par exemple, les résultats de la version en italien de FASTY n'étaient pas donnés dans l'article que nous avons choisi et il est difficile d'avoir accès à la documentation liée aux algorithmes des logiciels commercialisés comme PolyPredixTM (Bérard et Niemeijer, 2004; Niemeijer, 2005). La prédiction de lettres et de mots est néanmoins une application du TAL qui a naturellement suivi les avancées qui ont été réalisées dans ce domaine (Antoine et Maurel, 2007). L'approche symbolique a plusieurs fois été employée, mais sans donner de bien meilleurs résultats que celle basée sur les statistiques plus avancées. Les chercheurs et concepteurs ont pris conscience qu'il était nécessaire de combiner les deux approches afin de trier les différentes prédictions lexicales et ainsi ne proposer que les plus probables en fonction du contexte en cours de saisie.

La Table I.1 reprend dans l'ordre chronologique quelques-uns des systèmes que nous avons expliqués dans cette section. Nous avons également repris les techniques de prédiction, la langue et leurs taux d'économie de saisie respectifs (cf. KSR, Sous-section 3.3) obtenu avec une liste de prédictions proposant cinq mots. Les systèmes de complétion VITIPI (Boissière et Dours, 2001; Boissière *et al.*, 2012) et Soothsayer (Stoop et van den Bosch, 2014) qui n'affichent pas de liste d'hypothèses ne sont donc pas inclus. C'est aussi le cas de ceux qui prédisent les lettres (et les affichent avec une inter-

46. Pour un état des connaissances plus complet, le lecteur peut consulter les articles de Garay-Vitoria et Abascal (2006) ainsi que d'Antoine et Maurel (2007).

face utilisateur particulière) comme le modèle pentagramme de SibyLettre (Schadle, 2003) et le bigramme de KeyGlass (Raynal, 2007) ou l'arbre lexicographique de Word-Tree (Badr et Raynal, 2009; Badr, 2011).

SYSTÈME	TECHNIQUE	LANGUE	KSR ₅
HandiAS			
(Maurel <i>et al.</i> , 1998; Maurel et Le Pévédic, 2001)	Automate fini probabiliste		39,2 %
	Automate fini probabiliste (évolutif) + dictionnaire de l'utilisateur	Français	43,53 %
	Unigramme		43,9 %
	Bigramme		51,2 %
SibyMot (Schadle, 2003)	Trigramme	Français	55,8 %
	Trigramme + <i>chunk</i>		57,1 %
	Bigramme + trigramme (n-POS) + transducteur à états finis + dictionnaire de l'utilisateur + prédiction composée (mots composés en allemand)	Français	≈ 48 %
FASTY (Trost <i>et al.</i> , 2005)		Allemand	52,02 %
		Néerlandais	≈ 51 %
		Suédois	≈ 48 s%
	Bigramme		57,1 %
	Trigramme	Anglais	57,7 %
(Trnka <i>et al.</i> , 2006)	Trigramme + <i>topic modeling</i>		57,9 %
PCA			
(Blache et Rauzy, 2007)	Unigramme + modèle utilisateur + prédiction morphosyntaxique	Français	≈ 50 %
	Quadrigramme		57,87 %
Sibylle	+ modèle utilisateur		58,5 %
(Wandmacher et Antoine, 2007a,b; Wandmacher, 2008; Antoine, 2011)	+ modèle cache		58 %
	+ ASL + Interpolation Linéaire, IL	Français	58,49 %
	+ ASL + Interp. Géométrique, IG		58,61 %
	+ ASL + IL + Interp. <i>confidence weighting</i>		58,3 %
	+ ASL + IG + Interp. <i>confidence weighting</i>		58,92 %

TABLE I.1 – Systèmes de prédiction et leur KSR, adaptée de Wandmacher (2008)

Comme le fait remarquer Wandmacher (2008), il est difficile de comparer directement les résultats de ces systèmes qui utilisent tous des corpus (d'entraînement et de test) différents. Les résultats de SibyMot ainsi que de Sibylle sont basés sur le corpus d'entraînement *Le Monde*, ceux de VITIPI (non repris ici) reposaient sur un corpus de phrases météorologiques et FASTY a été testé sur des corpus variés (journalistique, roman, etc.), ce qui a également été le cas d'autres systèmes comme celui évalué par Trnka (2010) dans son travail de fin d'étude. Les performances de Sibylle que nous avons reprises sont celles obtenues avec des données de test dont le registre est identique à celles qui ont servies pour l'entraînement, c'est-à-dire le corpus journalistique *L'Humanité* et plus précisément un extrait de 58 457 mots datant de janvier 1999.

Nous insistons néanmoins sur le fait que derrière ces performances se cache un point tout aussi important, à savoir : les attentes réelles des utilisateurs. Les mots n'ont

pas tous la même importance pour un usager. Ce dernier peut vouloir éviter l’affichage de mots incongrus et ainsi s’attendre à ce que le système lui propose le nom d’un ami (comme « Rob ») à la place de mots (« robot », « Robert », etc.) qu’il ne connaît pas forcément. De telles propositions pourraient perturber l’attention de personnes déjà fragilisées par le handicap et affecter leur communication. Par conséquent, il peut être intéressant de faire des évaluations objectives et subjectives afin de vérifier si les deux aboutissent aux mêmes résultats. Dans la section suivante, nous verrons qu’elles comportent certains biais qui ne doivent pas remettre en question leur utilité respective.

3 Limites linguistiques, cognitives et ergonomiques

L’aide à la communication est un domaine pluridisciplinaire lié à la linguistique(-informatique), la psychologie cognitive ainsi que l’ergonomie (Blache et Rauzy, 2007). Cette section présente certaines limites relatives à la configuration et à l’évaluation des systèmes de prédiction ou de l’interface utilisateur dans laquelle ils sont intégrés.

Ces contraintes concernent plusieurs aspects comme la modélisation du langage en TAL (Sous-section 3.1), l’usage réel de la prédiction par les personnes handicapées relatif à l’IHM (Sous-section 3.2), l’apport des différentes métriques (Sous-section 3.3) et enfin la problématique de la représentation des caractères liée à leur encodage, à la disposition des claviers ainsi qu’à l’océrisation (Sous-section 3.4).

3.1 Modélisation du langage

En plus d’avoir étudié la notion de grammaire universelle liée au langage naturel, Noam Chomsky (1956) est l’un des premiers à avoir proposé une description formelle et rigoureuse des langues qui repose sur une conception logique et mathématique. Une grammaire G est définie par un quadruplet $\langle V_t, V_{nt}, S, RP \rangle$ où :

- V_t (Vocabulaire terminal) est un ensemble fini de symboles terminaux non vides, par exemple : un mot du langage ou une chaîne de caractères ;
- V_{nt} (Vocabulaire non terminal) est un ensemble fini de symboles non terminaux, par exemple : une catégorie (morpho)syntaxique ;
- S (Symbole) est un symbole initial ou axiome de V_{nt} ;
- RP (Règles de Production) est un ensemble fini de productions ou règles de réécriture de la forme $\alpha \rightarrow \beta$ avec α et β représentant toute séquence de symboles de $V (= V_t \cup V_{nt})$. Autrement dit, la séquence de symboles α peut être remplacée par la séquence de symboles β .

Le linguiste a établi une hiérarchie ou classification des grammaires basée sur leurs règles de production. Elle décompose les langages formels en quatre catégories (de 0 à 3) de la plus libre à la plus contraignante.

Type 0 – Grammaires générales (*recursively enumerable grammar*) : aucune restriction sur RP, excepté le fait qu'il doit exister au moins un symbole à gauche non vide. Elles ont la forme $\alpha \rightarrow \beta$ (où $\alpha \in V_{nt}^*$ et $\beta \in V^*$). Elles correspondent aux machines de Turing et sont très théoriques.

Type 1 – Grammaires contextuelles (*context-sensitive grammar*) : le remplacement d'un symbole non terminal peut dépendre des symboles autour de lui (son contexte). Elles ont la forme $\alpha A \beta \rightarrow \alpha \gamma \beta$ (où $A \in V_{nt}$ et $\alpha, \beta, \gamma \in V^*$). Elles correspondent aux machines de Turing à entrée bornée.

Type 2 – Grammaires non-contextuelles (*context-free grammar*) : les symboles non terminaux sont traités indépendamment de leur contexte. Elles ont la forme $A \rightarrow \alpha$ (où $A \in V_{nt}$ et $\alpha \in V^*$). Elles correspondent aux automates à pile.

Type 3 – Grammaires régulières (*regular grammar*) : elles sont linéaires à gauche ou linéaires à droite. Elles ont la forme $A \rightarrow Ba$ ou $A \rightarrow aB$ (où $A, B \in V_{nt}$ et $a \in V_t$). Elles correspondent aux automates finis.

Ces grammaires entretiennent une relation d'inclusion stricte : type 3 \subset type 2 \subset type 1 \subset type 0, ce qui signifie, par exemple, que les grammaires générales peuvent exprimer les langages générés par les trois autres catégories. En TAL, seuls les deux derniers types de grammaires sont utilisés. Nous en avons vu des exemples dans la sous-section précédente. Elles sont néanmoins limitées et génèrent différentes ambiguïtés. Par exemple, les grammaires de type 2 ne traitent pas toutes les exceptions du langage. Les grammaires CFG (Figure I.11) illustrent une ambiguïté syntaxique qui, bien que la phrase soit grammaticalement correcte, pose problème puisqu'elle peut être interprétée de deux façons différentes : 1) étant moi-même en pyjama, je mange un canard (Figure I.11a) ; 2) je mange un canard qui est en pyjama (Figure I.11b).

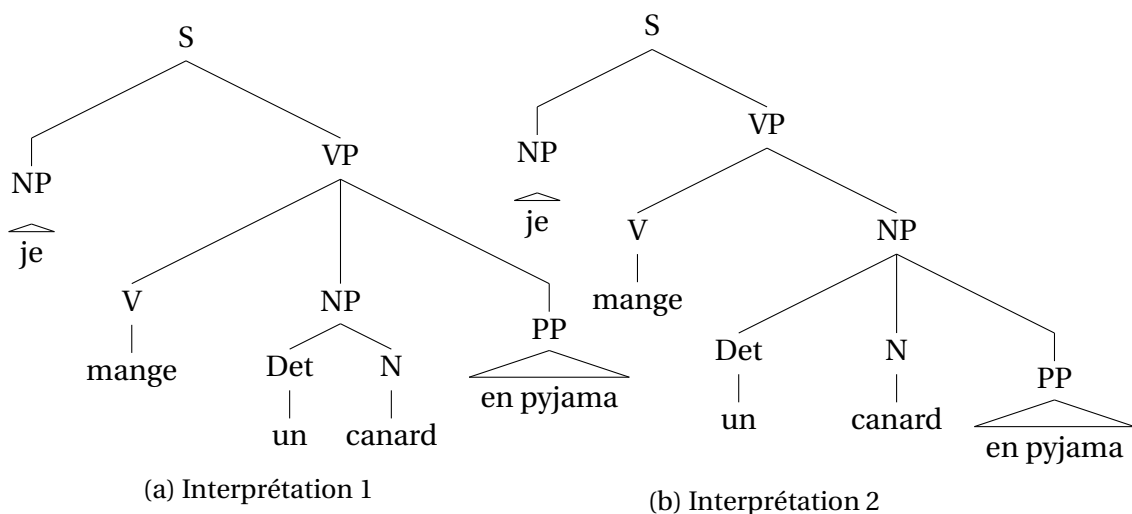


FIGURE I.11 – CFG : exemple de la ph. « je mange un canard en pyjama »

Un autre inconvénient de ce type de grammaire est la difficulté de prendre en compte l'accord entre les mots puisqu'il faut alors multiplier les règles. En étudiant la notion de grammaticalité d'une phrase, Chomsky (1969, pp. 17–19) énonce le fait que (1) et (2) sont des (parties de) phrases que l'on pourrait trouver en anglais, mais qui, dans un modèle statistique, seraient rejetées en raison de leur faible probabilité d'apparition alors que (1) est syntaxiquement correcte.⁴⁷ Ses travaux ont tenté de démontrer l'inadéquation des modèles de grammaires probabilistes à représenter le langage.

- (1) Colorless green ideas sleep furiously.
- (2) * Furiously sleep ideas green colorless.

Afin de modéliser certains phénomènes linguistiques, il a distingué la structure profonde de la structure de surface obtenue par des transformations.⁴⁸ Cette grammaire générative et transformationnelle a évolué. D'autres linguistes ont proposé des extensions. Nous pouvons citer la grammaire des cas de Fillmore qui énonce des rôles sémantiques. En 1970, Chomsky a lancé la théorie X-barre qui postule que tous les syntagmes ont la même structure imposant ainsi des restrictions sur les symboles non terminaux avant d'introduire le programme minimaliste dans les années 1990.

Bien que les théories linguistiques actuelles ne permettent toujours pas de modéliser entièrement le langage, nous avons néanmoins vu qu'il était possible d'en engendrer un à partir d'un nombre restreint de règles sans avoir appris toutes les phrases possibles d'une langue donnée et donc sans devoir toutes les énumérer grâce à la récursivité des règles de production. Cette approche s'oppose aux modèles statistiques qui ne reposent sur aucune règle, mais sur la constitution de corpus qui se veulent le plus représentatifs possible des tâches auxquelles ils sont destinés.

Comme nous l'avons déjà évoqué, Antoine et Maurel (2007, p. 29) ont montré que les prototypes de prédiction avaient « suivi l'historique du TAL ». Les systèmes symboliques consistant à analyser syntaxiquement les phrases en cours de saisie et reposant sur une base de connaissances étaient encore souvent utilisés durant les années 1990. Ils ont progressivement été remplacés par les grammaires probabilistes et les divers systèmes hybrides ainsi que par les modèles statistiques qui existaient déjà dans les logiciels commerciaux, mais sous la forme de simples fréquences.

Le prototype de recherche sur lequel sera basé notre étude de cas utilise une approche statistique et plus précisément un modèle stochastique avancé. Il ne fera donc pas appel à la modélisation du langage que nous venons de voir. Les limites de certaines mesures spécifiques à l'évaluation de modèles statistiques vont être présentées.

47. (1) « D'incolores idées vertes dorment furieusement »; (2) « * Furieusement dormir idées vert incolore » où * désigne une phrase agrammaticale. (1) est dépourvue de sens, mais grammaticale.

48. Ainsi, des phrases au passif (« un canard est mangé par Marie ») ou des inversions (« manges-tu un canard? ») peuvent être représentées par un mouvement du verbe lors du passage vers la structure de surface. Ces transformations permettent d'expliquer certaines différences entre les langues.

En linguistique informatique, il existe plusieurs métriques permettant d'évaluer un modèle et souvent utilisées dans le cadre de la reconnaissance de la parole.

Out Of Vocabulary (OOV) Les mots inconnus ou hors vocabulaire ont déjà été plusieurs fois évoqués (cf. point 2.1.2). Il est possible de mesurer leur pourcentage afin d'évaluer le modèle de langage utilisé. Ainsi, Trnka et McCoy (2007) avaient étudié le taux de OOVs dans leur système de prédiction de mots afin de mesurer la spécialisation de leurs différents corpus. Ils ont montré que plus ce taux était haut, plus le corpus était spécialisé au sein d'un domaine donné.

Perplexité (PP) Cette métrique est utilisée pour évaluer la complexité d'une tâche en fonction d'un modèle de langage. Formulée en (I.6), il s'agit de l'inverse de la moyenne géométrique des probabilités des mots du corpus de test. Dans sa thèse sur la prédiction de mots, Wandmacher (2008, p. 38) affirme qu'obtenir une perplexité moyenne de 100 revient à devoir deviner entre 100 symboles différents, ce qui est plus facile qu'entre 1 000 éléments; plus la perplexité est faible et meilleure sera la prédiction. Ses résultats ont néanmoins montré que la perplexité ne corrélait pas bien avec le taux d'économie de saisie (cf. KSR, Sous-section 3.3). Trnka (2010) explique par exemple que les mots communs entre le corpus d'entraînement et le corpus de test peuvent être prédits parfaitement alors qu'aucune lettre n'a été entrée.

$$PP(m_1...m_n) = \sqrt[n]{\prod_{i=1}^n \frac{1}{P(m_i|m_1...m_{i-1})}} \quad (I.6)$$

Entropie croisée Liée à la perplexité, l'entropie croisée d'un modèle permet de mesurer la qualité de prédiction d'un événement sur un corpus de test non connu. Formulée en (I.7), elle se calcule en nombre de bits moyen (nécessaires pour l'encodage des données de test). Tout comme la perplexité (qui est l'exponentielle de l'entropie croisée), plus cette métrique est petite, meilleur sera le modèle de langage.

$$-\frac{1}{n} \cdot \sum_i \log_2 P(m_i|m_1...m_{i-1}) \quad (I.7)$$

Selon Antoine (2011), les métriques classiques en modélisation du langage sont rarement utilisées dans le domaine de la CAA. Bien qu'Antoine et Maurel (2007, p. 39) disent avoir observé « une corrélation entre la perplexité ou l'entropie du modèle et la KSR qu'il permet d'atteindre » dans une de leurs études (Wandmacher et Antoine, 2007a), les auteurs se réfèrent ensuite à Copestake (1997) pour affirmer que « ces métriques issues de la théorie de l'information ne constituent pas des indicateurs explicites de l'aide fournie par la prédiction ». Par exemple, la perplexité a été utilisée que par quelques-uns des auteurs cités (Wandmacher, 2008; Trnka, 2010). Les métriques employées dans le cadre de la CAA seront détaillées plus loin (cf. Sous-section 3.3).

3.2 Usage de la prédiction

Il nous semble important de s'intéresser à l'usage que font les utilisateurs de ces systèmes puisqu'ils sont conçus pour leurs besoins spécifiques et qu'ils doivent s'adapter à eux. À noter que contrairement à la Section 2 sur les techniques de prédiction, les quelques études au sujet de leur utilisation réelle sont relativement éparées.

Laffont *et al.* (2007) ont analysé l'utilisation du système de communication alternative de commerce Dialo[®] chez une dizaine de personnes handicapées en situation de vie afin de se rapprocher le plus possible de leur usage réel. L'étude a duré deux mois. Ils ont montré que la prédiction de mots apportait un bénéfice modéré lorsqu'elle était utilisée avec la synthèse vocale, la prédiction ne faisant pas systématiquement gagner du temps. Les personnes évaluées présentaient diverses pathologies (SLA, IMC, LIS et anoxie cérébrale). Leurs résultats ont révélé qu'elles n'utilisaient un mot correctement prédit que dans 10 % des cas. Ils l'expliquent par le fait que le conflit cognitif entre écriture du message et lecture des prédictions peut être difficile à gérer.

Les auteurs sont donc arrivés aux mêmes constatations que Niemeijer (2005), qui avait déjà comparé les performances théoriques d'un système avec celles de réels utilisateurs et avait effectivement constaté une diminution de leurs résultats dans le deuxième cas. Son étude a montré qu'ils obtenaient en moyenne 41 à 43 % de réduction de clics avec leur système et 51 % par simulation. Comme le notent Trost *et al.* (2005), les résultats de certaines métriques dépendent également de l'ergonomie puisqu'une mauvaise interface utilisateur peut occasionner une charge cognitive supplémentaire qui pourrait, par exemple, ralentir la production de texte. Il existe en réalité peu de recherches sur l'utilisation des systèmes de prédiction sur le long terme.

Les systèmes étudiés sont souvent des prototypes de recherche conçus dans le domaine académique et dont beaucoup n'ont pas dépassé ce stade. Nous pouvons citer KOMBE (Guenthner *et al.*, 1994) et HandiAS (Maurel et Le Pévédic, 2001) qui, comme le font remarquer Blache et Rauzy (2007, p. 48), n'ont pas été « véritablement distribués au grand public ». D'autres applications continuent à évoluer et à être testées par de réels utilisateurs handicapés. C'est le cas du logiciel Sibylle qui est toujours opérationnel et utilisé quotidiennement au Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelles de Kerpape en France.⁴⁹ Le système VITIPI a aussi fait l'objet d'expériences auprès d'enfants au Centre Hospitalier de Montfavet à Avignon.

L'IHM a un impact sur l'utilisation des aides à la saisie de texte. Nous avons par exemple présenté des systèmes prédictifs dont l'interface proposait un réagencement ou une adaptation des lettres (cf. point 2.1.1) comme Sibylettré (Figure I.3) et KeyGlass (Figure I.4). Un autre exemple est le clavier Dasher qui est également dynamique, disponible dans une soixantaine de langues et qui, selon Antoine et Maurel (2007), repose

49. Il comprend aujourd'hui plusieurs claviers hautement configurables (cf. Figure I.5). Nous verrons que le moteur de prédiction de mots est intégré dans le système CVK de l'Hôpital de Garches.

sur des probabilités d'occurrence. Les auteurs expliquent que même s'il a été prouvé qu'un clavier dynamique améliore la vitesse de saisie, son usage dépend de la nature du handicap puisque cette disposition particulière peut aussi augmenter la charge cognitive. Ce type de systèmes ne pourrait donc pas convenir à tous les utilisateurs.

Bien qu'il reconnaisse que le système Sibylle augmente la vitesse de saisie, Guerrier (2015, p. 31) note qu'il faut « un effort de concentration supplémentaire pour lire les propositions » et qu'il existe un risque de « fatigue mentale [...] après quelques heures d'utilisation » chez les handicapés moteurs de type IMC (tétraplégiques athétosiques). Il critique également Dasher pour des raisons à la fois cognitives et ergonomiques ; d'après lui, il serait préférable de ne pas présenter de trouble visuel et d'être suffisamment concentré pour pouvoir sélectionner la lettre désirée. Cette augmentation de la charge mentale se retrouverait dans KeyGlass même s'il permet une réduction de l'effort physique (ce que ferait aussi Dasher vu qu'il « est utilisable sans faire de clic »).

Nous avons vu que ces systèmes de prédiction de lettres visaient à faciliter leur accès en les rapprochant de l'utilisateur en fonction de leur fréquence. Paradoxalement, le but de la disposition du clavier QWERTY (1868) et son adaptation française l'AZERTY était d'éloigner les lettres les plus utilisées les unes des autres « pour éviter que les marteaux de la machine à écrire ne s'entrecroisent entre eux quand les utilisateurs écrivaient trop rapidement » (Badr, 2011, pp. 11–12). Elles sont devenues la norme sur les ordinateurs, téléphones et tablettes malgré l'émergence de nombreuses propositions comme la disposition Dvorak (1936), un clavier anglais statique et optimisé par agencement visant à accélérer la vitesse de saisie (Martin et Pecci, 2007).⁵⁰

Certains prédicteurs s'intègrent dans toute application ou éditeur de texte existant, tandis que d'autres proposent une interface utilisateur spécifique constituant un système d'aide à la communication à part entière. La présentation de la liste de prédictions aurait aussi un impact sur les habitudes des utilisateurs et donc sur les résultats. Badr (2011) a constaté qu'il était fréquent que les propositions soient triées par ordre de fréquence, ce qui permettrait de réduire le temps de recherche étant donné que les mots les plus probables se trouveraient en haut de la liste de prédictions. Garay-Vitoria et Abascal (2006) ont néanmoins noté qu'un classement alphabétique était plus performant si le nombre de propositions est élevé.

Concernant l'usage de la prédiction associée à la synthèse vocale, nous avons vu que Laffont *et al.* (2007) rapportaient que celle-ci n'était pas systématiquement utilisée par les personnes handicapées, leur étude ayant montré que moins d'un tiers des personnes recrutées préféraient se servir de l'ordinateur pour communiquer oralement. Nous pensons que l'une des raisons serait la gêne que pourraient occasionner l'utilisa-

50. Pour plus de détails, cf. leur état de l'art sur les claviers physiques et logiciels. Comme le fait remarquer Badr (2011), le premier type nécessite les dix doigts alors que le deuxième n'en demande qu'un (ou l'utilisation d'un dispositif de pointage), la disposition étant aussi problématique dans ce cas.

tion de ces technologies auprès d'un public qui n'en a peut-être pas l'habitude. Antoine et Maurel (2007) avaient par exemple constaté que parce que la saisie sur clavier virtuel était plus discrète que la saisie par dictée vocale, cette dernière était relativement moins utilisée par les patients.

Les auteurs expliquent aussi que même si la personne tétraplégique conserve son élocution, beaucoup préfèrent se servir d'un clavier à défilement plutôt qu'un système de reconnaissance de la parole qui constitue une modalité de saisie à part entière. Dans sa thèse soutenue à l'Hôpital Raymond Poincaré, Pouplin (2016) a montré que les blessés médullaires ne profitaient pas de la même manière d'une prédiction dans une CAA suivant la hauteur de la lésion dont ils souffraient. Pouplin *et al.* (2014) ont testé le clavier dynamique CiviKey (*Custom Virtual Keyboard*)⁵¹ avec le système Sibylle chez des personnes tétraplégiques. D'après eux, le clavier associé à la prédiction ne permet pas d'améliorer la vitesse de saisie par rapport à un clavier statique sans prédiction.

Des retours d'expérience concernant Sibylle ont aussi montré que l'usage de la prédiction pouvait conduire à la production de plus de textes et une diminution des fautes d'orthographe chez des enfants atteints d'une IMC (Wandmacher et Antoine, 2007b). Cette réduction des erreurs résultant de la prédiction est un bénéfice que Copestake (1997) avait déjà noté en soulignant qu'il pouvait être utile pour la synthèse vocale. Trnka *et al.* (2007) rapportent que des études auraient, au contraire, constaté un taux de communication plus mitigé en raison de l'augmentation de la charge cognitive. Comme l'expliquent Renaud *et al.* (2010, p. 16), elle est présente dans tous les systèmes de prédiction et due au fait que les utilisateurs doivent systématiquement :

- “ 1) change their focus between the target document and the word list display, and possibly between the screen and keyboard;
2) visually scan the word list to decide whether their intended word is present;
and 3) select the intended word with the keyboard or mouse. ”

En plus d'insister sur la difficulté d'estimer l'impact de la prédiction sur le taux de communication, les études sur les performances de ces systèmes évoquent la complexité de réaliser des évaluations objectives avec de réelles personnes handicapées (Trnka *et al.*, 2006, 2007). Les auteurs avaient d'abord fait des tests avec des personnes non handicapées et justifient ce choix en raison de deux facteurs : la grande variabilité dans les capacités physiques et cognitives des utilisateurs, mais aussi dans les *designs* de l'interface utilisateur car les résultats peuvent dépendre de la facilité d'utilisation du système de prédiction. Ils expliquent que c'est la diversité des handicaps qui a poussé les concepteurs de CAA à développer des applications pour des besoins spécifiques.

51. Le logiciel *open source* CVK dispose d'un site (<http://www.civikey.com/>) et un *repository* (<https://github.com/Invenietis/ck-certified>) Il est utilisable avec un dispositif de pointage et un système de défilement. Il est éditable selon les besoins et couvre les caractères latins, cyrilliques, chinois et arabes.

Concernant les mesures subjectives, il n'existe aucune méthode pour évaluer la satisfaction des utilisateurs dans le cas spécifique d'un système de prédiction de mots. La *Quebec User Evaluation of Satisfaction with assistive Technology (QUEST)* ou Évaluation de la Satisfaction envers une Aide Technique (ESAT) est un questionnaire général qui comporte 12 énoncés de satisfaction sur les ajustements, le confort, l'efficacité, etc. pour tous les systèmes de CAA. Cette échelle est très rarement citée dans la littérature sur la prédiction de mots. Elle l'a par exemple été dans les recherches de Laffont *et al.* (2007) qui l'ont utilisée pour évaluer l'ensemble du système de communication alternative (comprenant la synthèse vocale) et plus récemment par Pouplin (2016).

L'auteur français rapporte aussi que les ergothérapeutes préconiseraient régulièrement les systèmes de prédiction de mots. Le processus de préconisation serait segmenté en plusieurs phases allant de l'expression d'un besoin et de la demande à l'apprentissage de l'aide technique si celle-ci a été validée après une série d'évaluations. Selon l'ASBL Comalso, il n'existerait pas de processus de préconisation des nouvelles technologies et des systèmes de prédiction de mots similaire en Belgique. Il est toutefois possible que les logopèdes et les ergothérapeutes qui connaissent ces types de logiciels les recommandent aux patients qui ont toujours la lecture et l'écriture.

Après avoir étudié une population ayant des niveaux lésionnels différents, Pouplin (2016, p. 199) précise qu'à sa connaissance, « il n'existe aucune étude, dans la littérature, recensant les besoins et attentes des personnes tétraplégiques envers les logiciels de prédiction de mots ». C'est pourquoi il explique que si elles s'attendent à ce que ces systèmes leur apportent non pas « une augmentation de la vitesse de saisie », mais « une diminution du nombre d'erreurs, ou une augmentation du confort d'utilisation de l'outil informatique », cela pourrait se manifester par un écart entre « les attentes des utilisateurs et l'apport des logiciels de prédictions de mots ». Il est donc important de les impliquer et de privilégier des démarches de conception centrées utilisateurs.

Il faut néanmoins souligner les difficultés liées à l'impossibilité de mener des études à large échelle sur une population homogène en situation écologique et sur le long terme. Les chercheurs et concepteurs de CAA ont ainsi testé leur système sur des petits groupes d'utilisateurs présentant diverses pathologies (Laffont *et al.*, 2007), sur des personnes non handicapées ou « pseudo-impaired » (Trnka *et al.* (2007) avaient par exemple calculé un délai supplémentaire d'au moins 1,5 seconde par touche); d'autres auteurs ont simulé le handicap moteur et les mouvements incontrôlés avec un dispositif spécifique (Guerrier, 2015). Ces conditions particulières ont alors pour but de reproduire des situations réelles et d'effectuer les tests en laboratoire.

Il est important de toujours détailler et justifier le protocole expérimental afin de pouvoir juger de sa validité. Dans la sous-section suivante, nous présenterons les métriques utilisées lors des simulations théoriques destinées à évaluer les systèmes de prédiction ainsi que les tâches demandées aux sujets lors de tests utilisateurs.

3.3 Apport des métriques

Avant d'aborder les limites de la représentation des caractères et l'étude de cas, nous présentons les métriques objectives qui sont couramment employées dans le cadre d'une évaluation d'un système d'aide à la saisie. En fonction de l'objectif de l'étude, il est possible de mesurer divers facteurs comme la pertinence de la prédiction, le taux d'erreur ou la vitesse de saisie (Badr, 2011). Afin de contextualiser leur usage ainsi que leurs limites, nous expliciterons certaines conditions expérimentales.

Nous avons vu que les techniques permettant d'optimiser la saisie relevaient à la fois du domaine du TAL et de l'IHM (Antoine et Maurel, 2007). Le premier a pour but de réduire le nombre de saisies et le deuxième de diminuer le temps d'accès aux touches. Certaines mesures sont donc davantage centrées sur le système (*system-centred*), tandis que d'autres prennent en compte le comportement de l'utilisateur (*user-centred*). Bien qu'il existe des écarts entre une évaluation objective et subjective, Antoine et Maurel (2007, p. 39) soulignent que ce « décalage avec l'estimation des performances de la prédiction ne doit pas remettre pas en cause l'utilité des métriques utilisées ».

Hit ratio ou hit rate Cette métrique permet d'évaluer la qualité d'un prédicteur. Il s'agit du pourcentage d'apparition du mot recherché dans la liste de prédictions, il est donc dépendant de la taille de celle-ci. Garay-Vitoria et Abascal (2006) définissent ce taux comme le quotient entre le nombre de fois où les mots sont correctement prédits et le nombre de mots écrits (II.4) ; plus celui-ci est haut et meilleure sera la pertinence de la prédiction. D'après eux, cette métrique doit néanmoins être combinée à d'autres car elle ne donne pas de précision sur le nombre d'appuis effectués ou économisés. Un mot peut avoir été correctement prédit après que presque toutes les lettres aient été saisies par l'utilisateur, ce qui va à l'encontre de l'objectif de conception d'un prédicteur performant puisque très peu de saisies peuvent donc avoir été économisées.

Keystroke Saving Rate (KSR) Le taux d'économie de saisie (exprimé en %) est, par conséquent, la métrique la plus souvent utilisée pour estimer la performance de la prédiction. Elle est définie formellement en (I.8) où « k_p est le nombre d'appuis effectivement réalisés par l'utilisateur lors de la saisie d'un message, k_a le nombre d'appuis qui auraient été nécessaires sans aide à la composition de mots et n est la taille de la liste de prédictions (normalement $n = 5$) » (Wandmacher et Antoine, 2007b, p. 77). Dans les études de certains auteurs comme Vescovi (2004), le KSR est dépendant de l'interface utilisateur du prédicteur car il calcule les lettres saisies et les appuis nécessaires pour accepter le mot (II.3). Le nombre d'opérations varie donc toujours en fonction de la façon dont on sélectionne les mots dans la liste, des appuis supplémentaires étant parfois nécessaires (cf. Chapitre II, point 3.2.1). Copestake (1997) avait estimé que les performances de la prédiction resteraient toujours dans une fourchette de 50 à 60 %. Ses calculs théoriques s'avèrent fondés puisque nous avons vu que les études citées

présentaient un KSR proche des 50 % bien que certaines commencent à dépasser les 60 % en fonction de certains paramètres comme les (registres des) corpus utilisés.

$$\text{KSR}_n = \left(1 - \frac{k_p}{k_a}\right) \cdot 100 \quad (\text{I.8})$$

Keystroke Per Character (KSPC) Le nombre moyen d'appuis par caractère saisi est employé pour évaluer le taux d'erreur, un facteur qui peut déterminer le choix d'un système prédictif et qui est parfois utilisé en traduction automatique associé à la distance de Levenshtein. Le KSPC est le quotient entre le nombre d'opérations au clavier (ici, le nombre d'appuis nécessaires à l'utilisateur) et le nombre de caractères du texte.

Word Per Minute (WPM) Pour évaluer la vitesse de saisie, il existe différentes métriques comme le nombre de caractères par seconde (ou par minute), mais aussi le nombre de mots par minute qui peut dépendre de l'interface. Dans le cas d'un système à défilement du curseur, Ghedira (2015) insiste sur la lenteur des utilisateurs puisqu'entre 1 et 5 mots sont validés par minute, le délai typique de défilement par caractère variant généralement entre 0,2 et 6 secondes en fonction de leur confort. Si le temps de scan de chaque item est trop rapide, le nombre d'erreurs et le risque de stress seront plus importants; alors que si celui-ci est plus long, la vitesse (et donc la rapidité de la communication) sera limitée. À noter que ce délai jugé optimal varie également au cours du temps selon « l'attention », « l'apprentissage » et « la fatigue » du patient. Bien que généralement stable, le réglage se fait toujours de manière « empirique ». La vitesse de saisie est extrêmement difficile à mesurer objectivement étant donné qu'elle varie en fonction de la situation et de l'individu (Bérard et Niemeijer, 2004). Il faut également du temps à l'utilisateur pour apprendre et s'habituer au système.

Des résultats peuvent être obtenus via deux méthodes différentes (Wester, 2003) : l'une, formelle; l'autre, empirique. La première est moins coûteuse. Elle est basée sur des heuristiques et *guidelines*. Les experts l'utilisent afin de déterminer les performances théoriques d'un système. Elle peut reposer sur un simulateur automatique. La deuxième prend plus de temps et demande plus de moyens. Elle tente de se rapprocher de l'usage réel des utilisateurs en étudiant leur comportement. Elle consiste à directement les interroger que ce soit par un questionnaire, un entretien, etc.

Ces méthodes permettent d'obtenir deux types de résultats : quantitatifs ou qualitatifs. Dans le cas d'un système de prédiction, il s'agit souvent de statistiques puisque les métriques ci-dessus sont utilisées dans la majorité des études citées (parfois sous des appellations différentes). Dans des études comme celle de Raynal (2007, p. 103), des simulations théoriques sur ordinateur sont effectuées dans le but de d'abord « déterminer le système de prédiction le plus performant à intégrer » afin de mieux cibler ses évaluations avec des personnes handicapées, ce qui permet d'adapter le prototype.

Les chercheurs et concepteurs intègrent alors un module qui permet de simuler automatiquement la meilleure utilisation possible, c'est-à-dire ne calculant pas les erreurs que l'utilisateur produirait et sélectionnant directement la bonne prédiction dès qu'elle est susceptible d'apparaître dans la liste. Un simulateur peut ainsi garder une trace des prédictions et des saisies d'un ensemble de mots (ou de phrases) qu'il est possible d'analyser et de comparer grâce à des fichiers logs ou *log files*.

Nous avons vu dans la sous-section précédente les difficultés liées à la réalisation d'évaluations subjectives auprès de réels utilisateurs handicapés. Différentes tâches peuvent être proposées. Badr (2011, p. 56) en détaille deux : « une recopie d'un texte ou d'un ensemble de mots, une saisie libre d'un texte de leur propre imagination ». Dans la première tâche, les utilisateurs doivent recopier des phrases ou des mots dictés ou affichés à l'écran. Il est important de bien les positionner ainsi que l'écran de feedback.

La deuxième est une tâche de création se rapprochant de l'utilisation habituelle de ces systèmes. Elle comporte plusieurs biais car il est plus difficile de contrôler les conditions de l'expérimentation comme le comportement des utilisateurs, les erreurs ou le texte saisi. Ces évaluations permettent néanmoins d'étudier les réactions des utilisateurs confrontés à un système de saisie en prenant en compte des facteurs tels que le type de handicap ou l'impact de la charge cognitive et visuelle sur les résultats.

Dans sa partie sur les limitations de la prédiction de mots, Copestake (1997, p. 39) avait commencé par aborder la difficulté de comparer leurs résultats avec ceux provenant de la littérature à cause de l'absence d'un corpus commun, mais également parce qu'il n'existe pas de mesure standard qui soit acceptée par tous. Par conséquent, il est difficile de comparer la configuration et l'évaluation de systèmes de prédiction. Comme l'avaient déjà noté Trnka *et al.* (2006) :

“ Although there is a large existing research base of word prediction for AAC, the results of each researcher can't be adequately compared due to differences in evaluation methods. Firstly, there is little agreement between researchers on the size of the prediction list. Some researchers wait until one character has been entered to begin prediction and some begin predicting solely on the basis of context. However, the most difficult aspect of comparing results amongst different researchers is that there is no standard corpus to be used for evaluation.

”

Nous concluons cette sous-section en rappelant qu'en moyenne, plus d'une saisie sur deux peut être évitée (le KSR étant aujourd'hui supérieur à 50 %), mais également en insistant sur la nécessité de réaliser des évaluations objectives et subjectives. Celles-ci nous donnent des indications sur les performances généralement attendues pour ce type de systèmes ainsi que sur certaines préférences des utilisateurs servant à l'amélioration et à l'adaptation continue de leur différente interface.

3.4 Représentation des caractères

Dans cette sous-section, nous abordons trois aspects limitatifs relatifs à la représentation des caractères ou leur intelligibilité : l'encodage, la disposition des claviers et l'océrisation. Nous reviendrons sur ces trois dimensions dans notre étude de cas.

L'ergonomie des différents claviers a souvent fait l'objet d'état de l'art dans les études présentant des systèmes de CAA et de prédiction (Antoine et Maurel, 2007; Badr, 2011; Guerrier, 2015). L'encodage des éléments constitutifs d'un mot, les caractères et diacritiques n'est jamais évoqué et pourtant nous verrons qu'il s'agit d'une problématique à part entière qui, même au XXI^e siècle, peut nuire à l'ergonomie de l'interface pour tous les utilisateurs. Les ordinateurs traitent différents types d'information représentée sous la forme de bits, c'est-à-dire de chiffres binaires (0 ou 1). Elle regroupe les données non numériques qui comprennent à la fois les caractères alphanumériques (les lettres majuscules et minuscules ainsi que les chiffres) et les caractères spéciaux. Afin de représenter cet ensemble d'information, il existe différents standards :

- ASCII (*American Standard Code for Information Interchange*) est le codage standard américain. Il s'agit d'une représentation normalisée où chaque caractère alphanumérique est codé sur 7 bits permettant de représenter 128 valeurs. Ce standard ne permet pas d'écrire tous les caractères du français.
- Latin-1 (ISO 8859-1) sur 8 bits, un octet, soit 256 valeurs. Il s'agit d'une extension de l'ASCII qui comporte 191 caractères de l'alphabet latin et qui est utilisé pour la plupart des langues d'Europe occidentale.
- Latin-9 (ISO 8859-15). Cette norme modifie légèrement l'ISO 8859-1. Pour le français, nous notons l'ajout du caractère œ et du symbole monétaire €.
- Unicode sur 8, 16 puis 32 bits. Chaque caractère est représenté par une valeur numérique unique. Ce standard a pour but de fournir une représentation universelle du texte. Il est par exemple utilisé pour le chinois qui compte plusieurs dizaines de milliers de caractères.

Différentes formes de codage de caractères sont donc utilisées. L'encodage le plus connu est l'UTF (*Unicode Transformation Format*), dont l'UTF-8, 16 et 32. Il peut arriver que les caractères ne s'affichent pas correctement ; le *mojibake* est un terme japonais qui désigne ce changement (*bake*) de caractères (*moji*).⁵² Ce type de problème est dû à la différence de codage de caractères entre celui qui est utilisé pour l'enregistrement d'un texte et celui servant à son affichage. Nous verrons dans l'étude de cas que nous avons été confronté à cette limitation dans le système prédictif étudié bien que le concepteur nous ait dit que tant que le modèle de langage était entraîné avec le même codage de texte que celui utilisé dans l'application, le système devait fonctionner. Il n'avait pas testé l'Unicode et cela a eu un impact sur les résultats du prédicteur.

52. Cet emprunt lexical vient du fait qu'il s'agit d'un phénomène très répandu dans cette langue.

Un problème d'encodage peut ainsi entraîner une mauvaise lisibilité des listes de prédictions. La disposition virtuelle des caractères doit aussi être prise en compte. Plusieurs recherches portant sur les systèmes de prédiction ont réalisé leurs évaluations sur des claviers limités aux 26 lettres de l'alphabet latin (en plus du caractère espace) parce que les diacritiques, les chiffres et la ponctuation n'avaient pas été implémentés dans leur prototype (Raynal, 2007; Badr, 2011). Ces deux études terminent en reconnaissant que cette limitation n'est pas « représentative de la saisie de texte pour la langue française » (Raynal, 2007, p. 119) et qu'au quotidien « ces claviers ne peuvent pas être utilisés » (Badr, 2011, p. 137). Dans leurs perspectives de travaux futurs, elles envisagent de refaire des évaluations de leur système sur un clavier plus complet.

Il convient donc de noter que l'agencement des claviers logiciels est limité. Bien qu'ils soient couramment utilisés, les caractères spéciaux, les lettres accentuées, les chiffres, les majuscules et certaines touches de fonction nécessitent souvent des appuis supplémentaires. Ces caractères sont parfois organisés dans des onglets séparés comme ce fut le cas pour la PCA qui comprend plusieurs menus : abc, ABC, 012, àÇÉ et &# (cf. Figure I.9a). D'autres prototypes comme Sibylle les ont également intégrés à des pavés ou sous-claviers (Figure I.9b). Dans la version K de Sibylle (Figure I.5), les lettres accentuées et les signes de ponctuation les plus fréquents en français (sauf l'apostrophe) sont directement accessibles depuis le clavier principal.

Enfin, pour terminer notre état de l'art, nous avons décidé d'aborder l'océrisation, c'est-à-dire la reconnaissance optique de caractères ou *Optical Character Recognition* (OCR). Il s'agit d'un procédé automatique qui consiste à reconnaître les chaînes de caractères des images et les traduire en texte. En fonction du logiciel utilisé, la qualité peut être différente et le nombre d'erreurs variable. Étant donné qu'un système de prédiction repose sur de gros corpus, une solution est d'utiliser ceux disponibles sur certains sites (gérés par des organisations, des universités, etc.), constitués grâce aux réseaux sociaux (Stoop et van den Bosch, 2014) ou à partir de ressources océrisées comme nous proposons de le faire dans notre étude de cas avec Google Books Ngram.

Issu d'un projet de numérisation à large échelle, Google Books est un exemple parmi d'autres. Si la phase d'océrisation n'a pas été réalisée correctement, nous pouvons supposer que cela ait un impact sur les prédictions. Le fait de confondre des mots dans un texte est susceptible d'affecter le calcul de leur fréquence. La qualité des techniques d'océrisation dépend souvent de nombreux facteurs. Les erreurs peuvent concerner des accents, majuscules, apostrophes et tirets, mais également des lettres.⁵³ Cet état de l'art aura donc permis d'aborder de nombreux aspects et limites des systèmes de prédiction de mots à la fois linguistiques, cognitives et ergonomiques.

53. Des chercheurs ont par exemple relevé des erreurs comme la confusion du « s » et « f » (« case » contre « café », « funk » contre « sunk », « fame » contre « same ») voire certaines incohérences telles que la surabondance de « Figure » à la place de « figure » montrant que le corpus contiendrait de la littérature scientifique (<https://www.wired.com/2015/10/pitfalls-of-studying-language-with-google-ngram/>).

Chapitre II

Étude de cas

This system allows me to communicate much better than I could before, and I can manage up to three words a minute. I can either speak what I have written or save it on disk. I can then print it out or call it back and speak it sentence by sentence.

Stephen Hawking (2013)

Dans ce chapitre, nous énoncerons plusieurs questions (Section 1) concernant les deux systèmes que nous proposons d'étudier et qui seront présentés en détail (Section 2). Nous détaillerons ensuite la méthodologie suivie (Section 3) avant de présenter nos résultats (Section 4) et de discuter de leur validité (Section 5).

1 Objectif

Bien que le système utilisé il y a encore quelques mois par Stephen Hawking dépassait probablement les trois mots par minute, la vitesse de saisie que nous cherchons à atteindre est à relativiser et difficile à définir en raison, notamment, des nombreuses possibilités de reconfiguration de ces systèmes. Nous avons néanmoins vu que les études les plus récentes estiment qu'au moins une saisie sur deux peut être évitée. Afin d'évaluer notre configuration, nous formulons quelques questions de recherche qui permettront de mieux cerner l'impact des différents paramètres qui nous intéressent.

Question 1 – Corpus Quel est l'impact du corpus d'entraînement sur les résultats? Nous en expérimenterons plusieurs et nous les comparerons afin de montrer si les performances du prédicteur proposé en dépendent. Nous testerons également la généralité de ces différents corpus. Est-ce que des corpus spécifiques seront plus efficaces que des corpus génériques? Par exemple, est-ce qu'un corpus caractérisant l'idiolecte de l'utilisateur sera plus adapté qu'un corpus littéraire? Nous supposerons aussi qu'un petit corpus de quelques milliers de mots sera moins performant qu'un plus gros en contenant des millions. Nous utiliserons également plusieurs corpus de test.

Question 2 – Prédicteurs Différents algorithmes de prédiction seront testés. Est-ce qu'un modèle de langage lissé trigramme obtiendra de meilleurs résultats qu'un modèle unigramme ou bigramme? Quel gain apportent les différents paramètres de lissage de l'interpolation linéaire? La combinaison d'un modèle de langage général avec un modèle utilisateur est-elle efficace? Et quels seront les résultats d'un prédicteur utilisant un modèle de langage lissé et encodé en format ARPA?

Question 3 – Prédictions Est-ce que nous pouvons supposer que plus leur nombre augmente, plus il y a de chances que le mot prédit soit le bon et donc celui recherché par l'utilisateur? Nous pouvons nous demander ce qu'il en est de l'usage réel qu'en font les personnes handicapées.

Question 4 – Interface utilisateur Nous avons vu que de nombreuses études apparaissent afin de trouver des solutions ergonomiques pour optimiser la présentation des prédictions. Nous évaluerons donc un utilisateur handicapé sur son interaction avec l'interface et sur son degré de satisfaction afin de proposer des pistes d'amélioration.

La première question concerne l'impact du choix du corpus; la deuxième, celle des différents algorithmes de prédiction; la troisième, celle du nombre de suggestions proposées à l'utilisateur; et la quatrième, celle de l'ergonomie relative à l'expérience utilisateur. Nous répondrons aux trois premières questions à l'aide de deux simulateurs (point 3.2.1) : le premier est intégré dans Presage et calcule automatiquement le Keystroke Saving Rate (KSR) sur un corpus textuel de test; le deuxième est un script que nous avons écrit afin d'obtenir des résultats plus précis. Enfin, la dernière question reposera sur un test utilisateur avec une personne handicapée (point 3.2.2).

2 Outils

Dans cette section, nous présentons les systèmes qui font l'objet de notre étude et qui ont l'avantage d'être tous les deux *open source*, Presage étant licencié sous *GNU General Public License version 2.0 (GPLv2)* et ACAT sous l'*Apache License version 2.0*.

2.1 Presage

Le système de prédiction a été conçu par Vescovi (2004). La version que nous testerons est la dernière disponible sur SourceForge (0.9.1, 2015).¹ Le prototype a changé de nom en 2008, Soothsayer étant alors devenu Presage à cause d'un logiciel de commerce ayant exactement la même fonction.²

1. <http://presage.sourceforge.net/>

2. Une news annonce ce changement (<http://presage.sourceforge.net/?q=node/38>). Il s'agit de la version SoothSayer Word Prediction d'Applied Human Factors (<http://www.ahf-net.com/soothsayer/>).

2.1.1 Description générale

Écrit en C++, Presage est rapide et multi-source car il est doté de plusieurs algorithmes qui génèrent des prédictions en modélisant le langage et en combinant des sources d'informations redondantes à l'aide de probabilités. Son architecture modulaire ainsi que son système de plugins prédictifs permettent de facilement le modifier et d'étendre ses modèles de langage. Il serait possible de développer des algorithmes de prédiction statistique, syntaxique ou sémantique. Vescovi (2004) citait le possible recours à WordNet. Presage est capable de prendre en compte le contexte et on pourrait constamment l'améliorer en étendant ses fonctionnalités et ses capacités prédictives. D'après un commentaire figurant dans le code source :

“ Presage is fundamentally different from predictive input technologies commonly found on mobile phones, which might more accurately be described as 'disambiguating text entry' rather than 'predictive text entry' systems. Such systems do not try to guess what the user intends to write in the future, only to determine what they most-likely intend to write in the present, given their past input. Presage [...] only reverts to word completion mode if the prediction did not contain the desired token.

”

La création de ressources spécialisées dans une application particulière ou un environnement personnalisé pour un utilisateur spécifique ou un type d'utilisateur pourrait produire des améliorations significatives dans la précision de la prédiction. Le degré élevé de modularité et de flexibilité du logiciel permet aussi au système d'être utilisé comme cadre pour tester et comparer différents mécanismes de prédiction. Presage ne comprend pas de clavier virtuel, mais un simple éditeur de texte multiplateforme (Windows, Linux, Mac OS, etc.) est intégré (cf. Figure I.1b).

2.1.2 Algorithmes de prédiction

Presage possède six algorithmes de prédiction qui peuvent être (dés)activés via un fichier .xml (cf. point 3.1.5), chacun demandant différents types de ressources :

PRÉDICTEUR	RESSOURCE
1 Generalized (default/user) smoothed n-gram	database.db / lm.db
2 ARPA	arpa(.lm) / vocab(.txt)
3 Abbreviation expansion	abbreviations.txt
4 Dejavu	dejavu_memory.txt
5 Dictionary	word.txt
6 Recency	–

TABLE II.1 – Prédicteurs de Presage

Nous avons décidé de limiter notre étude aux trois premiers prédicteurs.

Generalized (default/user) smoothed n-gram calcule la prédiction la plus probable à partir du contexte en cours de saisie et du modèle markovien de langage. Par défaut, il s'agit d'un modèle trigramme lissé (II.1) avec interpolation linéaire reposant sur deux bases de données qui regroupent des n-grammes, c'est-à-dire tous les mots constituant le vocabulaire du système. Le fichier database.db (default) est généré à partir d'un corpus textuel en utilisant, par exemple, l'outil text2ngram. Le fichier lm.db (user) lié au *ngram language model* de l'utilisateur est créé automatiquement lorsque le module d'apprentissage automatique est activé et que l'utilisateur saisit des mots.

$$P(m_3|m_1, m_2) = \lambda_1 \cdot f(m_3|m_1, m_2) + \lambda_2 \cdot f(m_3|m_2) + \lambda_3 \cdot f(m_3) \quad (\text{II.1})$$

où $0 < \lambda_i \leq 1$, $\sum_i \lambda_i = 1$, $C(m_0...m_n)$ est le total des n-grammes $m_0, ..., m_n$ et C est la somme de toutes les fréquences d'utilisation des tokens de la base de données :

$$\begin{aligned} f(m_3|m_1, m_2) &= \frac{C(m_1, m_2, m_3)}{C(m_1, m_2)} \\ f(m_3|m_2) &= \frac{C(m_2, m_3)}{C(m_2)} \\ f(m_3) &= \frac{C(m_3)}{C} \end{aligned}$$

ARPA est un prédicteur statistique utilisant un fichier ASCII. Les modèles de langage sont généralement stockés dans un format ARPA (cf. Table II.4), qui présente chaque n-gramme avec sa probabilité sous la forme d'un logarithme (\log_{10}) et un coefficient α appelé le poids *backoff* (II.2), issu de la technique de lissage de Katz. Dans un modèle avec un repli, si le n-gramme (n) recherché a une fréquence nulle, l'estimation se fera sur la base du n-gramme de rang inférieur ($n - 1$). Jurafsky et Martin (2009) citent deux outils permettant d'obtenir ce type de fichier : le SRILM toolkit³ et le Carnegie Mellon University (CMU) toolkit⁴ (Rosenfeld et Clarkson, 1997).

$$\log P * (m|h) \quad mh \quad \log \alpha(h) \quad (\text{II.2})$$

où h désigne l'historique et la technique de *backoff* de Katz est formalisée :

$$\begin{aligned} P_{\text{bo}}(m_n|m_{n-2}, m_{n-1}) &= \begin{cases} \frac{C^*(m_{n-2}, m_{n-1}, m_n)}{C(m_{n-2}, m_{n-1})} & \text{si } C(m_{n-2}, m_{n-1}, m_n) > 0 \\ \alpha_{m_{n-2}, m_{n-1}} P_{\text{bo}}(m_n|m_{n-1}) & \text{sinon} \end{cases} \\ P_{\text{bo}}(m_n|m_{n-1}) &= \begin{cases} \frac{C^*(m_{n-1}, m_n)}{C(m_{n-1})} & \text{si } C(m_{n-1}, m_n) > 0 \\ \alpha_{m_{n-1}} P_{\text{bo}}(m_n) & \text{sinon} \end{cases} \end{aligned}$$

3. <http://www.speech.sri.com/projects/srilm/>

4. <http://www.speech.cs.cmu.edu/SLM/toolkit.html>

Abbreviation expansion permet d'accélérer la saisie grâce à l'encodage d'une liste d'abréviations extensible (cf. Table II.5). À titre d'exemple, le fichier par défaut contient les abréviations suivantes : 2 *to* / 4 *for* / 2*moro tomorrow* / u *you* / w8 *wait* / b4 *before* / c *see* / r *are* / afaik *as far as I know* / asap *as soon as possible* / ianal *I am not a lawyer*.

2.2 Assistive Context-Aware Toolkit (ACAT)

ACAT est disponible depuis août 2015. La version que nous testerons est la dernière disponible sur GitHub (1.0.0, 31/10/2016).⁵ Il s'agit d'une interface utilisateur développée en C# par les chercheurs d'Intel Corporation. Presage est inclus par défaut dans la version adaptée au public.

2.2.1 Description de l'interface

Initialement conçu pour les besoins d'une seule personne, le physicien Stephen Hawking (Denman *et al.*, 2016), atteint d'une Sclérose Latérale Amyotrophique (SLA), ACAT est un logiciel de communication qui a l'avantage de permettre différentes modalités de saisie en fonction du handicap de l'utilisateur. Il intègre plusieurs claviers virtuels utilisables avec la souris ou par défilement bloc/ligne/colonne grâce au système de reconnaissance de mouvements des sourcils, des joues ou de la bouche via une caméra ou webcam. À noter qu'un calibrage de quelques secondes est nécessaire avant chaque utilisation si ce mode de saisie est activé.

La version de base d'ACAT est en anglais, mais vu qu'il s'agit d'un logiciel ouvert à tous les développeurs, des packs de langue pour le français, l'espagnol et le portugais sont maintenant disponibles.⁶ Ils incluent la traduction d'une partie de l'interface, différentes dispositions de clavier et leurs spécificités comme les lettres accentuées ou ligatures (ââæèêëïïôùûüç) ainsi qu'une base de données de n-grammes de mots pour la prédiction (cf. Figure I.2a). Elle n'a pas été traitée (présence de majuscules, signes de ponctuation, etc.) et a souvent été réalisée à partir d'un seul livre (libre de droit).

Le système comprend un tableau de bord (Figure II.1) permettant d'accéder à douze applications, dont plusieurs sont regroupées en deux modes (*Talk* ou *App*) et déclinées selon quatre dispositions de clavier (QWERTY, ABC, Alt et Alt ABC).⁷ Comme dans le logiciel Sibylle (Wandmacher et Antoine, 2007b), l'interface de chacune des applications regroupe des (sous-)claviers (cf. Figures II.2 et II.3) afin de pouvoir sélectionner des caractères, des chiffres, des signes de ponctuation, des mots et des phrases prédéfinies pour une communication rapide ou urgente. Le passage d'un clavier à l'autre se fait à l'aide du système de défilement par bloc ou de touches de saut.

5. ACAT dispose d'un site (<https://01.org/acat>) et un *repository* (<https://github.com/01org/acat>).

6. Des manuels d'utilisation, un guide pour les développeurs et des vidéos sont également accessibles sur le site (<https://01.org/acat/documentation-list>).

7. Le clavier QWERTY est remplacé par un AZERTY lorsque le pack français est téléchargé.

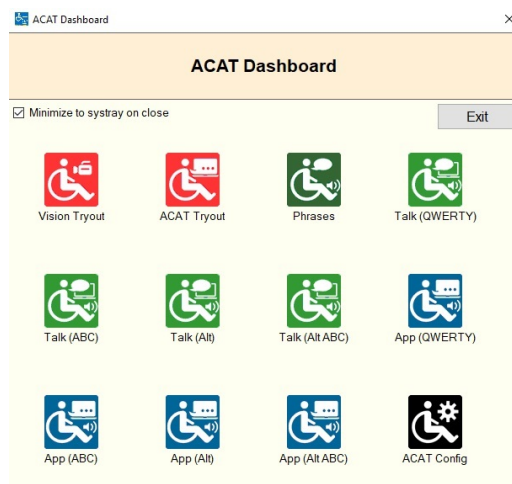


FIGURE II.1 – ACAT : tableau de bord

Contrairement à l'anglais, les langues romanes – dont le français – utilisent plusieurs diacritiques et ligatures. Alors que le système Sibylle traite les lettres accentuées comme de simples lettres, les différents claviers proposés dans ACAT requièrent un appui supplémentaire une fois la voyelle sélectionnée (ou consonne dans le cas du 'c' cédille en français). Cette disposition est configurable à l'aide de fichiers .xml fournis dans les différents packs, ce qui permet d'adapter facilement ACAT à d'autres langues.⁸

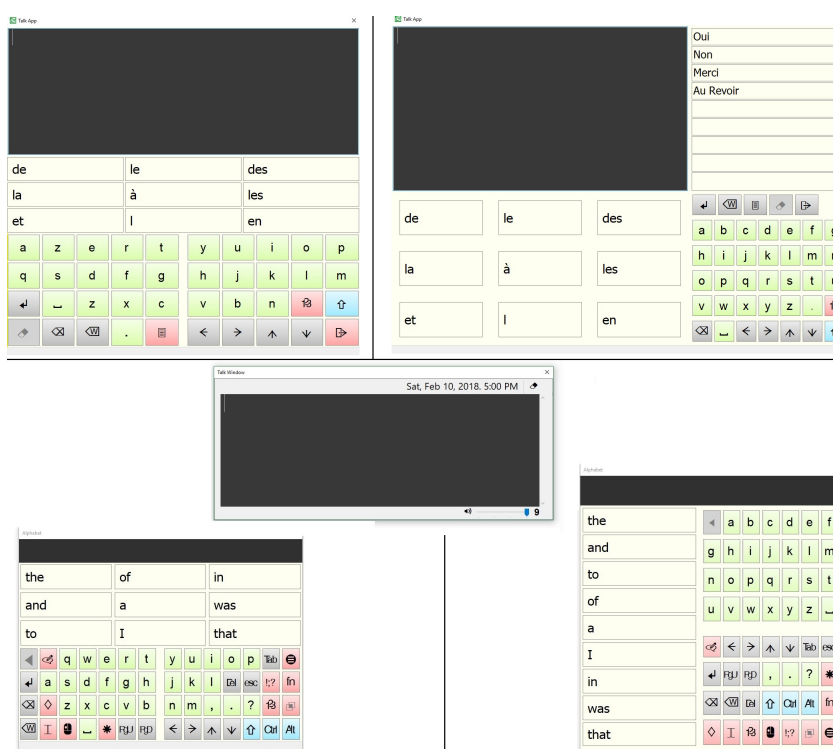


FIGURE II.2 – Claviers : mode Talk AZERTY et ABC ou App QWERTY et ABC

8. Il convient de mentionner que les packs de langue d'ACAT ont été réalisés par des développeurs anonymes (et donc pas par Intel). La prise en compte des diacritiques comme de simples lettres nécessiterait une modification des fichiers .xml des claviers et des images des deux thèmes (cf. point 2.2.2).

Les claviers de lettres comprennent toujours l'espace comme séparateur de mots ainsi que différents boutons pour effacer une lettre, un mot ou toute la fenêtre qui sert d'éditeur de texte pour la synthèse vocale (*Talk Window*). En mode *Talk*, le seul signe de ponctuation accepté est le point bien que le point, la virgule et le tiret (ou signe moins) sont disponibles depuis le pavé numérique (Figure II.3). Le mode *App* offre un clavier plus complet (proposant la virgule, le point et le point d'interrogation), de nombreux raccourcis, des touches de saut permettant d'accéder aux autres applications ainsi qu'un sous-clavier reprenant presque tous les signes de ponctuation (Figure II.3).



FIGURE II.3 – Sous-claviers : pavé numérique et ponctuation

En fonction de l'application, la liste des prédictions lexicales en propose 8 ou 9. Elle est présentée soit verticalement, soit à la fois horizontalement et verticalement. Le balayage automatique du clavier se distingue par sa couleur jaune. Pour sélectionner un élément en mode défilement, l'utilisateur doit donc cliquer trois fois (sauf si la liste est affichée verticalement). Il sélectionne d'abord le bloc (la liste de prédictions ou une partie du clavier), ensuite la ligne et enfin la colonne. Presage fait en sorte que lorsque l'utilisateur accepte une prédiction, l'espace typographique soit automatiquement ajoutée et que s'il ajoute un point, celui-ci vienne directement se coller au mot.

Comme son nom l'indique, l'application *Phrases* permet d'encoder une liste de phrases qui seront directement synthétisées. Elle est accessible avec le système de défilement et contient les exemples suivants : *Yes* (Oui) / *No* (Non) / *Thank you* (Merci) / *Good bye* (Au Revoir). Il n'y a pas de limite quant à la longueur des phrases et l'utilisateur peut les trier dans l'ordre alphabétique (inverse). Ce sous-clavier est également accessible depuis les touches de saut des claviers des différentes applications.

Tout comme le mécanisme de remplacement d'abréviation qui a pour but d'accélérer la vitesse de communication (cf. point 3.1.4), la gestion des phrases fréquemment utilisées est, selon les manuels, une fonctionnalité souvent intégrée dans les systèmes commerciaux de CAA comme PolyPredixTM ou Mind Express.⁹ Cette liste peut servir à encoder des messages d'urgence comme ceux proposés par Antoine et Maurel (2007, p. 28) : « je suis mal installé sur mon fauteuil », « je veux aller aux toilettes », etc. Nous avons utilisé une vingtaine de ces phrases pour constituer un des corpus de test.

9. Les deux logiciels cités proposent une prédiction de phrases. Le manuel du Mind Express 4 est disponible en ligne sur le site de la société Jabbla (<http://www.jabbla.com>).

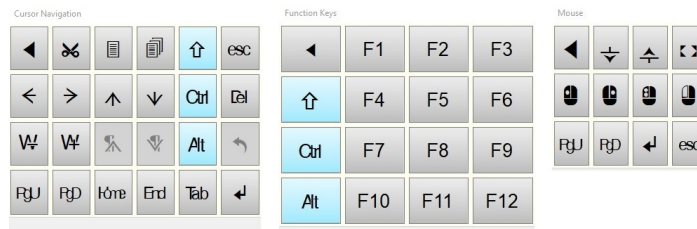


FIGURE II.4 – Autres sous-claviers : navigation, touches de fonction et souris

ACAT est doté d'une synthèse vocale (*text to speech*), dont il est possible de changer la voix, la vitesse d'élocution ou le volume. Certains boutons permettent d'effectuer différentes tâches sur ordinateur comme accéder aux mails ou à des documents et dossiers spécifiques, naviguer sur Internet via des commandes de déplacement, etc. Il existe plusieurs logiciels qui proposent des fonctions similaires, nous pouvons par exemple citer les applications proposées avec la commande oculaire Irisbond.¹⁰

2.2.2 Flexibilité et paramétrage utilisateur

Les possibilités offertes par le paramétrage utilisateur sont particulièrement importantes dans le cadre d'une application d'aide au handicap en raison de la diversité de ceux-ci. Certains paramètres sont directement accessibles depuis l'interface utilisateur via l'application ACAT Config qui les classe en neuf catégories : *General*, *Actuators*, *Text-to-speech*, *Word Predictors*, *Applications*, *Tools*, *Spell Checkers*, *Color Scheme* et *Language*. Contrairement aux autres applications, celle-ci n'est utilisable qu'avec une souris et un clavier.¹¹ Il est ainsi possible de paramétrer des raccourcis clavier ou de changer la vitesse de défilement, le nombre d'itérations de plusieurs éléments, etc. Concernant la prédiction, diverses options sont proposées, dont les suivantes :

Exemple 1 : 'WordPredictionNGram' modifie le nombre de n-grammes.

Exemple 2 : 'EnableWordPredictionDynamicModel' (dés)active l'apprentissage.

Exemple 3 : 'WordPredictionCount' modifie le nombre de mots dans la liste.

Bien qu'il ne soit pas possible pour l'utilisateur de réagencer les touches des claviers, des paramètres plus précis relatifs aux packs de langues sont accessibles depuis des fichiers .xml, ce qui évite de devoir recompiler le code source. L'utilisateur peut choisir entre deux thèmes : noir (par défaut) ou blanc. Les options de paramétrage du rendu visuel du clavier sont intéressantes car certains utilisateurs présentent également des troubles perceptifs. Les touches du clavier ont une couleur qui dépend de leur fonction. À noter que les touches de fonction (Fn) et de combinaison (Shift, Ctrl, Alt) utilisées pour faire des raccourcis clavier sont disponibles (cf. Figures II.2 et II.4).

10. Nous avons pu assister à une démonstration lors d'une réunion organisée par Comalso. Conçue en Espagne (<http://www.irisbond.com/>), elle devrait bientôt être commercialisée en Belgique.

11. Un menu de réglage est néanmoins accessible par défilement depuis les claviers en mode *App*.

3 Méthodologie

Cette section présentera notre méthodologie pour la configuration (Sous-section 3.1) et l'évaluation (Sous-section 3.2) des systèmes que nous venons de détailler. Un aspect de ce mémoire consiste en la création et le traitement de ressources, peu disponibles, en français. À noter qu'à notre connaissance, le corpus Google Books Ngram n'a jamais été utilisé dans le cadre d'un système de prédiction.

3.1 Configuration

Pour cette étape, nous avons d'abord analysé les bases de données existantes dans Presage et ACAT. La configuration par défaut de Presage varie en fonction des langues des corpus : pour l'anglais, le corpus utilisé est *The Picture of Dorian Gray* d'Oscar Wilde; pour l'espagnol, *El Quijote* de Miguel de Cervantes Saavedra; et pour l'italien, *L'Innocente* de Gabriele d'Annunzio.¹² Nous avons préalablement réalisé plusieurs tests avec le simulateur de Presage afin d'en comprendre le fonctionnement.

3.1.1 Corpus

De nombreux corpus écrits ou oraux sont maintenant disponibles sur le Web (Fairon, 2010). Certains ont été annotés par des linguistes, c'est le cas du *Brown Corpus* ou du *Penn Treebank* qui compte plusieurs millions de mots.

Une grande majorité concerne l'anglais même si de plus en plus de corpus français font leur apparition. En plus de pouvoir servir de ressources pour des approches *data-driven*, la plupart sont conçus pour des applications spécifiques comme la reconnaissance de la parole, la traduction automatique ou l'étude des variations (socio)linguistiques. Il est possible de trouver des corpus monolingues ou multilingues dans différents formats et sur des sujets variés. Des organisations comme le Linguistic Data Consortium (LDC)¹³ ou l'European Language Resources Association (ELRA)¹⁴ se chargent de les réunir et de les distribuer moyennant parfois une licence payante.

Dans notre étude, nous proposons de tester différents types de corpus comme l'avaient fait Antoine (2011) – uniquement pour les corpus de test – ainsi que Trnka et McCoy (2007) qui avaient aussi fait varier les corpus d'entraînement. Les corpus que nous avons utilisés n'avaient pas tous le même format. Ainsi, Google Books Ngram était déjà présenté sous la forme de n-grammes dans des fichiers .csv séparés et nous avons également récupéré des textes (.txt). Nous leur avons appliqué divers prétraitements et des versions en format standard ARPA ont été réalisées (cf. points 3.1.2 et 3.1.3).

12. Ce sont tous des textes provenant du Projet Gutenberg (<http://www.gutenberg.org>), un site connu pour sa mise à disposition de livres tombés dans le domaine public.

13. <https://www.ldc.upenn.edu/>

14. <http://catalog.elra.info/en-us/>

Avant de décrire nos corpus de test, nous présentons les trois corpus français d'entraînement ci-dessous, à savoir : Google Books Ngram, À se tordre et Perso. Pour ces deux derniers corpus, le nombre de mots indiqué est celui du fichier .txt prétraité.

CORPUS (ID)	GENRE	NBRE DE MOTS	DESCRIPTION
Google Books Ngram	Littéraire	46 786 461 uni/bi/trigrammes	310 fichiers Version 20090715
À se tordre	Littéraire	38 123	Roman (1891) d'Alphonse Allais
Perso	Idiolecte	18 377 : 14 071 4 306	Total : 44 articles de blog + interview écrite

TABLE II.2 – Corpus français d'entraînement

Google Books Ngram

En 2004, Google a lancé un vaste projet de numérisation qui met à la disposition des internautes plusieurs millions de livres. Quelques années plus tard, une application linguistique relative à Google Books propose d'observer l'évolution – sur plus de cinq siècles (années 1500 à 2000) – de la fréquence de mots ou groupes de mots : Google Books Ngram Viewer. En plus d'avoir mis ce corpus en ligne,¹⁵ les développeurs ont introduit le terme *culturomics* (culturomique) pour désigner un nouveau champ d'application de la lexicométrie (Michel *et al.*, 2011).¹⁶ Nous avons choisi d'utiliser la version française constituée en 2009 en raison de la taille des fichiers plus réduite et donc plus facile à manipuler que ceux de 2012. Selon Google, le corpus français contient 45 milliards de mots et les livres ayant une basse qualité d'océrisation (cf. Chapitre I, Sous-section 3.4) ainsi que les périodiques sont exclus; les n-grammes inclus apparaissent au moins quarante fois dans le corpus. Nous avons décidé de restreindre le corpus (points 3.1.2 et 3.1.3) en ne prenant, par exemple, que les occurrences de n-grammes apparaissant dans les livres des années 2000 à 2008.

À se tordre

À se tordre : histoires chatnoiresques est un roman de l'écrivain français Alphonse Allais, écrit en 1891. Le livre est tombé dans le domaine public et le texte brut est disponible sur le site du Projet Gutenberg.¹⁷ Nous l'avons utilisé pour pouvoir le comparer à nos autres corpus. Nous avons également voulu vérifier les conseils d'un utilisateur sur le *repository* d'ACAT qui le conseille et dit que cela devrait bien fonctionner.¹⁸

15. L'ensemble des données sous forme de n-grammes est sous la *Creative Commons Attribution 3.0 Unported License* et disponible (<http://storage.googleapis.com/books/ngrams/books/datasetsv2.html>); des informations supplémentaires sont accessibles (<https://books.google.com/ngrams/info>).

16. Les auteurs ont également lancé leur site (<http://www.culturomics.org>).

17. <http://www.gutenberg.org/ebooks/13834>

18. <https://github.com/01org/acat/wiki/Changing-language-and-creating-new-dictionaries>

Perso

Nous avons constitué ce petit corpus à partir des articles (années 2010 à 2016) du blog d'une personne ayant une Infirmité Motrice Cérébrale (IMC) athétosique après avoir obtenu son autorisation par mail. Nous avons eu recours à une technique de Web scraping et nous avons rajouté une interview écrite (2017) de notre utilisateur de test afin d'avoir un point de comparaison entre les corpus littéraires et celui représentant l'idiolecte de potentiels utilisateurs. L'ajout de l'interview avait pour but de renforcer la taille et la représentativité de ce type de registre.

Corpus de test

Afin d'évaluer le système de prédiction, nous avons utilisé différents corpus français de test. Trnka et McCoy (2007) ont montré que des évaluations de type *in-domain*, *out-of-domain* et *mixed-domain* avaient été effectuées par plusieurs chercheurs. Nous avons choisi de réaliser des évaluations de type *out-of-domain*. Autrement dit, nous testerons les prédictions sur un texte ne faisant pas partie du corpus d'entraînement. Les auteurs disent que cette approche ne permet pas toujours d'évaluer de nouvelles techniques, les résultats pouvant être dus à la trop grande différence entre les corpus, mais elle permettrait néanmoins de se rapprocher des performances de réels utilisateurs qui ont des besoins de communication variés.

Corpus A Le premier corpus est constitué d'une centaine de phrases. Il s'agit de séquences de 3 à 15 mots qui proviennent de sources différentes. Nous avons téléchargé un corpus anglais destiné à évaluer des techniques de saisie de textes, constitué par MacKenzie et Soukoreff (2003).¹⁹ Nous avons sélectionné au hasard 40 phrases (parmi les 500) que nous avons traduites en français. Elles sont très générales et ne comprennent pas forcément de verbe. Nous avons également repris 20 phrases préenregistrées dans la version K du système de prédiction Sibylle. Elles sont simples et adaptées au contexte de communication alternative. Enfin, les 40 phrases restantes le sont également et proviennent de nos contacts. Lors d'une réunion avec Comalso, nous avons obtenu ces phrases de la directrice et d'une étudiante en logopédie qui réalisait son mémoire sur l'évaluation du système polysémique Minspeak® (cf. Table II.7).

Corpus B.a et B.b Pour le deuxième corpus, nous avons envisagé de reprendre une partie de nos corpus d'entraînement, mais l'une des particularités de Google Books Ngram est qu'il est impossible d'avoir accès aux textes bruts qui ont été réellement utilisés. Ce corpus étant également déjà sous la forme de n-grammes, nous n'avons donc pas pu le diviser afin de constituer un corpus de test. Les textes tokenisés (cf. *infra*) Perso (B.a) et À se tordre (B.b) ont donc servi de corpus de test aux autres, mais pas quand ils étaient déjà utilisés pour l'entraînement.

19. <http://www.yorku.ca/mack/PhraseSets.zip>

3.1.2 Prétraitements

Pour la normalisation et la tokenisation, il existe différents outils qui intègrent la possibilité de recourir aux expressions régulières ou regex.²⁰ Nous n'avons pas tokenisé tous les corpus de la même façon, certains l'étant déjà intégralement ou partiellement. À noter que la tokenisation est une étape délicate qui, bien qu'elle soit souvent nécessaire, n'est jamais vraiment détaillée dans nos sources.

Google Books Ngram

Chaque fichier de Google Books Ngram se présente comme suit :

« ngram TAB year TAB match_count TAB page_count TAB volume_count NEWLINE ».

Afin de réduire la taille des nombreux fichiers (.csv) et nettoyer les données, nous avons effectué un premier traitement automatique à l'aide du script Perl disponible à l'Annexe B.²¹ Ce prétraitement visait à ne garder que les colonnes *ngram* (\$col[0]) et *match_count* (\$col[2]). Nous avons décidé de conserver les accents français, de mettre toutes les données en minuscule et de supprimer une partie des lignes non exploitables, c'est-à-dire les mots contenant des caractères spéciaux non français (comme le ñ espagnol), les chiffres²² ou certaines erreurs (cf. *infra*). Quant aux occurrences de n-grammes datant d'avant l'an 2000 (\$col[1]), elles ont toutes été ignorées.

```
if ($col[0] !~ /[a-z]/ or $col[0] =~ /^[^a-z\s^]|[\^\^]|(.) (\1{2,})|\b([b-z])\b
    (\1{0,})/g or $col[1] =~ /\b1/) {
    next readfile;
}
```

La tokenisation est une étape cruciale car elle détermine les mots qui entreront dans la base de données éliminant ceux qui pourraient être proposés à l'utilisateur dans la liste de suggestions. Il a donc fallu déterminer avec précision ce qu'était un mot français – et ce qui ne l'était pas – en opérant une série de choix. Nous avons retiré toute la ponctuation. La segmentation sur la base de l'espace aurait produit des virgules, points, etc. accolés aux mots comme le mentionnent Jurafsky et Martin (2009, p. 103).

Néanmoins, la non prise en compte des majuscules ou la suppression de mots avec des apostrophes (ou tirets) a également un impact direct sur les futures prédictions. Étant donné le taux important de n-grammes et d'erreurs (dues sans doute à la mauvaise océrisation des accents, majuscules, apostrophes et tirets), nous avons fait en sorte que les prédictions soient le plus correctes possible en réduisant le bruit afin

20. L'outil en ligne RegExr (<http://regexr.com/>) est utile pour *matcher* correctement les expressions.

21. Ce script a été utilisé une première fois sur chacun des 310 fichiers français de 2009 (10 fichiers d'unigrammes, 100 de bigrammes et 200 de trigrammes). Il a été réutilisé sur les fichiers obtenus de chaque n-gramme afin de fusionner les derniers doublons (entre fichiers) en faisant leur somme.

22. Nous pensons qu'il est inutile de prédire des chiffres bien que la version K de Sibylle en propose dans la liste de suggestions (et que les systèmes apprennent automatiquement ceux de l'utilisateur).

que l'utilisateur ne doive pas recourir à un correcteur orthographique.

Par conséquent, nous avons tenté d'éliminer les lignes non françaises ou comportant des erreurs, en nous basant sur une analyse fréquentielle des n-grammes de lettres, une méthode utilisée en cryptanalyse. Ainsi, des tokens comportant plus de trois lettres identiques consécutives ont été discriminés, tandis que des séquences de deux lettres identiques ont été conservées. Dans ce corpus, nous avons également retiré les mots d'une lettre (sauf l'auxiliaire « a » et la préposition « à »).

Il ne faut pas oublier que lorsque l'utilisateur sélectionne une prédiction, une espace typographique s'ajoute automatiquement (cf. Chapitre I, point 1.3.2). Or, s'il s'agit d'une lettre précédant une apostrophe ou un tiret (par exemple : « c » pour « c'est » ou « t » pour « t-il »), un appui serait perdu vu qu'il devrait l'effacer pour écrire le mot suivant. Pour réunir tous les fichiers générés via le script Perl, un fichier Batch contenant « copy *.csv importfichier.csv » permet de le faire en une fois.

Bien que nous ayons éliminé les doublons de chaque fichier, il restait encore des n-grammes communs entre ceux-ci. Pour fusionner les derniers éléments redondants et calculer leur somme, nous avons utilisé le logiciel OpenRefine²³ qui permet de trier les colonnes et d'effectuer divers traitements à l'aide des facettes et des fonctions GREL (General Refine Expression Language).²⁴ L'historique des opérations en format JSON (JavaScript Object Notation) est disponible à l'Annexe B.

À se tordre

Après avoir enlevé la table des matières, le préambule et la licence de fin du Projet Gutenberg, nous avons traité ce corpus avec le script utilisé pour l'interview (cf. *infra*).

Perso

Pour constituer ce petit corpus (.txt), nous avons écrit un script Python afin de faire du Web scraping avec la bibliothèque BeautifulSoup et tokeniser le tout avec la bibliothèque Natural Language Toolkit (NLTK) (Bird *et al.*, 2009).²⁵ Le script et la liste des articles utilisés sont respectivement disponibles aux Annexes B et C. Nous avons également traité l'interview avec NLTK afin d'enlever les majuscules, la ponctuation (les points, les virgules, les apostrophes, les tirets, etc.) tout en gardant les accents français, les chiffres et les entités nommées (vu qu'il était difficile de les discriminer automatiquement), mais en enlevant les retours chariot.

Corpus de test

Bien que le simulateur utilisé tokenise les corpus de test (cf. points 3.1.5 et 3.2.1), nous avons préféré utiliser le script de prétraitement afin d'assurer leur compatibilité.

23. <http://openrefine.org/>

24. Verborgh et De Wilde (2013) ont écrit un ouvrage de référence très complet.

25. <http://www.nltk.org/>

3.1.3 Formats de données

Le corpus Google Books Ngram étant en format .csv, nous l'avons importé dans une base de données SQLite après avoir créé des tables d'unigrammes (`_1_gram`), bigrammes (`_2_gram`) et trigrammes (`_3_gram`) similaires à celles en anglais (cf. Figure I.2a). La liste des commandes SQL est disponible à l'Annexe B. Pour l'extraction des statistiques n-grammes des corpus textuels À se tordre et Perso, nous avons utilisé l'outil `text2ngram`²⁶ fourni dans le dossier d'installation de Presage. Il permet de créer automatiquement une base de données composée de tables contenant les n-grammes. Chaque table possède une ou plusieurs colonnes remplies de mots (*word*, *word_1*, *word_2*, etc.) ainsi qu'une colonne contenant les fréquences (*count*).

Malgré tous les traitements effectués sur le corpus Google Books Ngram, il restait des mots non français (anglais, espagnol, latin, etc.) ou inexistantes. Afin de restreindre le taux de bruit, nous recommandons de déterminer un seuil de suppression des fréquences fixé empiriquement à 200 pour les unigrammes, compte tenu du nombre de n-grammes qui nous semblaient incorrects et inexploitable. Ces mêmes mots devraient être supprimés dans les bigrammes et les trigrammes. Nous proposons aussi d'enlever les bigrammes inférieurs à 50 et les trigrammes inférieurs à 10.²⁷ À noter que les n-grammes les plus fréquents sont les *stopwords* et que nous n'avons pas pu tester dans Presage cette base de données ainsi nettoyée en raison de bugs dans le système. Afin d'accélérer la vitesse de lecture de nos bases de données, des index ont été créés.

	GOOGLE BOOKS NGRAM		À SE TORDRE		PERSO	
Unigramme	1 269 020	64 062	6 878	6 943	2 927	2 966
Bigramme	12 908 527	3 425 581	25 359	25 949	11 331	11 468
Trigramme	32 608 914	31 183 453	35 054	35 247	15 989	16 209

TABLE II.3 – Nombre de n-grammes utilisés pour le prédicteur par défaut et ARPA

Le CMU toolkit a été utilisé pour générer les fichiers .arpa et .vocab des trois corpus. Leurs statistiques ainsi que celles des bases de données sont reprises dans la Table II.3. La différence dans le nombre de n-grammes entre le modèle de langage par défaut et ARPA est due à plusieurs raisons. La première est que l'outil utilisé limite le vocabulaire à 65 535 mots, nous avons donc supprimé tous les unigrammes dont la fréquence était inférieure à 2 000. La deuxième est que l'outil génère les fichiers à partir de format .txt, ce que nous ne disposons pas pour Google Books Ngram. Il est donc possible que pour ce corpus, les fichiers aient été moins bien générés. Enfin, la troisième et dernière raison expliquant des différences est liée au fait que nous avons supprimé les lignes qui contenaient des chiffres directement dans les bases de données À se tordre et Perso.

26. <https://homepages.inf.ed.ac.uk/lzhang10/ngram.html>

27. Vu que la moyenne des fréquences diminue en fonction du nombre de n-grammes (unigrammes : 3 846,589 / bigrammes : 278,848 / trigrammes : 67,832), il faudrait diminuer leur seuil.

GOOGLE BOOKS NGRAM (.CSV)	À SE TORDRE (.ARPA)
<pre>_1_gram.csv a,26055066 abandon,161680 abandonna,13666 abandonnai,1196 abandonnaient,3783 _2_gram.csv a,aucun,46904 a,aucune,58929 a,aucunement,1276 a,aussi,96198 a,beaucoup,62225 _3_gram.csv a,aucun,antécédent,220 a,aucune,action,210 a,aussi,abandonné,16 a,beaucoup,abusé,34</pre>	<pre>\data\ ngram 1=6943 ngram 2=25494 ngram 3=35247 \1-grams: -2.3690 a -0.2327 -4.5815 abandon -0.0783 -4.5815 abandonnait -0.0939 \2-grams: -2.9146 a aucun -0.0165 -2.9146 a aussi -0.0147 -2.9146 a beaucoup 0.0434 \3-grams: -1.1224 a aucun mérite -1.1224 a aussi des -1.1224 a beaucoup de</pre>

TABLE II.4 – Formats de données des corpus

Des extraits de fichiers .csv et .arpa sont présentés dans la Table II.4. Comme nous l'avons mentionné lorsque nous avons décrit les prédicteurs (cf. point 2.1.2), le format ARPA permet de stocker tous les n-grammes ainsi que leur fréquence sous la forme d'un logarithme. Les fréquences variant entre 0 et 1, il y a un risque de *numerical underflow* en les multipliant. Il est possible de leur appliquer directement diverses techniques de lissage et interpolations que nous ne détaillerons pas, mais dont le lecteur pourra trouver des informations dans le livre de référence de Jurafsky et Martin (2009).

3.1.4 Abréviations

Une méthode courante pour écrire plus rapidement consiste à utiliser l'écriture abrégée, autrement dit, recourir à différents systèmes abrégatifs tels que les abréviations, les acronymes, la contraction, la troncation, etc. Nous les employons, par exemple, dans le « langage SMS » qui fait progressivement l'objet d'études sociolinguistiques. Nous pouvons citer la constitution d'un vaste corpus de SMS (sms4science), un projet international coordonné par l'Université Catholique de Louvain (UCL).²⁸

Presage dispose d'un prédicteur qui permet d'afficher avec une probabilité certaine (de 1.0) les abréviations listées dans un fichier .txt prédéfini. Nous en proposons quelques-unes dans la Table II.5. Notre choix s'est d'abord porté sur celles qui évitent de générer de l'ambiguïté (c'est-à-dire lorsque des mots autres que l'abréviation peuvent être proposés) et ensuite sur les mots couramment utilisés. Nous avons constaté que moins l'abréviation est longue, plus il y a de risques qu'elle soit ambiguë.

28. www.sms4science.org

C'est pourquoi nous n'avons pas tenu compte de « pr » (« pour ») ou « m » (« monsieur ») alors que nous avons gardé « mme » (« madame »). Nous avons privilégié les abréviations courtes quand il existe deux possibilités non ambiguës, par exemple, « qqn » (et non « qqun ») ou « tjs » (« tjrs »). Nous avons évité les accents, les chiffres (« 2m1 »), les majuscules et les caractères spéciaux (« A+ ») vu que ceux-ci sont moins accessibles et requièrent plus de saisies en raison des touches de saut.

ABRÉVIATIONS ET MOTS			
ajd / auj	aujourd'hui	msg	message
annif	anniversaire	mtn	maintenant
aprem	après-midi	pb	problème
bb	bébé	pcq	parce que
bcp	beaucoup	pdt	pendant
bjr	bonjour	pk / pq	pourquoi
bsr	bonsoir	qd	quand
bxl	Bruxelles	qqch	quelque chose
cmt	comment	qqn	quelqu'un
cv	ça va	rdv	rendez-vous
doct	docteur	resto	restaurant
dsl	désolé	slt	salut
fb	Facebook	stp / svp	s'il te / vous plaît
kdo	cadeau	svt	souvent
lgtps	longtemps	tjr / tjs	toujours
mlle	mademoiselle	tlm	tout le monde
mme	madame	tps	temps
mr	monsieur	we / wk	week-end

TABLE II.5 – Exemple d'abréviations

À noter toutefois que le recours aux abréviations est inutile si l'utilisateur ignore lesquelles sont encodées, il faut donc les définir avec lui en fonction de ses habitudes. Bien qu'elles puissent accélérer la saisie, il peut s'avérer contraignant d'imposer la mémorisation d'une liste d'abréviations (Boissière et Dours, 2001), cet effort entraînant une augmentation de la charge cognitive (Antoine et Maurel, 2007; Ghedira, 2009). Comme le note Schadle (2003), la création de paires (abréviation/mot) est rapidement limitée, une abréviation ne pouvant remplacer qu'un seul mot. Dans ACAT, il est possible de les encoder via un fichier .xml ou depuis l'interface utilisateur. Deux modes sont supportés : écrit (*Write*) et oral (*Speak*). Dans le premier cas, le mot est écrit en toutes lettres dans l'éditeur de texte sans être présenté dans la liste de prédictions (avec une majuscule automatique en début de phrase, mais sans l'espace automatique); dans le deuxième, il est directement synthétisé et effacé de la fenêtre de parole.

```
<Abbreviation word="AJD" replaceWith="aujourd'hui" mode="Write" />
<Abbreviation word="AUJ" replaceWith="aujourd'hui" mode="Speak" />
```

3.1.5 Paramétrage

Les ressources que nous avons constituées sont reliées à un fichier appelé `presage.xml` qui est disponible à l'Annexe B. Pour résumer le fonctionnement de Presage, nous avons réalisé un schéma (Figure II.5), qui reprend ses différents composants.

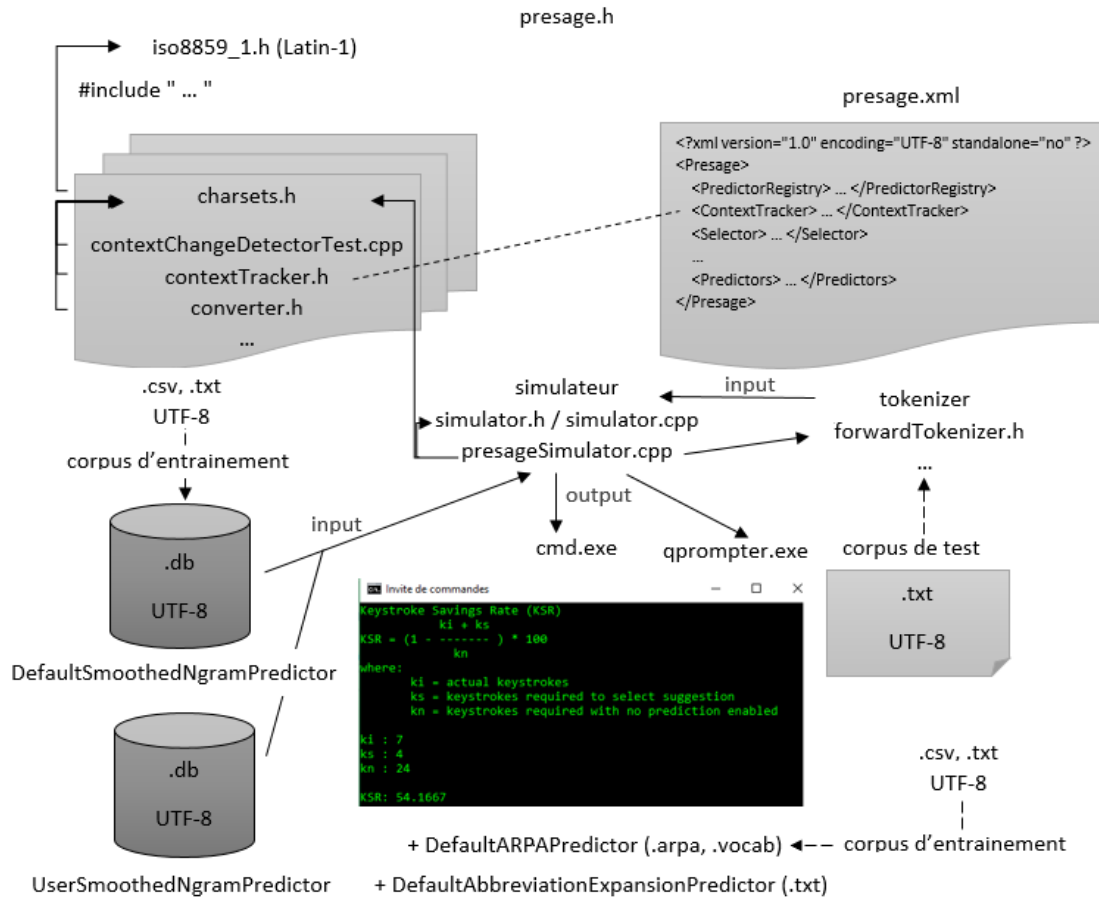


FIGURE II.5 – Presage : fonctionnement

Le fichier `.xml` contient des balises qui renvoient aux modules de base (fichiers `.h` et `.cpp`) du système (exemples : `PredictorRegistry`, `ContextTracker`, `Selector`, etc.). Il permet de choisir et de paramétrer les différents prédicteurs qui sont utilisés dans Presage via l'interface utilisateur `qprompter.exe` (cf. Figure I.1b) ainsi que dans `presage_simulator.exe`. Sous Windows, ce simulateur fonctionne uniquement dans l'Invite de commande (`cmd.exe`) et sera présenté dans le point 3.2.1.²⁹

Nous avons rencontré beaucoup de difficultés en utilisant ce simulateur sur des corpus de test français suite, notamment, à des problèmes d'encodage des caractères (cf. Chapitre I, Sous-section 3.4). Nous supposons que la raison vient du fait qu'il existe un fichier `iso8859_1.h (Latin-1)` et que tous nos corpus sont formatés en UTF-8. En raison de certains biais dans les résultats, nous avons décidé de concevoir un autre simulateur. À noter que le corpus de test est traité automatiquement par un tokenizer.

29. Nous y détaillerons également nos choix quant aux différents changements appliqués au fichier `.xml` dans le cadre de nos simulations.

3.2 Évaluation

Afin d'évaluer notre configuration, nous proposerons deux types d'évaluation : d'une part, une évaluation quantitative basée sur l'utilisation de plusieurs métriques via des simulations (point 3.2.1) ; d'autre part, une évaluation qualitative réalisée avec une personne handicapée et consistant en un test utilisateur (point 3.2.2).

3.2.1 Simulations

Un simulateur appelé `presage_simulator` a été conçu par Vescovi pour faciliter les tests et ainsi évaluer l'efficacité des différentes configurations de manière théorique. Nous avons vu dans le Chapitre I (Sous-section 3.3) que les éventuelles erreurs que ferait l'utilisateur ne sont pas calculées, que la prédiction correcte (avec ou sans espace automatique) est calculée dès qu'elle apparaît dans la liste et qu'il s'agit donc du résultat que l'on obtiendrait dans le meilleur des cas. Nous rappelons que cette méthode a souvent été utilisée avant une évaluation avec de réels participants. Nous pouvons citer Niemeijer (2005) qui avait testé deux versions de son système sur trois textes littéraires anglais (de 2 000 à 4 000 caractères), dont deux sont disponibles en ligne.³⁰

Le simulateur de Presage permet d'estimer automatiquement le Keystroke Saving Rate (KSR) en simulant l'interaction entre l'utilisateur et le système de prédiction lors de l'insertion d'un texte, c'est-à-dire le corpus de test. Celui-ci est inséré lettre par lettre dans le système qui garde une trace du nombre de touches pour le composer et qui permet aussi d'examiner chaque suggestion. À la fin de la simulation, les résultats sont exprimés en fonction des valeurs k_i , k_s , k_n et KSR (II.3) où k_i est le nombre d'appuis effectivement réalisés par l'utilisateur pour composer le mot désiré, k_s le nombre d'appuis requis pour sélectionner une suggestion et k_n le nombre d'appuis qui auraient été nécessaires pour composer le texte sans utiliser un système prédictif (Vescovi, 2004).

$$\text{KSR} = \left(1 - \frac{k_i + k_s}{k_n}\right) \cdot 100 \quad (\text{II.3})$$

Contrairement à la formule (I.8), ce KSR prend en compte l'interface utilisateur. En fonction de celle-ci, la valeur de k_s est parfois différente. Pour sélectionner une prédiction, l'utilisateur peut devoir effectuer deux appuis s'il s'agit par exemple d'un système à défilement linéaire ligne-colonne. Certains caractères peuvent aussi nécessiter plusieurs appuis. C'est le cas de quelques-unes des lettres accentuées du français se situant parfois sur d'autres pavés ou sous-claviers accessibles via des touches de saut. Les lettres avec un tréma, les ligatures et les majuscules nécessitent le maintien d'une touche enfoncée ainsi qu'une ou plusieurs autres touches sur un clavier standard. En ne comptant qu'un appui par lettre et mot correctement prédit, le simulateur calcule

30. Les courts textes utilisés lors de leurs tests étaient *Fishing For Jasmine* et *High and Lifted Up* accessibles sur le site *eastoftheweb* (<http://www.eastoftheweb.com/short-stories/>).

donc le KSR maximum que l'on pourrait obtenir avec une interface optimale.

Nous nous sommes rendu compte que le simulateur ne calculait pas le nombre de caractères, mais le nombre de bits, ce qui faussait tous les résultats, certains caractères français valant plusieurs appuis. Nous avons aussi remarqué que le nombre total de lettres (k_n) n'était jamais correct. Nous avons donc décidé de concevoir un autre simulateur qui donne les bons résultats. Il traite les prédictions et les fichiers logs obtenus avec le simulateur de Vescovi. Ce deuxième script Python est disponible à l'Annexe B. En plus du KSR (II.3) utilisé par Vescovi (2004), il calcule le taux d'économie de saisie (I.8) employé, par exemple, par Wandmacher (2008) et Trnka (2010) ainsi que le hit ratio ou taux d'utilisation de la prédiction formulé en (II.4). Ces métriques sont exprimées en pourcentage et ont été arrondies à trois chiffres après la virgule.

$$\text{Hit ratio} = \frac{\text{nombre de mots prédits}}{\text{nombre total de mots}} \cdot 100 \quad (\text{II.4})$$

La Table II.6 montre un exemple de saisie dans Presage. Il a été réalisé avec la base de données Google Books Ngram sur une des phrases du premier corpus de test : « on a sonné à la porte ». Après chaque lettre saisie, le système affiche ici cinq prédictions que l'utilisateur peut sélectionner avec la souris ou les touches de fonction F1 à F5 via qprompter (cf. Figure I.1b).³¹ Dans l'Invite de commandes, le simulateur de Vescovi n'affiche pas correctement les lettres accentuées ($\acute{e} = \tilde{A}\odot$, $\grave{a} = \tilde{A}$). Il donne les résultats suivants : $k_i = 8$, $k_s = 4$, $k_n = 24$ et KSR = 50%. Avec notre simulateur, nous obtenons : $k_i = 5$ (les lettres o-s-o-n-p ont été saisies), $k_s = 6$ (tous les mots ont été prédits), $k_n = 21$ (toutes les lettres et espaces valent un clic) et KSR = 47,619%. Il donne également le nombre total de lettres et espaces économisées (16), de mots (6) et le hit ratio (100%).

MOTS À PRÉDIRE	LETTRES SAISIES	F1	F2	F3	F4	F5
on	-	de	la	et	le	les
	o	ou	on			
a	-	peut	ne	a		
sonné	-	pu	vu	fait	été	déjà
	s	souvent	su	souligné	surtout	sur
	so	sous	son	soin	soi	soutenu
	son	sont	sonné			
à	-	le	pour	à		
la	-	la				
porte	-	fois	fin	suite	vie	recherche
	p	première	place	politique	production	porte

TABLE II.6 – Exemple de saisie de la ph. « on a sonné à la porte »

31. L'utilisation de touches (de fonction) comme accès direct à la liste de prédictions ne convient pas à tous les types de handicaps. L'interface de Presage n'est pas un système de CAA.

En 2007, Vescovi explique sur son site qu'il a corrigé un bug lié à son simulateur. Les résultats de sa thèse seraient en réalité « meilleurs » que ceux qu'il a présentés, autrement dit, les 45 à 50 % obtenus sur un petit corpus d'entraînement avec une liste de six prédictions représenteraient en réalité aux alentours des 70 %.³² Vescovi (2004) avait testé son prototype sur différents types de textes : des extraits du roman *La coscienza di Zeno* d'Italo Svevo (ainsi que d'autres écrits de cet auteur), des articles du journal *La Repubblica* et un ensemble de ses mails. Ses résultats montraient que dans ce dernier cas, ils étaient inférieurs aux autres corpus de 2 à 5 %. Le calcul du simulateur se fait à partir du fichier .xml que nous avons déjà présenté (cf. point 3.1.5).

Inclus dans les balises <DELTAS> du prédicteur par défaut et du modèle de langage de l'utilisateur, les paramètres λ_i contrôlent la façon dont les unigrammes, bigrammes et trigrammes sont combinés. Nous verrons qu'ils déterminent des changements dans les performances et c'est pourquoi chaque essai de Vescovi (2004) avait été répété sur chaque type de textes avec un ensemble de paramètres distincts choisis de manière à couvrir un large spectre. Nous avons laissé les paramètres de lissage du modèle trigramme à '0.01 0.1 0.89' sauf lorsque la deuxième question a été abordée. D'après le mail que nous a envoyé Vescovi, son plugin d'apprentissage automatique était activé lors de ses tests, Presage ayant eu la capacité d'apprendre dès le début. Initialement, ce module n'existait pas ; il a été ajouté après pour pouvoir garder le modèle de langage inchangé. Nous avons choisi de ne pas l'activer. Tous nos tests ont été effectués avec les modules d'apprentissage automatique <ONLINE_LEARNING> et <LEARN> désactivés afin de garantir la même probabilité pour toutes les passations.

Par défaut, il s'agit d'une liste de '6' <SUGGESTIONS>. Nous avons modifié le nombre de suggestions en fonction de nos évaluations. Dans l'état de l'art, nous avons vu que même si certains auteurs avaient testé l'impact du nombre de prédictions, la plupart des résultats sont donnés à partir d'une liste de cinq suggestions. C'est pourquoi nous avons décidé d'en faire notre *baseline*. L'option <REPEAT_SUGGESTIONS> commande l'activation du rejet implicite. Lors de nos simulations, nous avons choisi de la désactiver même si elle permettrait de nous rapprocher de l'expérience de réels utilisateurs de systèmes de CAA. La charge cognitive et visuelle peut faire en sorte qu'ils ne choisissent pas toujours le mot désiré dès qu'il apparaît dans la liste de prédictions. Nous avons également mis <GREEDY_SUGGESTION_THRESHOLD> à '1' afin d'afficher les propositions contenant au moins une lettre comme « a » et « à ».

Étant donné que nous avons dû mettre le corpus Google Books Ngram en minuscule afin d'essayer de prendre en compte certains problèmes d'océrisation liés aux doublons, nous avons adapté et normalisé tous nos autres corpus de la même façon. Ce choix nous éloigne des pratiques des utilisateurs de systèmes de CAA puisqu'aucune prédiction avec majuscule et signe de ponctuation ne sera proposée. Nous étions donc

32. <http://multithread.org/?q=node/47>

obligé de traiter nos corpus de test avec le même script utilisé pour nos corpus textuels. L'absence de majuscule ne nous a pas permis de calculer le taux de noms propres. Il convient de rappeler que le simulateur utilise un fichier qui tokenise automatiquement le corpus de test. À noter que bien que le mécanisme d'abréviation fonctionne, le simulateur ne calcule pas le nombre de saisies économisé avec ce prédicteur.

3.2.2 Test utilisateur

Afin d'évaluer Presage intégré à ACAT, nous avons effectué un test utilisateur avec une personne handicapée ayant une IMC athétosique. Le participant était un adulte de 26 ans qui ne parle pas, ne présente pas de déficience intellectuelle, a une bonne vue et un bon niveau de français. Il s'agit d'un utilisateur régulier de systèmes de prédiction (clavier visuel Windows, Skippy) et de CAA (Mind Express). Il utilise généralement 8 prédictions. Ces informations proviennent du questionnaire disponible à l'Annexe D.

La tâche demandée était une recopie avec l'interface utilisateur et la prédiction. La consigne était de l'utiliser dès que le mot correct était affiché dans la liste. L'utilisateur évalué devait recopier une vingtaine de phrases (avec les lettres accentuées, mais sans la ponctuation) issues du corpus de test A et présentées dans la Table II.7. Il avait déjà utilisé ces phrases lors d'une évaluation du système Minspeak[®]. Bien qu'il utilise rarement les abréviations, nous lui avons montré la liste encodée dans ACAT. L'une des consignes était de les utiliser lorsqu'il pensait que c'était nécessaire en fonction des possibilités du texte à recopier afin de voir s'il les utiliserait sans y être contraint.

PHRASES À RECOPIER	
1 bonjour un verre d eau s il vous plaît	11 je ne resterai malheureusement pas longtemps aujourd'hui
2 comment t appelles tu	12 je ne connais pas la réponse mais je vais me renseigner
3 il ne se sent pas bien depuis ce matin	13 comment font les dauphins pour communiquer
4 à quelle heure ferme le magasin	14 c'est un travail de longue haleine
5 la voiture ne démarre plus	15 il vient seulement d'être mis au courant
6 quel temps fera-t-il mercredi	16 cela sera fait quand les poules auront des dents
7 fais attention où tu marches	17 les astres l'ont toujours fascinée
8 finalement je ne vais pas y aller	18 le travail acharné vainc tout
9 je mange des fruits tous les jours	19 les petits ruisseaux font les grandes rivières
10 cette conférence était très intéressante	20 ce n'est pas grave d'être en cage quand on ne sait pas voler

TABLE II.7 – Extrait du corpus de test A

En ce qui concerne l'interface ACAT, nous avons testé le clavier blanc AZERTY en mode *App*. L'utilisateur que nous avons testé ayant des capacités et des besoins spécifiques, nous avons adapté les modalités de saisie à son handicap qui correspond à une situation réelle possible. La crispation de son visage ne permettait pas d'utiliser la webcam et le système de reconnaissance de mouvements des sourcils, des joues ou de la bouche qui est associé au défilement bloc/ligne/colonne. Pour sélectionner les caractères et les prédictions, il a préféré utiliser une souris avec ses pieds (Figure II.6) vu qu'il a du mal à contrôler son corps. L'ordinateur était posé sur une table. Son fauteuil électrique a été surélevé et incliné de façon à reproduire sa position habituelle.

Le test a été réalisé sur un ordinateur portable qui n'était pas celui du participant. Étant donné qu'ACAT ne dispose pas de simulateur, ni de système de *log files*,³³ nous avons utilisé les outils Morae v3.3.3 de TechSmith.³⁴ Ils sont employés dans le cadre de test utilisateur afin d'étudier l'*user experience*. Nous avons utilisé Morae Recorder pour réaliser des enregistrements audiovisuels sans le son et Morae Manager pour convertir les fichiers .rdg en vidéos. Ces logiciels ont donc permis d'enregistrer l'activité à l'écran et les mouvements de la souris représentés par un tracé gris (Figure II.6). Le participant a utilisé uniquement le clavier virtuel positionné en bas à gauche, la liste des phrases à recopier (en format .txt) étant affichée à droite et la fenêtre de parole en haut.

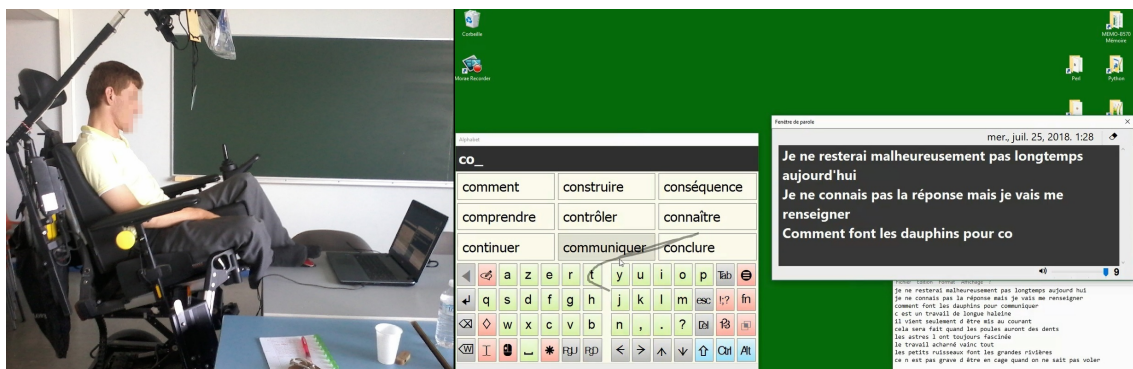


FIGURE II.6 – ACAT et Presage : test utilisateur

Les paramètres de Presage ont été choisis en fonction des résultats théoriques obtenus avec les simulateurs et répondant aux questions relatives aux corpus, prédicteurs et prédictions. Ces choix de configuration (ainsi que leur justification) seront donc explicités dans la partie présentant les résultats (cf. Sous-section 4.4). L'option d'apprentissage automatique n'a pas été testée et est restée désactivée afin de garantir la même probabilité tout au long de l'expérience. Le test utilisateur a duré environ une heure : une séance d'installation et d'explications de 10 minutes suivie de deux sessions (avec une pause), chacune destinée à retranscrire 10 phrases. La première a duré 28 minutes et la deuxième 20. Un formulaire de consentement a aussi été signé.

33. Les options 'DebugLogMessagesToFile', 'DebugMessagesEnable' et 'AuditLogEnable' activées, le système est devenu instable et nous avons dû réinstaller ACAT. D'autres bugs ont été recensés.

34. <https://www.techsmith.com/morae.html> (le lien vers notre vidéo est disponible à l'Annexe B).

4 Résultats

Dans cette section, nous présentons nos résultats quantitatifs et qualitatifs qui permettent de répondre aux quatre questions (cf. Section 1).

4.1 Question 1 – Corpus

Les premiers tests ont été effectués avec le modèle trigramme lissé par défaut. Nous avons fait varier les corpus d'entraînement et de test, dont les tailles et genres diffèrent. La Table II.8 montre les résultats obtenus sur le premier corpus de test.

CORPUS D'ENTRAÎNEMENT CORPUS DE TEST	GOOGLE BOOKS NGRAM PHRASES	À SE TORDRE PHRASES	PERSO PHRASES
k_i	835	1 516	1 567
k_s	573	505	474
KSR	57,932 %	39,617 %	39,020 %
k_p	2 512	1 831	1 780
Taux d'économie	75,052 %	54,705 %	53,181 %
Hit ratio	90,094 %	79,402 %	74,528 %

TABLE II.8 – Résultats : corpus d'entraînement et de test A

Le gros corpus Google Books Ngram obtient les meilleurs résultats en comparaison avec les deux autres corpus, c'est-à-dire une différence de près de 18 % pour le KSR, de plus de 20 % pour le taux d'économie et de 10 à 15 % pour le taux d'utilisation de la prédiction. La différence entre les corpus À se tordre (littéraire) et Perso (idiolecte) est très faible. La Table II.9 montre les résultats obtenus sur les autres corpus de test.

CORPUS D'ENTRAÎNEMENT CORPUS DE TEST	GOOGLE BOOKS NGRAM PERSO	À SE TORDRE PERSO	PERSO À SE TORDRE
k_i	24 375	49 368	120 567
k_s	16 576	13 824	24 595
KSR	59,592 %	37,647 %	29,071 %
k_p	76 971	51 978	84 094
Taux d'économie	75,948 %	51,287 %	41,089 %
Hit ratio	90,204 %	75,228 %	64,653 %

TABLE II.9 – Résultats : corpus d'entraînement et de test B

Le corpus d'entraînement Perso n'est pas du tout adapté au registre littéraire du corpus de test À se tordre. Cela peut s'expliquer par le grand décalage temporel entre les deux corpus. Le corpus d'entraînement est aussi plus petit que le corpus de test. Les résultats ont également montré que lorsque le Google Books Ngram était utilisé comme corpus d'entraînement, plus d'une saisie sur deux était évitée. Les autres corpus obtiennent à chaque fois de moins bons résultats sur les différents corpus de test.

4.2 Question 2 – Prédicteurs

Plusieurs tests ont été effectués afin d'évaluer les performances des différents prédicteurs (cf. point 2.1.2), le nombre de n-grammes et les paramètres de lissage.

4.2.1 Generalized default smoothed n-gram et n-grammes

Nous avons d'abord fait varier le nombre de n-grammes pour évaluer le modèle lissé unigramme ($\lambda_1 = 1$), bigramme ($\lambda_1 = 0,01 / \lambda_2 = 0,99$) et trigramme par défaut.

GOOGLE BOOKS NGRAM						
N	1	2	(3)			
KSR	43,800 %	54,108 %	57,932 %			
T. d'économie	60,382 %	71,287 %	75,052 %			
Hit ratio	87,264 %	90,408 %	90,094 %			
À SE TORDRE				PERSO		
N	1	2	(3)	1	2	(3)
KSR	34,926 %	39,438 %	39,617 %	34,896 %	38,810 %	39,020 %
T. d'économie	49,566 %	54,556 %	54,705 %	48,909 %	53,002 %	53,181 %
Hit ratio	77,044 %	79,559 %	79,402 %	73,742 %	74,685 %	74,528 %

TABLE II.10 – Résultats : prédicteur par défaut et nombre de n-grammes

Plus le n-gramme est élevé, meilleurs sont les résultats en termes de KSR et de taux d'économie quels que soient les corpus d'entraînement sur le corpus de test A.

4.2.2 Generalized default smoothed n-gram et lissage

Nous avons aussi fait varier les paramètres de lissage (λ_i) de la même façon que l'avait fait Vescovi (2004, pp. 89–91). Le paramétrage par défaut utilisé dans tous nos autres tests est le suivant : $\lambda_1 = 0,01 / \lambda_2 = 0,1 / \lambda_3 = 0,89$. Les résultats obtenus avec Google Books Ngram – le meilleur corpus d'entraînement – étant relativement proches (Tables II.8 et II.9), nous avons choisi d'utiliser le corpus de test A.

λ_1	0,7	0,5	0,4	0,3	0,2	0,1
λ_2	0,2	0,3	0,3	0,3	0,3	0,2
λ_3	0,1	0,2	0,3	0,4	0,5	0,7
k_i	1 063	991	961	931	912	879
k_s	563	566	569	570	570	572
KSR	51,419 %	53,480 %	54,287 %	55,153 %	55,721 %	56,647 %
k_p	2 284	2 356	2 386	2 416	2 435	2 468
T. d'économie	68,240 %	70,391 %	71,287 %	72,184 %	72,751 %	73,737 %
Hit ratio	88,522 %	88,993 %	89,465 %	89,622 %	89,622 %	89,937 %

TABLE II.11 – Résultats : prédicteur par défaut et paramètres de lissage

De meilleurs résultats sont donc obtenus si le paramètre de lissage du trigramme λ_3 (le n-gramme le plus élevé) est plus important. Nos résultats confirment que le choix de Vescovi est pertinent étant donné que le paramétrage par défaut se rapproche de celui de la dernière colonne, c'est-à-dire : $\lambda_1 = 0,1$ / $\lambda_2 = 0,2$ / $\lambda_3 = 0,7$. Les meilleurs résultats restent ceux obtenus via le paramétrage par défaut, suivis par ceux à l'extrémité droite du tableau. Ces paramètres ont donc un impact sur la prédiction, l'écart entre les lambdas les plus extrêmes étant de plus de 5 à 6 % pour le KSR, de 6 à 7 % pour le taux d'économie et de 1 à 2 % pour le taux d'utilisation de la prédiction.

4.2.3 Generalized default / user smoothed n-gram

Après avoir modifié les paramètres de lissage du modèle de langage général, nous avons repris celui utilisé par défaut et avons ajouté le modèle de langage de l'utilisateur avec les mêmes paramètres de lissage. Avec Google Books Ngram, nous avons testé trois petits corpus d'entraînement de genre identique, mais de taille différente, le Blog étant un texte de 14 071 mots et l'Interview de seulement 4 306 mots (cf. Table II.2).

CORPUS D'ENTRAÎNEMENT	GOOGLE BOOKS NGRAM		
CORPUS D'ENTRAÎNEMENT (UTILISATEUR)	PERSO	BLOG	INTERVIEW
KSR	59,037 %	58,769 %	58,559 %
Taux d'économie	76,486 %	76,157 %	75,978 %
Hit ratio	91,823 %	91,509 %	91,666 %

TABLE II.12 – Résultats : prédicteur par défaut et modèle de langage de l'utilisateur

Les bases de données Blog et Interview ont été créées uniquement pour ce test avec l'outil text2ngram. Nous supposons que l'utilisateur ait saisi manuellement ces textes lorsque le mode d'apprentissage automatique était activé. Les résultats sont sensiblement meilleurs (de 1 à 2 %) que ceux obtenus avec le corpus de test A et les mêmes paramètres, mais sans ce prédicteur et donc sans ce deuxième corpus d'entraînement (Table II.8). Le corpus Perso – qui est plus gros car constitué des deux autres corpus – a obtenu les meilleurs résultats même s'il n'y a pas de réelle différence entre les trois.

4.2.4 ARPA et lissage

Nous avons testé les fichiers .vocab et .arpa de nos trois corpus en appliquant différentes techniques de lissage : *Absolute-Discounting* (A-D), *Good-Turing* (G-T) et *Witten-Bell* (W-B). Pour le corpus Google Books Ngram, nous avons obtenu exactement les mêmes résultats avec les trois méthodes d'interpolation. Nous pensons que la raison est due à la possibilité que les fichiers .arpa de Google Books Ngram n'aient pas été correctement générés à partir des fichiers .csv, le CMU toolkit fonctionnant avec différents formats de fichiers convertis à partir d'un texte continu en .txt.

GOOGLE BOOKS NGRAM						
LISSAGE	A-D / G-T / W-B					
KSR	57,245 %					
T. d'économie	74,096 %					
Hit ratio	88,679 %					
À SE TORDRE				PERSO		
LISSAGE	A-D	G-T	W-B	A-D	G-T	W-B
KSR	39,199 %	39,288 %	39,438 %	38,571 %	38,512 %	38,721 %
T. d'économie	54,227 %	54,317 %	54,496 %	52,674 %	52,614 %	52,823 %
Hit ratio	79,088 %	79,088 %	79,245 %	74,213 %	74,213 %	74,213 %

TABLE II.13 – Résultats : prédicteur ARPA

Les résultats obtenus avec le prédicteur ARPA et Google Books Ngram sur le corpus de test A montrent toujours de très bons résultats par rapport aux deux autres corpus. Tous sont sensiblement moins bons que ceux avec la base de données, c'est-à-dire le prédicteur par défaut basé sur le modèle lissé avec interpolation linéaire (Table II.8).

4.3 Question 3 – Prédictions

Par défaut, le système propose 5 suggestions. Afin de déterminer l'impact du nombre de prédictions lexicales sur les résultats, nous l'avons fait varier de 1 à 9. Le corpus d'entraînement utilisé était Google Books Ngram sur le corpus de test A.

NBRE DE PRÉDICTIONS	1	3	(5)	7	9
KSR	45,354 %	51,239 %	57,932 %	60,113 %	61,458 %
Taux d'économie	60,501 %	67,523 %	75,052 %	77,532 %	78,996 %
Hit ratio	79,716 %	85,691 %	90,094 %	91,666 %	92,295 %

TABLE II.14 – Résultats : prédicteur par défaut et nombre de prédictions

Les résultats ci-dessus montrent que plus le nombre de prédictions est grand, plus les résultats sont effectivement élevés. Le nombre de prédictions ayant un impact sur les résultats, cela confirme notre hypothèse ainsi que les résultats provenant de la littérature scientifique comme nous le détaillerons dans la section suivante.

L'écart entre les nombres de prédictions les plus extrêmes (1 et 9) est supérieur à 16 % pour le KSR, 18 % pour le taux d'économie et 12 % pour le taux d'utilisation de la prédiction. Il est intéressant de noter que même avec une seule prédiction proposée, le corpus d'entraînement Google Books Ngram obtient toujours de meilleurs résultats que les corpus À se tordre et Perso sur tous les corpus de test (Tables II.8 et II.9).

Il ne faut néanmoins pas oublier que dans un système de CAA, il est nécessaire de tenir compte des aspects ergonomiques et cognitifs des utilisateurs. Nous rappelons que plus le nombre de prédictions est élevé, plus la charge cognitive est importante. C'est la raison pour laquelle nous avons décidé de nous limiter au test de 9 prédictions.

4.4 Question 4 – Interface utilisateur

Les résultats quantitatifs obtenus avec les deux simulateurs ont permis d'identifier une configuration de Presage qui regroupe les meilleurs paramètres et que nous appelons 'Mixte' (Figure II.7). Nous l'avons reproduite dans ACAT avec le fichier .xml disponible à l'Annexe B. Tous ces résultats ont été obtenus avec le corpus de test A, mais en raison de sa taille, nous avons demandé à notre utilisateur de test de recopier uniquement une vingtaine de phrases issues du premier corpus (cf. point 3.2.2).³⁵

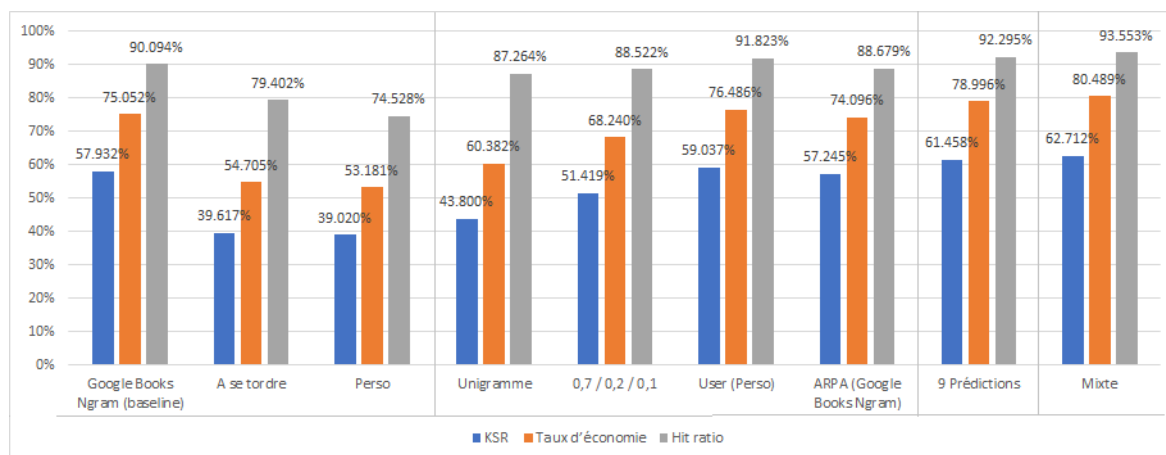


FIGURE II.7 – Résultats : corpus, prédictors, prédictions

Corpus Le gros corpus littéraire 'Google Books Ngram' était dans tous les cas plus performant que les autres corpus d'entraînement ('À se tortre' et 'Perso'). C'est la raison pour laquelle il avait déjà servi de *baseline* dans la plupart de nos tests (lissage, modèle utilisateur et nombre de prédictions), nous avons donc choisi de le réutiliser. Dans ACAT, nous avons néanmoins testé la petite version recommandée au point 3.1.3.

Prédictors En raison des résultats de certains tests (par exemple 'Unigramme' ou '0,7 / 0,2 / 0,1'), nous avons laissé le modèle trigramme lissé par défaut (0,01 / 0,1 / 0,89) et avons ajouté le modèle utilisateur, les deux corpus d'entraînement (Google Books Ngram et Perso) étant sous la forme de bases de données. Le prédictor utilisant les abréviations proposées via le fichier Abbreviations.xml (cf. point 3.1.4) a pu être testé.

Prédictions Par défaut, nous avons vu qu'ACAT proposait généralement neuf prédictions, c'est-à-dire le nombre qui a obtenu les meilleurs résultats. Nous avons laissé le nombre de prédictions à 9. Ce paramètre n'est accessible que via l'interface graphique.

À noter que la satisfaction subjective de l'utilisateur a été évaluée au moyen du System Usability Scale (SUS) disponible à l'Annexe D. Le score obtenu est de 70 %.³⁶

35. Ces 20 phrases proviennent de l'étudiante ayant fait son mémoire sur le système Minspeak®. Elles sont composées de 145 mots et 768 caractères ou 637 sans espace (cf. Table II.7).

36. À titre de comparaison, l'échelle du SUS est divisée de la façon suivante : (25) pire imaginable / (39) mauvais / (52) acceptable / (73) bon / (86) excellent / (100) meilleur imaginable.

Durant l'*user experience* et après le visionnage des deux enregistrements, nous avons noté plusieurs observations sur la prédiction et l'interface utilisateur :

- Des mots proposés n'existent pas ou dans d'autres langues (« the », « depth », etc.). Des prédictions désignant des entités nommées voire des sigles comme « cmjn » (« Cyan, Magenta, Jaune et Noir ») ont peu de chance d'être utilisées.
- Le prédicteur basé sur le modèle de l'utilisateur fonctionnait. Par exemple, après le verbe « appelles », le système a proposé deux noms propres (dont celui de l'utilisateur) se trouvant dans les textes composant le corpus Perso.
- Même si l'utilisateur connaissait les abréviations encodées dans le système, il n'en a employées que quatre sur les dix occasions. « cmt » a remplacé « comment » sans avoir été affiché dans la liste ; en entrant « sv », le système a proposé « svp » puis l'a remplacé par « s'il vous plaît » après le clic de sélection.
- L'utilisateur n'a pas toujours sélectionné la prédiction dès qu'elle apparaissait dans la liste, mais parfois quand plusieurs mots ayant la même racine que le mot recherché étaient proposés (« intéressant(e)(s) », « intéressé(e)(s) », etc.).
- La boîte qui contient les prédictions est trop petite, elle ne peut pas contenir les longs mots. Elle affiche par exemple « malheureu... » ou « gouvernem... ».
- L'option de majuscule automatique était activée et fonctionnait bien.
- L'option d'espace automatique après avoir sélectionné une prédiction était activée et fonctionnait également (sauf après la sélection d'une abréviation).
- L'utilisateur a parfois utilisé la touche pour effacer une lettre.
- L'utilisateur n'a laissé que deux erreurs de retranscription (au tout début) : il a oublié « d eau » (ph. 1) ; il a écrit « ne sont » à la place de « ne se sent » (ph. 3).
- En sélectionnant une lettre accentuée depuis le sous-clavier, l'utilisateur a appuyé plusieurs fois sans faire exprès sur la touche I (située alors derrière), c'est-à-dire sur le raccourci du sous-clavier de navigation (cf. Figure II.4).
- La fenêtre de parole n'est pas fixe et se déplace automatiquement à d'autres endroits de l'écran. Lorsqu'ACAT fonctionne, il est difficile de déplacer les fenêtres d'autres programmes car l'interface utilisateur propose alors un menu avec des raccourcis de navigation adapté à chacune des applications.
- Le temps d'attente après la sélection d'une lettre accentuée via le sous-clavier du pack français est plus long que celui des autres lettres.
- La touche entrée permettant de faire un retour à la ligne et de synthétiser la phrase ralentit aussi le système. Après avoir été à la ligne, nous avons parfois constaté un problème d'affichage des prédictions qui apparaissaient et disparaissaient au moment d'écrire la première lettre de la nouvelle phrase rendant la sélection des mots impossible. À noter que la synthèse vocale fonctionnait.
- ACAT propose une option intéressante ('SeedWordPredictionOnNewSentence') qui ignore le contexte après une nouvelle phrase et qui était activée.

5 Discussion

Cette section reviendra sur les différents résultats obtenus via les simulations théoriques et le test utilisateur. Nous les confronterons ensuite à ceux rapportés dans la littérature scientifique. Tout d'abord, il faut souligner le fait qu'il est très difficile de comparer les études entre elles, notamment en raison de méthodologies hétérogènes. Ainsi, citons les conditions de test qui ne sont pas toujours explicites et la distinction entre deux métriques souvent confondues : le KSR et le taux d'économie de saisie.

Ils varient selon que l'on évalue l'interface complète ou seulement la prédiction. Les résultats du KSR sont effectivement toujours moins bons que ceux obtenus avec le taux d'économie, l'écart variant de 15 à 20 %. Bien qu'il ne donne pas d'information sur le nombre d'appuis effectués ou économisés, le hit ratio ou taux d'utilisation de la prédiction est généralement resté élevé, c'est-à-dire dans les 70, 80 ou 90 % en fonction de certains paramètres comme les corpus, les prédicteurs ou le nombre de prédictions.

Nos premiers tests ont montré que le corpus Google Books Ngram permet d'économiser plus d'une saisie sur deux que ce soit via le KSR ou avec le taux d'économie. Ces performances sont également observées sur deux corpus de test de taille différente. Nous avons constaté un très grand écart – de plus ou moins 20 % – dans les résultats obtenus avec Google Books Ngram et les deux autres corpus d'entraînement. Les données d'apprentissage ont donc un impact important sur les résultats.

Le genre du corpus d'entraînement ne semble toutefois pas avoir eu une grande influence sur les résultats si nous comparons les corpus – de taille plus comparable – À se tordre (littéraire) et Perso (idiolecte) qui ont obtenu des pourcentages assez proches sur un corpus de test identique. Quant à la taille du corpus d'entraînement, nous avons vu qu'un corpus de plusieurs millions de n-grammes comme Google Books Ngram était plus performant qu'un corpus constitué d'un seul livre comme À se tordre.

Il est également préférable de tester un corpus d'entraînement littéraire sur un corpus personnel que l'inverse. D'après Gelšvartas *et al.* (2016), Presage est un prédicteur de texte à usage général, c'est-à-dire souvent entraîné à partir de corpus littéraires, dont l'un des inconvénients est la présence de discours indirect. Or, les utilisateurs de systèmes de CAA devraient avoir tendance à employer plus de discours direct pour communiquer oralement. Pour le corpus littéraire À se tordre, nous aurions pu envisager de faire un tri spécifique selon le type de discours rapporté (direct ou indirect).

Dans le chapitre I (point 2.1.1), nous avons vu que Trnka et McCoy (2007) ainsi que Boissière *et al.* (2012, 2015) s'opposaient sur la question du *core vocabulary* du corpus d'entraînement. Notre corpus Google Books Ngram est très générique, il contient énormément de termes qui ne seront probablement jamais utilisés. D'un point de vue théorique, il a effectivement obtenu les meilleurs résultats, mais nous continuons de nous demander si ce type de prédictions est utile et ne perturbe pas l'utilisateur.

La constitution du corpus Google Books Ngram nous amène à critiquer un certain manque de contextualisation étant donné l'impossibilité de connaître l'origine exacte des livres qui le constituent. Le fait que seulement 4 % de tous les livres soient repris n'est pas représentatif. Une mauvaise tokenisation comme la présence de l'occurrence 'internet' avant 1950 à la place de l'abréviation 'internat.' pour 'international' témoigne d'une approche purement lexicale qui ne tient pas compte de la polysémie.³⁷

Il est donc intéressant de se demander si Google Books Ngram est un corpus réellement accessible, interprétable et utilisable dans n'importe quel contexte. Son utilisation pose la question de la fiabilité des sources numériques. Il aurait été possible d'utiliser la version 2012 de Google Books Ngram qui, d'après le site où il est hébergé, aurait été mieux tokenisée. Nous aurions également pu tester une autre tokenisation de nos corpus d'entraînement et de test en laissant, par exemple, les apostrophes et tirets.

Pour constituer un corpus représentant l'idiolecte d'un utilisateur, des mails ou des tweets auraient pu être utilisés comme l'avaient fait Vescovi (2004) ou Stoop et van den Bosch (2014).³⁸ En raison des droits d'auteur, nous avons renoncé à tester une récente autobiographie à la première personne. Il aurait été intéressant de vérifier l'adaptation lexicale, mais aussi en termes de langue et de genre. Wandmacher (2008) avait par exemple comparé des corpus français, allemands et anglais de différents registres.

L'auteur affirmait qu'utiliser un corpus d'entraînement journalistique écrit par plusieurs auteurs et comprenant des sujets divers était plus avantageux que des corpus issus du Web en raison de leur qualité et de leur origine souvent moins bien contrôlées. Les résultats obtenus avec notre corpus Perso pourraient s'expliquer par le fait que nous n'avons pas corrigé les éventuelles erreurs dans le blog et l'interview. Nous aurions également pu comparer ce corpus avec un autre de type journalistique.

Nous avons constitué un quatrième corpus appelé Frenchngrams³⁹ que nous n'avons pas pu tester à cause de bugs dans le simulateur de Presage (et pas dans ACAT). Il était constitué de fichiers d'unigrammes, bigrammes et trigrammes contenant chacun les 10 000 n-grammes les plus fréquents d'un corpus regroupant 400 livres français les plus populaires du Projet Gutenberg (le 04/12/2016). Il aurait fallu recompiler l'ensemble du code source pour l'évaluer et le comparer à nos autres corpus.

D'après les informations sur GitHub, avaient été omis les ouvrages n'étant pas français à l'origine, les pièces de théâtre où chaque nom de personnage est retranscrit afin de savoir qui parle, les ouvrages n'existant pas en format .txt et les petits ouvrages (inférieurs à 400–500 Ko). Le corpus brut consistait en 2 millions de lignes, 21 858 058 de mots, 125 679 247 caractères (soit 5,75 caractères par mot), totalisant 129 Mo. Le corpus (.csv) était déjà en minuscule et dépourvu de caractères spéciaux (non français).

37. <http://dejavu.hypotheses.org/469> ; <http://socioargu.hypotheses.org/1963>

38. Leur corpus de tweets est disponible en ligne (<https://github.com/woseseltops/soothsayer>).

39. Frenchngrams est le nom du *repository* (<https://github.com/gturret/frenchngrams>) contenant les fichiers bruts. Nous avons aussi déposé notre base de données tokenisée sur GitHub (cf. Annexe B).

Nous avons ensuite testé plusieurs prédicteurs. Nous confirmons qu'un modèle de langage lissé trigramme obtient de meilleurs résultats qu'un modèle unigramme ou bigramme. Plus le nombre de n-grammes est bas, plus le gain obtenu avec le n-gramme de rang supérieur est grand.⁴⁰ Les paramètres de lissage d'un modèle avec interpolation linéaire ont un impact sur les performances du prédicteur en termes de KSR et de taux d'économie de saisie, le gain pouvant atteindre plus de 5 %.

Quant à la combinaison d'un modèle de langage par défaut avec un modèle utilisateur, nous avons vu que plus le deuxième corpus d'entraînement est gros, meilleurs seront les résultats. L'efficacité est toutefois limitée. Il faudrait que le système soit utilisé régulièrement. Nous avons aussi testé nos corpus d'entraînement en format ARPA avec trois techniques de lissage, mais nous n'avons pas constaté d'amélioration par rapport au prédicteur par défaut basé sur le modèle lissé avec interpolation linéaire.

Bien que nous n'ayons pas comparé la prédiction de mots issus de plusieurs langues, Vescovi (2004) a fait remarquer que l'efficacité des méthodes statistiques était susceptible de diminuer en fonction de la complexité morphologique et syntaxique. L'auteur explique qu'un même prédicteur peut être plus performant s'il est utilisé sur l'anglais qui est relativement plus simple que l'italien.⁴¹ Comme le français est une langue romane qui comporte beaucoup de suffixes, il serait plus difficile de trouver la bonne suggestion et cela pourrait également expliquer les performances obtenues.

Il convient de noter que les algorithmes statistiques proposés dans Presage sont relativement simples par rapport à certaines techniques décrites dans la littérature scientifique. Ils ne recourent par exemple pas au *topic modeling* étudié par Trnka *et al.* (2006), ni à l'Analyse Sémantique Latente (ASL) évaluée par Wandmacher et Antoine (2007a,b). Étant donné l'architecture de Presage, il aurait néanmoins été possible de développer de nouveaux prédicteurs sous la forme de modules extensibles.

Gelšvartas *et al.* (2016) ont évalué Presage avec et sans l'ajout d'un prédicteur sémantique destiné à prendre en compte les mots qui ne se trouvent pas dans le corpus. Leurs données d'entraînement étaient constituées de phrases anglaises récoltées dans des questionnaires médicaux.⁴² Ils ont obtenu de meilleures performances en spécialisant le modèle trigramme par défaut avec leur modèle construit à partir de leurs données et ils ont montré qu'il n'était pas nécessaire d'étendre le nombre de n-grammes.

Le corpus Google Books Ngram en format quadrigramme aurait été trop lourd pour ACAT, mais nous aurions pu faire varier la taille d'un même corpus d'entraînement en fonction du nombre de n-grammes. Selon Leshner *et al.* (1999), leur augmenta-

40. Avec le corpus Google Books Ngram, l'écart entre le modèle unigramme et bigramme est d'au moins 10 % alors que celui entre le modèle bigramme et trigramme est d'environ 5 %.

41. Il note par exemple que les adjectifs anglais sont plutôt invariables alors qu'il existe plusieurs terminaisons en italien, ce qui influence donc le nombre d'alternatives pour chaque mot.

42. Leur but était de concevoir un prédicteur spécialisé pour les patients ayant la maladie de Huntington, une affection neurodégénérative provoquant des troubles moteurs et cognitifs importants.

tion a un impact positif sur la précision du prédicteur. Ils ont suggéré de faire l'apprentissage sur plus de 3 millions de mots pour un modèle bigramme et trigramme, ayant montré qu'un modèle unigramme entraîné sur 500 000 mots anglais ne passait que de 45,7 à 47,2 % alors qu'un modèle d'ordre supérieur passe de 50 à 53,6 % voire 54,3 %.

D'autres auteurs avaient montré qu'un modèle trigramme était plus performant qu'un modèle de rang inférieur, nous pouvons citer pour le français Schadle (2003). Vescovi (2004) avait fait varier les paramètres de lissage, mais en raison des problèmes liés à son simulateur, nous ne proposons pas de comparer ses résultats aux nôtres. Comme l'avaient constaté Wandmacher et Antoine (2007a,b); Wandmacher (2008); Antoine (2011), les modèles utilisateurs n'apportent pas de véritable amélioration (cf. Table I.1). Et peu d'études détaillent le format de leurs corpus (.arpa, .csv, etc.).

En ce qui concerne les prédictions, nous supposons que plus leur nombre augmente, plus il y a de chances que le mot prédit soit le bon et donc celui recherché par l'utilisateur. Les résultats du hit ratio confirment cette hypothèse. Nous constatons une augmentation de plus de 10 % du taux d'utilisation selon que le système propose une ou (plus de) cinq prédictions. L'amélioration étant moins importante à partir de cinq, nous pensons que le choix doit se faire en fonction des préférences de l'utilisateur.

Copestake (1997) a par exemple testé jusqu'à vingt suggestions alors que Garay-Vitoria et Abascal (1997) ainsi qu'Hunnicut et Carlberger (2001) avaient constaté une amélioration des résultats, c'est-à-dire une diminution du pourcentage du nombre d'appuis, en augmentant le nombre de mots prédits de un à dix. Comme nous l'avons déjà vu, le temps et l'effort cognitif nécessaires pour sélectionner une proposition augmentent si le nombre de prédictions est élevé (Garay-Vitoria et Abascal, 2006).

Bérard et Niemeijer (2004) ont également observé qu'au plus le nombre de mots suggérés était grand, plus leur système de prédiction était efficace. Ils ont fait un test avec cinq à quinze suggestions et ils en recommandent finalement entre cinq et dix.⁴³ Niemeijer (2005) précise néanmoins qu'un système classique en propose cinq à sept alors que les préférences de ses utilisateurs de test varient entre huit et quatorze. Badr (2011, p. 61) affirme qu'il s'agit d'un facteur « qui peut rendre la liste plus utilisable ».

Pouplin (2016) insiste sur le fait que plus le nombre de prédictions est important, plus le KSR augmente et par conséquent, plus le temps de recherche d'un mot dans la liste augmente, ce qui diminuerait la vitesse de communication. Comme l'a souligné Ghedira (2015), il faut donc toujours trouver le juste équilibre entre la maximisation de la performance et la minimisation de la charge cognitive ressentie par l'utilisateur, c'est-à-dire faire le choix entre un système plus performant ou moins fatigant.

Le test utilisateur a donné des résultats différents de ceux obtenus via Presage.

43. Pour cinq, dix et quinze prédictions, les auteurs ont obtenu une augmentation du pourcentage de réduction de clics, celui-ci étant respectivement de 46, 51 et 53 % avec un gain dans la réduction des mouvements du curseur allant jusqu'à 30 %.

Même si nous ne l'avons pas calculé, le KSR théorique sera différent à cause des sous-claviers contenant les lettres accentuées. Les enregistrements montrent que l'utilisateur ne sélectionne pas toujours la prédiction dès qu'elle apparaît dans la liste. Il existe donc un écart entre cette métrique et l'usage qu'en fait en pratique l'utilisateur. Bien que les abréviations soient plus rapides, la personne testée a préféré ne pas les utiliser.

En ce qui concerne la forme de la liste de prédictions, Pouplin (2016, p. 58) avait déjà relevé le fait que Niemeijer (2005) « préconisait une double présentation sur clavier virtuel, à la fois horizontale avec un nombre limité de mots et une présentation verticale avec un nombre plus important de mots ». D'après Pouplin (2016), ces « observations qualitatives » n'auraient pas fait l'objet d'une évaluation spécifique. Il a également constaté qu'il n'existait pas de réelle étude concernant la position idéale de la liste de prédictions qui permettrait d'améliorer la vitesse de saisie.

Bien que Garay-Vitoria et Abascal (2006) aient observé dans la littérature que la présentation verticale était généralement mieux acceptée, nous avons constaté que beaucoup de systèmes libres ou payants ont tendance à présenter les prédictions horizontalement. Citons le clavier visuel Windows, Mind Express, les smartphones, etc. Par défaut, la présentation dans ACAT est à la fois horizontale et verticale. Nous n'avons pas vérifié le temps et l'efficacité du système de défilement bloc/ligne/colonne.

Nous avons déjà abordé la complexité de réaliser une étude en situation écologique lorsque nous avons vu certaines limites (cf. Chapitre I, Sous-sections 3.2 et 3.3). En raison de diverses contraintes, notre étude s'est limitée au test et à l'adaptation des systèmes pour un seul utilisateur. La grande variabilité au sein d'un même handicap (ici IMC) ne permet pas de tirer des conclusions facilement généralisables à l'ensemble des personnes qui seraient dans une situation de handicap plus ou moins similaire.

L'utilisateur testé a été plutôt satisfait de l'interface d'ACAT, le score du SUS étant presque bon. Selon ses réponses, il était également satisfait des prédictions, de leur affichage ainsi que celui du clavier virtuel. Il a trouvé qu'il s'agissait d'un moyen pratique et facile à utiliser. Néanmoins, il a aussi relevé des problèmes liés aux accents et à des blocages pendant son utilisation (c'est-à-dire le déplacement incontrôlé de la fenêtre de parole, l'apparition et la disparition des prédictions après un retour à la ligne, etc.).

Pour ACAT, nous recommandons un réagencement des claviers français. Étant donné qu'un 'c' est par exemple plus fréquent qu'un 'ç', le fait de le traiter comme une simple lettre et non de passer par un sous-clavier permettrait de gagner du temps en économisant un appui et d'éviter de cliquer sur une touche non désirée (caractère ou raccourci). Nous estimons donc qu'il faudrait intégrer les lettres accentuées aux autres. À noter que le sous-clavier de ponctuation n'est accessible que via le mode *App*.

L'ergonomie peut être améliorée. Même si nous pouvons supposer que les utilisateurs emploient le mode *Talk* uniquement pour la communication orale et la rapidité, cette présentation limite leur usage. Nous proposons d'inclure d'office ces signes

de ponctuation dans toutes les applications. Il serait intéressant de connaître les habitudes des utilisateurs de systèmes de CAA. Nous pouvons par exemple nous demander s'ils emploient la ponctuation pour écrire des messages destinés à être synthétisés.⁴⁴

Plusieurs bugs dans Presage ont également été recensés. Bien que fonctionnant dans ACAT, la base de donnée Frenchngrams ainsi que celle contenant la plus petite version de Google Books Ngram – conseillée et employée dans le test utilisateur – n'ont pas pu être évaluées avec le simulateur de Vescovi, ni dans qprompter à cause d'une erreur dans un fichier C++. Étant donné que nous avons décidé de ne pas recompilier tout le code source, nous n'avons pas pu expérimenter ces corpus et les comparer.

Après avoir testé le simulateur de Presage et ACAT, nous recommandons aux deux systèmes de fournir des *log files* par session qui donnent toutes les informations permettant de mieux les évaluer. Par exemple, dans la version K du logiciel Sibylle, un fichier .xml donne la date, l'heure du début et de la fin de la session ainsi que le mot et la partie prédite de celui-ci, ce qui suffit pour connaître le taux d'économie de saisie et le taux d'utilisation de la prédiction. Avec la phrase « on a sonné à la porte » :

```
<Session>
  <DatesTimes start_date="18/07/2018 13:19:53" end_date="18/07/2018 13:21:19"
    loading_time_ms="5537" />
  <Environement jre_version="1.8.0_171-b11" os="Windows 10" />
  <Usage clic_count="0" word_prediction_use_count="6" switch_use_mode_count="1"
    />
  <Plugins loaded_plugins_count="0" paying_plugins_count="0" />
  <WordPredictions>
    <WordPrediction base_word="sonné" predicted_part="né " />
    <WordPrediction base_word="porte" predicted_part="porte " />
    <WordPrediction base_word="on" predicted_part="n " />
    <WordPrediction base_word="à" predicted_part="à " />
    <WordPrediction base_word="a" predicted_part="a " />
    <WordPrediction base_word="la" predicted_part="la " />
  </WordPredictions>
</Session>
```

Enfin, nous voulons insister sur l'importance et les enjeux d'une évaluation « cadrée ». Comme le faisaient déjà remarquer Boissière et Dours (2001), il faudrait constituer un benchmark, mais aussi « définir des critères d'évaluation mesurables, communs aux différents systèmes, afin d'effectuer des comparaisons objectives ». C'est pourquoi nous avons par exemple utilisé deux formules pour les résultats. Parmi les difficultés rencontrées, nous pouvons citer la grande diversité des études et projets.

44. Nous avons remarqué qu'écrire « le téléphone sonne » avec ou sans '?' ne changeait pas l'intonation de la synthèse vocale dans ACAT. Or, l'intonation a un impact sur la compréhension des actes illocutoires. En fonction de celle-ci, il peut s'agir d'une affirmation ou d'une question. L'utilisateur pourrait trouver plus rapide d'écrire « le téléphone sonne? » plutôt que « est-ce que le téléphone sonne? ».

Conclusion

The development of full artificial intelligence could spell the end of the human race.

Stephen Hawking⁴⁵

L'objectif de ce mémoire était d'étudier le fonctionnement des systèmes de prédiction de mots et l'interface utilisateur qui leur est associée. Le chapitre I a permis de passer rapidement en revue les différentes communications alternatives existantes. Nous avons ensuite présenté plusieurs systèmes de prédiction de mots en fonction de leur approche (statistique, symbolique ou hybride), mais aussi expliqué leurs limites. Le chapitre II était centré sur deux systèmes : Presage et ACAT. Après avoir créé et traité diverses ressources en français, nous avons développé un simulateur afin d'évaluer les paramètres du prédicteur et déterminer la meilleure configuration théorique. L'interface complète a ainsi fait l'objet d'un test utilisateur avec une personne handicapée.

Les prototypes de recherche et les logiciels d'aide à la communication à destination des personnes en situation de handicap sont nombreux et utilisent différentes techniques de modélisation du langage. Des options de paramétrage se sont développées et sont présentes dans beaucoup de ces systèmes. Nous pouvons citer la gestion de leur dictionnaire ainsi que l'apprentissage automatique de nouveaux mots, l'activation ou désactivation de certains prédicteurs, la présentation et le nombre de prédictions, etc. Des progrès en TAL et IHM sont encore réalisés. Néanmoins, nous pouvons nous demander jusqu'où il faut aller. Faut-il, par exemple, viser un KSR de 80, 85 % au détriment d'autres facteurs comme l'augmentation de la charge cognitive et visuelle ?

En étudiant un système de prédiction et son interface, nous avons vu que ces outils comportent des limites linguistiques, cognitives et ergonomiques. D'un point de vue technique, les prédicteurs statistiques comme ceux implémentés par défaut dans Presage ne tiennent pas compte des aspects morpho-lexicaux, syntaxiques et sémantiques du langage puisqu'ils reposent en grande partie sur la représentativité des corpus avec lesquels ces modèles ont été construits. L'origine et la constitution de ces données d'entraînement doivent alors être contrôlées. L'étape de prétraitement et le tri du dictionnaire pose la question de la définition de son contenu. Nous avons par exemple montré qu'un gros corpus générique était plus efficace qu'un plus petit.

45. <http://www.bbc.com/news/technology-30290540> (article de la BBC du 2 décembre 2014).

Nos résultats ont confirmé ceux trouvés dans la littérature, à savoir qu'un modèle de langage lissé trigramme est plus efficace qu'un modèle n-gramme d'ordre inférieur et que plus le nombre de prédictions est grand, meilleurs seront le KSR, le taux d'économie de saisie et le hit ratio. Il est aussi possible d'obtenir des améliorations avec un modèle de langage utilisateur, mais cela prend du temps car le système de prédiction doit s'adapter et donc être fréquemment utilisé. Bien que le corpus Google Books Ngram évalué permette d'économiser plus d'une saisie sur deux, notre test utilisateur a montré que certaines prédictions avaient peu de chance d'être utilisées et qu'elles n'étaient pas systématiquement choisies dès qu'elles apparaissaient dans la liste.

Les logopèdes sont donc souvent confrontés à la question de l'adaptation du lexique. Il faut aussi rappeler que l'utilisation des prédictions dépend de la charge cognitive et visuelle du patient. En plus de l'effort physique, l'utilisateur doit se concentrer sur l'écriture de son texte, éviter de faire des erreurs, lire les prédictions et faire appel à ses capacités de mémorisation lorsqu'il pense pouvoir utiliser une abréviation. Ainsi, malgré l'implémentation d'un prédicteur permettant d'économiser davantage de saisies (et donc de pallier la lenteur de la communication), la personne testée a préféré ne pas les utiliser. D'un point de vue ergonomique, l'interface peut également avoir un impact sur les résultats de certaines métriques utilisées dans le cadre de la prédiction.

En analysant ACAT, nous avons constaté que le simple ajout d'un sous-clavier contenant les lettres accentuées est susceptible d'influencer le KSR et les mesures liées aux erreurs ainsi qu'au temps. Bien que nous ne l'ayons pas calculé, il existe un écart entre les évaluations quantitatives réalisées à l'aide d'un simulateur théorique calculant la meilleure utilisation possible du système et les évaluations qualitatives effectuées avec de réels participants. Ces méthodes sont délicates et coûteuses car elles nécessitent des moyens, des contacts, etc. Notre étude s'est donc limitée au test d'un utilisateur. Néanmoins, elle a permis d'identifier certains problèmes ergonomiques, de compléter l'évaluation du système de prédiction et de proposer des recommandations.

Presage et ACAT présentent des possibilités intéressantes. Les tests ont montré des résultats qui nous semblent acceptables pour des logiciels libres malgré les quelques bugs recensés. Le dépôt en ligne sur des forges logicielles comme SourceForge ou GitHub est pratique pour les développeurs, mais elles offrent une visibilité limitée auprès des utilisateurs qui en ont réellement besoin. C'est pourquoi nous recommandons, par exemple, que des versions stables soient publiées sur un site plus attirant afin que les personnes handicapées puissent les tester et donner leur avis pour continuer à les améliorer.⁴⁶ Il est très important de faire intervenir les différentes personnes qui sont concernées par ces technologies et de promouvoir leur accessibilité.

46. Par exemple, le MOOC sur l'accessibilité numérique de l'équipe d'Inria Learning Lab encourageait fortement les participants à donner leur avis sur leur nouveau lecteur vidéo. Ce MOOC et ce lecteur servant d'outil pédagogique étaient accessibles à tout le monde, y compris aux personnes en situation de handicap (auditif, visuel, etc.). Il s'agit d'un projet combinant à la fois la recherche et la vulgarisation.

Enfin, nous terminerons ce mémoire en abordant le TAL, les systèmes d'aide à la communication pour les personnes handicapées ainsi que la prédiction d'un point de vue éthique afin de mettre en avant certains risques et leurs implications.⁴⁷

Il faut savoir que « les réflexions éthiques en TAL » concernaient majoritairement « l'anonymisation des données personnelles » (Lefeuve-Halftermeyer *et al.*, 2016, p. 48). Après avoir étudié en profondeur les divers apports des systèmes de prédiction et de CAA, nous pouvons nous demander ce qu'il en est des risques liés à ces technologies langagières d'aide à la personne. Des ateliers comme ETeRNAL⁴⁸ ainsi que des études telles que celles d'Antoine *et al.* (2014a,b) ou de Lefeuve-Halftermeyer *et al.* (2016) commencent à apparaître afin d'aborder les risques technologiques. L'une des premières étapes de leur démarche s'inscrit dans la recherche des facteurs de risque.

Ces études s'interrogent par exemple sur la modification des compétences langagières due à une certaine dépendance. Nous avons vu que l'usage du système de prédiction Sibylle pouvait conduire à la production de plus de textes et une diminution des fautes d'orthographe chez des enfants atteints d'une Infirmité Motrice Cérébrale (IMC). Antoine *et al.* (2014b) se sont par exemple demandé « si l'aide favorise une maîtrise plus rapide du système de la langue, ou si l'amélioration de la qualité des productions ne masque pas un abandon de cette capacité au profit du système... ». Autrement dit, un risque attendant à l'usage de ces technologies comme outil d'assistance ou de compensation serait la perte à terme d'une compétence suppléée par la technologie.

Les utilisateurs pourraient avoir tendance à se fier davantage aux propositions de la machine plutôt qu'à leurs propres analyses. Ainsi, bien que ces systèmes soient conçus pour réduire leur handicap en leur permettant d'accélérer leur communication, ils peuvent aussi l'augmenter en les aidant à tout moment dans leurs choix orthographiques comme le font les correcteurs et les traducteurs automatiques qui tenteraient de se substituer aux réelles connaissances des utilisateurs. C'est la raison pour laquelle les études citées soulignent également la question de la capacité d'innovation et de création linguistique ainsi que celle de la nécessité d'une évaluation de la criticité.

Concernant la capacité d'innovation et la création linguistique, elles peuvent être limitées avec un système de prédiction basé soit sur un corpus de mots prédéfini et leurs statistiques, soit sur l'usage d'une norme et de règles symboliques (Lefeuve-Halftermeyer *et al.*, 2016, p. 54). Lorsque nous avons créé et traité les ressources de Presage, nous avons déterminé l'ensemble des mots qui seront proposés à l'utilisateur. Nous avons envisagé de tester un corpus constitué de documents écrits en Facile A Lire et à Comprendre (FALC). Or, en supposant que les corpus aient un impact sur les compétences des utilisateurs, nous nous sommes demandé s'il fallait aller jusqu'à simplifier le corpus d'un système de prédiction pour obtenir de bonnes performances.

47. Cette partie n'aurait pas pu être écrite sans les échanges avec J.-Y. Antoine que nous remercions.

48. Éthique et TRaitemeNt Automatique des Langues (<http://eternal.loria.fr/>).

L'évaluation est une question délicate car les résultats varient d'une personne à l'autre et que tous n'ont pas les mêmes connaissances. D'une part, la simplification de textes, même si elle peut être efficace, limite les capacités d'expression de l'utilisateur. Les personnes atteintes de certaines pathologies neurodégénératives ayant eu l'habitude de pratiquer le langage ne retrouveront pas toute sa richesse. D'autre part, nous ne pouvons pas forcer les personnes quelles qu'elles soient à s'exprimer selon une 'norme' bien établie sans entraîner le risque que mentionnent Antoine *et al.* (2014a), à savoir « un usage restreint de langue avec pour conséquence éventuelle la création d'un genre du discours du handicapé reconnaissable et dû au système d'assistance ».

La simplification de textes ainsi que le choix du corpus et donc des mots qu'il contient peuvent donc être considérés dans un sens plus large et constituer des facteurs de risque. La personne est déjà handicapée et on lui limite ses possibilités d'interaction même avec un but justifiable : saisir plus vite et ne pas le troubler avec des prédictions qu'il ne comprendrait pas. Les chercheurs et concepteurs sont donc confrontés à un dilemme. Si la prédiction dépasse les compétences langagières de l'utilisateur, elle est susceptible de le perturber avec des mots qu'il ne connaît pas, ce qui pourrait empêcher l'apprentissage chez des enfants. Mais si elle est trop simple, elle réduit ses possibilités d'expression en deçà de ses capacités et augmente donc son handicap.

C'est pourquoi le rôle des logopèdes et des ergothérapeutes est donc d'ajuster au mieux le lexique du système aux capacités du patient. La vulnérabilité des personnes handicapées peut avoir un impact sur l'estimation de la criticité, c'est-à-dire sur la réalité du risque combinée avec sa sévérité. Selon Antoine *et al.* (2014a,b), des prédictions inappropriées pourraient avoir « une influence négative sur l'état psychologique du patient (énervement, fatigue cognitive) ». En testant Google Books Ngram avec un réel utilisateur, nous avons noté plusieurs propositions lexicales « mal contextualisées ». Pour un usage en situation écologique, nous pensons que nos traitements ne sont pas suffisants étant donné le nombre de mots qui ne seront probablement jamais utilisés.

Il existerait donc plusieurs risques relatifs à la prédiction. Nous avons vu le risque cognitif ou de régression se traduisant par une perte d'autonomie, le risque psychologique lié au ressenti par rapport à l'usage de la prédiction, mais aussi le risque sociétal où le rôle du chercheur serait de réfléchir, d'évaluer et puis de décider. Les concepteurs de systèmes de prédiction ont tendance à penser en termes de gains immédiats, observables et mesurables objectivement (comme le taux d'économie de saisie ou la vitesse de saisie) en ne parlant pas des risques sur le long terme et des éventuels « effets plus globaux sur la rééducation et l'évolution de la pathologie » (Antoine *et al.*, 2014b).

Annexe A

Accessibilité du Web

L'annexe reprend chaque directive AnySurfer suivie des recommandations du WCAG 2.0 avec le niveau d'importance (A indispensable, AA important et AAA optionnel) concernant les personnes qui ont un handicap moteur et l'accessibilité du Web.

Pour accéder aux informations de chacune d'elles, il faut entrer soit le lien vers le site belge AnySurfer (<http://www.anysurfer.be/fr/en-pratique/directives/directive/>) suivi du numéro et de l'intitulé de la directive, soit le lien vers le site du WCAG 2.0 (<https://www.w3.org/TR/WCAG20/>) suivi de l'ancre de la page (#).

Exemple – AnySurfer <http://www.anysurfer.be/fr/en-pratique/directives/directive/1-1-la-totalite-du-site-est-utilisable-au-moyen-du-clavier>

Exemple – WCAG 2.0 <https://www.w3.org/TR/WCAG20/#keyboard-operation>

1. NAVIGATION		
1-1-	la-totalite-du-site-est-utilisable-au-moyen-du-clavier	
2.1.1	clavier (#keyboard-operation)	Niveau A
2.1.2	pas de piège au clavier (#keyboard-operation)	Niveau A
2.4.3	parcours du focus (#navigation-mechanisms)	Niveau A
2.4.7	visibilité du focus (#navigation-mechanisms)	Niveau AA
4.1.2	nom, rôle et valeur (#ensure-compat)	Niveau A
3. MISE EN FORME		
3.1 TEXTE		
3-1-2-	les-espaces-et-autres-caracteres-ne-sont-pas-utilises-pour-produire-des-effets-visuels	
1.1.1	contenu non textuel (#text-equiv)	Niveau A
1.3.2	ordre séquentiel logique (#content-structure-separation)	Niveau A
3.2 MISE EN PAGE		
3-2-1-	le-contenu-de-la-page-suit-un-ordre-logique	
1.3.1	information et relations (#content-structure-separation)	Niveau A
1.3.2	ordre séquentiel logique (#content-structure-separation)	Niveau A

4. INTERACTIVITÉ		
4.1 FORMULAIRES		
4-1-1-	les-labels-et-elements-des-formulaires-sont-intimement-lies	
3.3.2	étiquettes ou instructions (#minimize-error)	Niveau A
4.1.2	nom, rôle et valeur (#ensure-compat)	Niveau A
4-1-3-	letiquette-des-champs-de-saisie-de-date-indique-le-format-requis	
3.3.2	étiquettes ou instructions (#minimize-error)	Niveau A
4-1-5-	les-regroupements-delements-dans-une-liste-deroulante-se-font-avec-optgroup	
1.3.1	information et relations (#content-structure-separation)	Niveau A
4-1-7-	chaque-formulaire-a-un-bouton-denvoi-visible	
3.3.2	étiquettes ou instructions (#minimize-error)	Niveau A
4.2 CONTRAINTES DU TEMPS		
4-2-1-	les-actions-ne-sont-pas-strictement-liees-a-des-contraintes-de-temps	
2.2	délai suffisant (#time-limits)	Niveau A(AA)
2.2.1	réglage du délai (#time-limits)	Niveau A
2.2.2	mettre en pause, arrêter, masquer (#time-limits)	Niveau A
2.2.3	pas de délai d'exécution (#time-limits)	Niveau AAA
2.2.4	interruptions (#time-limits)	Niveau AAA
2.2.5	nouvelle authentification (#time-limits)	Niveau AAA
5. PLUG-INS ET DOCUMENTS TÉLÉCHARGEABLES		
5-1-	flash	
5-2-	microsoft-silverlight	

TABLE A.1 – AnySurfer et WCAG 2.0

Ce tableau a pour but de montrer des recommandations issues de standards et de normes existantes tout en insistant sur les principales difficultés que rencontrent plus spécifiquement les personnes en situation de handicap moteur.¹ Elles concernent surtout les aspects relatifs à la navigation, la mise en forme et l'interactivité des formulaires ainsi que les contraintes temporelles. Afin de prendre en compte de nouvelles interfaces comme les spécificités du mobile, il est intéressant de noter que le W3C a publié une mise à jour de ces recommandations, appelées WCAG 2.1 (<https://www.w3.org/TR/WCAG21/>), le 5 juin 2018.

1. Nous aurions pu également prendre comme exemple les recommandations du UAAG 1.0 (<http://www.w3.org/TR/UAAG10/>).

Annexe B

Code source

Conformément à la licence du libre accès en vigueur pour tous les logiciels et ressources utilisées, nous avons mis tous nos fichiers (scripts, corpus, logs, vidéo, etc.) dans un *repository* sur GitHub (https://github.com/galami/WordPrediction_AAC).

Perl

Le script Perl ci-dessous a été écrit afin d'effectuer un premier traitement du corpus Google Books Ngram. Les regex ne tiennent pas compte des lignes avec des *ngrams* ne contenant aucune lettre de l'alphabet et ceux contenant des caractères spéciaux (non français), des chiffres, des répétitions de minimum trois lettres identiques, des mots d'une lettre (sauf /a/ et /à/) ainsi que tous les *ngrams* datant d'avant l'an 2000. Les diacritiques françaises /ââçèèëëïîôùûüÿ/ et les ligatures /æœ/ ont été conservées.

```
#!/usr/bin/perl

use utf8;
use strict;
use warnings;
use Term::ProgressBar; # Facultatif : affiche une barre d'état (*)

open( my $file , '<:utf8' , $ARGV[0] ) or die( "Impossible d'ouvrir le fichier
    $ARGV[0]" );

my $num_lines = ( stat $file )[7]; # (*)
my ( $compteur_modif, $progress_byte, $next_update ) = ( 0, 0, 0 );
my $progress = Term::ProgressBar->new(
    {   name => "[Lecture] ",
        count => $num_lines,
        ETA   => 'linear' ,
    }
);
```

```

my %count; # Table de hachage

readfile:while ( my $line = <$file> ) {

    $progress_byte += length($line); # (*)
    $next_update = $progress->update($progress_byte) if ( $progress_byte >
    $next_update && $progress_byte < $num_lines );

    $line = lc $line; # Minuscule

    my @col = split( /\t/, $line ); # TSV

    if ($col[0] !~ /[aa-z]/ or $col[0] =~ /^[a-z\s^ââçéèëëïïôùüÿæ][\^]|(.)
    (\1{2,})|\\b([b-zâçéèëëïïôùüÿæ])\\b(\1{0,})/g or $col[1] =~ /\b1/) {
        next readfile;
    }

    $count{$col[0]} += $col[2]; # Somme pour fusionner les doublons
}

close( $file );

$\ = "\n";
open( my $file2, '>:utf8', $ARGV[1] ) or die( "Impossible d'ouvrir le fichier
    $ARGV[1]" );
print { $file2 } (join ' ', ( $_, $count{$_} )) for sort keys %count; # CSV (+ tri)
close( $file2 );

$progress->update($num_lines) if $num_lines >= $next_update; print "\n"; # (*)

```

Dans un fichier 2009, le scalaire \$col[0] désigne la colonne *ngram*, \$col[1] *year*, \$col[2] *match_count*, \$col[3] *page_count* et \$col[4] *volume_count*. La somme des valeurs associées aux clés de la table de hachage est calculée et la table est ensuite triée suivant l'ordre lexicographique croissant. Lors de la deuxième tokenisation avec ce script, nous l'avons simplement modifié en utilisant les lignes ci-dessous.

```

my @col = split( /\t/, $line ); # CSV
$count{$col[0]} += $col[1]; # Somme pour fusionner les doublons entre fichiers

while (my ($key,$count) = each %count) {
    print { $file2 } "$key,$count"; # CSV (pas de tri)
}

```

Le tri de la nouvelle table de hachage ainsi que l'ajout des virgules (représentant les colonnes) pour les fichiers *_2_gram* et *_3_gram* ont été réalisés avec Notepad++.

JSON

L'historique des différentes étapes d'OpenRefine en format JSON :

1. Reorder rows permanently;
2. Blank down cells in column Column 1;
3. Join multi-valued cells in column Column 2;
4. Text transform on cells in column Column 2 using expression grel :
sum(foreach(value.split(','),x, x.toNumber()));
5. Split column Column 1 by separator : remplace les espaces par deux colonnes (_2_gram) qui seront word_1, word ainsi que les espaces par trois colonnes (_3_gram) qui seront word_2, word_1, word dans la base de données.

À noter que nous l'avons uniquement utilisé pour la deuxième tokenisation du fichier _1_gram de Google Books Ngram (sans l'étape 5), OpenRefine étant moins rapide que Perl pour traiter les autres fichiers (_2_gram et _3_gram).

```
[
  {
    "op": "core/row-reorder",
    "description": "Reorder rows",
    "mode": "record-based",
    "sorting": {
      "criteria": [
        {
          "errorPosition": 1,
          "caseSensitive": false,
          "valueType": "string",
          "column": "Column 1",
          "blankPosition": 2,
          "reverse": false
        }
      ]
    }
  },
  {
    "op": "core/blank-down",
    "description": "Blank down cells in column Column 1",
    "engineConfig": {
      "mode": "row-based",
      "facets": []
    }
  },
  {
    "columnName": "Column 1"
  }
]
```

```

    "op": "core/multivalued-cell-join",
    "description": "Join multi-valued cells in column Column 2",
    "columnName": "Column 2",
    "keyColumnName": "Column 1",
    "separator": ", ",
  },
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Column 2 using expression",
    "expression": "grel:sum(forEach(value.split(','),x, x.toNumber()))",
    "engineConfig": {
      "mode": "row-based",
      "facets": []
    },
    "columnName": "Column 2",
    "expression": "grel:sum(forEach(value.split(','),x, x.toNumber()))",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  }
  {
    "op": "core/column-split",
    "description": "Split column Column 1 by separator",
    "engineConfig": {
      "mode": "row-based",
      "facets": []
    },
    "columnName": "Column 1",
    "guessCellType": true,
    "removeOriginalColumn": true,
    "mode": "separator",
    "separator": "\\s",
    "regex": true,
    "maxColumns": 2
  }
]

```

Python

Le premier script a été écrit pour faire du Web scraping et extraire les articles du Blog qui constitue une partie du corpus Perso. La liste des articles utilisés est disponible à l'Annexe C. Nous remercions S. Hengchen pour ses conseils en Web scraping.

```
# -*- coding: utf-8 -*-

import os, sys
from bs4 import BeautifulSoup # Web scraping
from nltk.tokenize import RegexpTokenizer # Tokenisation

tokenizer = RegexpTokenizer(r'\w+')

link = str(input("Entrez le lien : "))

import urllib.request
with urllib.request.urlopen(link) as url:
    html = url.read().decode("utf8")

def tag_visible(element):
    if element.parent.name in ["p"]: # Si dans un paragraphe
        return True
    return False

def text_from_html(body):
    soup = BeautifulSoup(body, 'html.parser')
    soup = soup.find("div", {"class": "entry-content clearfix"})
    texts = soup.findAll(text=True)
    visible_texts = filter(tag_visible, texts)
    return " ".join(t.strip().lower() for t in visible_texts) # Minuscule

file = open("corpus.txt", "a", encoding="utf-8")

htmllist = tokenizer.tokenize(text_from_html(html))
article = " ".join(htmllist)

file.write(article + " ")

print(len(article.split())) # Facultatif : nbre de mots
print(len(article)) # Facultatif : nbre de lettres

file.close()
```

Le deuxième script sert à traiter et nettoyer les fichiers logs de `presage_simulator`. En plus de résoudre les problèmes liés à l'encodage des caractères, ce nouveau simulateur permet de calculer le KSR de Vescovi (2004), le taux d'économie de saisie et le hit ratio en pourcentage. À noter que nous avons compté l'espace automatique dans le calcul du k_i et du `taux_econo_total` pour chaque mot (sauf le dernier). Nous remercions J. Jabon de nous avoir aidé à concevoir la première partie de ce simulateur.

```
#!/usr/bin/python3
# -*-coding:Latin-1 -*

import sys

fname = str(sys.argv[1])
print("File to treat :", fname, "\n")

with open(fname, 'rb') as f:
    content = f.readlines()

datas_tmp, datas = [], []
line_nr, ki, ks, kn, ksr, kp, taux_econo_total, wd_nr = 0, 0, 0, 0, 0, 0, 0, 0

for l2 in content :
    line_nr+=1
    try :
        l = l2.decode(encoding='UTF-8', errors='strict')
        if(l.startswith("====>")) :
            nm = l[len("====> "):].strip()
            d = {};
            d["line_nr"] = line_nr
            d["name"] = nm
            d["wd"] = []
            datas_tmp.append(d);
        elif(l.startswith("=====")):
            break
        elif(l.startswith("    >")) :
            d[ ">" ] = l[len("    > "):].strip()
        else :
            d["wd"].append(l.strip())
    except UnicodeDecodeError:
        continue

for i in range(1, len(datas_tmp)) :
    if datas_tmp[i]["name"] != datas_tmp[i-1]["name"] :
        datas.append(datas_tmp[i-1])
datas.append(datas_tmp[len(datas_tmp)-1])
```

```

for i in datas :
    kp = (len(i["name"])-len(i[>]))
    ki+=len(i[>])
    taux_econo_total+=kp
    kn+=len(i["name"])+1 # Une espace après chaque mot
    if kp != 0 : # Si le nbre de lettres économisées est différent de 0
        ks+=1 # Le mot a été prédit et il faut un clic pour saisir la prédiction
        if len(i["name"]) == len(i[>]) and i != datas[-1] :
            # Si toutes les lettres à prédire ont été saisies
            ki+=1 # On compte l'espace automatique (sauf dernier mot)
        if kp != 0 and i != datas[-1] : # Si la prédiction a été utilisée
            taux_econo_total+=1 # On compte l'espace automatique (sauf dernier mot)
        wd_nr+=1 # Nbre de mots
        # ===== Affichage des résultats (par mot) =====
        print(i["name"]+' > '+i[>]) # Mot à prédire > lettre(s) saisie(s)
        if i != datas[-1] : # Tous les mots, sauf le dernier
            print("nbre de lettre(s) a predire :", len(i["name"])+1)
            if kp == 0 : # Le mot n'a pas été prédit
                print("nbre de lettre(s) saisie(s) :", len(i["name"])+1, "/ nbre de
                lettre(s) economisees :", kp, "\n")
            else : # Le mot a été prédit
                print("nbre de lettre(s) saisie(s) :", len(i[>]), "/ nbre de
                lettre(s) economisees :", kp+1, "\n")
            else : # Le dernier mot (sans espace)
                print("nbre de lettre(s) a predire :", len(i["name"]))
                print("nbre de lettre(s) saisie(s) :", len(i[>]), "/ nbre de lettre(s)
                economisees :", kp, "\n")

kn-=1 # Enlève l'espace en trop du dernier mot
ksr = (1-(ki+ks)/kn)*100 # Formule du KSR de Vescovi
hit_ratio = (ks/wd_nr)*100 # # Taux d'utilisation de la prédiction

print("===== \n")
print("ki :", ki) # Actual keystrokes
print("ks :", ks) # Keystrokes required to select suggestion
print("ksr :", ksr, "% \n")
print("nbre total de lettres economisees :", taux_econo_total)
# taux_econo_total = (1-(ki/kn))*100
print("taux d'economie de saisie :", (taux_econo_total/kn)*100, "% \n")
print("kn :", kn) # Keystrokes required with no prediction enabled
print("nbre total de mots :", wd_nr)
print("hit ratio :", hit_ratio, "%")

```

SQL

Cette annexe reprend les différentes commandes SQL utilisées ou conseillées : la création des tables par n-gramme pour le corpus Google Books Ngram, la création des index ainsi que la suppression des mots dont la fréquence est inférieure au seuil déterminé dans la table des unigrammes et qui sont présents dans les tables des bigrammes ainsi que des trigrammes (uniquement pour Google Books Ngram).

```
-- Création des tables
CREATE TABLE _1_gram (word TEXT, count INTEGER, UNIQUE(word))
CREATE TABLE _2_gram (word_1 TEXT, word TEXT, count INTEGER, UNIQUE(word_1, word
))
CREATE TABLE _3_gram (word_2 TEXT, word_1 TEXT, word TEXT, count INTEGER, UNIQUE
(word_2, word_1, word))

-- Création des index
CREATE INDEX idx__1_gram ON _1_gram (word)
CREATE INDEX idx__2_gram ON _2_gram (word_1, word)
CREATE INDEX idx__3_gram ON _3_gram (word_2, word_1, word)

-- Google Books Ngram : statistiques (avant suppression)
SELECT AVG(count) FROM -- _1_gram : 3 846,589 _2_gram : 278,848 _3_gram : 67,832
SELECT SUM(count) FROM -- _1_gram; _2_gram; _3_gram;
-- 4 881 398 607; 3 599 520 908; 2 211 946 087;
SELECT word, max(count) FROM -- _1_gram; _2_gram; _3_gram;
-- 272 494 238 (de); 48 281 184 (de la); 1 928 766 (et de la);

-- Méthode empirique pour déterminer les seuils de suppression
SELECT * FROM -- _1_gram _2_gram _3_gram
WHERE count < -- 200 50 10
ORDER BY count DESC

-- Suppression des unigrammes < 200 (les mêmes dans les bigrammes et trigrammes)
CREATE TABLE under200 AS SELECT * from _1_gram WHERE count < 200
DELETE FROM _1_gram WHERE word IN (SELECT word FROM under200); -- 1 045 917 rows
DELETE FROM _2_gram WHERE word IN (SELECT word FROM under200); -- 491 286 rows
DELETE FROM _2_gram WHERE word_1 IN (SELECT word FROM under200); -- 305 284 rows
DELETE FROM _3_gram WHERE word IN (SELECT word FROM under200); -- 108 316 rows
DELETE FROM _3_gram WHERE word_1 IN (SELECT word FROM under200); -- 70 067 rows
DELETE FROM _3_gram WHERE word_2 IN (SELECT word FROM under200); -- 41 267 rows
CREATE TABLE under50 AS SELECT * from _2_gram WHERE count < 50
DELETE FROM _2_gram WHERE count < 50 -- 8 926 604 rows
CREATE TABLE under10 AS SELECT * from _3_gram WHERE count < 10
DELETE FROM _3_gram WHERE count < 10 -- 13 295 814 rows
DROP TABLE under200; DROP TABLE under50; DROP TABLE under10;
```

XML

Le fichier de configuration de Presage (presage.xml) de Vescovi (2004). Les paramètres sont ceux utilisés lors du premier test avec le corpus Google Books Ngram.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>

<Presage>
  <PredictorRegistry>
    <LOGGER>ERROR</LOGGER>
    <!-- PREDICTORS
      Space separated list of predictors to use to generate predictions
    -->
    <PREDICTORS>DefaultSmoothedNgramPredictor
DefaultAbbreviationExpansionPredictor</PREDICTORS>
  </PredictorRegistry>
  <ContextTracker>
    <LOGGER>ERROR</LOGGER>
    <!-- SLIDING_WINDOW_SIZE
      Size of buffer used by context tracker to detect context changes
    -->
    <SLIDING_WINDOW_SIZE>80</SLIDING_WINDOW_SIZE>
    <!-- LOWERCASE_MODE
      Instruct context tracker to track text as lowercase
    -->
    <LOWERCASE_MODE>yes</LOWERCASE_MODE>
    <!-- ONLINE_LEARNING
      Controls presage online machine learning feature.
      Presage is context-aware and capable of dynamic online learning.
      Setting this to yes/true will enable online learning mode.
      Setting this to no/false will disable online learning mode.

      When online learning mode is disabled, it is still
      possible to instruct presage to learn through its API.
    -->
    <ONLINE_LEARNING>no</ONLINE_LEARNING>
  </ContextTracker>
  <Selector>
    <LOGGER>ERROR</LOGGER>
    <!-- SUGGESTIONS
      Controls how many suggestions are returned in each prediction.
    -->
    <SUGGESTIONS>5</SUGGESTIONS>
    <!-- REPEAT_SUGGESTIONS
      Allow the same suggestion to be offered in subsequent
      predictions, even if no context change has been detected.
```

```

-->
<REPEAT_SUGGESTIONS>no</REPEAT_SUGGESTIONS>
<!-- GREEDY_SUGGESTION_THRESHOLD
  Select only tokens whose completion length is greater than
  the specified greedy suggestion threshold.
  i.e. If this option is set to 2 and the current prefix is
       "cu", then the word "cub" will not offered as a
       suggestion, because the completion's length is only one
       character long. Tokens "curb" or "cube" or "cubicle" or
       "cucumber" will however be offered, because these
       words' completions are at least 2 characters long.
-->
<GREEDY_SUGGESTION_THRESHOLD>1</GREEDY_SUGGESTION_THRESHOLD>
</Selector>
<PredictorActivator>
  <LOGGER>ERROR</LOGGER>
  <!-- PREDICT_TIME
    Maximum time allowed for predictors to return their prediction.
  -->
  <PREDICT_TIME>1000</PREDICT_TIME>
  <!-- MAX_PARTIAL_PREDICTION_SIZE
    Desired size of each prediction prior to combination phase.
  -->
  <MAX_PARTIAL_PREDICTION_SIZE>60</MAX_PARTIAL_PREDICTION_SIZE>
  <!-- COMBINATION_POLICY
    policy used by predictor to combine predictions returned
    by the active predictors into one prediction.
  -->
  <COMBINATION_POLICY>Meritocracy</COMBINATION_POLICY>
</PredictorActivator>
<ProfileManager>
  <LOGGER>ERROR</LOGGER>
  <!-- AUTOPERSIST
    Automatically saves configuration to active profile.
  -->
  <AUTOPERSIST>false</AUTOPERSIST>
</ProfileManager>
<Predictors>
  <DefaultSmoothedNgramPredictor>
    <PREDICTOR>SmoothedNgramPredictor</PREDICTOR>
    <LOGGER>ERROR</LOGGER>
    <DBFILENAME>C:\Program Files\presage\share\presage\db_ggbooks_1.db</
DBFILENAME>
    <!-- delta_0, delta_1, ..., delta_{n-1}
      Deltas also control the value of n in the n-gram model.
    -->
    <DELTAS>0.01 0.1 0.89</DELTAS>

```



```

    <LEARN>false</LEARN>
    <DatabaseConnector>
        <LOGGER>ERROR</LOGGER>
    </DatabaseConnector>
</DefaultSmoothedNgramPredictor>
<UserSmoothedNgramPredictor>
    <PREDICTOR>SmoothedNgramPredictor</PREDICTOR>
    <LOGGER>ERROR</LOGGER>
<!-- ${HOME} is special. It expands to:
    - $HOME on Unix
    - %USERPROFILE% on Windows
-->
    <DBFILENAME>${HOME} /. presage / db_perso .db</DBFILENAME>
    <!-- delta_0 , delta_1 , ... , delta_{n-1}
Deltas also control the value of n in the n-gram model.
-->
    <DELTAS>0.01 0.1 0.89</DELTAS>
    <LEARN>false</LEARN>
    <DatabaseConnector>
        <LOGGER>ERROR</LOGGER>
    </DatabaseConnector>
</UserSmoothedNgramPredictor>
<DefaultRecencyPredictor>
    <PREDICTOR>RecencyPredictor</PREDICTOR>
    <LOGGER>ERROR</LOGGER>
    <LAMBDA>1</LAMBDA>
    <N_0>1</N_0>
    <CUTOFF_THRESHOLD>20</CUTOFF_THRESHOLD>
</DefaultRecencyPredictor>
<DefaultDictionaryPredictor>
    <PREDICTOR>DictionaryPredictor</PREDICTOR>
    <LOGGER>ERROR</LOGGER>
    <DICTIONARY>/usr/share/dict/words</DICTIONARY>
    <!-- fixed probability assigned to prediction -->
    <PROBABILITY>0.000001</PROBABILITY>
</DefaultDictionaryPredictor>
<DefaultAbbreviationExpansionPredictor>
    <PREDICTOR>AbbreviationExpansionPredictor</PREDICTOR>
    <LOGGER>ERROR</LOGGER>
    <ABBREVIATIONS>C:\Program Files\presage\share\presage/
abbreviations_fr.txt</ABBREVIATIONS>
</DefaultAbbreviationExpansionPredictor>
<DefaultDejavuPredictor>
    <PREDICTOR>DejavuPredictor</PREDICTOR>
    <LOGGER>ERROR</LOGGER>
    <MEMORY>C:\Program Files\presage\share\presage/dejavu_memory_en.txt<
/MEMORY>

```

```

        <TRIGGER>3</TRIGGER>
    </DefaultDejavuPredictor>
    <DefaultARPAPredictor>
        <PREDICTOR>ARPAPredictor</PREDICTOR>
        <LOGGER>ERROR</LOGGER>
        <ARPAFILENAME>C:\Program Files\presage\share\presage\lggbooksa.arpa<
/ARPAFILENAME>
        <VOCABFILENAME>C:\Program Files\presage\share\presage\lggbooks.vocab
</VOCABFILENAME>
        <TIMEOUT>100</TIMEOUT>
    </DefaultARPAPredictor>
</Predictors>
</Presage>

```

Le fichier de configuration d'ACAT (PresageWordPredictorSettings.xml) de Denman *et al.* (2016). Il s'agit de la version du pack français que nous avons adaptée. Afin d'accélérer la vitesse de l'interface ACAT, nous avons décidé d'utiliser la base de données db_ggbooks_2, c'est-à-dire une plus petite version de Google Books Ngram puisqu'elle a été traitée avec le code SQL (cf. *supra*).

```

<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http:
/ /www.w3.org/2001/XMLSchema">
    <DatabaseFileName>db_ggbooks_2.db</DatabaseFileName>
    <LearningDatabaseFileName>db_perso.db</LearningDatabaseFileName>
</Settings>

```

Annexe C

Corpus Blog

Pour accéder à chacun des articles, il faut entrer le lien vers le blog (<http://blog.royaumehandi.fr/>) suivi de la date et de l'intitulé de l'article. Les statistiques sont les résultats obtenus avec la fonction `len()` du script Python (après tokenisation).

Exemple <http://blog.royaumehandi.fr/2012/01/les-claviers-virtuels>

Excepté le dernier article, tous sont issus des catégories « Conseils pour les personnes handicapées » et « Informations personnelles », les autres catégories étant plus littéraires. Le script n'a pas pris en compte les poèmes pour des raisons stylistiques.

DATE	INTITULÉ DE L'ARTICLE	NBRE DE MOTS	NBRE DE LETTRES
2016/02/	les-situations-de-la-vie-de-tous-les-jours-propre-a-mon-handicap	957	5 233
2012/01/	les-claviers-virtuels	249	1 474
2012/01/	le-multicontrol	68	401
2011/12/	linfantilisation	234	1 324
2011/12/	la-piscine	152	884
2010/12/	ma-facon-de-communiquer	283	1 558
2016/01/	mon-aventure-avec-pole-emploi	34	197
2016/01/	l'autonomie	373	2 009
2015/11/	propositions-d'améliorations-pour-les-associations-daide-a-domicile	312	1 854
2015/11/	ma-nouvelle-vie-et-mon-experience-avec-mon-association-daide-a-domicile	1 176	6 440
2015/06/	mon-point-de-vue-sur-l'accessibilité-en-france	497	2 962
2014/08/	ma-future-location	312	1 664
2014/08/	mes-aides-a-domicile	264	1 389

2012/06/	un-gouvernement-exemplaire	253	1 386
2012/05/	mariage-homosexuel	133	729
2012/03/	constat-apres-un-dialogue	188	1 121
2011/03/	cap-emplois-et-creation-d'une-entreprise	191	1 010
2011/02/	sexe-et-handicap	286	1 547
2011/02/	le-choix	143	820
2011/02/	handisport-et-television	193	1 081
2011/02/	des-modeles-a-suivre	172	974
2011/02/	handicap-et-travail	242	1 361
2011/02/	sensibilisation-sur-le-handicap	217	1 247
2011/02/	une-petite-remarque-amusante	122	661
2011/02/	avion-et-handicap	247	1 289
2011/02/	ces-anges-non-reconnus	189	1 041
2011/01/	domination	80	447
2011/01/	petite-constatation	74	409
2011/01/	représenter-les-miens	134	771
2011/01/	la-difference	432	2 430
2010/12/	une-famille-d'un-enfant-a-mobilite-reduite-vu-par-un-handicape	423	2 184
2010/12/	handicap-ennemi-de-la-television	199	1 092
2010/12/	l'informatique-et-le-handicap	357	2 035
2010/12/	ma-normalite	337	1 770
2010/12/	description-de-mon-handicap-par-rapport-a-mes-experiences	281	1 580
2010/12/	ma-scolarite	374	2 032
2010/12/	l'amusement-passif	306	1 632
2010/12/	le-profit-dans-le-domaine-du-handicap	280	1 533
2010/11/	un-medley-de-mes-pensees	471	2 386
2010/11/	mon-regard-sur-mon-handicap	511	2 683
2010/07/	mon-handicap	1 183	7 998
2010/07/	mon-experience-personnelle-avec-les-aides-de-vie-scolaire	262	1 368
2010/07/	nouvelle-horizon-problemes	97	501
	a-propos-2	746	4 173
TOTAL		14 071	80 727

TABLE C.1 – Liste des articles utilisés

Annexe D

Questionnaire

Ce questionnaire est anonyme et réalisé dans le cadre du test utilisateur du mémoire « Configuration et évaluation d'un système de prédiction de mots au sein d'un logiciel de Communication Améliorée et Alternative (CAA) pour personnes handicapées : Étude de Presage et ACAT ». Il est composé de quatre parties :

1. Profil utilisateur
2. Usage des systèmes de prédiction et CAA
3. Test des systèmes de prédiction et CAA
4. System Usability Scale (SUS)

Profil utilisateur

- **Initiales (nom – prénom) :** M B
- **Sexe :** masculin
- **Âge :** 26 ans
- **Handicap :** Infirmité Motrice Cérébrale (IMC) athétosique
- **Bonne vue :** oui
- **Niveau de français :** expérimenté

Usage des systèmes de prédiction et de CAA

- **Quel(s) système(s) de prédiction utilisez-vous :** clavier visuel Windows, [Skippy]
 - **Fréquence :** presque toujours
- **Pourquoi utilisez-vous un système de prédiction :** par rapidité, pour plus de confort d'utilisation
- **Êtes-vous satisfait des systèmes de prédiction que vous utilisez :** oui

- **Combien de prédictions utilisez-vous généralement** : il y a 8 prédictions dans le clavier visuel Windows
- **Quel est le nombre minimum et le nombre maximum de prédictions** : en général, il n'y a que 8 prédictions dans le clavier Windows
- **Quel(s) paramètre(s) modifiez-vous** : je ne touche pas aux réglages, pour moi, il est suffisamment adapté. Peut-être moins dans Windows 7 car les prédictions n'apparaissent pas automatiquement après avoir tapé un mot, je dois appuyer sur une lettre pour que les prédictions apparaissent
- **Utilisez-vous les abréviations** : oui
 - **Fréquence** : rarement
- **Utilisez-vous la ponctuation** : oui
 - **Fréquence** : presque toujours
- **Utilisez-vous les lettres avec accent** : oui
 - **Fréquence** : presque toujours
- **Quel(s) système(s) de CAA utilisez-vous** : Mind Express
 - **Fréquence** : presque toujours
- **Quelle(s) modalité(s) de saisie utilisez-vous** : contacteurs, joystick
- **Utilisez-vous la synthèse vocale** : oui
 - **Fréquence** : presque toujours

Test des systèmes de prédiction et de CAA

Après avoir testé le système de prédiction Presage¹ et le logiciel de CAA Assistive Context-Aware Toolkit (ACAT)², nous vous invitons à répondre à quelques questions afin de pouvoir proposer des pistes d'amélioration. Nous nous intéressons à la manière dont les utilisateurs interagissent avec ces systèmes et leur ressenti.

- **Aviez-vous déjà utilisé ces deux logiciels** : non
- **Estimez-vous que vous avez rencontré des difficultés pour atteindre l'objectif (la recopie du texte)** : oui, pour moi, il y avait beaucoup de bugs
- **Étiez-vous satisfait des prédictions** : oui
- **Étiez-vous satisfait de l'affichage des prédictions et du clavier virtuel** : oui
- **Quels seraient les deux qualités des logiciels utilisés** : un moyen pratique, facile à utiliser
- **Quels seraient les deux défauts des logiciels utilisés** : le blocage pendant l'utilisation, la problématique des accents

1. Vescovi, GNU General Public License version 2.0 (<http://presage.sourceforge.net/>).

2. Intel Corporation, Apache License version 2.0 (<https://01.org/acat>).

System Usability Scale (SUS)

Pour chacune des dix affirmations suivantes, veuillez cocher (X) vos ressentis en fonction des différentes situations présentées sur une échelle de 1 (pas du tout d'accord) à 5 (tout à fait d'accord).

		1	2	3	4	5
1	Je pense que j'aimerais utiliser fréquemment ce système			X		
2	J'ai trouvé ce système inutilement complexe		X			
3	J'ai trouvé ce système facile à utiliser				X	
4	Je pense que j'aurais besoin du support d'un spécialiste pour utiliser ce système	X				
5	J'ai trouvé que les différentes fonctions de ce système étaient bien intégrées				X	
6	J'ai trouvé ce système trop incohérent		X			
7	Je pense que ce système sera facile à apprendre pour beaucoup de personnes					X
8	J'ai trouvé ce système très contraignant à utiliser			X		
9	Je me suis senti(e) en confiance lorsque j'ai utilisé ce système					X
10	J'ai dû apprendre beaucoup de choses avant de me sentir familiarisé(e) avec ce système					X

TABLE D.1 – SUS : questionnaire de satisfaction

Il convient de noter que pour calculer le score d'utilisabilité par répondant et obtenir une note sur 100, il faut passer par différentes étapes de calcul :

- pour les items 1, 3, 5, 7 et 9, soustraire 1 pour chaque score;
- pour les items 2, 4, 6, 8 et 10, calculer 5 - (moins) chaque score;
- faire la somme des 10 scores (les 10 items) ainsi recalculés;
- multiplier la somme par 2,5.

Table des matières

Liste des figures	iv
Liste des tables	v
Liste des abréviations	vi
Introduction	1
I État de l'art	4
1 Communication améliorée et alternative	4
1.1 Représentation du langage	4
1.2 Aide au handicap	6
1.3 Problématique et positionnement	8
1.3.1 Technologies et communication	8
1.3.2 Complétion et liste de prédictions	9
2 Systèmes de prédiction	12
2.1 Approche statistique	13
2.1.1 Modèle markovien de langage	14
2.1.2 Techniques stochastiques avancées	16
2.2 Approche symbolique	19
2.2.1 Aspects morpho-lexicaux	19
2.2.2 Aspects syntaxiques	22
2.2.3 Aspects sémantiques	23
2.3 Approche hybride	25
2.4 Synthèse des systèmes	29
3 Limites linguistiques, cognitives et ergonomiques	33
3.1 Modélisation du langage	33
3.2 Usage de la prédiction	37
3.3 Apport des métriques	41
3.4 Représentation des caractères	44
II Étude de cas	46
1 Objectif	46
2 Outils	47

2.1	Presage	47
2.1.1	Description générale	48
2.1.2	Algorithmes de prédiction	48
2.2	Assistive Context-Aware Toolkit (ACAT)	50
2.2.1	Description de l'interface	50
2.2.2	Flexibilité et paramétrage utilisateur	53
3	Méthodologie	54
3.1	Configuration	54
3.1.1	Corpus	54
3.1.2	Prétraitements	57
3.1.3	Formats de données	59
3.1.4	Abréviations	60
3.1.5	Paramétrage	62
3.2	Évaluation	63
3.2.1	Simulations	63
3.2.2	Test utilisateur	66
4	Résultats	68
4.1	Question 1 – Corpus	68
4.2	Question 2 – Prédicteurs	69
4.2.1	Generalized default smoothed n-gram et n-grammes	69
4.2.2	Generalized default smoothed n-gram et lissage	69
4.2.3	Generalized default / user smoothed n-gram	70
4.2.4	ARPA et lissage	70
4.3	Question 3 – Prédictions	71
4.4	Question 4 – Interface utilisateur	72
5	Discussion	74
	Conclusion	80
	A Accessibilité du Web	84
	B Code source	86
	C Corpus Blog	98
	D Questionnaire	100
	Table des matières	104
	Bibliographie	105

Bibliographie

- (Antoine, 2011) Jean-Yves ANTOINE, Prédiction de mots et saisie de requêtes sur interfaces limitées : Dispositifs mobiles et aide au handicap. *In Recherche d'information contextuelle, assistée et personnalisée*, Recherche d'information et web, pages 273–298. Hermès-Lavoisier, 2011.
- (Antoine *et al.*, 2014a) Jean-Yves ANTOINE, Marie-Elisabeth LABAT, Anaïs LEFEUVRE-HALFTERMEYER et Christian TOINARD, Vers une méthode de maîtrise des risques dans l'informatisation de l'aide au handicap. *In Envirorisk'2014, Le forum de la gestion des risques technologiques, naturels et sanitaires*. Cépaduès, 2014a.
- (Antoine *et al.*, 2014b) Jean-Yves ANTOINE, Anaïs LEFEUVRE-HALFTERMEYER et Willy ALLEGRE, Pour une réflexion éthique sur les conséquences de l'usage des NTIC : Le cas des aides techniques (à composante langagière ou non) aux personnes handicapées. *In Journée ATALA Ethique et TAL*, 2014b.
- (Antoine et Maurel, 2007) Jean-Yves ANTOINE et Denis MAUREL, Aide à la communication pour personnes handicapées et prédiction de texte : Problématique, état des lieux et retour sur trente ans de recherche en communication augmentée. *TAL Traitement Automatique des Langues*, 48(2):9–46, 2007.
- (Asnani *et al.*, 2015) Kavita ASNANI, Douglas VAZ, Tanay PRABHUDESAI, Surabhi BORGIKAR, Megha BISHT, Sharvari BHOSALE et Nikhil BALAJI, Sentence Completion Using Text Prediction Systems. *In Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, pages 397–404. Springer, 2015.
- (Badr, 2011) Georges BADR, *Modèle théorique et outil de simulation pour une meilleure évaluation des claviers logiciels augmentés d'un système de prédiction de mots*. Thèse de doctorat, Université Toulouse 3 Paul Sabatier, 2011.
- (Badr et Raynal, 2009) Georges BADR et Mathieu RAYNAL, WordTree : Results of a Word Prediction System Presented Thanks to a Tree. *In International Conference on Universal Access in Human-Computer Interaction*, pages 463–471. Springer, 2009.

- (Bérard et Niemeijer, 2004) Christian BÉRARD et David NIEMEIJER, Evaluating Effort Reduction through Different Word Prediction Systems. *In International Conference on Systems, Man and Cybernetics*, volume 3, pages 2658–2663, The Hague, Netherlands, Octobre 2004. IEEE.
- (Bianco *et al.*, 2007) Anick BIANCO, Philippe BLACHE, Julie MARTY et Stéphane RAUZY, Les outils de communication alternative. *In Les Dysarthries*, pages 1–12. Solal, 2007.
- (Bird *et al.*, 2009) Steven BIRD, Ewan KLEIN et Edward LOPER, *Natural Language Processing with Python : Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., 2009.
- (Blache et Rauzy, 2007) Philippe BLACHE et Stéphane RAUZY, Le moteur de prédiction de mots de la plateforme de communication alternative. *TAL Traitement Automatique des Langues*, 48(2):47–70, 2007.
- (Boissière et Dours, 2001) Philippe BOISSIÈRE et Daniel DOURS, Comment VITIPI un système d'assistance à l'écriture pour les personnes handicapées peut offrir des propriétés intéressantes pour le TALN? *In Proceedings of TALN*, volume 1, 2001.
- (Boissière *et al.*, 2015) Philippe BOISSIÈRE, Nadine VIGOUROUX, Mustapha MOJAHID, Damien SAUZIN et Frédéric VELLA, TALN et IHM : Une approche transdisciplinaire pour la saisie de textes de personnes en situation de handicaps. *In 22eme Conference sur le Traitement Automatique des Langues Naturelles (TALN 2015)*, 2015.
- (Boissière *et al.*, 2012) Philippe BOISSIÈRE, Nadine VIGOUROUX, Mustapha MOJAHID et Frédéric VELLA, Adaptation of AAC to the context communication : A real improvement for the user illustration through the VITIPI word completion. *In International Conference on Computers for Handicapped Persons*, pages 451–458. Springer, 2012.
- (Caldwell *et al.*, 2008) Ben CALDWELL, Michael COOPER, Loretta Guarino REID et Gregg VANDERHEIDEN, Web Content Accessibility Guidelines (WCAG) 2.0, 2008. W3C Recommendation, <https://www.w3.org/TR/WCAG20/>.
- (Cataix-Nègre, 2017) Élisabeth CATAIX-NÈGRE, *Communiquer autrement : Accompagner les personnes avec des troubles de la parole ou du langage*. De Boeck Supérieur, 2ème édition, 2017.
- (Chomsky, 1956) Noam CHOMSKY, Three models for the description of language. *IRE Transactions on Information Theory*, 2:113–124, 1956.
- (Chomsky, 1969) Noam CHOMSKY, *Structures syntaxiques*. Éditions du Seuil, 1969.

- (Copestake, 1997) Ann COPESTAKE, Augmented and alternative NLP techniques for augmentative and alternative communication. *Natural Language Processing for Communication Aids*, pages 37–42, 1997.
- (Denman *et al.*, 2016) Pete DENMAN, Lama NACHMAN et Sai PRASAD, Designing for "a" user : Stephen Hawking's UI. In *Proceedings of the 14th Participatory Design Conference : Short Papers, Interactive Exhibitions, Workshops*, pages 94–95. ACM, 2016.
- (European Commission, 2017) EUROPEAN COMMISSION, Progress Report on the implementation of the European Disability Strategy (2010–2020), 2017. Commission Staff Working Document, <http://ec.europa.eu/social/main.jsp?langId=en&catId=89&newsId=2725>.
- (Fairon, 2010) Cédrick FAIRON, Constitution et étude de corpus spécialisés sur le Web. *Le discours et la langue*, 2(1), 2010.
- (Garay-Vitoria et Abascal, 1997) Nestor GARAY-VITORIA et Julio ABASCAL, Intelligent Word-Prediction to Enhance Text Input Rate (A Syntactic Analysis-Based Word-Prediction Aid for People with Severe Motor and Speech Disability). In *Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 241–244. ACM, 1997.
- (Garay-Vitoria et Abascal, 2006) Nestor GARAY-VITORIA et Julio ABASCAL, Text prediction systems : A survey. *Universal Access in the Information Society*, 4(3):188–203, 2006.
- (Gelšvartas *et al.*, 2016) Julius GELŠVARTAS, Rimvydas SIMUTIS et Rytis MASKELIŪNAS, User adaptive text predictor for mentally disabled huntington's patients. *Computational Intelligence and Neuroscience*, 2016.
- (Ghedira, 2009) Souhir GHEDIRA, *Aide à la communication pour personnes handicapés moteurs : Modélisation et optimisation du système homme-machine*. Thèse de doctorat, Université Paul Verlaine, 2009.
- (Ghedira, 2015) Souhir GHEDIRA, Optimisation de la communication pour personnes handicapées. In *22eme Conference sur le Traitement Automatique des Langues Naturelles (TALN 2015)*, 2015.
- (Guenthner *et al.*, 1994) Franz GUENTHNER, Stephan LANGER, Karin KRÜGER-THIELMANN, Nathalie RICHARDET, Paul SABATIER et Robert PASERO, KOMBE : Communication aids for the handicapped, 1994.
- (Guerrier, 2015) Yohan GUERRIER, *Proposition d'une aide logicielle pour la saisie d'information en situation dégradée : Application à des utilisateurs IMC athétosiques dans des contextes liés au transport et aux activités journalières*. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambresis, 2015.

- (Hawking, 2013) Stephen HAWKING, *My brief history*. Random House, 2013.
- (Hunnicutt et Carlberger, 2001) Sheri HUNNICUTT et Johan CARLBERGER, Improving Word Prediction Using Markov Models and Heuristic Methods. *AAC Augmentative and Alternative Communication*, 17(4):255–264, 2001.
- (Jurafsky et Martin, 2009) Daniel JURAFSKY et James H. MARTIN, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2ème édition, 2009.
- (Laffont *et al.*, 2007) Isabelle LAFFONT, Claude DUMAS, Dephine POZZI, Maria RUQUET, Anne Claire TISSIER, Frédéric LOFASO et Olivier DIZIEN, Home trials of a speech synthesizer in severe dysarthria : Patterns of use, satisfaction and utility of word prediction. *Journal of Rehabilitation Medicine*, 39(5):309–405, 2007.
- (Lefevre-Halftermeyer *et al.*, 2016) Anaïs LEFEUVRE-HALFTERMEYER, Virginie GOVAERE, Jean-Yves ANTOINE, Willy ALLEGRE, Samuel POUPLIN, Jean-Paul DEPARTE, Samia SLIMANI et Aurore SPAGNULO, Typologie des risques pour une analyse éthique de l'impact des technologies du TAL. *TAL et Ethique*, 57(2):47–71, 2016.
- (Leshner *et al.*, 1999) Gregory W. LESHER, Bryan J. MOULTON, D. Jeffery HIGGINBOTHAM, Kathleen F. MCCOY et Christopher PENNINGTON, Effects of ngram order and training text size on word prediction. *In Proceedings of the RESNA'99 Annual Conference*, pages 52–54. Citeseer, 1999.
- (MacKenzie et Soukoreff, 2003) I. Scott MACKENZIE et R. William SOUKOREFF, Phrase sets for evaluating text entry techniques. *In CHI'03 extended abstracts on Human factors in computing systems*, pages 754–755. ACM, 2003.
- (Martin et Pecci, 2007) Benoît MARTIN et Isabelle PECCI, État de l'art des claviers physiques et logiciels pour la saisie de texte. *Revue d'Interaction Homme-Machine (RIHM)*, 8(2):147–205, 2007.
- (Maurel et Le Pévédic, 2001) Denis MAUREL et Brigitte LE PÉVÉDIC, The syntactic prediction with token automata : Application to HandiAS system. *Theoretical Computer Science*, 267(1):121–129, 2001.
- (Maurel *et al.*, 1998) Denis MAUREL, Brigitte LE PÉVÉDIC et Jonathan YAVCHITZ, La prédiction lexicale et syntaxique à partir du contexte gauche : Application au système handias. *Revue Informatique et Statistique dans les Sciences Humaines*, 33:275–284, 1998.

- (Merlo, 2015) Aurelie MERLO, Une aide à la communication par pictogrammes avec prédiction sémantique. *In 22eme Conference sur le Traitement Automatique des Langues Naturelles (TALN 2015)*, 2015.
- (Michel *et al.*, 2011) Jean-Baptiste MICHEL, Yuan Kui SHEN, Aviva Presser AIDEN, Adrian VERES, Matthew K GRAY, Joseph P PICKETT, Dale HOIBERG, Dan CLANCY, Peter NORVIG, Jon ORWANT *et al.*, Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011.
- (Niemeijer, 2005) David NIEMEIJER, In memoriam of Christian Bérard : Striving for effort reduction through on-screen keyboard word prediction. *In Assistive technology : From virtuality to reality*, Lille, France, Septembre 2005.
- (Pasero *et al.*, 1994) Robert PASERO, Nathalie RICHARDET et Paul SABATIER, Guided sentences composition for disabled people. *In Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 205–206. Association for Computational Linguistics, 1994.
- (Pino, 2014) Alexandros PINO, Augmentative and Alternative Communication Systems for the Motor Disabled. *In Disability Informatics and Web Accessibility for Motor Limitations*, pages 105–152. IGI Global, 2014.
- (Pouplin, 2016) Samuel POUPLIN, *Evaluation de l'efficacité des logiciels de prédiction de mots sur la vitesse de saisie de texte sur l'outil informatique pour les personnes blessées médullaires cervicaux*. Thèse de doctorat, Université Paris-Saclay, 2016.
- (Pouplin *et al.*, 2014) Samuel POUPLIN, Johanna ROBERTSON, Jean-Yves ANTOINE, Antoine BLANCHET, Jean Loup KAHLOUN, Philippe VOLLE, Justine BOUTEILLE, Frédéric LOFASO et Djamel BENSMAIL, Effect of dynamic keyboard and word-prediction systems on text input speed in persons with functional tetraplegia. *Journal of Rehabilitation Research and Development*, 51(3):467–480, 2014.
- (Raynal, 2007) Mathieu RAYNAL, Le système KeyGlass : Système d'ajout dynamique de touches sur clavier logiciel. *TAL Traitement Automatique des Langues*, 48(2):97–121, 2007.
- (Renaud *et al.*, 2010) Alfred RENAUD, Fraser SHEIN et Vivian TSANG, Grammaticality judgement in a word completion task. *In Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing : Writing Processes and Authoring Aids*, pages 15–23. Association for Computational Linguistics, 2010.
- (Rosenfeld et Clarkson, 1997) Roni ROSENFELD et Philip CLARKSON, Statistical Language Modeling using the CMU-Cambridge Toolkit. 1997.

- (Schadle, 2003) Igor SCHADLE, *Sibylle : Système linguistique d'aide à la communication pour les personnes handicapées*. Thèse de doctorat, Université de Bretagne Sud, 2003.
- (Sevens *et al.*, 2015) Leen SEVENS, Vincent VANDEGHINSTE, Ineke SCHUURMAN et Frank VAN EYNDE, Natural language generation from pictographs. *In Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 71–75. Association for Computational Linguistics, 2015.
- (Stoop et van den Bosch, 2014) Wessel STOOP et Antal VAN DEN BOSCH, Improving word prediction for augmentative communication by using idiolects and sociolects. *Dutch Journal of Applied Linguistics*, 3(2):136–153, 2014.
- (Tenny, 2016) Carol TENNY, A linguist looks at AAC : Language representation systems for augmentative and alternative communication, compared with writing systems and natural language. *Writing Systems Research*, 8(1):84–119, 2016.
- (Trnka, 2010) Keith TRNKA, *Word Prediction Techniques for User Adaptation and Sparse Data Mitigation*. Thèse de doctorat, University of Delaware, 2010.
- (Trnka et McCoy, 2007) Keith TRNKA et Kathleen F. MCCOY, Corpus studies in word prediction. *In Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 195–202, 2007.
- (Trnka *et al.*, 2007) Keith TRNKA, Debra YARRINGTON, John MCCAW, Kathleen F. MCCOY et Christopher PENNINGTON, The effects of word prediction on communication rate for AAC. *In Proceedings of NAACL HLT 2007, Companion Volume*, pages 173–176, 2007.
- (Trnka *et al.*, 2006) Keith TRNKA, Debra YARRINGTON, Kathleen F. MCCOY et Christopher PENNINGTON, Topic Modeling in Fringe Word Prediction for AAC. *In International Conference on Intelligent User Interfaces (IUI)*, pages 276–278, 2006.
- (Trost *et al.*, 2005) Harald TROST, Johannes MATIASEK et Marco BARONI, The Language Component of the FASTY Text Prediction System. *Applied Artificial Intelligence*, 19(8):743–781, 2005.
- (Verborgh et De Wilde, 2013) Ruben VERBORGH et Max DE WILDE, *Using OpenRefine*. Packt Publishing, Birmingham, 2013.
- (Vescovi, 2004) Matteo VESCOVI, *Soothsayer : Un sistema multi-sorgente per la predizione del testo*. Thèse de doctorat, Politecnico di Milano, 2004.
- (Wandmacher, 2008) Tonio WANDMACHER, *Adaptive word prediction and its application in an assistive communication system*. Thèse de doctorat, Université François – Rabelais, 2008.

- (Wandmacher et Antoine, 2007a) Tonio WANDMACHER et Jean-Yves ANTOINE, Methods to integrate a language model with semantic information for a word prediction component. *EMNLP-CoNLL 2007*, pages 506–513, 2007a.
- (Wandmacher et Antoine, 2007b) Tonio WANDMACHER et Jean-Yves ANTOINE, Modèle adaptatif pour la prédiction de mots : Adaptation à l'utilisateur et au contexte dans le cadre de la communication assistée pour personnes handicapées. *TAL Traitement Automatique des Langues*, 48(2):71–95, 2007b.
- (Wester, 2003) Malin WESTER, *User Evaluation of a Word Prediction System*. Thèse de doctorat, Uppsala University, 2003.
- (Wood, 1996) Matthew E. J. WOOD, *Syntactic Pre-Processing in Single-Word Prediction for Disabled People*. Thèse de doctorat, University of Bristol, 1996.