

SibyLettre

Prédiction de lettre pour la communication assistée

SCHADLE Igor, ANTOINE Jean-Yves, LE PÉVÉDIC Brigitte,
POIRIER Franck

Laboratoire VALORIA/Équipage, Université de Bretagne Sud (EA 2593)
{igor.schadle, jean-yves.antoine, brigitte.le-pevedic franck.poirier}@univ-ubs.fr

Résumé : Cet article présente la partie prédiction de lettre du projet Sibylle, un système d'aide à la communication pour personne handicapée. Le handicap concerne les personnes privées de l'usage de la parole et dont les facultés motrices sont très réduites. Le principe de la communication assistée sur ordinateur est de composer des messages par l'intermédiaire d'un clavier virtuel présenté à l'écran. Cette aide pose un problème de lenteur de saisie, en particulier dans le cas où la sélection des lettres utilise un système de défilement automatique. L'objectif du projet Sibylle est de baser le système sur une prédiction de mot qui permet d'économiser le nombre de saisies. La prédiction sera réalisée en intégrant des connaissances linguistiques. La prédiction de mot est cependant une tâche difficile, c'est pourquoi notre projet intègre également une prédiction de lettre dont le rôle est de présenter prioritairement les lettres les plus probables sur le clavier simulé. Cette partie, SibyLettre, est l'objet de ce présent article. Nous y présentons le modèle sur lequel est basée la prédiction (le modèle statistique n-gramme), le clavier simulé adapté (un clavier dynamique) ainsi qu'une application réalisée. Cette application est actuellement utilisée au centre de rééducation fonctionnelle de Kerpape par des enfants IMC (Infirmités Motrices Cérébrales). Par rapport aux systèmes habituels, SibyLettre permet un gain de l'ordre de deux fois moins de temps pour deux fois moins de validations.

Mots-clés : Handicap, aide à la communication, AAC, prédiction de lettre, modèle statistique n-gramme, clavier simulé dynamique.

1 Introduction

Cet article présente un système d'aide à la communication pour personne handicapée. Le handicap étudié concerne les personnes dont l'usage de la parole est fortement ou complètement altéré, et dont les facultés motrices sont très réduites (hémiplégie, tétraplégie). C'est le cas de patients IMC (Infirmités Motrices Cérébrales), SLA (Sclérose Latérale Amyotrophique) ou Locked-In Syndrom. Quelle que soit la maladie, le handicap physique est sévère, la communication est privée de son support oral habituel et les modalités d'interaction sont limitées.

Un moyen de communication alternatif est le recours aux systèmes dits de communication assistée ou AAC (Alternative and Augmentative Communication). On appelle système de communication assisté tout système de suppléance dont le rôle est d'augmenter ou de restaurer, ne serait-ce que partiellement, la fonction de communication. Cette aide rend une certaine autonomie à la personne handicapée et améliore sa qualité de vie.

Pour le handicap considéré ici, le principe des aides à la communication repose sur l'écriture de phrases à l'aide d'un tableau de symboles. Le message est construit en sélectionnant successivement sur le tableau les différents symboles qui le composent. L'intervention de la personne handicapée se limite à la désignation des symboles ; une personne « valide » ou une machine est chargée de relever les éléments désignés. Les symboles sont souvent les lettres de l'alphabet orthographique (éventuellement réduit) ; cet usage n'est cependant pas exclusif et le tableau peut contenir tout autre élément de langage (alphabet phonétique, symboles Bliss, icônes, etc.).

Communication assistée sur ordinateur

Un médium particulièrement intéressant est l'ordinateur grâce à la polyvalence dont il dispose et son relatif faible coût. On distingue alors l'interface matérielle d'accès à l'ordinateur, de la partie logicielle (Figure 1).

L'interface matérielle dépend du geste libre laissé par le handicap, elle est chargée de remplacer le clavier réel, interface rendue inadaptée. Il peut s'agir d'un joystick, d'une commande oculaire, d'une commande par souffle, d'un simple bouton poussoir, etc. Une caractéristique importante est le degré de liberté qu'elle autorise pour manipuler l'ordinateur.

La partie logicielle réalise à proprement parlé le système d'aide à la communication. Son interface est constituée principalement d'un clavier simulé (clavier présenté à l'écran) et d'un traitement texte, ou d'une simple zone d'édition pour afficher le texte en cours. Une synthèse vocale permet éventuellement de prononcer le texte composé.

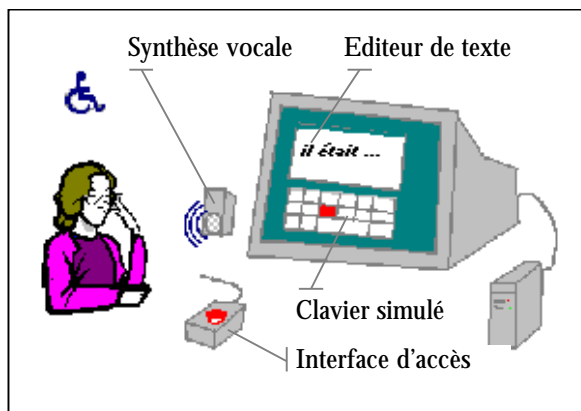


Figure 1: Système de communication assistée sur ordinateur.

La saisie sur clavier simulé est étroitement liée au degré de liberté offert par l'interface matérielle. Dans le meilleur des cas, il est équivalent à celui de la souris et l'utilisateur manipule directement le pointeur de la souris. Il en va tout autrement pour les commandes « tout ou rien » (comme le bouton poussoir) qui sont l'équivalent d'un simple clic. Dans ce cas, la sélection des symboles (i.e. les touches

du clavier) ne peut être réalisée que par un système de défilement automatique du curseur. Le curseur met en surbrillance successivement les touches une à une ; lorsque le curseur désigne la lettre désirée, l'utilisateur n'a plus qu'à valider. Ce mode de saisie est excessivement lent et c'est à ce type de saisie pénalisante que sont soumis les personnes handicapées concernées par notre système SibyLettre.

En conséquence, le problème majeur des systèmes de communication assistée est la lenteur de composition, en particulier dans le cas des commandes « tout ou rien ». La tâche de saisie est généralement longue et fastidieuse. A titre de comparaison, le Tableau 1 présente la vitesse de quelques modalités de communication (Le Pévédic, 1997). Notons que la vitesse donnée pour une personne handicapée (5 mots/min), déjà bien inférieure à celui d'une personne « valide », varie fortement en fonction de la gravité du handicap et peut se réduire à seulement quelques caractères par minute.

Type de communication	Vitesse
Communication orale	150
Communication manuscrite	12-16
Saisie secrétaire	25
Saisie clavier un doigt	11
Handicapé avec aide technique	5

Tableau 1: Exemples de vitesses de communication (mots/minute)

Pour permettre une saisie plus rapide, deux approches complémentaires sont envisageables, l'une cherche à minimiser le nombre de saisies, l'autre à accélérer la sélection sur le clavier simulé.

Dans la première approche, l'objectif consiste à économiser le nombre de saisies. Ceci peut être réalisé par différentes méthodes. Le rappel de phrases pré-enregistrées par touche de fonction est efficace mais limité, il a son utilité pour les phrases types ou pour une communication d'urgence. Un moyen efficace est l'utilisation d'un système d'abréviations, en particulier si le système est capable de décoder l'abréviation (Ricco, 2001), ou mieux, de créer des abréviations à la volée (McCoy, 1995). Une autre approche propose d'élaborer la communication à partir de scénarios ; la saisie est alors partiellement guidée par des schémas prédéfinis, l'utilisateur n'ayant plus qu'à saisir les mots-clé de la phrase (Richardet, 1998). Actuellement, les systèmes les plus performants sont basés sur une prédiction de mot. Le système VITIPI (Boissière, 2000) écrit automatiquement les lettres dès qu'il n'y a plus d'ambiguïté, tandis qu'un système comme HandiAS (Maurel, 2001) propose une liste de mots qui évite à l'utilisateur de taper la fin des mots.

Pendant, l'économie du nombre de lettres à saisir reste relativement limitée. Les systèmes VITIPI et HandiAS annoncent des gains en économie de saisies compris entre 26% et 43%. Ceci laisse encore plus de la moitié des lettres à saisir par l'utilisateur dans des conditions difficiles. C'est pourquoi, la deuxième approche pour accélérer la vitesse de saisie cherche à faciliter la sélection des lettres sur le clavier simulé. Elle concerne plus particulièrement la sélection avec défilement automatique. Dans ce cas, l'objectif consiste à minimiser le nombre de défilements pour atteindre la lettre désirée. Ce problème est loin d'être trivial, les travaux de Cantegril (2001) ont ainsi clairement montré que les systèmes de sélection habituellement proposés sont loin d'être optimaux. Les deux principales méthodes pour minimiser le nombre de défilements sont l'utilisation du défilement « ligne/colonne » et la prédiction de lettre. Le système SibyLettre présenté dans cet

article s'inscrit dans cet approche. Nous reviendrons plus en détails sur la problématique liée au défilement automatique et aux différentes solutions envisageables dans la section 3.

Ainsi, malgré les progrès réalisés, les systèmes d'aide à la communication peuvent être encore améliorés et tel est l'enjeu des recherches actuelles dans ce domaine. La recherche est particulièrement active et tend vers une approche pluridisciplinaire alliant psychologues, linguistes, ergonomes et informaticiens. Citons notamment les universités de Dundee (Ecosse), Stanford et Delaware (EU), du Royal Institut KTH (Suède) et en France, les laboratoires du LIM (Marseille), de l'IRIT (Toulouse), du LI/E3i (Tours) et de l'IRIN (Nantes). Le projet Sibylle, que nous allons maintenant présenter, se situe dans cet axe de recherche.

2 Projet Sibylle¹

L'objet de notre projet Sibylle est la réalisation d'un outil d'aide à la communication. Nous nous plaçons dans le cadre des aides logicielles sur ordinateur.

L'objectif principal est d'élaborer un système de prédiction de mots efficace pour économiser le nombre de saisies. Les propositions seront accessibles à l'utilisateur par l'intermédiaire d'une liste de mots (Figure 2). La prédiction sera réalisée en contexte, en fonction des mots et des lettres déjà saisis. La deuxième aide complémentaire apportée par Sibylle est une prédiction de lettre. La liste est directement affichée dans le clavier simulé (Figure 2). La prédiction est basée sur les lettres du mot en cours.

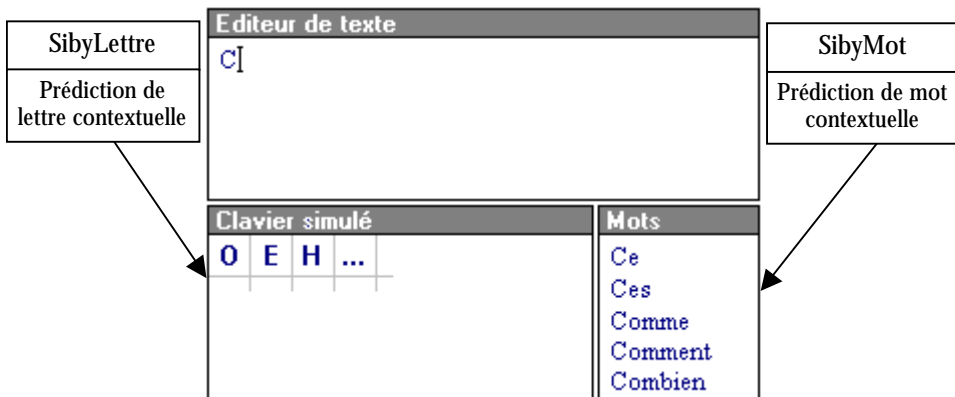


Figure 2 : Rôle des prédictions de mot et de lettre dans l'interface

Les figure suivantes présentent une simulation de saisie sur le début de phrase « Comme chaque ... ».

Au début de la phrase, le contexte est vide, la liste des mots propose les mots les plus courants commençant une phrase (déterminants « le, la, les », pronoms personnels « je » et « il ») ; le clavier simulé affiche lui en premier les lettres dont la probabilité d'apparition en début de mot sont les plus fortes « d, l, p, ... ». Supposons que l'utilisateur désire écrire le mot « Comme ». Ce mot n'apparaît pas dans la liste des mots, il tape la lettre sur le clavier simulé (Figure 3, Image 1).

¹ Dans l'antiquité, nom donné aux prêtresses qui prédisaient l'avenir.

Après chaque saisie, les deux listes sont actualisées. La liste des mots affiche de nouvelles propositions commençant par les lettres tapées, (ici le C), mots estimés comme les plus probables dans le contexte par le module de prédiction. Le mot « Comme » apparaît et l'utilisateur n'a plus qu'à le sélectionner (Figure 3, Image 2). Le caractère « espace » est automatiquement inséré. La saisie se poursuit ainsi pour les autres mots. Les images 2 et 3 de la Figure 3 montrent que la prédiction de mot tient compte des mots déjà saisis : dans les deux cas le mot en cours de saisie commencent par la lettre C mais les propositions affichées dans la liste des mots sont différentes.

Les lettres et les mots étant dans des listes différentes, se pose le problème de la sélection et du basculement d'une liste à l'autre. La sélection des lettres est réalisée simplement : après chaque saisie, le curseur se remet automatiquement en position sur la première lettre proposée. Quant à l'accès aux mots, il est réalisé par une touche « accès à la liste des mots » intégrée au clavier de lettres. La gestion de ce type de sélection alternée et l'ergonomie générale de l'interface seront présentées plus en détail dans la partie 6.

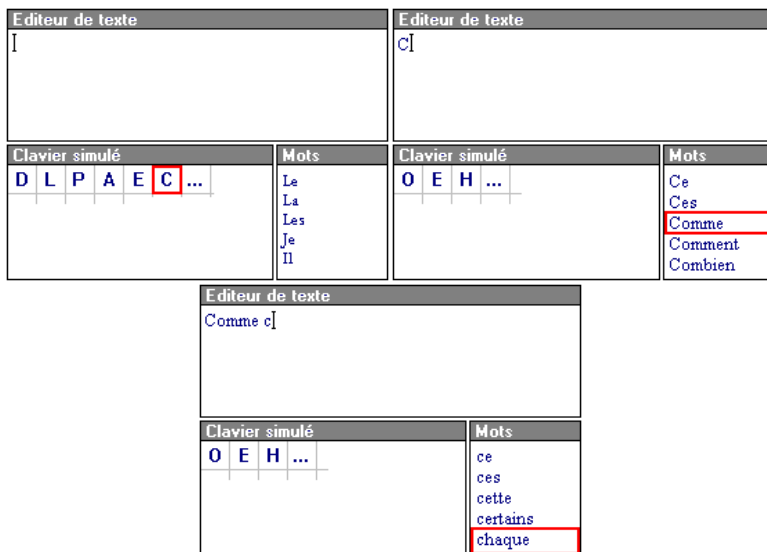


Figure 3 : Simulation de Sibylle sur la saisie de « Comme chaque... » avec l'actualisation du clavier de lettres et de la liste des mots en fonction du contexte

Ce projet s'effectue en collaboration avec le Centre Mutualiste de Rééducation et Réadaptation Fonctionnelle (CMRRF) de Kerpape. Le centre de Kerpape apporte sa compétence dans le domaine du handicap et son expérience dans les systèmes de communication assistée. Il a notamment participé aux projets d'aide à la communication PVI (Vaillant, 1997) et Axelia (Abraham, 2000) et contribue à la recherche dans le domaine du handicap. Son rôle est également de servir de centre de test et de validation des différents outils qui seront issus du projet. Dans le cadre de cette collaboration, de nombreux patients sont des enfants IMC, aux facultés motrices très réduites. Ils utilisent comme interface matérielle le bouton poussoir.

Dans cet article, nous présentons la première partie de notre projet, SibyLettre. SibyLettre s'adresse plus particulièrement aux personnes ayant recours à la sélection par défilement automatique. C'est une aide complémentaire à la prédiction de mot qui doit aider l'utilisateur dans la saisie des lettres que SibyMot ne permet pas d'économiser. Le rôle de SibyLettre est donc essentiel pour la saisie des mots inconnus du système et surtout au début des mots, là où le nombre de mots prédits est supérieur à celui que peut contenir la liste des mots.

Un des moyens d'optimiser la vitesse de sélection est de classer les lettres en fonction de leur fréquence d'usage dans la langue. Le balayage automatique commence ainsi par passer en revue les caractères les plus fréquents (typiquement l'espace, puis les lettres E, S, A, etc.). Le système que nous proposons, SibyLettre, est une généralisation de ce principe. Les lettres sont classées selon leur probabilité d'apparition contextuelle, en fonction des lettres déjà tapées du mot en cours. Par exemple, après la saisie de la première lettre C, le système propose en priorité les lettres O et E, ce qui reflète les nombreux mots commençant par CO ainsi que l'usage fréquent du déterminant « ce » et de ses dérivés. L'ordre des lettres est actualisé sur le clavier simulé après chaque saisie et ce afin de tenir compte du nouveau contexte. Le clavier était statique, il devient dynamique.

La suite de l'article est organisée de la manière suivante. D'abord nous rappelons les différents paramètres qui permettent d'améliorer la sélection par défilement automatique, en particulier celui du classement des lettres sur lequel est fondé SibyLettre (section 3). Ensuite, nous présentons le modèle utilisé pour la prédiction de lettre (section 4), puis le fonctionnement du clavier dynamique (section 5). Une dernière partie est consacrée à l'application Sibylle actuellement utilisée au centre de Kerpape et qui intègre SibyLettre (section 6).

3 Sélection par défilement automatique

Comme nous l'avons mentionné en introduction, lorsque d'interface d'accès n'autorise que l'équivalent du simple clic, le déplacement du curseur est réalisé par un défilement automatique et le clic sert à valider l'élément mis en surbrillance. Ce mode de sélection est excessivement lent. Par exemple, dans le cas du défilement linéaire, où les touches sont parcourues une à une, l'accès à la lettre médiane d'un clavier de 65 touches est réalisé en 33 défilements. Dans ce contexte, améliorer la vitesse de saisie consiste à réduire le temps d'accès à la lettre désirée. La métrique couramment utilisée est le nombre de défilements moyen (NDM). Elle ne tient pas compte de la vitesse de défilement du curseur qui dépend essentiellement de l'utilisateur et s'exprime en secondes (généralement entre 1 et 5 secondes). Nous présentons dans cette partie les principaux paramètres qui permettent de minimiser le nombre de défilements moyen.

Type de balayage

Pour réduire le temps d'accès, la solution adoptée par les logiciels du commerce est d'utiliser un balayage adapté : le défilement « ligne / colonne » (désormais LC). La sélection d'une lettre se fait alors en deux temps : sélection de la ligne où se trouve la lettre désirée, puis sélection de la lettre dans la ligne. Plus formellement, ce balayage utilise le principe de dichotomie et peut être généralisé pour une sélection en V validations. Le nombre de défilements moyen est donné par l'équation :

$$\text{NDM} = V \times (T^{1/V} + 1) / 2 \quad (1)$$

T est le nombre de touches du clavier,

V est le nombre de validations nécessaire.

Cette formule suppose que la disposition de la matrice des touches est optimale pour le type de balayage (cf. infra).

Il est donc théoriquement possible d'obtenir un meilleur gain avec des V supérieurs. Cependant, le gain est obtenu au détriment d'une charge cognitive accrue (ce type de balayage nécessite en effet un effort de concentration et une charge intellectuelle plus importants pour comprendre le cheminement du balayage) et d'un nombre de validations supérieur (source d'erreurs). Pour un clavier de 64 touches, on trouve pour V=1 NDM=32,5 (défilement linéaire), pour V=2 NDM=9 (défilement LC, avec une matrice optimale de 8x8), pour V=3 NDM=7,5 (avec un découpage optimal à chaque étape en blocs de 4). Le défilement LC est généralement un bon compromis.

Disposition de la matrice des touches

L'équation précédente suppose déjà une optimisation pour la recherche dichotomique : la disposition de la matrice des touches du clavier. L'équation complète doit tenir compte du nombre de lignes et de colonnes. Par exemple, pour le défilement LC, l'équation est la suivante :

$$\text{NDM} = \sum_{i=1}^{i=N_c} i/N_c + \sum_{j=1}^{j=N_l} j/N_l = (N_c+1)/2 + (N_l+1)/2 \quad (2)$$

Nc est le nombre de colonnes, Nl le nombre de lignes.

A partir de cette équation, on démontre que la matrice des touches doit être carrée (pour le défilement LC). Les travaux de Cantegrit (2001) ont ainsi montré que les dispositions habituelles 3x10 (3 lignes par 10 colonnes) et 4x8, permettant d'afficher les 26 lettres de l'alphabet, sont clairement sous-optimales (Figure 4).



Figure 4 : Exemple d'interface (issue d'un logiciel existant) présentant les lettres en disposition 3x10, dans l'ordre alphabétique et sans les lettres accentuées

Nombre de touches

Une autre optimisation réalisable est de réduire le nombre de touches. Cette solution est effective dans certains logiciels avec l'omission des lettres accentuées (Fi). La saisie de texte sans accent ne correspond généralement pas au souhait des personnes handicapées, sauf quand elle constate une amélioration notable de sa

vitesse d'expression. Les lettres accentuées posent d'ailleurs un problème, elles n'ont ni leur place dans l'ordre « azerty », ni dans l'ordre alphabétique... Elles sont souvent difficiles d'accès et rend leur saisie pénalisante.

Ordre des lettres

Une dernière amélioration consiste à tenir compte de la probabilité d'apparition des lettres. L'équation du NDM devient :

$$\text{NDM} = \sum_{i=1}^{i=T} \text{PROB}(i) \times \text{DEF}(i) \quad (3)$$

PROB(i) est la probabilité d'apparition de la touche i,

DEF(i) est le nombre de défilements nécessaires pour accéder à la touche i. Il dépend de la position de la touche dans la matrice et du type de balayage.

D'après l'équation (3), pour minimiser le NDM, les lettres dont la probabilité d'apparition PROB(i) est la plus élevée doivent être placées sur les touches dont l'accès nécessite le moins de défilements DEF(i). Ainsi, sur clavier statique, le NDM est minimal pour l'ordre fréquentiel (ordre dépendant de la fréquence d'usage des lettres dans la langue, ce qui n'est pas le cas des ordres alphabétique (Figure 4) et « azerty »).

Cette solution peut elle-même être améliorée en classant les lettres selon leur probabilité d'apparition en contexte. L'ordre des lettres était statique, il devient dynamique. C'est le principe de SibyLettre. Ceci nécessite un modèle de prédiction de lettre que nous allons maintenant présenter.

4 Prédiction de lettre

4.1 Modélisation

Dans le cadre de Sibylle, il serait possible d'établir une prédiction de lettre à partir d'une approche lexicale, en particulier à l'aide des prédictions délivrées par SibyMot. Cependant, un des rôles attribué à SibyLettre est d'être efficace lorsque l'utilisateur saisit un mot nouveau, ce qui exclut cette approche. Nous avons donc opté pour une prédiction indépendante du lexique et choisi le modèle statistique n-gramme. La probabilité d'apparition des lettres (et du caractère « fin de mot » symbolisé par l'espace) est alors estimée à partir des n-1 dernières lettres du mot en cours. Ce modèle a pour avantage d'être robuste et est couramment utilisé en ingénierie linguistique (étiquetage morphosyntaxique, reconnaissance vocale, etc.).

Bases théoriques

Ce modèle est issu de la théorie de l'information. Celle-ci suppose que la probabilité d'apparition d'un événement dépend de toute séquence passée. La probabilité d'occurrence de la séquence L_1, \dots, L_m sera ainsi décrite comme le produit de probabilités conditionnelles :

$$\text{PROB}(L_1, \dots, L_m) = \prod_{i=1}^{i=m} \text{PROB}(L_i | L_1, \dots, L_{i-1})$$

Compte tenu du nombre de séquences possibles, ce résultat n'est pas directement exploitable. L'estimation de ces probabilités nécessiterait en effet un trop grand nombre de données. Une bonne approximation consiste à se limiter à un contexte de n observations. Ce modèle est appelé n-gramme. Il exprime le fait que la

probabilité d'apparition d'une séquence peut être donnée de manière satisfaisante par une combinaison de probabilités basée sur les n-1 observations précédentes :

$$\text{PROB}(L_1, \dots, L_m) \cong \prod_{i=1}^{i=m} \text{PROB}(L_i | L_{i-(n-1)}, \dots, L_{i-1})$$

L'approximation est d'autant meilleure que le paramètre n est élevé.

Dans le cadre de notre prédiction de lettre, les observations sont les lettres et la séquence L_1, \dots, L_{m-1} correspond au début du mot en cours. Cette séquence est connue et notre problème consiste plus simplement à classer les lettres en fonction des probabilités $\text{PROB}(L_m | L_{m-(n-1)}, \dots, L_{m-1})$, ou encore des fréquences d'occurrence $\text{REQ}(L_{m-(n-1)}, \dots, L_m)$ observées sur corpus.

Fiabilité des données

La nature statistique du modèle pose le problème de la représentativité des données utilisées pour l'estimation des probabilités. Ce problème est connu sous le nom d'éparpillement des données (Allen, 1997) : un grand nombre de données est concentré sur un faible nombre de cas. Ceci implique en particulier que les cas non observés se voient affectés d'une probabilité nulle biaisant l'estimation globale. Parmi les techniques employées pour obtenir des estimations fiables, nous avons adopté une technique de repliement : le module de prédiction classe les lettres en fonction des probabilités de l'ordre n non nulles puis, pour les lettres restantes, il fait appel aux probabilités sur l'ordre n-1 et ainsi de suite jusqu'à un classement complet.

Corpus de données

Le corpus utilisé pour estimer les probabilités est celui du journal Le Monde². Les données représentent tous les articles du journal sur une période de 5 ans (1995 à 1999). La prédiction ne portant que sur les mots, seules les lettres et l'espace ont été conservés, soit 65 caractères (cette cardinalité élevée s'explique par la présence des lettres étrangères). Nous avons également converti les majuscules en minuscules. Le corpus ainsi obtenu contient 600 millions de caractères, ce qui correspond à 110 millions de mots pour 400 000 formes fléchies. Outre sa grande taille, ce corpus présente l'avantage de permettre une estimation de la fréquence des mots (et des séquences de lettres) en usage dans la langue. Le problème de la représentativité du vocabulaire est, a priori, limité car SibyLettre s'applique aux séquences de lettres.

Pour les évaluations quantitatives, chacune des cinq années a servi de corpus de test (autour de 20% des données), l'apprentissage étant réalisé sur les quatre autres années. Cinq séries de tests ont ainsi été obtenues, ceci afin d'effectuer une validation croisée (Allen, 1997). Les résultats étant très proches, seule la moyenne est présentée par la suite.

Pour la partie évaluation qui suit, nous rappelons que le paramètre n définit la longueur de la séquence et n-1 la taille de la fenêtre contextuelle.

4.2 Evaluation de la prédiction

Perplexité

² « Le Monde » Text Corpus Version 1.0 Years 1995 to 1997 & 1998 and 1999

La mesure de perplexité est couramment utilisée pour évaluer les modèles statistiques. Elle permet de mesurer la complexité de la tâche étant donné un modèle (Boite, 2000) et réciproquement de mesurer l'efficacité d'un modèle pour une tâche donnée. Cette perplexité est définie comme étant 2^H , où H est la mesure d'entropie :

$$H = \sum_h \text{PROB}(h) \sum_{i=1}^N \text{PROB}(L_i|h) \log_2 \text{PROB}(L_i|h)$$

où L_i est une lettre et h une séquence du modèle.

Le Tableau 2 présente les valeurs de perplexité estimées sur le corpus pour différentes valeurs de n .

n	1	2	3	4	5
Perplexité	17,4	10,2	6,9	5,0	4,1

Tableau 2 : Perplexité en fonction de n

La mesure de perplexité est difficile à interpréter. Cependant, l'évolution de la perplexité de 17,4 ($n=1$) à 4,1 ($n=5$) montre que l'efficacité du modèle s'accroît de manière significative avec la taille du contexte.

Ces résultats peuvent également être mis en relation avec la perplexité des modèles de langage utilisant le même modèle statistique (n -gramme de mots). Roukos (1996) cite ainsi des valeurs de perplexité de 247 pour un trigramme de mots ($n=3$) estimé sur de l'anglais général, de 105 en style journalistique et de 20 dans le domaine de la radiologie. Ceci rappelle que la prédiction de mot est une tâche complexe offrant de nombreux choix possibles et laisse à supposer que la prédiction de lettre peut être envisagée comme une bonne aide complémentaire (rôle défini pour SibyLettre).

Prédiction moyenne (PM)

La pertinence de la prédiction délivrée par le modèle peut également être montrée par une mesure de prédiction moyenne (PM). Celle-ci consiste à relever, lors de la phase de test, le rang de la lettre attendue dans la liste proposée. Par exemple, avec « COM » le système délivre la liste « M, P, B, ... ». Si le mot à saisir est « COM...PTER », la lettre attendue est P et le rang dans la liste est 2. Un calcul de moyenne délivre la prédiction moyenne :

$$PM = \frac{1}{N} \times \sum_{i=1}^{i=N} \text{Rang}(i)$$

où N est le nombre d'observations et $\text{Rang}(i)$ le rang obtenu à l'observation i .

La Figure 5 donne la prédiction moyenne obtenue pour les différentes valeurs de n ainsi qu'une valeur limite (cf. infra) :

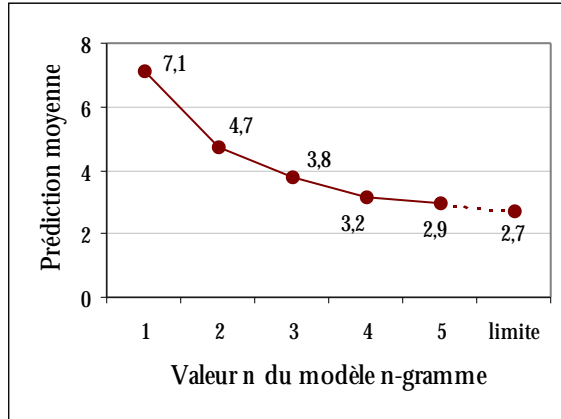


Figure 5 : Prédiction moyenne en fonction de n

Si l'on rappelle que pour $n=1$, le modèle n -gramme est basé sur la seule fréquence des lettres dans la langue (et ne tient donc pas compte du contexte), on peut constater que l'ordre fréquentiel est loin d'être optimal. La prédiction moyenne varie ainsi de 7,1 pour $n=1$ à 2,9 pour $n=5$, soit un gain de 59 %. Autrement dit, l'utilisation du contexte gauche du mot en cours permet d'améliorer considérablement la prédiction utilisée par le clavier simulé.

On peut également remarquer que le gain obtenu pour chaque valeur de n supplémentaire décroît rapidement. Ce gain est de 2,4 entre $n=1$ et $n=2$ et seulement de 0,3 entre $n=4$ et $n=5$. Ceci s'explique par la décroissance rapide du nombre d'occurrences des mots en fonction de leur longueur (Tableau 3), décroissance qui limite l'influence d'une prise en compte d'un contexte plus long. En effet, pour deux valeurs m et $m+1$ de n , les prédictions délivrées par le modèle sont identiques pour les lettres dont la position est inférieure à m . Par exemple, pour $m=4$ et $m+1=5$, les prédictions sont identiques sur les trois premières lettres des mots soit 50,3 % des lettres (Tableau 4). Le gain n'est donc réalisé que sur 49,7 % des lettres.

Longueur	1	2	3	4	5
% mots	9,0	21,9	13,4	10,0	8,9
% cumulé	9,0	30,9	44,3	54,3	63,2

Tableau 3 : Occurrences des mots en fonction de la longueur

Position	1	2	3	4	5
% lettres	17,3	17,3	15,7	12	9,6
% cumulé	17,3	34,6	50,3	62,3	71,9

Tableau 4 : Occurrences des lettres en fonction de la position dans le mot

La remarque précédente nous permet également de calculer une prédiction moyenne limite que le modèle peut au mieux approcher. Pour toutes les lettres dont la position est inférieure au n du meilleur modèle (dans notre cas 5), on utilise les prédictions de ce modèle ; pour les autres lettres, on suppose que la prédiction est idéale (rang égal à 1). La prédiction moyenne obtenue pour $n=5$ donne une valeur limite de 2,7. On remarquera que le résultat obtenu pour $n=5$ (2,9) est déjà très proche de cette valeur limite.

Prédiction moyenne en fonction de la position dans le mot (PMP)

Dans le cadre du projet Sibylle, en dehors des mots inconnus, la prédiction de lettre sera utilisée pour la saisie du début des mots. Ceci correspond à un contexte faible pour la prédiction de lettre. Aussi avons nous cherché à évaluer la perte de performances observée sur le début des mots.

Au calcul de prédiction moyenne défini précédemment, le paramètre « position de la lettre dans le mot » est ajouté. Sur l'exemple « COM...PTER » avec la liste « M, P, B, ... », le rang de la lettre dans la liste est 2 et la position dans le mot 4. Le calcul de moyenne est effectué pour chaque position j :

$$PMP(j) = \frac{1}{N} \times \sum_{i=1}^{i=N} \text{Rang}(i, j)$$

où N est le nombre d'observations et Rang(i, j) le rang obtenu à l'observation i pour la position j.

Les résultats sont donnés par la Figure 6 pour les 5 premières lettres des mots et pour les valeurs de n de 1 à 5. Par exemple, pour la 4^e lettre des mots, la PMP est de 7,3 avec n=1 (1^{ère} série, en haut) ; pour cette même position, la PMP est de 1,8 avec n=5 (dernière série, en bas).

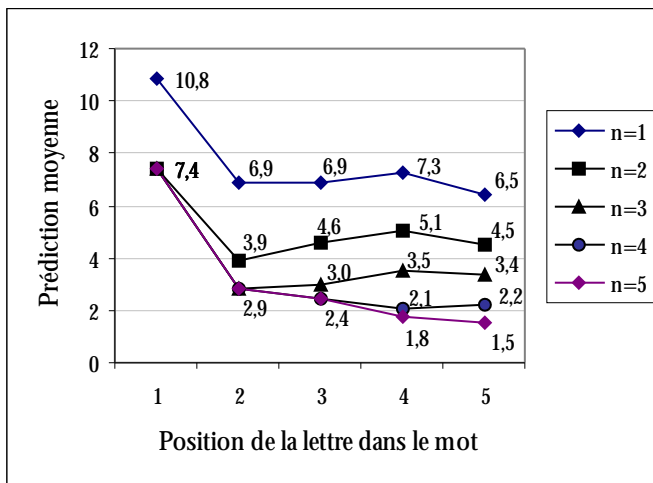


Figure 6 : Prédiction moyenne en fonction de n et de la position de la lettre dans le mot

L'analyse de chaque série de données (chaque valeur de n) permet de constater que la principale difficulté réside dans la prédiction de la première lettre. On observe un écart net entre la prédiction moyenne de la première lettre des mots et les suivantes ($PMP(1)=7,4$ et $PMP(2)=2,9$ pour $n>2$). Nous reviendrons sur cette remarque dans les perspectives pour proposer une ébauche de solution à ce problème. Notons cependant qu'une prise en compte du contexte (ici le fait que l'on est en début de mot) par notre système permet tout de même une amélioration de la prédiction par rapport à l'ordre fréquentiel statique, la PMP passe en effet de 10,8 (1^{ère} lettre des mots, n=1) à 7,4 (n>1), soit un gain de 31 %.

A l'inverse, la prédiction est efficace dès la deuxième lettre. Ceci permet d'envisager une collaboration entre les deux systèmes de prédiction (lettre et mot)

sur les mots courts. Nous reviendrons également plus en détail sur cette remarque dans la partie perspectives.

5 Clavier dynamique

Les logiciels de clavier simulé habituels ne disposent pas de prédiction dynamique des lettres. Avec la prédiction contextuelle (réalisée après chaque saisie), l'ordre des lettres devient variable et implique une interface gérant cet aspect dynamique. L'ajout au clavier simulé d'une liste de lettres affichant les premières propositions serait envisageable. Afin de ne pas surcharger l'interface, nous avons opté pour un système de clavier dynamique. Sur les 65 caractères prédits, nous avons choisi de ne présenter que les lettres de la langue française.

5.1 Fonctionnement du clavier dynamique

Après chaque saisie, le début du mot en cours est fourni au module de prédiction. En retour, celui-ci délivre la liste des lettres classées en fonction de leur probabilité d'apparition. La liste contient également un caractère « fin de mot », symbolisé par l'espace. Le clavier simulé est alors réorganisé pour refléter la prédiction, le curseur remis en position initiale (en haut à gauche du clavier) et le défilement automatique reprend.

Les lettres accentuées sont traitées comme les autres. Elles bénéficient de la prédiction et sont naturellement classées en fonction de leur probabilité d'apparition.

Le balayage utilisé est le défilement linéaire. Ce balayage est nécessité par l'aspect dynamique qui rend le défilement ligne / colonne inadapté. En effet, en terme de charge cognitive, le défilement LC impose de connaître à l'avance la ligne sur laquelle se trouve la lettre désirée, ce qui n'est pas le cas ici.

Le Tableau 5 et la Figure 7 présentent une simulation de la saisie des premières lettres du mot « COMP...TER ». Pour la première lettre, le contexte est vide (\emptyset), la liste proposée est D, L, P, A, E, C, ... le curseur doit défiler 6 fois pour atteindre la lettre désirée. Après cette première lettre, la saisie est plus rapide. Avec le contexte C, la lettre présentée en premier est la lettre O souhaitée. Un seul défilement et l'utilisateur n'a plus qu'à valider (la remise en position initiale du curseur est comptée comme un défilement). Les deux lettres suivantes sont saisies presque aussi rapidement, en deux défilements.

Contexte	Propositions	Saisie
\emptyset	DLPAEC...	C
C	OEH...	O
CO	NMU...	M
COM	MPB...	P

Tableau 5 : Contexte et propositions pour la saisie de COMP...TER

Lettres	Lettres
d l p a e c s	o e h a r i _
m r f i t u q	l u é y f g n
n o à v é b j	d s c m p è v
g h y k w z x	b t j z q w k
ç è ù _	x à ç ù

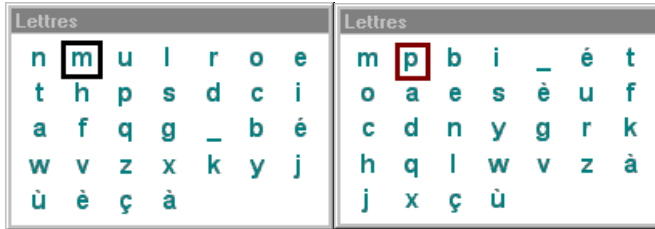


Figure 7 : Exemple de réorganisation dynamique sur le début de mot COMP...TER

Sur cet exemple, la saisie a été réalisée successivement en 6, 1, 2, 2 défilements. Ces valeurs sont à rapprocher avec celles obtenues théoriquement par la PMP (pour $n=5$, les résultats donnaient : $PMP(1..4)=7,4, 2,9, 2,4$ et $1,8$).

5.2 Evaluation du clavier dynamique

Après l'évaluation des capacités prédictives du modèle n-gramme, nous avons tout naturellement cherché à mesurer le gain apporté par cette prédiction sur le temps de sélection. A cet effet, nous avons comparé les différents modes de sélection possibles.

Paramètres

Ces modes de sélection peuvent être caractérisés par deux principaux paramètres. Ces paramètres sont le type de balayage (défilement linéaire ou défilement ligne / colonne) et l'organisation des lettres sur le clavier simulé. Nous avons distingué trois types d'organisation :

- L'ordre aléatoire (noté 0) ne tient compte ni de la prédiction ni de la fréquence des lettres. A ce titre, l'ordre alphabétique et l'ordre « azerty » entrent dans cette catégorie.
 - L'ordre fréquentiel est organisé en fonction de la fréquence des lettres observées dans la langue, indépendamment de leur contexte d'apparition. Cet ordre correspond au cas $n=1$ du modèle n-gramme.
 - Enfin, l'ordre dynamique, réorganisé après chaque saisie (cas $n>1$)
- Pour le défilement LC, la matrice des touches est supposée carrée.

Métrique

La métrique utilisée est le nombre de défilements moyen, défini dans la section 3. Nous rappelons ici l'équation :

$$NDM = \sum_{i=1}^{i=T} \text{PROB}(i) \times \text{DEF}(i)$$

$\text{PROB}(i)$ est la probabilité d'apparition de la touche i ,

$\text{DEF}(i)$ est le nombre de défilements nécessaires pour accéder à la touche i . Il dépend de la position de la touche dans la matrice et du type de balayage.

Notons que dans le cas du défilement linéaire, le curseur défile sur les lettres classées selon leur probabilité d'apparition, le nombre de défilements moyen est alors égal à la prédiction moyenne calculée en 4.2.

Résultats

Les résultats sont donnés dans le Tableau 6 avec les deux balayages et les différentes organisations (les ordres statiques et dynamiques ont été regroupés). La dernière colonne notée Val rappelle le nombre de validations nécessaires pour

sélectionner une lettre. Par exemple, on peut lire que le nombre de défilements moyen pour le défilement LC avec une organisation fréquentielle (colonne n=1) est de 4,3.

	statique		dynamique				Val
	0	n=1	n=2	n=3	n=4	n=5	
Défilement linéaire	33	7,1	4,7	3,8	3,2	2,9	1
Défilement LC	9	4,3	-	-	-	-	2

Tableau 6 : Nombre de défilements moyen en fonction du balayage et de l'ordre des lettres

Ces résultats permettent de montrer l'intérêt de la prédiction de lettre. Avec n=5, le NDM obtenu par SibyLettre est de 2,9. Par rapport aux systèmes habituels, comme le logiciel présenté dans la Figure 3 (ordre 0 avec défilement LC, NDM=9), le gain est important. Par rapport au mode de sélection optimal sur clavier statique (ordre fréquentiel avec défilement LC), le gain est de 32 % (2,9 contre 4,3). Ce gain est d'autant plus appréciable que la sélection nécessite deux fois moins de validations.

Ces résultats permettent également de mettre en évidence certaines observations de la section 3. Ainsi, il apparaît que les ordres alphabétique et « azerty » habituellement proposés ne sont pas adaptés (ordre 0 dans le Tableau 6, résultats en italique). Dans le cas d'un clavier statique, cette synthèse met également en évidence l'efficacité du défilement LC sur le défilement linéaire. Ainsi, pour l'ordre fréquentiel (colonne n=1), le nombre de défilements moyen est de 7,1 pour le défilement linéaire contre 4,3 pour le défilement LC, soit un gain de 42 %. Ce gain permet de justifier la validation supplémentaire introduite par le défilement LC dans le cas des ordres statiques mais reste cependant moins efficace que le système SibyLettre.

6 Application Sibylle

La phase de validation par les utilisateurs est une étape nécessaire de l'évaluation (Preece, 1994). Les résultats théoriques ne se traduisent en effet pas toujours par des gains effectifs. Dans le cas de SibyLettre, cette évaluation est même indispensable à cause de l'aspect dynamique susceptible d'accroître la charge cognitive. C'est pourquoi une application a été réalisée en vue de l'évaluation de SibyLettre par des personnes handicapées.

De manière plus générale, l'application Sibylle doit nous permettre d'évaluer les modules linguistiques (pertinence des prédictions en fonction de la saisie effective) et les solutions ergonomiques mises en œuvre.

Une partie de la conception de l'application Sibylle a été réalisée en concertation avec le centre de Kerpape. En particulier, nous avons pu comparer les divers logiciels disponibles à Kerpape, et observer leur utilisation par plusieurs personnes handicapées et dans différents contextes (communication courante, école de Kerpape, ergothérapie) (Scapin, 1990).

Nous avons réalisé une application opérationnelle et conviviale et ce pour plusieurs raisons. D'abord, il convient de rappeler que quelque soit l'outil de communication, la composition de texte demande aux personnes handicapées un effort. Ces personnes sont généralement habituées à un système de communication et l'utilisation d'un autre système d'aide demande une phase d'apprentissage. Cet effort supplémentaire sera d'autant mieux accepté que l'interface est conviviale et intègre des commodités et des fonctionnalités dont l'utilisateur a l'habitude. Ensuite,

il existe de nombreux logiciels dans le commerce et certaines personnes handicapées utilisent couramment l'ordinateur, elles ont une certaine exigence quant aux interfaces proposées. Enfin, l'expérience de Kerpage a montré qu'une interface réalisée juste dans le cadre d'une évaluation peut révéler des fautes d'ergonomie qui vont fortement pénaliser l'évaluation.

6.1 Présentation de l'interface

L'interface présentée (Figure 8) est celle actuellement utilisée au centre de Kerpage.

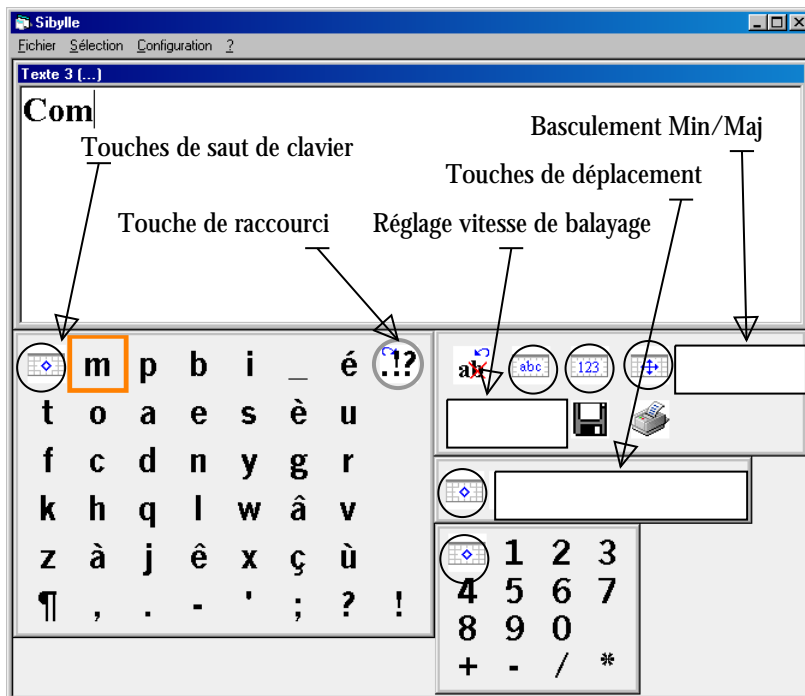


Figure 8 : Interface de Sibylle et quelques définitions de touche

L'interface est composée d'une zone d'édition de texte et du clavier simulé.

Le clavier simulé est composé de pavés de touches. L'objectif de cette organisation est toujours d'optimiser le nombre de défilements en balayage automatique. Le principe est ici de regrouper les touches par type en différents pavés (lettres, chiffres, touches de déplacement, fonctions générales). Le basculement d'un pavé à l'autre est réalisé par des touches de « saut de clavier ».

Selon le même principe de « saut », l'accès aux ponctuations (en bas du clavier de lettres) est réalisable par une touche de raccourci (Shneiderman, 1998) située en haut à droite du clavier de lettres. En validant cette touche, le curseur se positionne directement sur la première touche des ponctuations.

Nous avons également implanté un « clic long ». Il permet d'offrir un degré de liberté supplémentaire pour les personnes handicapées à même de contrôler leur geste. Le « clic long » est réalisé en maintenant enfoncé plus longtemps le bouton de validation (son activation et la durée sont paramétrables). Lorsque la durée d'appui du « clic long » est atteinte, le curseur change de couleur. En défilement linéaire, la fonction associée au clic long est le retour en position initiale du curseur.

La liste suivante présente quelques unes des fonctionnalités de Sibylle :

- Adaptation à l'interface matérielle : le logiciel s'adapte à l'interface matérielle d'entrée. Les deux modes d'interaction proposés sont le mode « souris », et le mode « balayage automatique ». En mode « souris », le logiciel propose un suivi de curseur (mise en relief de la touche sous le curseur souris). Pour le balayage automatique il est possible d'utiliser le défilement linéaire ou LC. Les trois ordres alphabétique, fréquentiel et dynamique sont implantés.
- Configuration de l'interface matérielle : le geste dirigeant l'interface matérielle est souvent mal contrôlé. Il est possible de corriger certaines erreurs à l'aide de paramètres : la durée minimale d'enfoncement des touches (pour éviter les appuis intempestifs), la durée minimale entre deux clics (évite des successions de clics non désirées), la vitesse de défilement du balayage. Actuellement ces paramètres doivent être réglés manuellement par un assistant (sauf la vitesse de défilement, réglable par la personne handicapée). Notons cependant qu'il existe des jeux de tests pour régler semi-automatiquement ces paramètres (Noirhomme, 2000) ; Le système EDITH (Pino, 2000) propose une vitesse de balayage adaptative.
- Configuration de la police de caractère : les troubles de la vue sont fréquents (en particulier chez les personnes âgées), la police de caractère des lettres du clavier est donc configurable (taille, style, casse et couleur).

De manière générale, l'interface est largement paramétrable (polices, couleurs, taille et position des claviers). Il est également possible de modifier les claviers, les touches, par l'intermédiaire de fichiers de configuration.

6.2 Evaluation qualitative

Cette dernière évaluation est consacrée à l'évaluation de l'interface auprès de personnes handicapées. L'évaluation a porté sur le clavier dynamique, le découpage en pavés de touches et le clic long. Cette évaluation est essentiellement qualitative.

L'application a été testée avec trois enfants IMC, à l'école du CMRRF de Kerpape. Ces enfants ont l'habitude d'utiliser les aides logicielles.

Deux des enfants ont accueilli très favorablement le logiciel avec le clavier dynamique. Le troisième a préféré retourner à un mode de sélection statique. Ce dernier résultat était cependant prévisible car cet enfant souffre de problèmes de poursuite oculaire ; ce problème de vue rend l'aspect dynamique très perturbant.

Pour les deux premiers enfants, les résultats sont très positifs. La phase d'apprentissage à cette nouvelle aide a été simple et rapide. Le principe du saut de touche, du saut de clavier et du clic long n'ont pas posé de problème particulier. Le gain apporté par Sibylle est essentiellement apprécié en termes de « confort », le défilement linéaire et son unique validation est particulièrement apprécié. Dans le cadre de l'école, le personnel encadrant a constaté que ces enfants composent plus de textes et font moins de fautes d'orthographe.

L'évaluation se poursuit actuellement avec d'autres personnes. L'application va également être utilisée en ergothérapie. La prochaine version de Sibylle implantera un mécanisme de boîte noire qui permettra de collecter des données en vue d'une évaluation quantitative.

7 PERSPECTIVES

7.1 Clavier dynamique avec souris

La bonne adaptation des utilisateurs au clavier dynamique (dans le cas où il est accepté), permet d'envisager d'étendre ce principe à des modes offrant plus de

degrés de liberté (souris, joystick), en réorganisant les lettres les plus probables autour de la lettre en cours. Pour les personnes manipulant ces interfaces physiques avec difficulté, ceci permettrait en effet de minimiser l'effort de déplacement.

7.2 Amélioration de la prédiction

La suite de notre projet Sibylle est d'intégrer une prédiction de mot. Dans cette perspective, nous envisageons d'améliorer la prédiction de la première lettre. La prédiction pourra se baser sur les prédictions de SibyMot ou à l'aide de connaissances de plus haut niveau, comme la prédiction de la prochaine classe syntaxique. On peut ainsi remarquer que pour la première lettre des mots (contexte vide), les premières lettres proposées par la prédiction sont d et l et correspondent aux mots les plus fréquents : de, le (et leurs formes fléchies).

7.3 Collaboration prédiction de lettre et de mot

La prédiction de lettre est très efficace pour les mots dont la fréquence est élevée dans la langue. Typiquement ces mots sont les mots grammaticaux qui sont généralement courts. Nous tiendrons compte de cette remarque pour l'affichage des mots proposés par le module de prédiction. Il est en effet inutile d'afficher un mot court dans la liste des mots (dont l'accès est pénalisant), alors que sa saisie complète lettre par lettre peut être aussi voire plus rapide.

8 CONCLUSION

Nous sommes partis du constat que la prédiction de mot est une tâche difficile qui laisse encore un grand nombre de lettres à saisir par l'utilisateur. Cette saisie est particulièrement longue dans le cas d'une sélection avec défilement automatique.

Nous avons montré qu'une prédiction de lettre avec modélisation statistique par n-gramme obtenait de bons taux de prédiction. Nous proposons ainsi d'intégrer au clavier simulé une prédiction de lettre dynamique. Les résultats théoriques obtenus affichent un gain de 32 % en temps de saisie et 50 % en nombre de validations, par rapport à une sélection utilisant un défilement ligne / colonne avec organisation fréquentielle.

9 REFERENCES

- Abraham M. (2000). Reconstruction de phrases oralisées à partir d'une écriture pictographique, Handicap 2000, pp 151-156, Paris.
- Allen, J. (1997). Natural Language Understanding, Allen, J., Chapter VII. Benjamins Cummings éditions, 1997.
- Boissière P. (2000). VITIPI : Un système d'aide à l'écriture basé sur un principe d'auto-apprentissage et adapté à tous les handicaps moteurs, Handicap 2000, pp 81-86, Paris.
- Boite, R. (2000). Traitement de la parole, Boite R. et al. Presses polytechniques et universitaires romandes, 2000.
- Cantegrit, B. (2001). Réflexions sur l'aide à la communication des personnes présentant un handicap moteur, Cantegrit B., Toulotte J.-M., TALN 2001, vol 2, pp 193-202, Tours.
- McCoy, K.F. (1995). Somme applications of natural language processing to the field of augmentative and alternative communication, McCoy, K. F., Demasco, P., in

- Proceedings of the IJCAI'95 Workshop on Developing AI Applications for Disabled People, pp 97-112, Montreal, Canada.
- Le Pévédic, B. (1997). Prédiction morphosyntaxique évolutive, Le Pévédic B., Thèse de doctorat, Ecole doctorale Sciences pour l'ingénieur de Nantes, 1997.
- Maurel, D. (2001). HandiAS : un système multilingue pour l'aide à la communication des personnes handicapées, Maurel D., Rossi N., Thibault N., TALN 2001, vol 2, pp 203-212, Tours.
- Ménier, G. (2001). Système adaptatif de prédiction de texte, Ménier G., Poirier F., TALN 2001, vol 2, pp 213-222, Tours.
- Noirhomme-Fraiture, M. (2000). A Laboratory of Ergonomic Analyses for Children Suffering form Cerebral Palsy, in Tools for Working with Guidelines, Noirhomme-Fraiture M., Charrière C., Vanderdonck J., TFWWG 2000, pp 35-49.
- Pino, P. (2000). EDITH : Adaptation automatique du temps de défilement aux caractéristiques et intentions de l'utilisateur, Handicap 2000, pp 125-130, Paris.
- Preece, J. (1994). Human-Computer Interaction, Preece J., Rogers Y., Sharp H., Benyon D., Holland S., Carey T., Addison-Wesley ed, Wokingham, 1994.
- Ricco, X. (2001). Vers un logiciel multilingue et gratuit pour l'aide aux personnes handicapées de la parole : HOOK (une interface du projet W), Ricco X., Dutoit T., TALN 2001, vol 2, pp 223-232, Tours.
- Richardet, N. (1998). Composition de phrases assistée - Un système d'aide à la communication pour handicapés, Richardet, N., Thèse de doctorat, Université de la Méditerranée, 1998.
- Roukos, S. (1996). Survey of the state of the Art in Human Language Technology, Chap. 1.6, Cambridge University Press, 1996. <http://cslu.cse.ogi.edu/HLTSurvey/ch1node8.html>.
- Scapin, D. L. (1990). Des critères ergonomiques pour l'évaluation et la conception d'interfaces, Scapin D. L., actes du XXVIème Congrès de la SELF, Montréal.
- Shneiderman B. (1998). Designing the User Interface : Strategies for Effective Human-Computer Interaction, Schneiderman B., Addison-Wesley 3rd ed., 1998.
- Vaillant, P. (1997). PVI : Système de traduction d'icônes en langue, Interaction entre modalités sémiotiques : de l'icône à la langue, Vaillant, P., Thèse de Sciences Cognitives, Université Paris-XI, Orsay, 1997.