

VITIPI : Un système d'aide à l'écriture basé sur un principe d'auto-apprentissage et adapté à tous les handicaps moteurs.

Philippe BOISSIERE

Daniel DOURS

IRIT – UPS
118 Route de Narbonne
31 062 TOULOUSE Cedex (France)

boissier@irit.fr

dours@irit.fr

RESUME :

Cet article tente d'apporter une solution au problème de la lenteur de saisie de texte par une personne handicapée des membres supérieurs. Les difficultés d'utilisation des systèmes existants et leurs insuffisances d'adaptation nous ont conduits à développer le système VITIPI. Après un rapide survol des problèmes rencontrés, nous expliquerons le principe de la modélisation du système VITIPI, puis nous aborderons les problèmes de mise en œuvre. Nous indiquerons enfin ses performances et nous montrerons qu'il peut être utilisé par toute personne, quel que soit le type de son handicap et le matériel qu'elle utilise.

MOTS CLES : Communication enrichie, assistance à l'écriture, apprentissage par transducteur

INTRODUCTION

Pour toute personne vivant dans notre société, l'écriture occupe une grande place. Au travail, au quotidien ou dans les loisirs, nul ne peut plus (ou très difficilement) s'en passer. Lorsque la personne est handicapée, l'écriture peut parfois servir à compenser (voire à remplacer) la communication verbale [1], à condition que la personne puisse écrire. Utiliser un papier et un crayon est certainement le moyen le plus simple, mais lorsque la personne est tétraplégique, l'écriture, passe aujourd'hui par des moyens informatiques. Parmi ces techniques, on peut maintenant utiliser la reconnaissance vocale lorsque la personne n'a pas de trouble de l'élocution. Si tel n'est pas le cas, il faut passer par la saisie du texte lettre après lettres. Or cette saisie peut encore poser deux types de problèmes : d'une part, l'outil de saisie n'est pas totalement adapté au handicap de la personne, d'autre part, même lorsqu'il est bien adapté, il subsiste encore une certaine lenteur à l'écriture à laquelle il faudra remédier.

Après une présentation rapide des problèmes rencontrés avec les systèmes existants, nous décrirons les principes du système VITIPI. Nous expliquerons comment nous l'avons modélisé en utilisant à la fois un modèle de n-grammes et de transducteur. Nous aborderons ensuite les problèmes de mise en œuvre de ce système. Enfin, nous indiquerons ses performances et nous montrerons que ce système peut être utilisé par toute personne, quel que soit le type de son handicap, et le matériel qu'elle utilise.

PROBLEMATIQUE

Notre propos n'est pas de faire ici une présentation exhaustive des systèmes d'aide à la communication ou à l'écriture qui ont été réalisés depuis près de 15 ans. Nous nous contenterons de mentionner les réalisations qui nous paraissent les plus connues parmi lesquelles nous mettrons en exergue les difficultés d'utilisation que nous avons essayées d'éviter pour notre système.

Les systèmes existants

Il nous paraît important de ne pas obliger l'utilisateur à apprendre un système de codage ou d'abréviations. Même s'il existe un programme pour traduire la sténotypie en français [2], encore faut-il pouvoir apprendre ce codage et utiliser le clavier spécialisé qui va avec. Un autre écueil à éviter est de proposer à l'utilisateur une liste de mots [3], [4], même s'ils ont été préalablement sélectionnés par des méthodes très performantes [5], [6], [7], [8], ils perturbent quand même la pensée de l'utilisateur dont l'attention peut s'égarer. De plus, il est assez difficile pour ces types de systèmes de pouvoir prédire des mots nouveaux ou altérés. Il est également important de souligner que la plupart de ces systèmes sont tributaires d'un traitement de textes, et s'adaptent très rarement à d'autres logiciels : tableurs, base de données, ..., ou autres outils multimédia (CD-ROM, Internet ...)

Le système VITIPI

Le système VITIPI tente d'apporter des solutions aux problèmes évoqués. Il produit un texte sans avoir à taper toutes les lettres qui le composent. Il prédit des parties de mots et les affiche dès qu'il n'y a plus d'ambiguïté. Ce procédé évite de présenter à l'utilisateur des mots qui n'ont rien à voir avec ceux qu'il désire. De plus le système est doté de procédures d'inférences qui permettent de corriger *en ligne* des fautes de frappes ou d'orthographe, et de prédire des mots nouveaux qui n'appartiennent pas au vocabulaire de base. Lorsqu'un mot nouveau apparaît, il est automatiquement intégré dans le vocabulaire de base et bénéficie des fonctionnalités offertes par les procédures d'inférences. VITIPI n'est pas tributaire d'un traitement de textes et peut fonctionner sous n'importe quelle application Window.

Lorsque l'on ne tient pas compte des mots qui précèdent, VITIPI peut prédire 26 % de lettres sur un vocabulaire

de 5.930 mots. Il corrige également 72 % de fautes de frappes et 75 % de fautes d'orthographe.

Cependant, ces résultats de prédiction, aussi intéressants soient ils, peuvent encore être améliorés. En effet, il existe un lien entre la taille du vocabulaire et le rendement du système (*nombre de lettres prédites par rapport au nombre total de lettres*). Plus le vocabulaire est grand, plus le système va rencontrer d'ambiguïtés. Il devra donc attendre les lettres de l'utilisateur pour lever ces ambiguïtés, ce qui fera baisser le rendement. Inversement, si le vocabulaire est petit, le rendement sera important. L'augmentation du rendement passe donc par une réduction de la taille du vocabulaire, mais cela ne doit pas pour autant pénaliser l'utilisateur. Il est intéressant de constater qu'après une succession de mots, seul un ensemble de mots relativement restreints peuvent apparaître. Nous avons donc conçu un système qui utilise cette propriété pour créer l'ensemble des mots susceptibles d'être écrits à un instant donné. Dans le modèle que nous avons adopté, cette création se fait de façon automatique.

MODELISATION DU SYSTEME

La construction de ce système repose sur deux processus qui coopèrent. Le premier processus sélectionne la liste des mots qui sont susceptibles d'être écrits à un instant donné afin de réduire le vocabulaire ; pour le réaliser, nous avons employé le modèle des n-grammes. Le second processus permet la prédiction de lettres dans les mots ainsi sélectionnés. Pour ce faire, nous avons utilisé la modélisation par transducteur.

Sélection du vocabulaire (Modèle n-gramme):

La sélection des mots du vocabulaire susceptibles d'être écrits peut se faire par deux approches différentes. La première approche présuppose l'existence de règles de syntaxe et de sémantique, clairement définies de façon exhaustive et non ambiguë. Ces règles, grâce à des outils développés en Intelligence Artificielle, comme les systèmes experts ou autres systèmes de production, permettent de générer en partie le langage. C'est semblait-il l'option choisie par les projets Kombe [9], [10] et Illico [11]. La seconde approche part au contraire du principe que les règles de syntaxe et de sémantique sont implicitement contenues dans les phrases. Il suffirait donc d'introduire dans le système une grande quantité de phrases (*correctes du point de vue syntaxique et sémantique*) et que celui-ci synthétise de lui-même les règles syntactico-sémantiques qui ne lui ont pas été fournies de façon explicite.

La première approche est selon nous très lourde à mettre en oeuvre et peu évolutive. Difficile à mettre en oeuvre car il faut pouvoir disposer de la totalité des règles syntactico-sémantiques destinées à modéliser la partie du langage qui nous intéresse. Peu évolutive car quand l'utilisateur veut changer de langue, que son langage évolue, ou tout simplement quand un mot nouveau apparaît, le système de règles est à revoir, il faut donc

pouvoir le mettre à jour et assurer sa cohérence. Ceci se fait rarement de façon automatique. La plupart du temps, cela requiert quelques compétences en intelligence artificielle que l'utilisateur n'a pas.

Nous avons donc choisi d'orienter notre recherche en nous servant de la seconde approche qui, bien qu'elle pose des problèmes sur le choix et la représentativité du corpus, paraît intéressante.

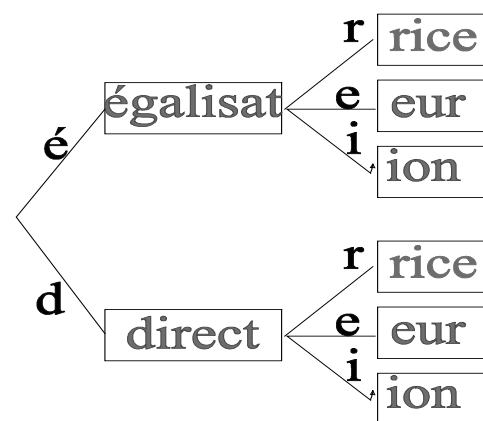
Pour modéliser cette approche, nous avons utilisé le modèle des *N-grammes*. Ce type de modèle a été utilisé pour la première fois en reconnaissance de la parole [12]. Grâce à ce modèle, lorsque l'on a écrit un mot, on peut retrouver dans le corpus l'ensemble des mots qui le suivent. Si l'on tient compte des deux mots déjà écrits, l'ensemble des mots qui peuvent suivre se réduit. En prenant en considération les *n-1* derniers mots, le nombre de mots qui peuvent suivre se réduit encore pour ne plus être réduit qu'à un seul, voire aucun. Cet ensemble de mots qui suivent une chaîne de *n-1* mots peuvent être sélectionnés pour constituer l'ensemble des mots susceptibles d'être écrits à un instant donné. Voyons à présent le modèle qui nous permet d'aider à la prédiction des lettres dans les mots ainsi sélectionnés.

Prédiction des lettres sélectionnées (Modélisation par transducteur)

Afin que VITIPI puisse afficher automatiquement les lettres prédites en fonction de celles entrées par l'utilisateur, nous avons modélisé la base lexicale par un transducteur. Sans entrer dans les détails de sa création et de sa minimisation qui ont déjà fait l'objet de plusieurs publications ([13], [14], [15], [16]), rappelons qu'un transducteur est composé de transitions, chacune comprenant : un état de départ et un d'arrivée, une entrée et une sortie. L'entrée de la transition correspond à la lettre frappée par l'utilisateur, et la sortie à celles que le système doit afficher. A titre d'exemple, les mots :

Directeur	Direction	Directrice
Egalisateur	Egalisation	Egalisatrice

engendrent le transducteur suivant (les lettres en gras sont automatiquement affichées par le système) :



Coopération des deux modèles :

Nous pouvons remarquer que l'on peut faire cohabiter dans le système une multitude de *transducteurs* (ou *sous-transducteurs*) contenant des vocabulaires différents. Dans un *sous-transducteur* on peut trouver :

- tous les mots qui suivent un mot donné
- tous les mots qui suivent un couple de mots
- ...
- tous les mots qui suivent une suite de n-1 mots

Pour pouvoir accéder à un sous transducteur donné, il suffit de considérer les mots du texte comme étant les lettres d'un alphabet. A chaque mot du vocabulaire issu du corpus est associé un numéro d'ordre unique. Ces numéros d'ordre sont considérés comme des entrées du transducteur. Ainsi la connaissance des mots écrits précédemment nous conduira au bon *sous-transducteur*.

Nous avons exposé les modèles utilisés pour construire VITIPI. Voyons à présent les problèmes soulevés par la mise en œuvre du système lorsque l'on génère des transducteurs à partir de corpus relativement importants, ou lorsque l'utilisateur écrit des mots ou des phrases qui ne figurent pas dans le corpus d'apprentissage.

MISE EN ŒUVRE DU SYSTEME

On pourrait craindre que la création de systèmes générés à partir de grands corpus occupent une place mémoire relativement importante, d'autant que nous prenons en compte des suites de mots dont la longueur peut aller jusqu'à 10 (*segments de 10 Mots*). Il n'en est rien car d'une part, la création arborescente du transducteur factorise par nature le début des mots ou des phrases qui ont des préfixes identiques, d'autre part, nous effectuons une minimisation qui regroupe la fin des mots ou des phrases qui ont le même suffixe. L'objectif de cette minimisation est bien sûr de réduire la place occupée par le transducteur mais surtout de faire apparaître des structures qui sont identiques pour des fins de mots ou de phrases.

Lorsque l'utilisateur se sert du système, la phrase écrite peut appartenir ou non, au corpus d'apprentissage. Si elle y appartient, le système parcourt *normalement* le transducteur avec les mots fournis par l'utilisateur. Si la phrase n'appartient pas au corpus, alors le système en parcourant le transducteur va rencontrer une transition indéterminée, il devra alors *inférer* une transition.

Il se peut qu'en saisissant la phrase, l'utilisateur écrive un mot qui est inconnu du système. Le mot peut être effectivement un mot nouveau que le système n'a jamais rencontré. Il déclenchera une inférence qui, par *analogie* avec les autres mots existant dans la base, permettra de déterminer la transition et de prédire la fin de ce mot nouveau. Il se peut que le mot soit connu du vocabulaire mais qu'il ait été altéré parce que l'utilisateur a fait une faute de frappe ou d'orthographe. Dans ce cas là, le système va envisager 4 procédures d'inférences qui vont déterminer la transition, corriger la faute et permettre

l'écriture de la fin du mot. Le détail de ces procédures d'inférences, et la façon dont le système en choisit une parmi celles dont il dispose, peut être trouvé dans [13], [14], [15], [16].

Lorsque l'utilisateur écrit son texte, il lui arrive forcément d'écrire des phrases qui n'appartiennent pas au corpus d'apprentissage. Dans ce cas là, deux cas peuvent se produire :

- soit il s'agit d'un mot inconnu du système et l'on est ramené aux cas évoqués précédemment. Si le mot est réellement nouveau, alors ce sera obligatoirement une phrase nouvelle qui une fois terminée, sera intégrée au corpus.
- soit le mot est connu du système, ainsi que le sont tous les mots écrits précédemment, mais l'ordre dans lequel ils apparaissent n'a jamais été rencontré dans le corpus d'apprentissage. Nous avons pour cela développé deux nouvelles procédures d'inférences que nous allons exposer ci dessous.

Imaginons par exemple, que le corpus d'apprentissage contienne les phrases suivantes :

Mardi, le ciel était très sombre.	Hier, le ciel était couvert.
Hier, le ciel était très clair.	Ce matin, le ciel est dégagé.
Nous sommes très heureux de vous voir.	Nous sommes très contents d'être ici.
Ils étaient bien heureux qu'il soit venu.	Ils étaient bien contents de l'apprendre.
Aujourd'hui vous êtes heureux il fait beau.	Aujourd'hui vous êtes contents de lui.
Sont-ils contents des vacances ?	

Si l'utilisateur veut écrire la phrase : « *Ce matin, le ciel était très clair.* ». Lorsqu'il commence à taper : « *Ce matin, le ciel é* », le système se bloque sur la lettre « *é* » car il ne connaît pas de phrase commençant ainsi, même s'il connaît tous les mots de la phrase. Il se trouve face à une transition indéterminée.

La première procédure, que nous appellerons inférence par *réduction*, essaye de retrouver dans le début de phrase que l'utilisateur écrit, un segment de phrase qui soit connu du corpus. Pour cela, il va retirer le premier mot de la phrase et passer le segment de phrase ainsi obtenu dans le transducteur. Dans notre exemple, le segment de phrase obtenu est : « *matin, le ciel é* ».

Si le système ne rencontre pas de transition indéterminée alors l'objectif est atteint, l'indétermination est levée et le système continue d'aider l'utilisateur sans être bloqué. Si par contre il trouve une transition indéterminée, alors cela signifie que le segment de phrase ne figure pas dans le corpus. Sur notre exemple, le système rencontre une transition indéterminée car le segment de phrase « *matin, le ciel é* » ne peut pas être trouvé dans le corpus.

S'il l'on bute toujours sur une transition indéterminée, alors on enlève encore le premier mot de ce segment de phrase pour obtenir le nouveau segment qui a son tour, sera repassé dans le transducteur. Ce processus de troncature est réitéré jusqu'à ce que l'on ne rencontre plus de transition indéterminée ou qu'il ne reste plus qu'un seul mot dans le segment de phrase. Sur l'exemple, nous obtenons le segment de phrase : « *le ciel é* » qui lui figure bien dans le corpus avec la phrase : « *Hier, le ciel était très clair.* ».

Pour illustrer la seconde inférence que nous appellerons inférence des *entrées similaires*, imaginons que l'utilisateur veuille écrire : « *Ils étaient bien content qu'il soit venu.* ». Lorsqu'il entre le début de phrase « *Ils étaient bien content q* » le système va se trouver bloqué par la lettre « *q* » car il ne connaît pas de segment de phrase commençant ainsi. Il est face à une transition indéterminée.

La seconde procédure, d'inférence des entrées similaires, va essayer de voir si en remplaçant un mot de ce segment par un autre mot similaire, on ne retomberait pas sur un segment de phrase appartenant au corpus.

Intuitivement, nous pouvons comprendre que les mots « *contents* » et « *heureux* » sont similaires. Pour aboutir à ce résultat, VITIPI va trouver que les mots et les segments de phrases qui sont devant le mot « *contents* » sont presque les mêmes que ceux que l'on trouve devant le mot « *heureux* ». En remplaçant « *contents* » par « *heureux* », le système ne rencontre plus de transition indéterminée et peut continuer la prédiction.

Après avoir expliqué le fonctionnement du système, essayons maintenant d'évaluer ses performances face à des situations connues ou inconnues.

PERFORMANCES DE VITIPI

Notre objectif était non seulement d'évaluer le fonctionnement de VITIPI lorsqu'il est confronté à des situations inconnues, mais aussi de tester ses capacités d'apprentissage. Pour cela, nous avons réalisé un corpus en prenant chaque jour les prévisions météorologiques disponibles sur le site internet de Météo France. Nous avons procédé de la façon suivante :

Le texte des prévisions du premier jour a été introduit dans le système pour créer un premier transducteur que nous avons minimisé. Avec ce premier transducteur, nous avons simulé l'écriture des prévisions du deuxième jour en comptabilisant le nombre de mots nouveaux ainsi que le pourcentage de lettres entrées par l'utilisateur. Nous avons ensuite pris les prévisions des deux premières journées et nous avons constitué un deuxième transducteur avec lequel nous avons simulé l'écriture de la troisième journée. Ce processus a été réitéré jusqu'au dix-neuvième transducteur pour simuler l'écriture de la vingtième journée. Les résultats obtenus sont les suivants.

En prenant le premier transducteur et en faisant passer les prévisions du deuxième jour, on obtient si on fait la prédiction avec des mots isolés un taux de 14 % de lettres affichées par le système. Par contre, en prenant en compte les 10 mots qui précèdent, et avec le même transducteur, le système affiche 16,5 % de lettres. Les résultats obtenus en faisant passer les prévisions du vingtième jour avec le dix-neuvième transducteur montrent qu'en mots isolés, le taux de prédiction est de 23 % et de 41 % si l'on tient compte des mots qui précèdent. L'apport du modèle de n-grammes pour la prédiction de lettres est donc intéressant.

Il est important de souligner que par souci de réalisme, tous ces transducteurs contenaient au départ les 5.930 mots du vocabulaire du Français de base. Sans ce vocabulaire, les taux de prédiction pour les mots isolés auraient été plus élevés. Par ailleurs, toutes les phrases présentées aux transducteurs pendant les différents tests étaient toutes différentes du corpus d'apprentissage. Par contre, lorsque l'on représente aux transducteurs toutes les phrases du corpus d'apprentissage, on obtient un taux de prédiction proche de 77 %.

Par ailleurs une version de VITIPI, fonctionnant uniquement avec des mots isolés, a été évaluée dans un lycée spécialisé de la région toulousaine [17]. La comparaison par rapport au système HANDIWORD montre que de la 6^e à la 5^e, les élèves préfèrent HANDIWORD car ils maîtrisent mal l'écriture et préfèrent voir le mot complet s'afficher à l'écran. Après la 4^e, VITIPI est préféré car ils comprennent mieux les fonctionnalités du système et ont une meilleure connaissance de l'écrit. VITIPI a permis d'améliorer jusqu'à 2 ou 3 fois la vitesse de frappe des élèves.

AIDE A LA SAISIE PAR DESIGNEUR

Même si les designers ordinaires peuvent très bien fonctionner avec notre système, nous en avons réalisé un, qui grâce à VITIPI optimise la sélection des lettres. Dans les designers « classiques » aussi appelés *systèmes à défilement*, les lettres sont disposées sous forme matricielle. Pour sélectionner une lettre dans le tableau, l'utilisateur va d'abord sélectionner la colonne, puis la ligne contenant la lettre. Pour sélectionner la colonne, un spot lumineux balaye *séquentiellement* la première ligne jusqu'à ce que l'utilisateur envoie un signal d'arrêt au système via un contacteur ou un dispositif adapté à son handicap. De même la sélection de la lettre dans la colonne choisie se fait *séquentiellement*. Si la lettre choisie se trouve au bout de la ligne et/ou au bout de la colonne, le temps mis pour sélectionner la lettre sera très long.

Pour optimiser le processus, VITIPI détermine les lettres qui sont susceptibles d'être écrites et les met en évidence par une coloration différente de la lettre dans la matrice. La sélection de la colonne ne se fera plus par balayage séquentiel, le designer se positionnant sur la colonne possédant au moins une lettre mise en évidence. Si

aucune des lettres mises en évidence n'a été sélectionnée par l'utilisateur, le système balayera les autres lettres. La sélection de la lettre à l'intérieur de la colonne sélectionnée se fera de la même façon en balayant *prioritairement* les lettres mises en évidence.

CONCLUSIONS ET PERSPECTIVES

Nous avons tenu à ce que le système VITIPI ne soit pas tributaire de traitement de textes, afin qu'il puisse s'adapter à n'importe quel autre logiciel (tableur, éditeur de programmes ou de courriers électroniques, ...). De plus, nous voulions que VITIPI s'adapte à toutes les formes de handicap et soit utilisable à partir de n'importe quel clavier (*ordinaire, spécialisé, ou désigneur*). Il fallait donc un système qui fasse l'interface entre les différents claviers, les différents logiciels, et qui envoie à ces derniers, les lettres produites par VITIPI.

Nous avons réalisé une interface sous forme de boîtier électronique, qui s'intercale entre le clavier et l'unité centrale, elle est perçue par les logiciels comme un clavier ordinaire. Le boîtier dispose d'un petit écran permettant de visualiser les mots produits par VITIPI, et les messages envoyés à l'utilisateur. Ce système a été commercialisé, mais le prix trop élevé a en partie découragé les acheteurs éventuels. Nous avons donc changé d'optique en réalisant l'interface sous une forme totalement logicielle qui s'intègre dans un environnement Windows sous PC.

BIBLIOGRAPHIE

1. BRANGIER E., PINO P., LE DREZEN A., LAMAZIERE J. Prothèse interactionnelle: Pallier les déficits interactionnels des handicapés lourds avec une interface de contrôle d'environnement. LLIA N° 123 Interfaces MAI 1997 PP 202 – 206.
2. DEROUAULT A.M. MERIALDO B., STEHLE J.L. Une expérience de transcription automatique sténotypie français. T.S.I vol 2 n°5 SEPT 1983
3. HUNNICUTT S. A lexical prediction for a text-to-speech system RAPPORT DU DEPT OF SPEECH COMMUNICATION STOCKHOLM STL-QSPR 2-1/1985
4. LE PEVEDIC B., MAUREL D. Un dictionnaire électronique évolutif par apprentissage RAPPORT DE RECHERCHE I.R.I.N - 75 DECEMBRE 1994
5. BERTENSTAM J., HUNNICUTT S. Adding Morphology to a Word Predictor TIDE : THE EUROPEAN CONTEXT FOR ASSISTIVE TECHNOLOGY P.312-315 PARIS AVRIL 1995
6. MAGNUSON T. Word Prediction as Linguistic Support for Individuals with Reading and Writing Difficulties TIDE : THE EUROPEAN CONTEXT FOR ASSISTIVE TECHNOLOGY P. 316 - 319 PARIS AVRIL 1995
7. LE PEVEDIC B. Prédiction Morphosyntaxique évolutive dans un système d'aide à la saisie de textes pour des personnes handicapées physiques Thèse de DOCTORAT I.R.I.N OCTOBRE 1997 (N° ED-82-269)
8. CARLBERGER J. Design and Implementation of a Probabilistic Word Prediction Program Master's Thesis Dept. of Speech, Music and Hearing, KTH, SE-100 44 Stockholm, Sweden. 1997
9. GUENTHNER F., KRÜGER-THIELMANN, PASERO R. ; SABATIER P., Communication aids for handicapped persons 2ND EUROPEAN CONFERENCE ON THE ADVANCEMENT OF REHABILITATION TECHNOLOGY, ECART'93 STOCKHOLM MAI 1993
10. GODBERT E., PASERO R. , SABATIER P., La connectivité en langage naturel : Modélisation de contraintes sur le nombre 13-EME CONF INTER : INTELLIGENCE ARTIFICIELLE SYSTEMES EXPERTS, LANGAGE NATUREL AVIGNON MAI 1993
11. PASERO R., SABATIER P., a Linguistic games for language Learning and tests, an ILLICO application Computer-Assisted Language Learning (CALL) 1998
12. JELINEK F., Self-organized language modeling for speech recognition Read-ings in Speech Recognition, Waibel and Lee (Editors). Morgan Kaufmann. 1989
13. BOISSIERE Ph. DOURS D., Writing-assistance system for disabled persons in a man-machine communication 5th EUROPEAN SIGNAL PROCESSING CONFERENCE P. 1651 - 1654 BARCELONE SEPTEMBER 1990
14. BOISSIERE Ph., VITIPI Un système auto-organisationnel pour faciliter le dialogue écrit homme-machine THESE D'UNIVERSITE, I.R.I.T UPS, TOULOUSE NOVEMBRE 1990
15. BOISSIERE Ph., DOURS D., De l'aide à l'écriture pour les personnes handicapées à un système d'aide à la communication et au contrôle de l'environnement. INFORMATIQUE 92 (EC2) P. 645-656 MONTPELIER MARS 1992
16. BOISSIERE Ph. DOURS D., VITIPI : Versatile Interpretation of Text Input by Persons with Impairments 5th national Conference on Computers for Handicapped Persons, pp.165-172, Linz July 1996,
17. DUBUS N. Evaluation de l'interface intelligente d'aide à la saisie informatique, VITIPI au lycée "Le Parc Saint-Agne" Journal d'Ergothérapie PP 95 - 100 MASSON, Mars 1996